aws

User Guide

AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery: User Guide

Table of Contents

What is Elastic Disaster Recovery?

AWS Elastic Disaster Recovery (AWS DRS) minimizes downtime and data loss with fast, reliable recovery of on-premises and cloud-based applications using affordable storage, minimal compute, and point-in-time recovery.

You can increase IT resilience when you use AWS Elastic Disaster Recovery to replicate on-premises or cloud-based applications running on supported operating systems. Use the AWS Management Console to configure replication and launch settings, monitor data replication, and launch instances for drills or recovery.

Set up AWS Elastic Disaster Recovery on your source servers to initiate secure data replication. Your data is replicated to a staging area subnet in your AWS account, in the AWS Region you select. The staging area design reduces costs by using affordable storage and minimal compute resources to maintain ongoing replication.

You can perform non-disruptive tests to confirm that implementation is complete. During normal operation, maintain readiness by monitoring replication and periodically performing non-disruptive recovery and failback drills. AWS Elastic Disaster Recovery automatically converts your servers to boot and run natively on AWS when you launch instances for drills or recovery. If you need to recover applications, you can launch recovery instances on AWS within minutes, using the most up-to-date server state or a previous point in time. After your applications are running on AWS, you can choose to keep them there, or you can initiate data replication back to your primary site when the issue is resolved. You can fail back to your primary site whenever you're ready.

Getting started with AWS Elastic Disaster Recovery

Topics

AWS Elastic Disaster Recovery initialization and permissions

In order to use AWS Elastic Disaster Recovery, the service must first be initialized for any AWS Region in which you plan to use Elastic Disaster Recovery.

Initializing AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery must be initialized upon first use from within the AWS Elastic Disaster Recovery Console. The initialization process occurs automatically once a user accesses the AWS Elastic Disaster Recovery Console. The user is directed to create the default replication settings, and upon saving the template, the service is initialized by creating the IAM roles which are required for the service to work. Learn more about creating the default replication settings as part of the quick start guide.

> **Important**
>
> AWS Elastic Disaster Recovery **is not** compatible with CloudEndure Disaster Recovery.

AWS Elastic Disaster Recovery can only be initialized by the Admin user of your AWS Account. During initialization, the following IAM roles will be created:

- **AWSServiceRoleForElasticDisasterRecovery**

- **AWSElasticDisasterRecoveryReplicationServerRole**
- **AWSElasticDisasterRecoveryConversionServerRole**
- **AWSElasticDisasterRecoveryRecoveryInstanceRole**
- **AWSElasticDisasterRecoveryAgentRole**
- **AWSElasticDisasterRecoveryFailbackRole**
- **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole**

Additional policies

You can create roles with granular permission for AWS Elastic Disaster Recovery. The service comes with the following predefined managed IAM policies:

- AWSElasticDisasterRecoveryConsoleFullAccess
- AWSElasticDisasterRecoveryReadOnlyAccess
- AWSElasticDisasterRecoveryAgentPolicy
- AWSElasticDisasterRecoveryAgentInstallationPolicy
- AWSElasticDisasterRecoveryFailbackPolicy
- AWSElasticDisasterRecoveryFailbackInstallationPolicy
- AWSElasticDisasterRecoveryInstancePolicy
- AWSElasticDisasterRecoveryServiceRolePolicy
- AWSElasticDisasterRecoveryLaunchActionsPolicy

Learn more about AWS Elastic Disaster Recovery roles and managed policies.

Manually initializing DRS

You can manually initialize AWS Elastic Disaster Recovery through the API. This can help you automate service initialization through script when initializing multiple accounts.

> **Note**
>
> You will need to create the replication settings template after initializing the service.

To initialize AWS Elastic Disaster Recovery manually, create the following IAM roles through the IAM CreateRoleAPI. Learn more about creating IAM roles in the AWS IAM documentation.

Creation of each role must include the following parameters:

Role name	Path	Trusted Entity
AWSElasticDisasterRecoveryAgentRole	/service-role/	drs.amazonaws.com
AWSElasticDisasterRecoveryFailbackRole	/service-role/	drs.amazonaws.com
AWSElasticDisasterRecoveryConversionServerRole	/service-role/	ec2.amazonaws.com
AWSElasticDisasterRecoveryRecoveryInstanceRole	/service-role/	ec2.amazonaws.com
AWSElasticDisasterRecoveryReplicationServerRole	/service-role/	ec2.amazonaws.com
AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole	/service-role/	ec2.amazonaws.com

Example using the AWS CLI: `aws iam create-role --path "/service-role/" --role-name AWSElasticDisasterRecoveryReplicationServerRole --assume-role-policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":{"Service":"ec2.amazonaws.com"},"Action":"sts:AssumeRole"}]}'`

After the roles have been created, attach the following AWS managed policies to the roles through the IAM AttachRolePolicy API. Learn more about adding and removing IAM identity permissions in the AWS IAM documentation.

1. Attach Managed Policy **AWSElasticDisasterRecoveryAgentPolicy** to Role **AWSElasticDisasterRecoveryAgentRole**
2. Attach Managed Policy **AWSElasticDisasterRecoveryFailbackPolicy** to Role **AWSElasticDisasterRecoveryFailbackRole**
3. Attach Managed Policy **AWSElasticDisasterRecoveryConversionServerPolicy** to Role **AWSElasticDisasterRecoveryConversionServerRole**
4. Attach Managed Policy **AWSElasticDisasterRecoveryRecoveryInstancePolicy** to Role **AWSElasticDisasterRecoveryRecoveryInstanceRole**
5. Attach Managed Policy **AWSElasticDisasterRecoveryReplicationServerPolicy** to Role **AWSElasticDisasterRecoveryReplicationServerRole**
6. Attach Managed Policy **AWSElasticDisasterRecoveryRecoveryInstancePolicy** and **AmazonSSMManagedInstanceCore** to Role **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole**

> **Note**
>
> Roles must also have a trust policy defined. The trust policy needs to define source identity and source account for security reasons, and allow the service to call SetSourceIdentity and AssumeRole. See the following policy examples.

Example 1: creating a role for the **AWSElasticDisasterRecoveryAgentRole** with trusted entity relationships via the CreateRole API:

Role: AWSElasticDisasterRecoveryAgentRole

```
$ aws iam create-role --path "/service-role/" --role-name
    AWSElasticDisasterRecoveryAgentRole --assume-role-policy-document file://agent-
source-drs-trust-policy.json
```

agent-source-drs-trust-policy.json

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
    "Service": "drs.amazonaws.com"
   },
   "Action": [
    "sts:AssumeRole",
    "sts:SetSourceIdentity"
   ],
   "Condition": {
    "StringLike": {
     "sts:SourceIdentity": "s-*",
     "aws:SourceAccount": "1234567891011"
    }
   }
  }
 ]
}
```

Example 2: creating a role for the **AWSElasticDisasterRecoveryFailbackRole** with trusted entity relationships via the CreateRole API:

Role: AWSElasticDisasterRecoveryFailbackRole

```
$ aws iam create-role --path "/service-role/" --role-name
    AWSElasticDisasterRecoveryFailbackRole --assume-role-policy-document file://
failback-source-drs-trust-policy.json
```

failback-source-drs-trust-policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "drs.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "1234567891011",
                    "sts:SourceIdentity": "i-*"
                }
            }
        }
    ]
}
```

Example 3: creating roles for the **AWSElasticDisasterRecoveryConversionServerRole**, **AWSElasticDisasterRecoveryRecoveryInstanceRole**, and **AWSElasticDisasterRecoveryReplicationServerRole** with trusted entity relationships via the CreateRole API:

Role: AWSElasticDisasterRecoveryConversionServerRole

```
$ aws iam create-role --path "/service-role/" --role-name
    AWSElasticDisasterRecoveryConversionServerRole --assume-role-policy-document
 file://source-drs-trust-policy.json
```

Role: AWSElasticDisasterRecoveryRecoveryInstanceRole

```
$ aws iam create-role --path "/service-role/" --role-name
    AWSElasticDisasterRecoveryRecoveryInstanceRole --assume-role-policy-document
 file://source-drs-trust-policy.json
```

Role: AWSElasticDisasterRecoveryReplicationServerRole

```
$ aws iam create-role --path "/service-role/" --role-name
    AWSElasticDisasterRecoveryReplicationServerRole --assume-role-policy-document
 file://source-drs-trust-policy.json
```

source-drs-trust-policy.json

```
{
    "Version":  "2012-10-17",
     "Statement": [
        {
             "Effect":  "Allow",
             "Principal": {
                 "Service":  "ec2.amazonaws.com"
            },
             "Action":  "sts:AssumeRole"
        }
    ]
}
```

Once the policies are attached to the roles, run the `aws drs initialize-service` command. This will automatically create the service-linked role (**AWSServiceRoleForElasticDisasterRecovery**), create instance profiles, add roles to instance profiles, and will finish service initialization.

Learn more about AWS Elastic Disaster Recovery roles and managed policies.

Programmatically initializing DRS

To programmatically initialize the service, create an IAM role with the following IAM policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:AttachRolePolicy",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:ArnEquals": {
                    "iam:PolicyARN": [
                        "arn:aws:iam::aws:policy/service-role/
AWSElasticDisasterRecoveryAgentPolicy",
                        "arn:aws:iam::aws:policy/service-role/
AWSElasticDisasterRecoveryFailbackPolicy",
```

```
                        "arn:aws:iam::aws:policy/service-role/
AWSElasticDisasterRecoveryConversionServerPolicy",
                        "arn:aws:iam::aws:policy/service-role/
AWSElasticDisasterRecoveryRecoveryInstancePolicy",
                        "arn:aws:iam::aws:policy/service-role/
AWSElasticDisasterRecoveryReplicationServerPolicy"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "drs:InitializeService",
                "drs:ListTagsForResource",
                "drs:GetReplicationConfiguration",
                "drs:CreateLaunchConfigurationTemplate",
                "drs:GetLaunchConfiguration",
                "drs:CreateReplicationConfigurationTemplate",
                "drs:*ReplicationConfigurationTemplate*",
                "iam:TagRole",
                "iam:CreateRole",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:GetRole",
                "iam:DeleteRole",
                "iam:DeleteServiceLinkedRole",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "sts:DecodeAuthorizationMessage",
```

```
                "ec2:DescribeSecurityGroups",
                "ec2:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/drs.amazonaws.com/
AWSServiceRoleForElasticDisasterRecovery"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:ListInstanceProfilesForRole",
                "iam:GetInstanceProfile",
                "iam:ListInstanceProfiles",
                "iam:AddRoleToInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::*:instance-profile/*",
                "arn:aws:iam::*:role/*"
            ]
        }
    ]
}
```

Once the policies are attached to the roles, run the `aws drs initialize-service` command. This will automatically create the service-linked role (**AWSServiceRoleForElasticDisasterRecovery**), create instance profiles, add roles to instance profiles, and will finish service initialization.

Learn more about AWS Elastic Disaster Recovery roles and managed policies.

Accessing the AWS Elastic Disaster Recovery Console

You can access AWS Elastic Disaster Recovery directly through the AWS Console or through the following links:

- Commercial AWS Regions: https://console.aws.amazon.com/drs/home
- AWS GovCloud Regions: https://console.amazonaws-us-gov.com/drs/home

Supported AWS Regions

The following AWS Regions are supported by AWS Elastic Disaster Recovery:

Region name	Region identity	Support in AWS Elastic Disaster Recovery
AWS GovCloud (US-West)	us-gov-west-1	Yes
AWS GovCloud (US-East)	us-gov-east-1	Yes
US East (Ohio)	us-east-2	Yes
US East (N. Virginia)	us-east-1	Yes
US West (N. California)	us-west-1	Yes
US West (Oregon)	us-west-2	Yes
Africa (Cape Town)	af-south-1	Yes
Asia Pacific (Hong Kong)	ap-east-1	Yes
Asia Pacific (Mumbai)	ap-south-1	Yes
Asia Pacific (Hyderabad)	ap-south-2	Yes
Asia Pacific (Osaka)	ap-northeast-3	Yes
Asia Pacific (Seoul)	ap-northeast-2	Yes
Asia Pacific (Singapore)	ap-southeast-1	Yes
Asia Pacific (Sydney)	ap-southeast-2	Yes
Asia Pacific (Jakarta)	ap-southeast-3	Yes
Asia Pacific (Melbourne)	ap-southeast-4	Yes
Asia Pacific (Tokyo)	ap-northeast-1	Yes

Region name	Region identity	Support in AWS Elastic Disaster Recovery
Canada (Central)	ca-central-1	Yes
Europe (Frankfurt)	eu-central-1	Yes
Europe (Zurich)	eu-central-2	Yes
Europe (Ireland)	eu-west-1	Yes
Europe (London)	eu-west-2	Yes
Europe (Milan)	eu-south-1	Yes
Europe (Spain)	eu-south-2	Yes
Europe (Paris)	eu-west-3	Yes
Europe (Stockholm)	eu-north-1	Yes
Middle East (UAE)	me-central-1	Yes
Middle East (Bahrain)	me-south-1	Yes
Israel (Tel Aviv)	il-central-1	Yes
South America (São Paulo)	sa-east-1	Yes

DRS technical training materials

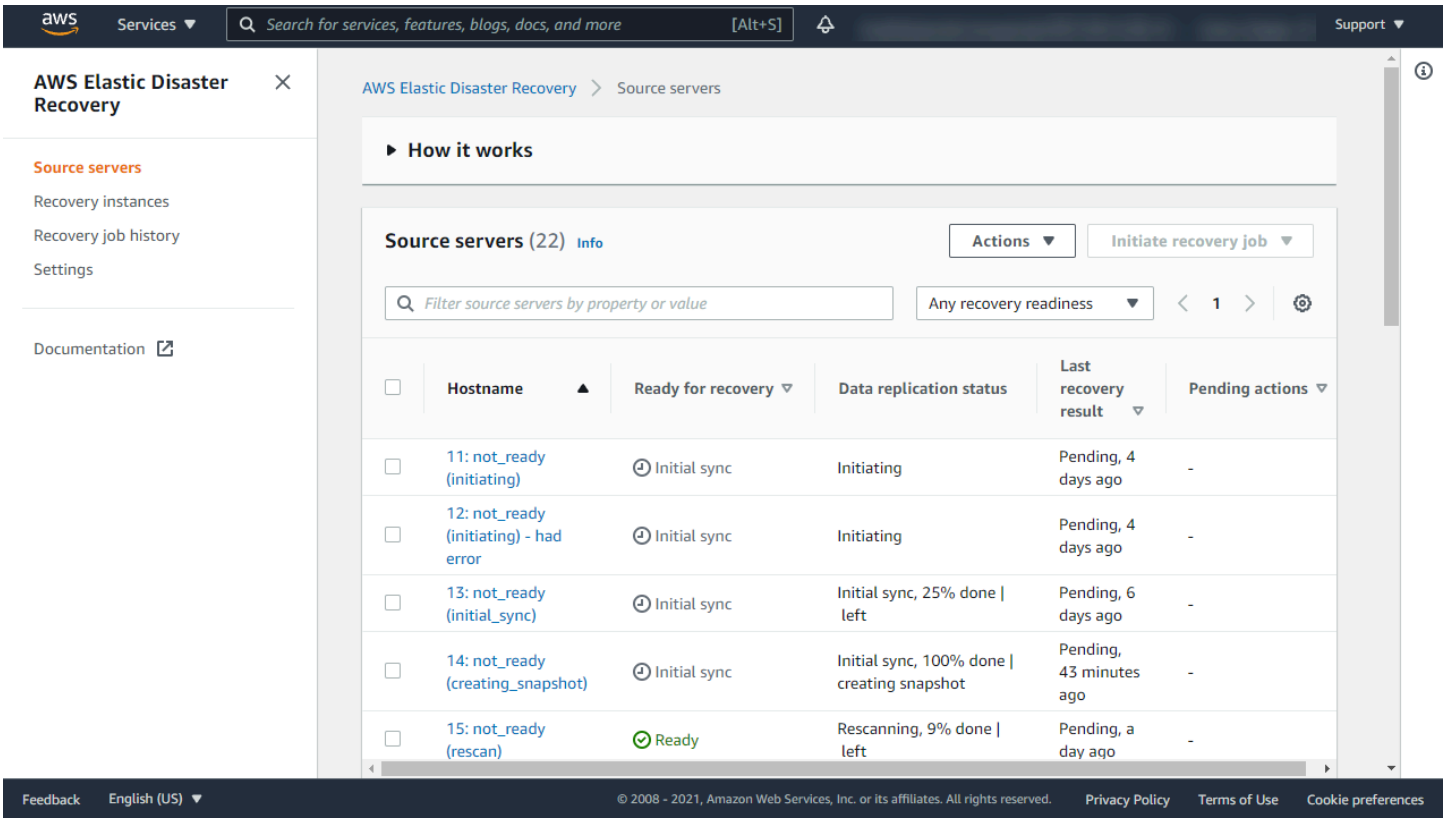
The following free technical trainings are available for DRS:

- AWS Elastic Disaster Recovery - A Technical Introduction

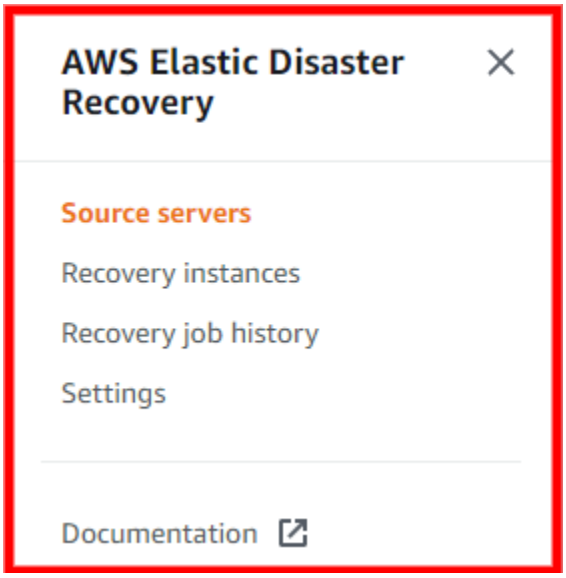
Using the AWS Elastic Disaster Recovery Console

AWS Elastic Disaster Recovery is AWS Region-specific. Make sure that you select the correct Region from the **Select a Region** menu when using AWS Elastic Disaster Recovery, just like you would with other AWS Region-specific services such as Amazon EC2.

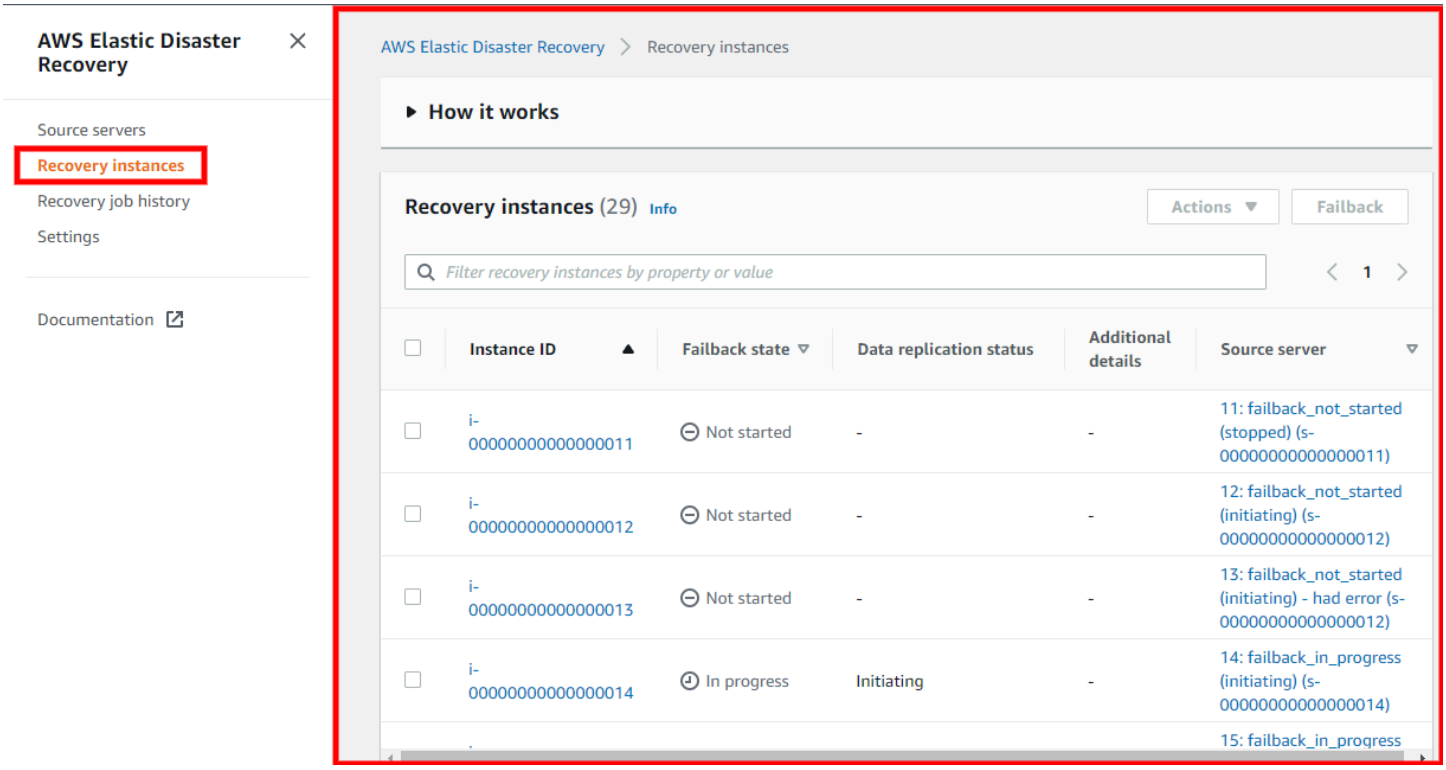
AWS Elastic Disaster Recovery is divided into several primary pages. Each page contains additional tabs and actions. The default view for the AWS Elastic Disaster Recovery Console is the **Source servers** page. This page automatically opens every time you open AWS Elastic Disaster Recovery.



You can navigate to other AWS Elastic Disaster Recovery pages through the left-hand **AWS Elastic Disaster Recovery** navigation menu.

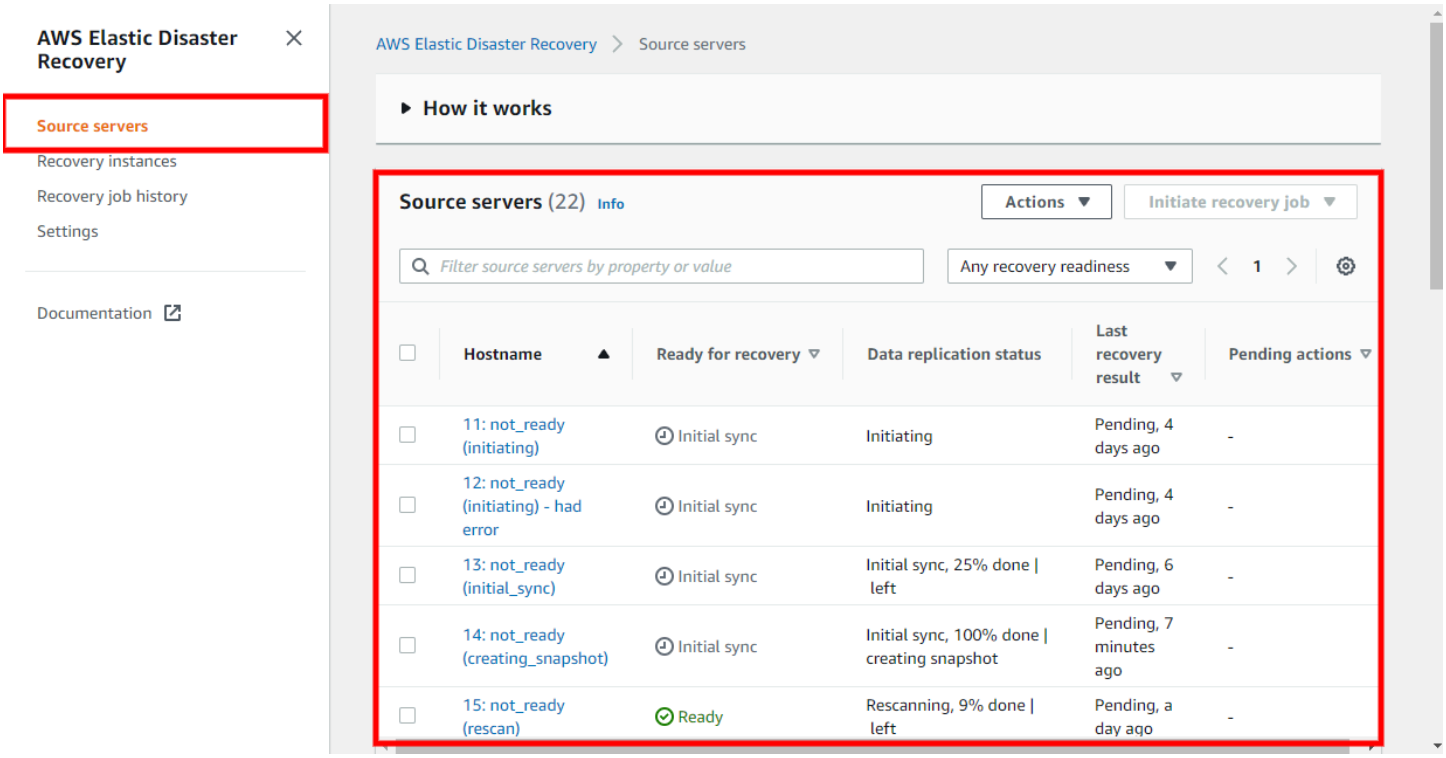


Each Elastic Disaster Recovery page will open in the right-hand main view. Here, you can interact with the various tabs, actions, and settings on the page.

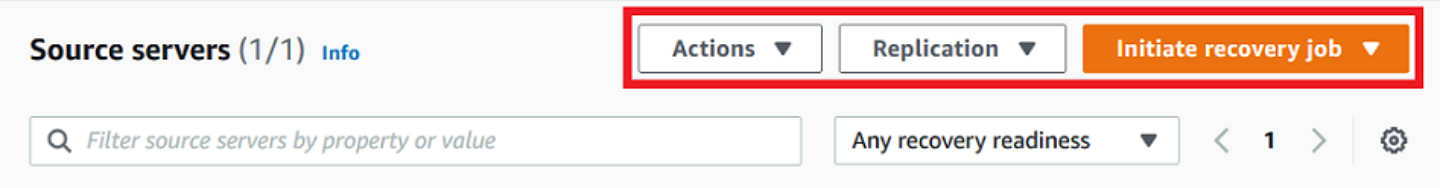


Source servers page

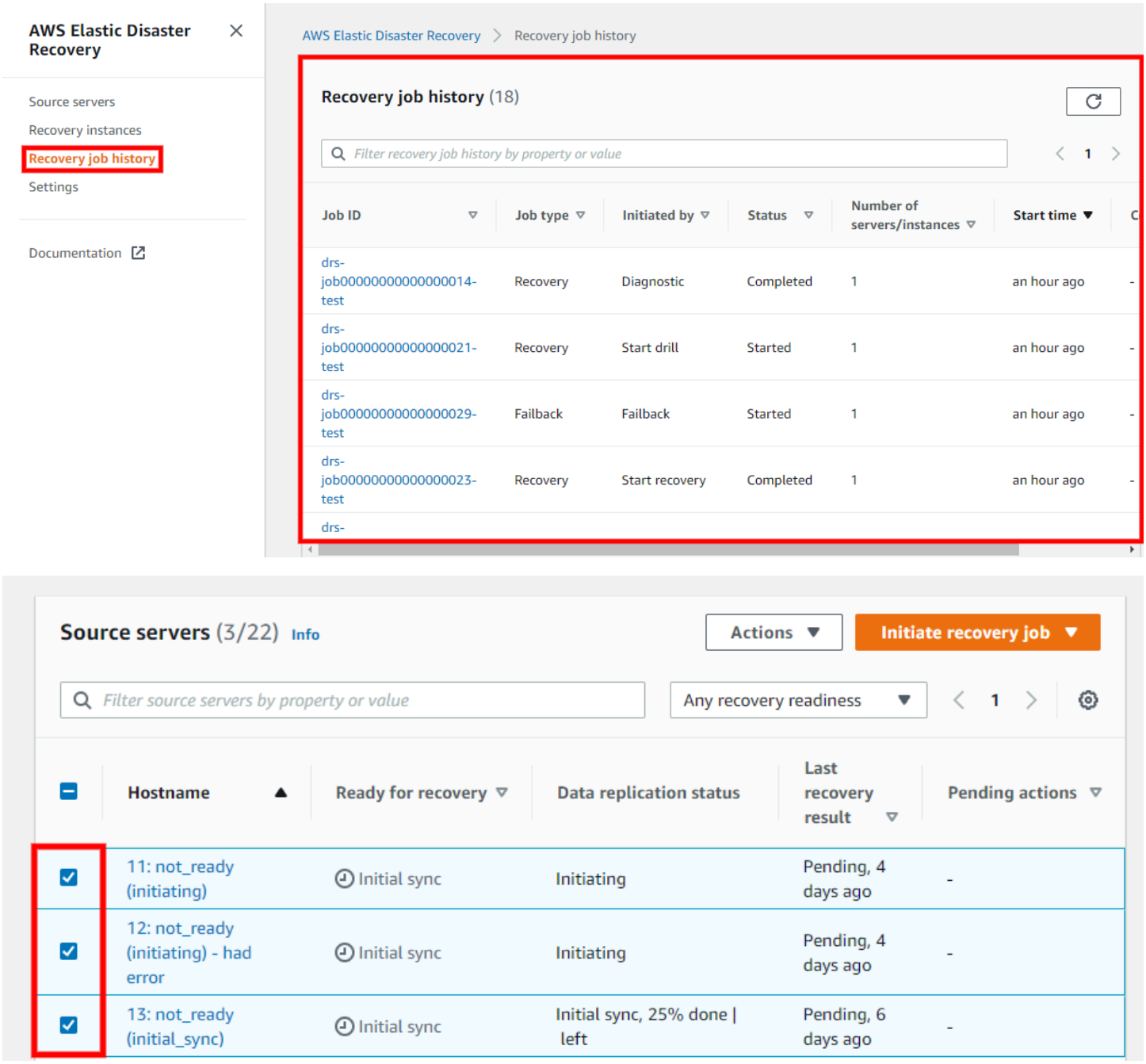
The Source Servers page lists all of the source servers you added to AWS Elastic Disaster Recovery and allows you to interact with your servers and perform a variety of actions. Learn more about the Source servers page.



You can control your source servers within the EAWS Elastic Disaster Recovery Console through the **Actions** and **Initiate recovery job** menus.



You can review the progress of all commands through the **Recovery job history** tab. Learn more about recovery job history.



The commands within the **Actions** and **Initiate recovery job** menus influence the specific source servers you have selected. You can select a single source server or multiple source servers for any command by checking the box to the left of the server name.

You can use the **Filter source servers...** box to filter servers based on a variety of parameters.

AWS Elastic Disaster Recovery color codes the state of each source server. Use the **Alerts** column to easily determine the state of your server.

- A server that is ready to launch Drill or Recovery instances will show the green checkmark and will state **Ready**.

  Ready

  A server that is ready to launch Drill or Recovery instances, but is experiencing a non-critical issue such as lag will show the blue warning sign and will state **Ready** and will display the lag duration to the right. You may need to take action to fix the lag.

  Ready | lag 2 hr

  A server that is still undergoing initial sync will show a gray circle with three dots and will state **Initial sync**.

  Initial sync

  A server that is disconnected will show the gray warning sign and will state **Disconnected**.

  Disconnected

  A server that is not ready due to a significant error, such as a stall, will show a red **X** and will state **Not ready**. The Not Ready state is only shown for servers that are not replicating and do not have any previously created Points in Time. Action must be taken in order to fix the issue.

  Not ready

When various commands are initiated, AWS Elastic Disaster Recovery will display information messages at the top of the **Source servers** page. AWS Elastic Disaster Recovery color codes these messages for clarity.

A green message means that a command was completed successfully.

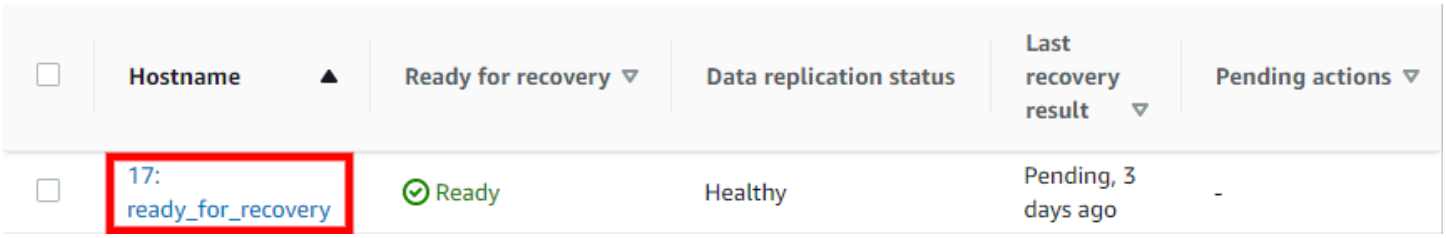
Example:

A red message means that a command was not completed successfully.

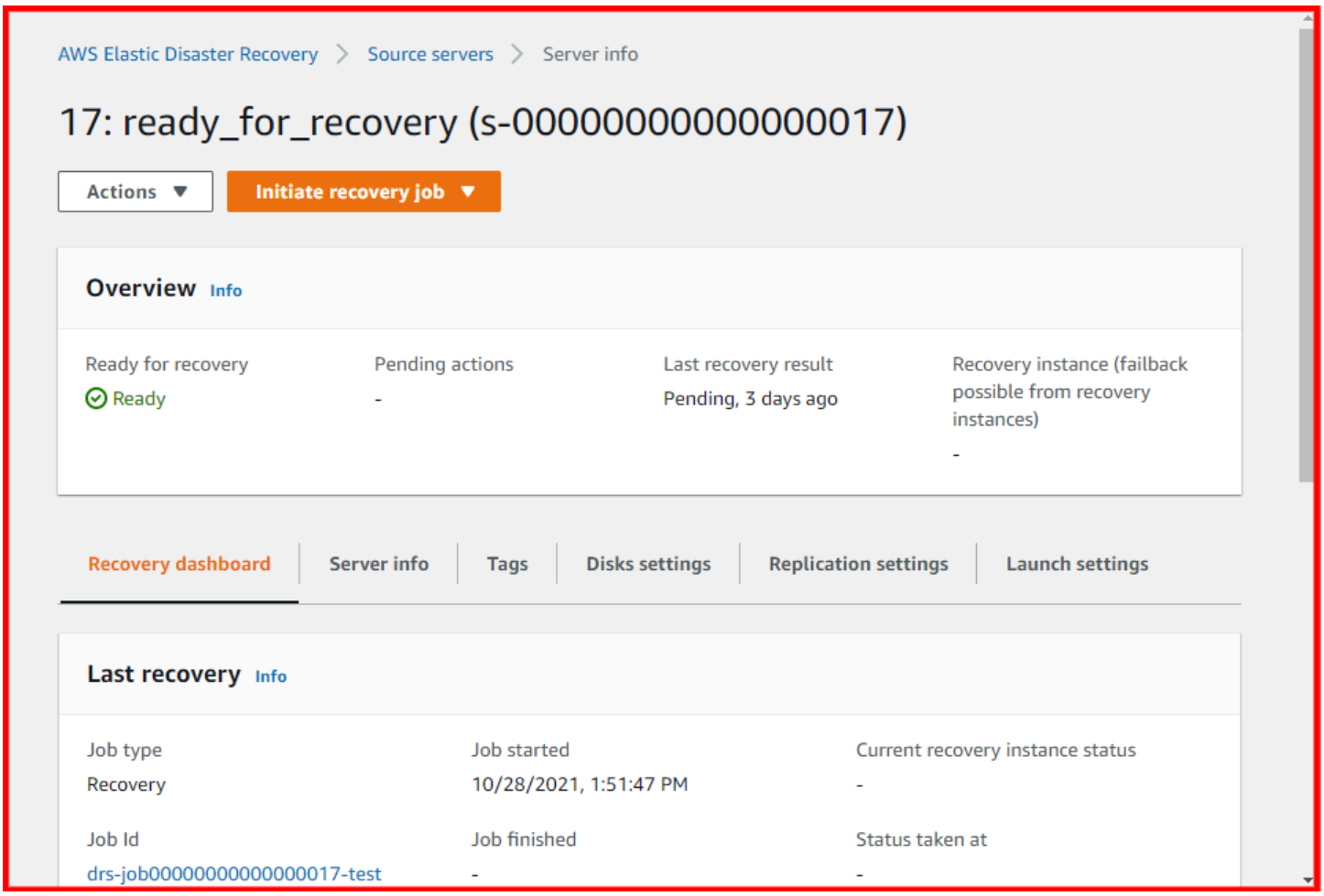
Example:

Each message shows details and links to supplemental information.

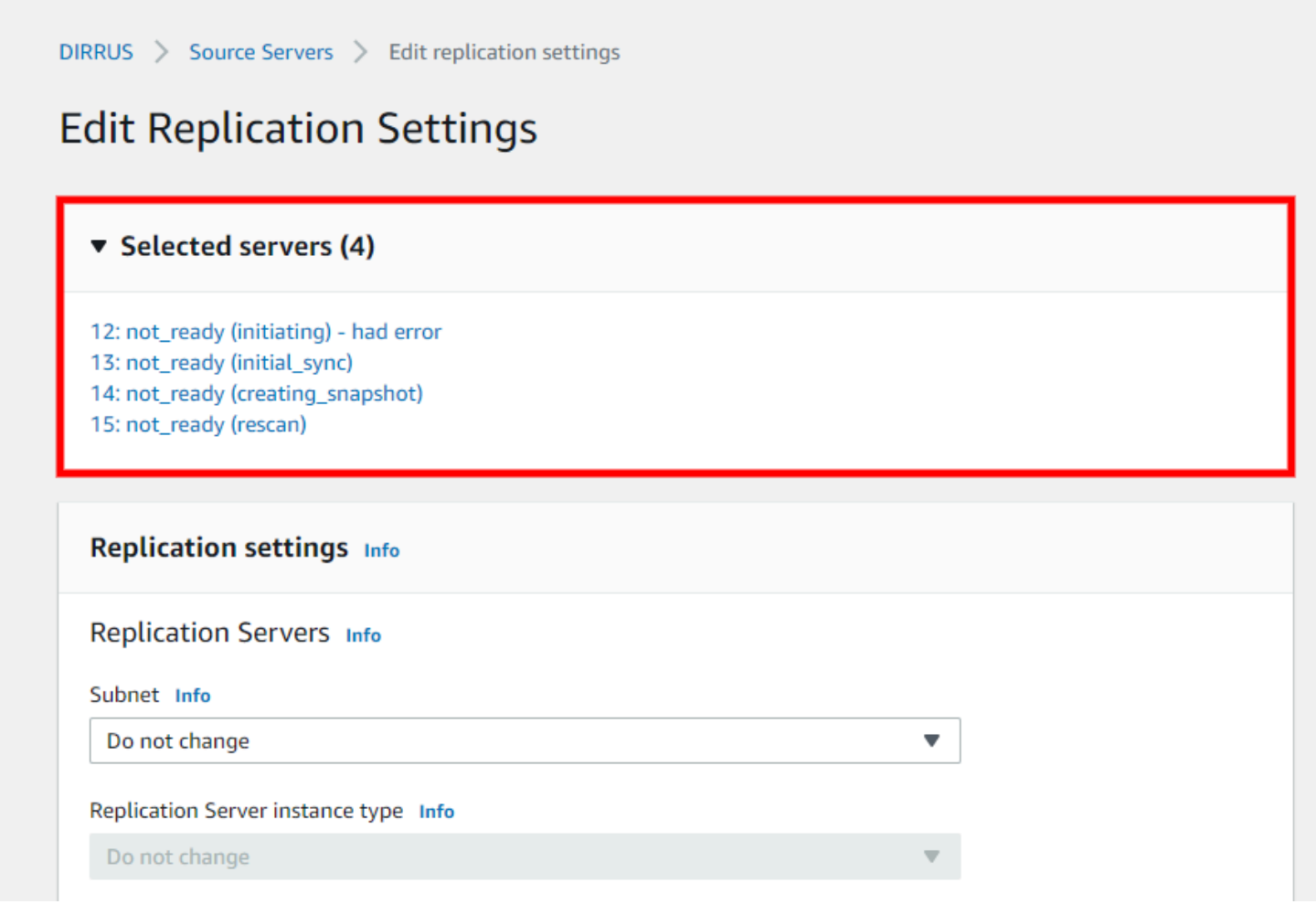
AWS Elastic Disaster Recovery allows you to interact with and manage each server. Choose the server hostname to be redirected to the server details view.



The **Server details** view tab shows specific details for an individual server. From here, you can see an overview of the server's recovery state, as well as various technical details, manage tags, manage disks, edit the server's replication settings, and edit the server's launch settings through the various tabs. Learn more about the Server Details view.



Certain Elastic Disaster Recovery commands, such as **Edit replication settings**, allow you to interact with multiple source servers at once. When multiple source servers are selected by checking the box to the left of the server name and the **Replication > Edit replication settings** option is chosen, AWS Elastic Disaster Recovery will indicate which servers are being edited.



In order for setting changes you have made in the AWS Elastic Disaster Recovery Console to take effect, be sure to click **Save** at the bottom of each Settings page.

Disaster recovery overview

The general process is:

1. Install the AWS Replication Agent on the source server.
2. Wait until initial sync is finished.
3. Launch drill instances. Perform acceptance drills on the servers
4. Initiate a failover by redirecting traffic.
5. Confirm that the Recovery instance was launched successfully.
6. To recover your data, initiate a failback.
7. Complete the failback

8. Return to normal operations.

Best practices

For a more complete discussion of best practices for planning, implementing, and maintaining disaster recovery for on-premises applications using AWS, see this white paper.

Planning

1. Being ready for a real recovery event requires pre-planning. Simply having your servers replicating to AWS, and even having launched them once is not enough. You should have a written recovery plan of what to do in the event of a real recovery event. To learn more, read this Checklist for your IT disaster recovery plan.
2. Once your source servers have reached the Healthy state (after initial sync has completed), you should launch Drill instances for each of your applications and ensure that each application as a whole is working as expected when running in your recovery AWS Region. As you go through this process, you will likely create the necessary network resources required (together with security groups and other related resources). While you can keep these recovery networks (and related resources) up and running even when not in use, it is recommended that once you have them set up properly, create a CloudFormation template that can create them on demand, should the need arise. You should discover and record the order in which servers, and applications need to be launched, and record this in the recovery plan.

Drilling

Regular drills are an integral part of any Disaster Recovery solution. With DRS, drilling is simple and nondisruptive (both to the servers at the source, and to the replication process itself). We recommend drilling as often as is practical, and at least several times a year, and updating the recovery plan with any findings and required changes. Testing and understanding failback is also important. Be sure to include it in your initial drill, and in at least some of your regular drills.

Regular testing can help ensure that your resources are properly prepared for both disasters and scheduled drills. Before conducting large-scale scheduled drills, make sure you meet all the prerequisites and run the required tests. To allow our support team to assist you in case of misconfiguration or other issues, conduct the preliminary testing a week or 2 before the scheduled drill.

> **Note**
>
> While your drill instances are up and running, you are paying for them as per your standard Amazon EC2 rates. Make sure to terminate the drill instances when the drill is done, and include this as a step in your recovery plan.

Monitoring

You can monitor the health of the ongoing replication using the DRS console or programmatically. In the AWS DRS console, go to the **Servers list** page, and look at the **Ready for recovery** column. Any server that is not showing as **Ready** with a green checkmark, may require attention. Servers that show **stalled** in the **Data replication status** column require your intervention to resolve. Servers that are showing **Lag**, may resolve themselves (unless they are also stalled). You should monitor and explore to see if the Lag is a persistent problem (for example, due to insufficient network bandwidth). You can use a scripted solution and the DRS API to respond to servers becoming stalled, or going into lag, or you can use Amazon EventBridge and the EventBridge events generated by AWS DRS.

Limits

Due to Amazon EBS limits on the rate at which EBS snapshots can be taken, the maximum number of servers that can be replicated using DRS in a single AWS account is limited to 300. To replicate more than the maximum number of servers, use multiple AWS accounts, or multiple target AWS Regions (you will need to set up DRS separately for each account/ Region.

Protecting Point-In-Time snapshots

DRS uses EBS snapshots to maintain recovery Points-In-Time. If these are deleted, then you can only recover from the latest state, as maintained on the replication server (and if it is terminated, then you can no longer recover at all). In the event of a breach, which includes not just corruption of your data at source, but also access to your AWS account, then the malicious actor could delete your Point-In-Time snapshots, unless you take extra measures to protect them.

Controlling agent installation permissions

You should control who can install the AWS Replication Agent in your account. Once an agent is installed you immediately begin accruing charges for DRS, and for replication resources (such as

EBS, etc.) The agent installation permissions should be as limited as is practical. The recommended way for controlling who can install agents is to create an IAM role, and to allow users to assume the role.

1. Create an IAM role (IAM docs link | IAM console link), based on the DRS managed permission for agent installation. If this role is to be used by someone outside of your AWS account make sure to use the external ID functionality. Send the role ARN to the users who need to install agents (ARN is not secret and can be sent via email). Use permission boundaries to further limit what can be done using that role. For example, you can control which AWS Region it can be used for, how long the temporary credentials created with the role are good for, specify tags that must be provided (or may not be provided) during agent installation, and more.
2. Users who install the agents assumes that role (must be a user of an AWS account, either yours, or another; you configure who the role is for in step 1). This creates temporary IAM credentials for that users which are used for agent installation. These credentials are limited to only the permissions required for agent installation (and further limited by the permission boundaries you defined), yet are associated with the user (for example, so their usage can be tracked using CloudTrail).

Recovery best practices

1. **Overview:** DRS makes successful failover possible, by handling ongoing replication, and the on-demand launching of actual Recovery instances. The re-routing of data is not done via DRS, and should be done using your preferred DNS routing service, such as Amazon Route 53. Your recovery plan should include details of which service to use, who in your organization owns this service, and what conditions must be met to perform the re-routing (for example: launch Recovery instances using DRS, perform successful launch-validation test, wait for system X, Y, and Z to also launch and pass test, then re-route).
2. **Termination protection for recovery instances:** When you launch recovery instances in case of a real event, you should prevent them from being inadvertently terminated. This should be done after you have performed launch-validation test, and before data re-routing. You can turn on termination protection directly from the Amazon EC2 console, by selecting the instances, and from the **Actions** menu choosing **Instance settings, change termination protection**, and choosing **Yes, Enable.** You should document this step in you recovery plan. Learn more about termination protection.
3. **Understanding failover costs:** Your EC2 recovery instances are created according to the launch settings you have configured for each source server. Recovery instances accrue EC2 and EBS

charges as per AWS rates for your account in the target AWS Region. While you use the Recovery instances, you also continue paying for DRS, and the replication resources it created.

4. **Failover dos and don'ts:** Do not use the **Disconnect from AWS** action in the DRS console for servers for which you launched Recovery instances, even in the case of a real recovery event. Performing a disconnect will terminate all replication resources related to these source servers, including your Point-In-Time (PIT) recovery points. You may need these PITs while you are in failover state, for regulatory reasons, or to re-launch a Recovery instances for any reason (for instance if you discover that the PIT from which you launched includes corrupt or malicious data, and you want to relaunch from an earlier PIT). You should realize that while you you use your Recovery instances as your primary, and new data is presumably written to them, these recovery instances are not themselves being replicated, and you are not creating any new PITs for these changes. It is possible to configure the Recovery instances as new source servers and replicate them cross-Region, so as to have DR for your recovery site (this carries with it additional costs, as is detailed in the linked page).
5. **Using recovery for migration:** Once you launch and use recovery instances on AWS for a real event, you may wish to go on using them permanently, instead of your original servers. The primary additional steps you need to do are:
   a. Set up cross region replication, so that these recovery instances become new source servers;
   b. Wait for these new source servers to have to full number of daily PITs that you need to maintain;
   c. Perform the **Disconnect from AWS** action on the original source servers, so as to avoid confusion, and to stop paying for DRS and related replication resources for these original source servers. You can also then choose **Delete** from the **Actions** menu, and this will cause DRS to forget everything it knows about these source servers, and for them to no longer appear in the Elastic Disaster Recovery console.
6. **Recover into existing instance:** In case you would like to recover into an instance that already exists, instead of launching a new one for recovery, drill or failback. Instance to recover into must be of the same operating system platform (Linux or Windows) as the source instance, it must be stopped and it must have the tag key *AWSDRS* and tag value *AllowLaunchingIntoThisInstance*. Learn more about recover into existing instance.

Failback best practices

1. **Mass failback:** If you are failing back more than several servers, and your source environment is VMware vCenter, then consider using DRS Mass Failback Automation client.

2. **Return to normal operation:** make sure that the failed-back servers at the source are replicating back to AWS, and appear as source servers in the DRS console. If they do appear in the DRS console and are not replicating, explore the reason (such as firewall settings, etc.) If they do not appear in the DRS console you may need to install / re-install the AWS Replication Agent on them. Make that you do not end up with two source server entities in the DRS console, one representing the original server, and one the failed-back server.
3. **Cleanup after return to normal operation:** Once you have completed failback, there may be multiple AWS resources left behind that you no longer need and that are costly to maintain:

   After performing a failback to on-premises environment, perform the following steps:

   - Clean Recovery instances: Terminate these instances from the **Recovery instances** page of the DRS Console.
   - Source servers: These appear in the Source Servers page of the DRS console. Make sure that you only have one source server in the DRS console for each actual server at the source. Source servers are billed by DRS and consume replication resources (billed by other AWS services) until you perform the **Disconnect from AWS** action. If you do have duplicate source servers, do not disconnect/delete the original ones until the new ones have accumulated all the Point-In-Time recovery points (PITs) you need. Performing the **disconnect from AWS** action will cause the PITs from the original sources servers to be discarded. If your source is also in AWS, then you will have more resources that need to be cleaned up. Learn more about cleaning up these resources.

> **Note**
>
> The cleanup process following a cross-region failback is different. Learn how to perform a cleanup following a cross-region failback.

Security best practices

You can review security best practices in the Security chapter.

Quick start guide

This section will guide you through first time Elastic Disaster Recovery setup, including:

Topics

First time setup

In order to use AWS Elastic Disaster Recovery (AWS DRS), you first need to set it up in each AWS Region in which you want to use it (the Region into which you will be replicating, and where you will launch Recovery instances). Setting up the service consists of defining default replication settings and creating the roles and permissions required for the service to operate.

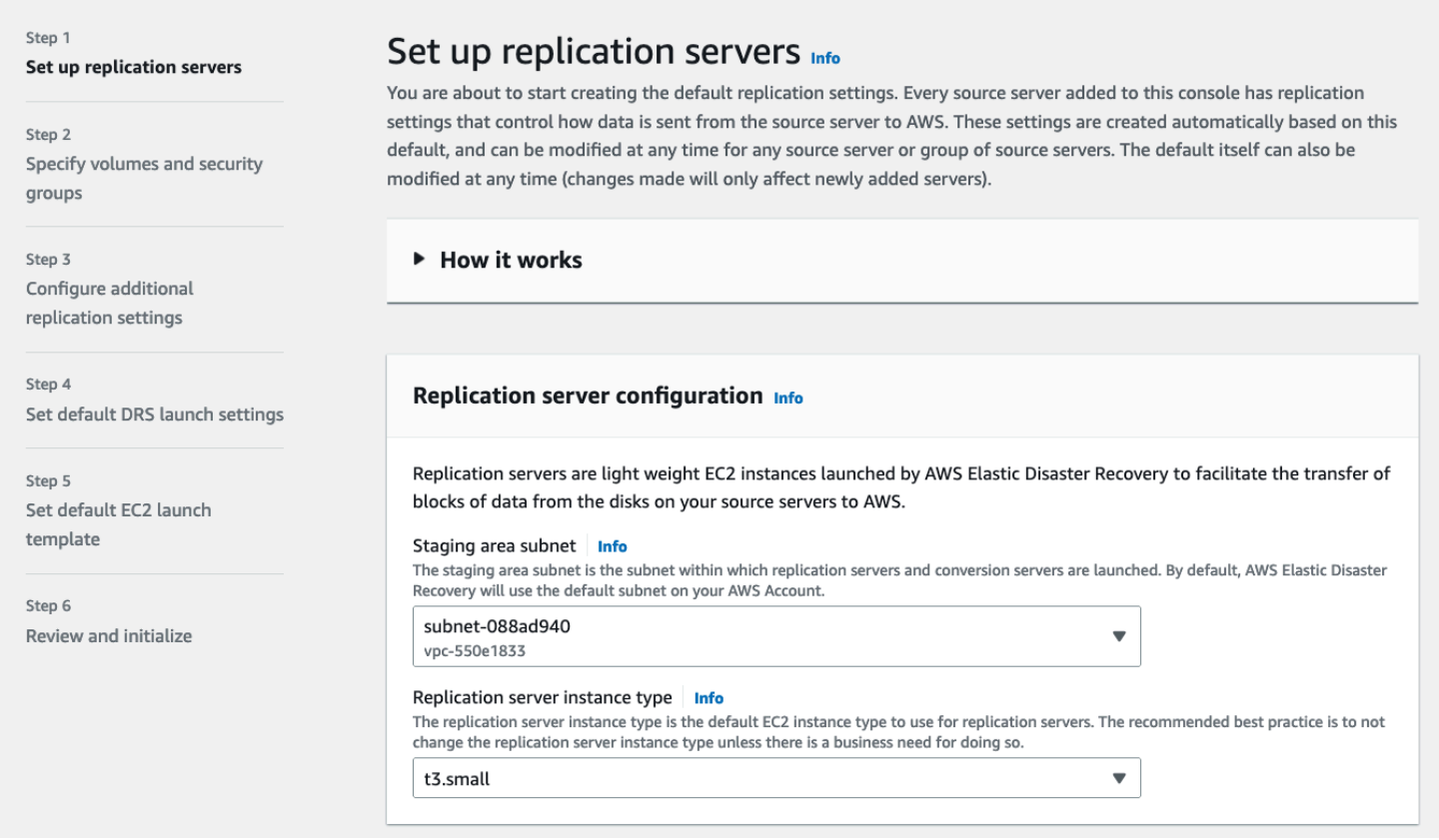
> **Note**
>
> You need to be the admin user of the AWS account, or have a role with the AWSElasticDisasterRecoveryConsoleFullAccess permission in order to set up the service

The first setup step for AWS DRS is setting the default replication settings. Choose **Set default replication settings** on the AWS Elastic Disaster Recovery landing page.

You will be guided through the steps of setting up your default replication settings, default launch settings, and EC2 template. These default settings will be applied to every source server that is added to AWS Elastic Disaster Recovery. You can change both the default settings and individual source server settings for one or more source servers at any time. Learn more about editing your replication settings and launch settings.

Note

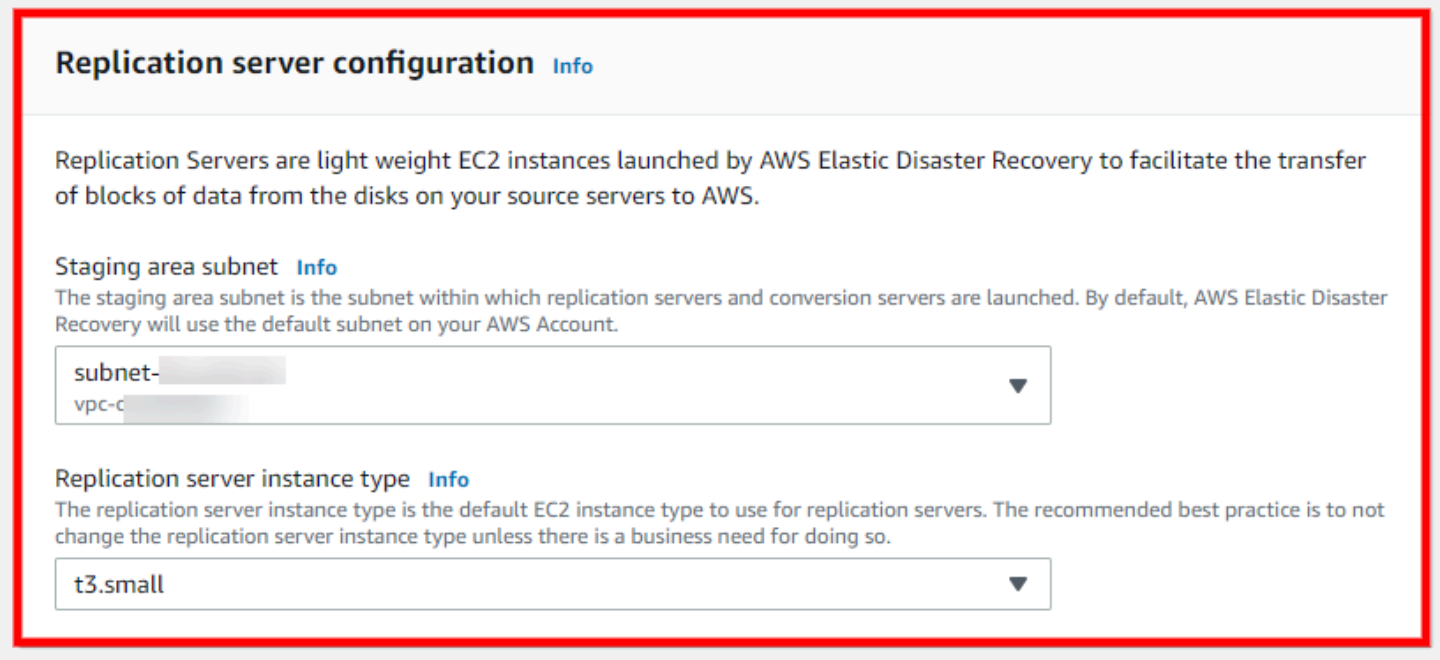
You can use the default setting, by simply choosing **Next** on each of the pages in this wizard or modify any of the setting to best fit your needs. To learn more about each setting, click the **Info** links next to each section.



Important

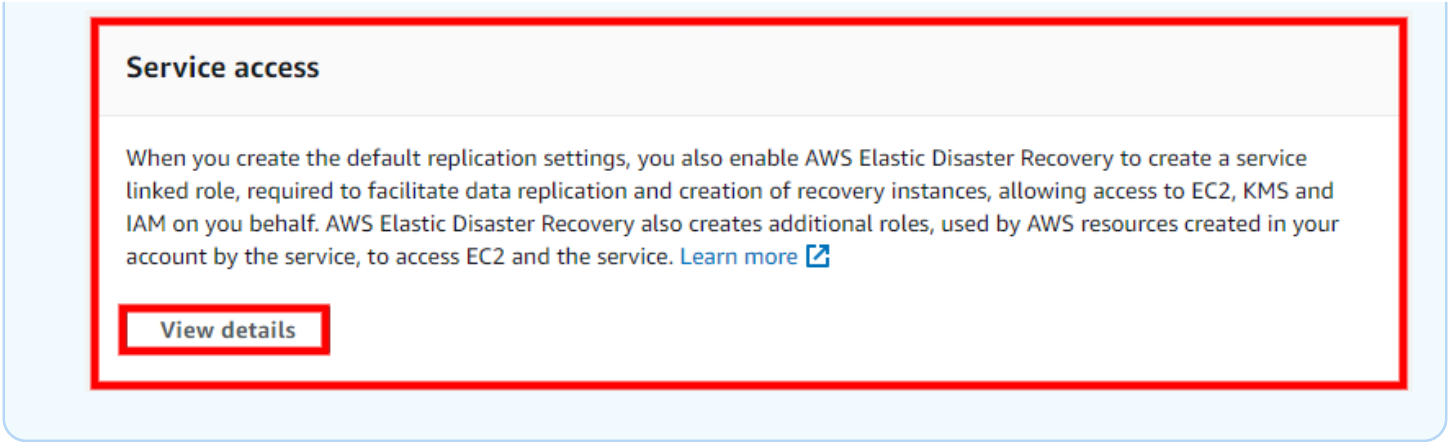
Before configuring your default settings, ensure that you meet the Network requirements for running AWS Elastic Disaster Recovery

On the first page of the wizard, you will be asked to **Set up replication servers**. Replication servers are lightweight Amazon EC2 instances that are used to replicate data between your source servers and AWS. Replication servers are automatically launched and terminated as needed. You can start using AWS Elastic Disaster Recovery with the default replication server settings or you can configure your own settings. Learn more about replication server settings.



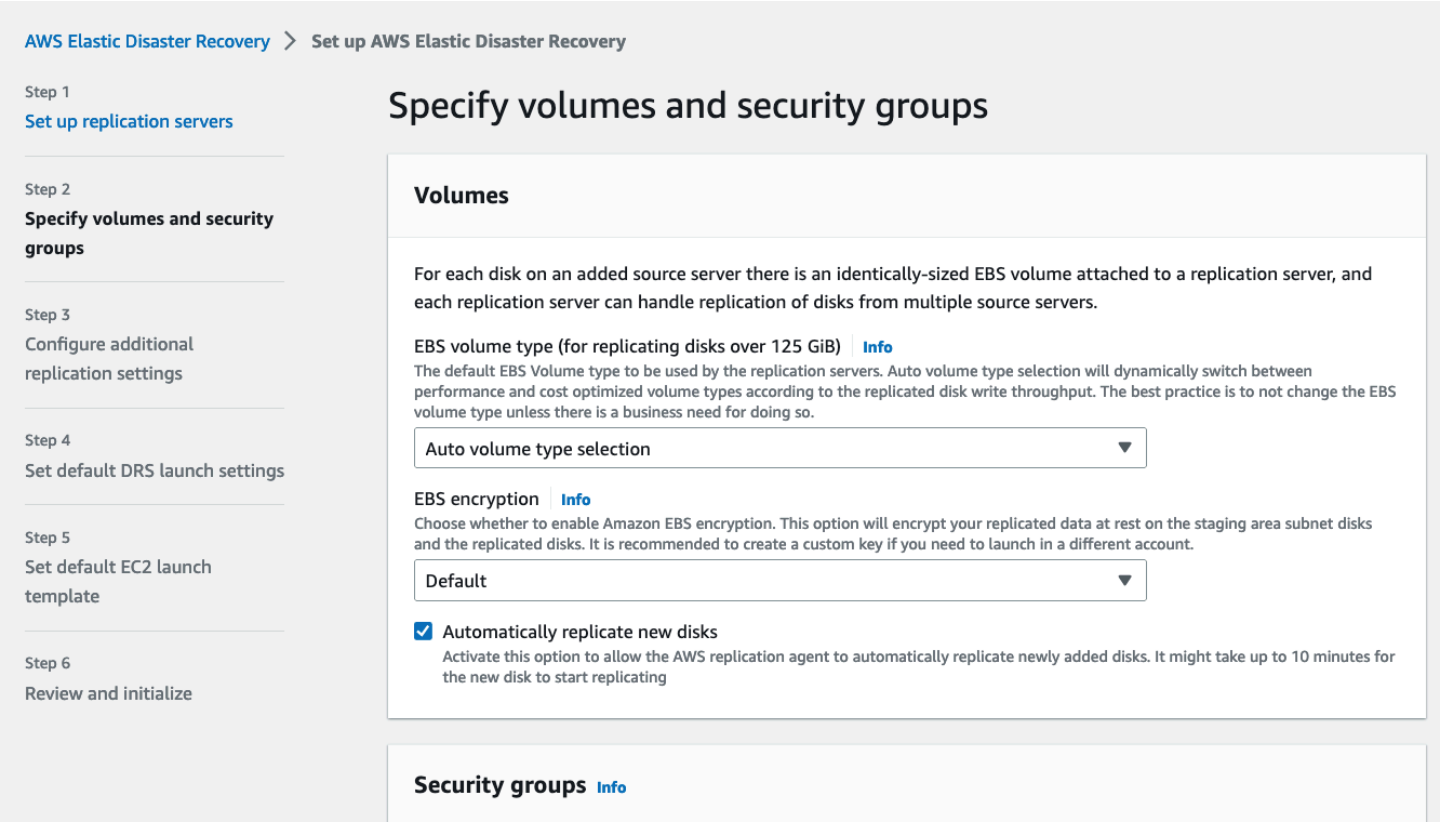
- Configurable replication server settings include:
  - The subnet within which the replication server will be launched
  - Replication server instance type

> **Note**
>
> During this step you will also be able to review the service linked role and additional policies created during Elastic Disaster Recovery initialization. Choose **View details** to learn more.

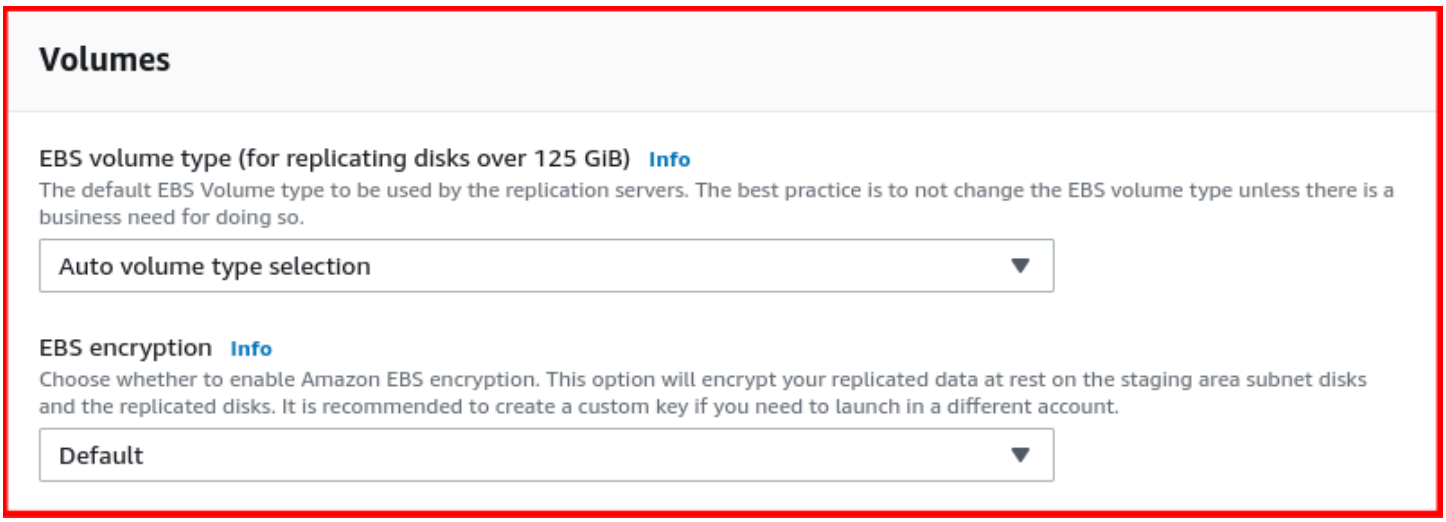


Click **Next** to proceed to the second page of the wizard.

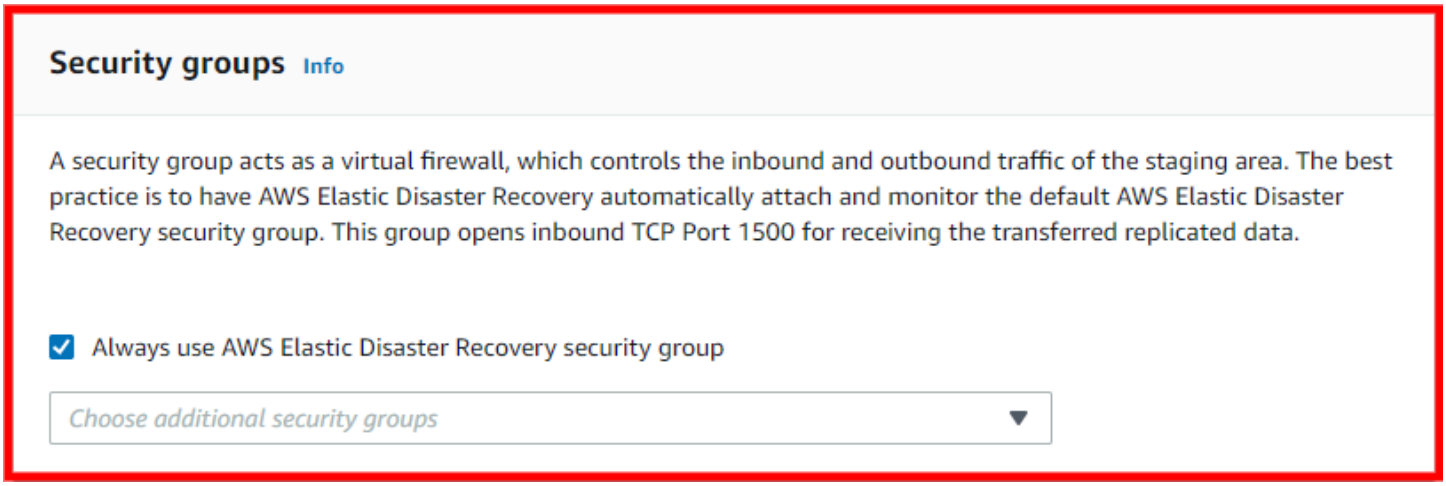
On the second page of the wizard you will be asked to **Specify volumes and security groups**.



For each disk on an added source server there is an identically-sized EBS volume attached to a replication server, and each replication server can handle replication of disks from multiple source servers. Learn more about volumes.



A security group acts as a virtual firewall, which controls the inbound and outbound traffic of the staging area. The best practice is to have AWS Elastic Disaster Recovery automatically attach and monitor the default AWS Elastic Disaster Recovery security group. This group opens inbound TCP Port 1500 for receiving the transferred replicated data. Learn more about security groups.

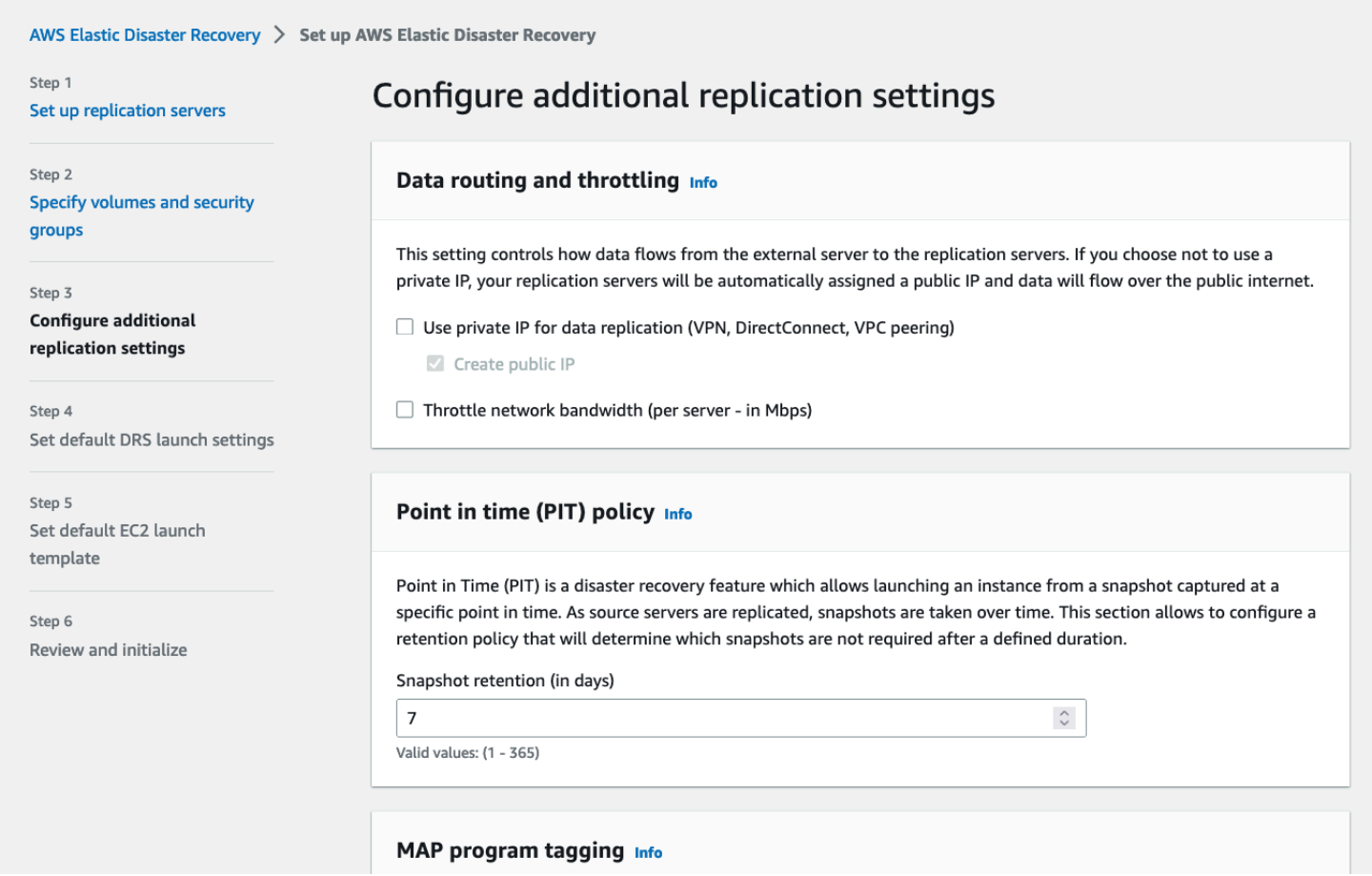


Configurable volumes and security groups settings include:

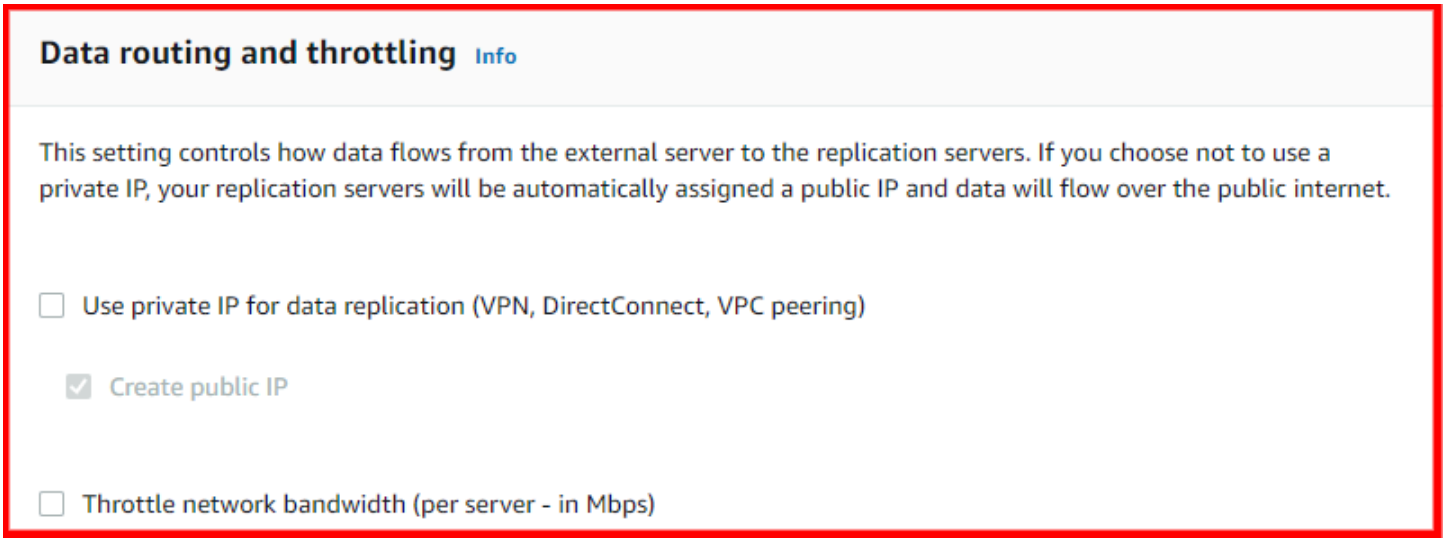
- EBS volume type
- EBS encryption
- Always use AWS Elastic Disaster Recovery security group

Click **Next** to proceed to the third page of the wizard.

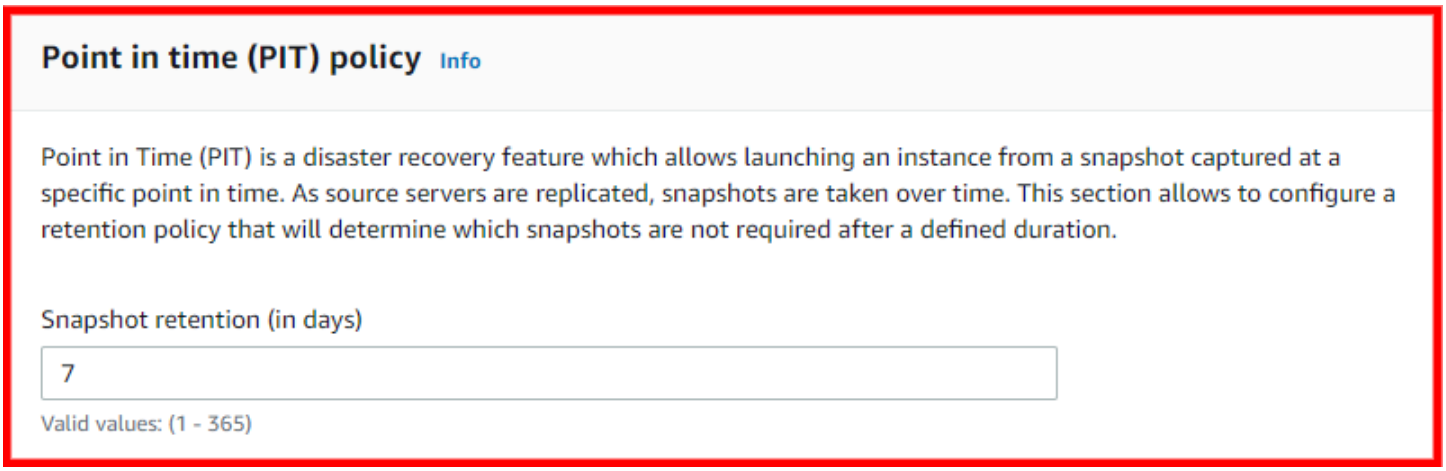
On the third page of the wizard you will be asked to **Configure additional replication settings**. These include **Data routing and throttling**, **Point in time (PIT) policy**, and **Tags**.



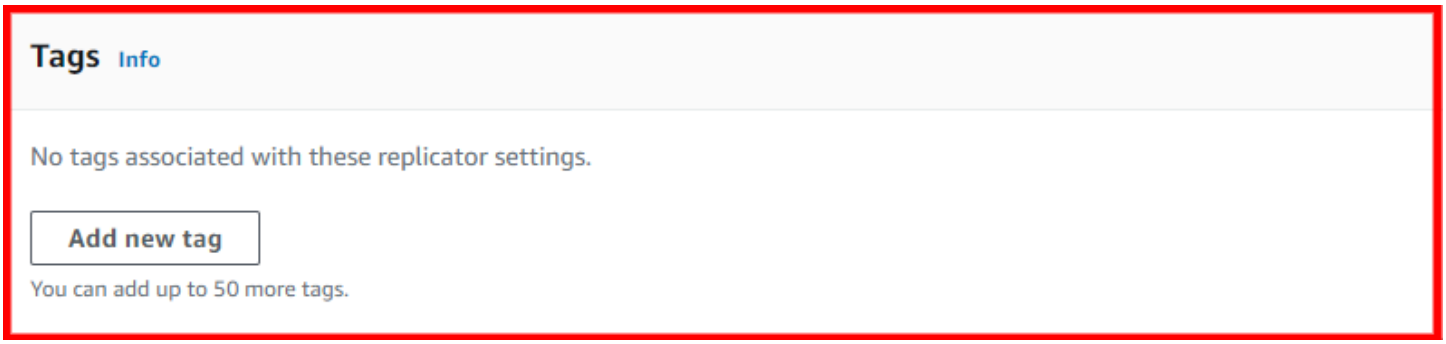
Data routing and throttling controls how data flows from the external server to the replication servers. If you choose not to use a private IP, your replication servers will be automatically assigned a public IP and data will flow over the public internet. Learn more about data routing and throttling.



Point in Time (PIT) is a disaster recovery feature which allows launching an instance from a snapshot captured at a specific point in time. As source servers are replicated, snapshots are taken over time. The **Point in time (PIT) policy** section allows to configure a retention policy that will determine which snapshots are not required after a defined duration.



The **Tags** section allows you to add custom tags to resources created by AWS Elastic Disaster Recovery in your AWS account.

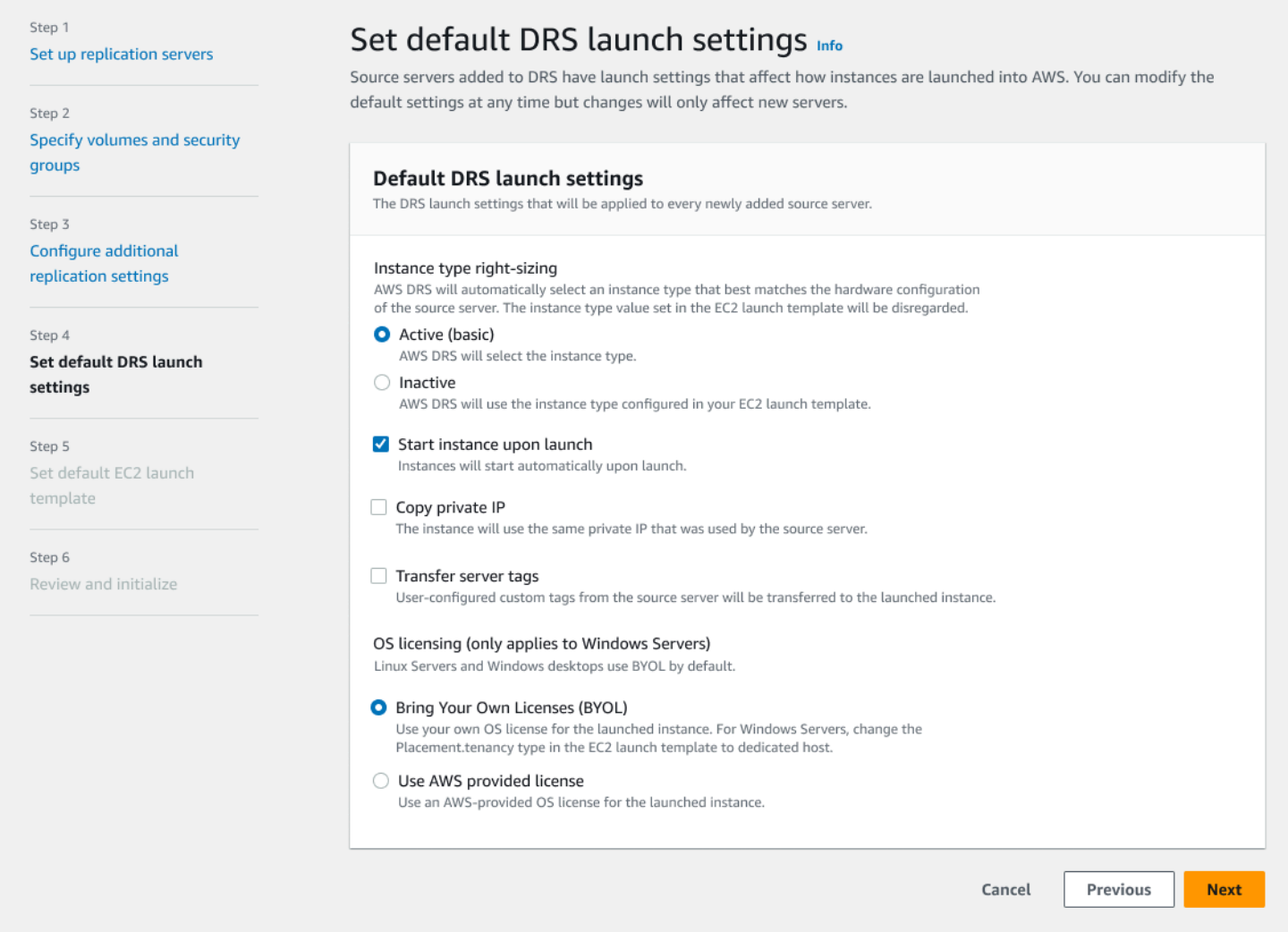


Configurable additional settings include:

- Use private IP for data replication
- Create public IP
- Throttle network bandwidth
- Snapshot retention
- Tags

Click Next to proceed to the fourth page of the wizard: **Set default DRS launch settings**.

Default launch settings define how drill or recovery instances are launched in AWS. You can start using AWS Elastic Disaster Recovery with the default launch settings or configure your own. Learn more about default DRS launch settings.

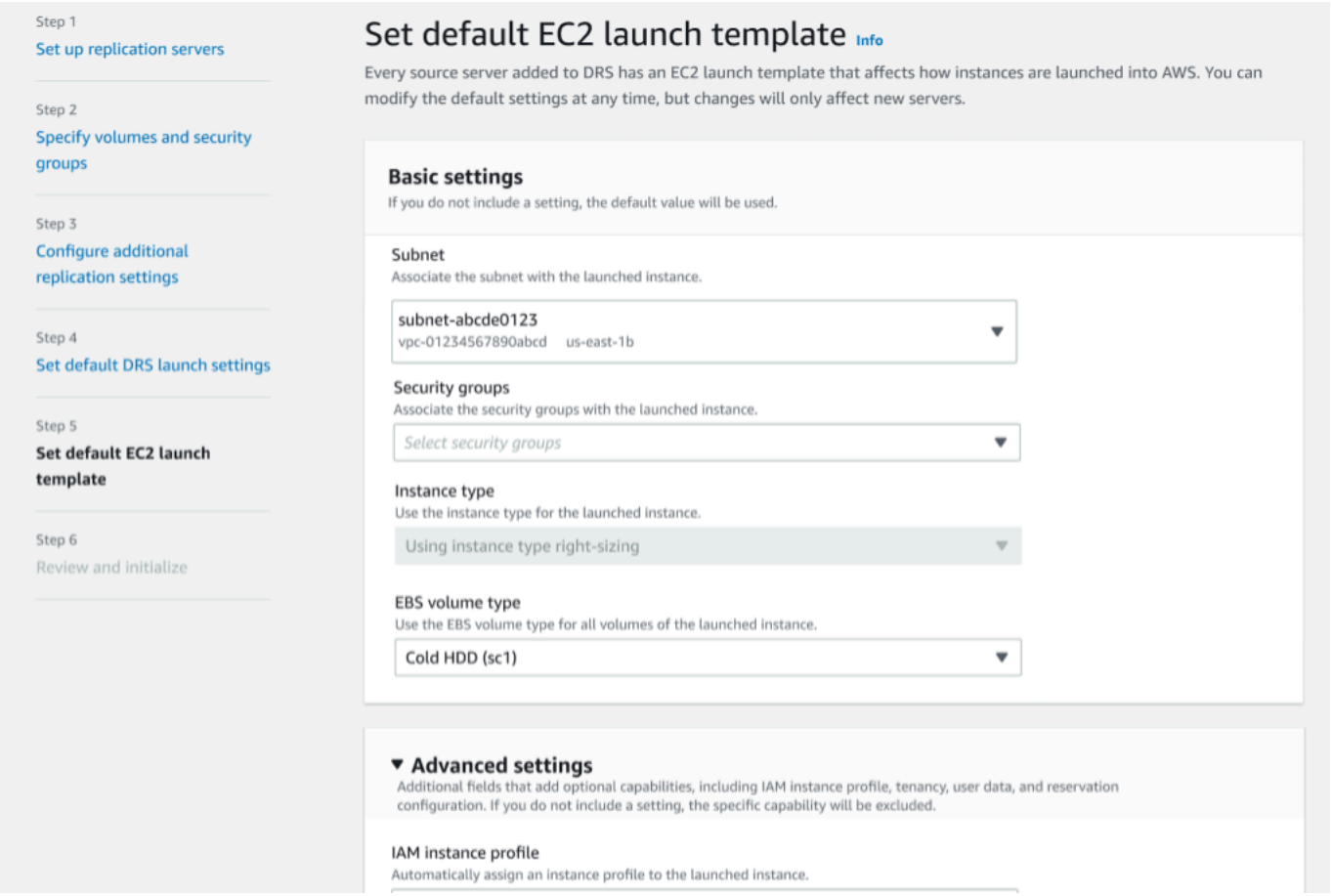


Configurable options include:

- Instance type right sizing
- Start instance upon launch
- Copy private IP
- Transfer server tags
- OS licensing

Click **Next** to proceed to the fifth page of the wizard: **Set default EC2 launch settings**. This page allows you to configure the default EC2 launch template which defines how instances are launched

in AWS. Changes you make to the template will only affect new servers, but you can edit the template for multiple servers according to your preferences. Learn more about default EC2 launch template. The EC2 launch template includes basic and advanced settings.



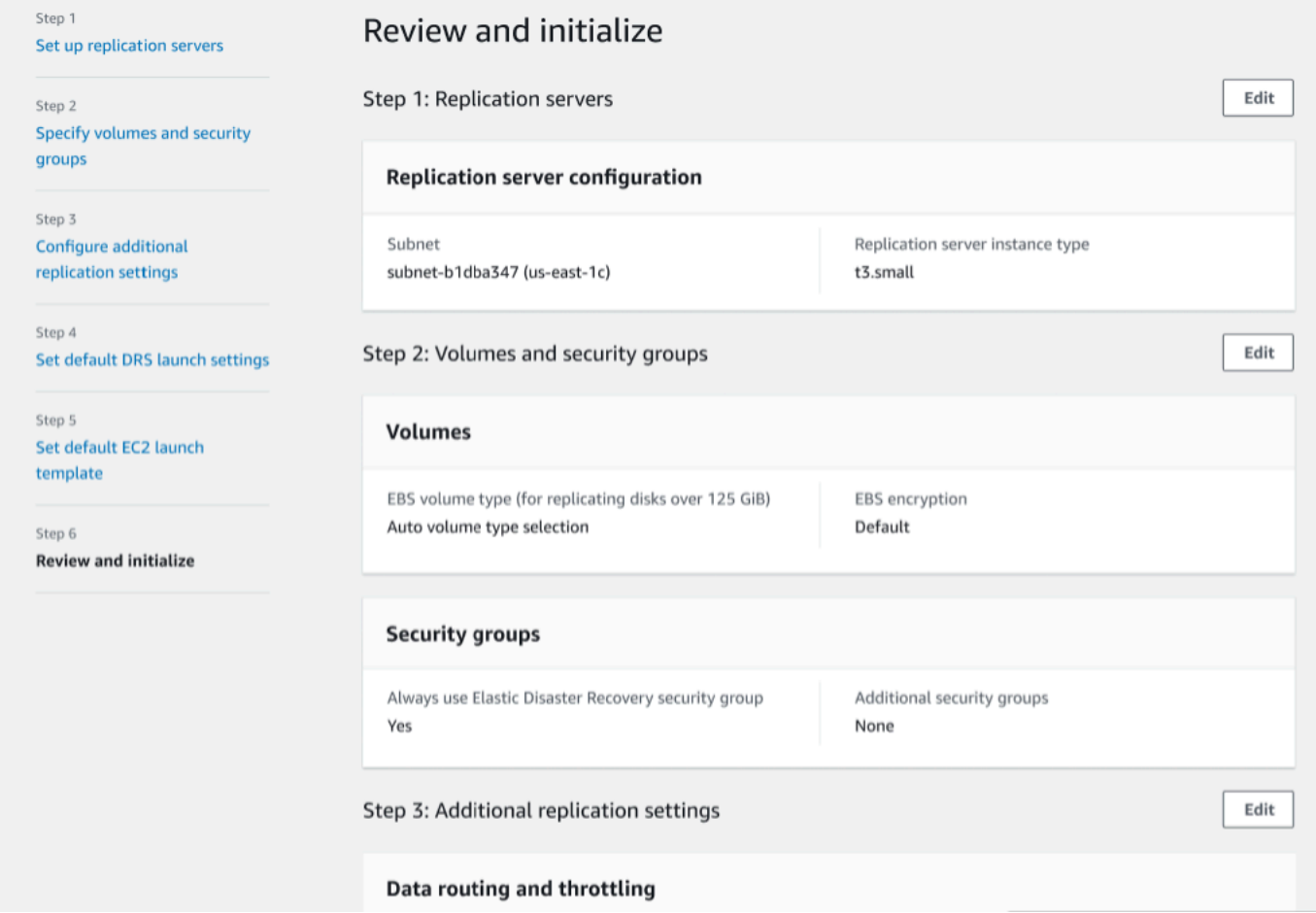
Basic configurable options include:

- Subnet
- Security groups
- Instance type
- EBS volume type

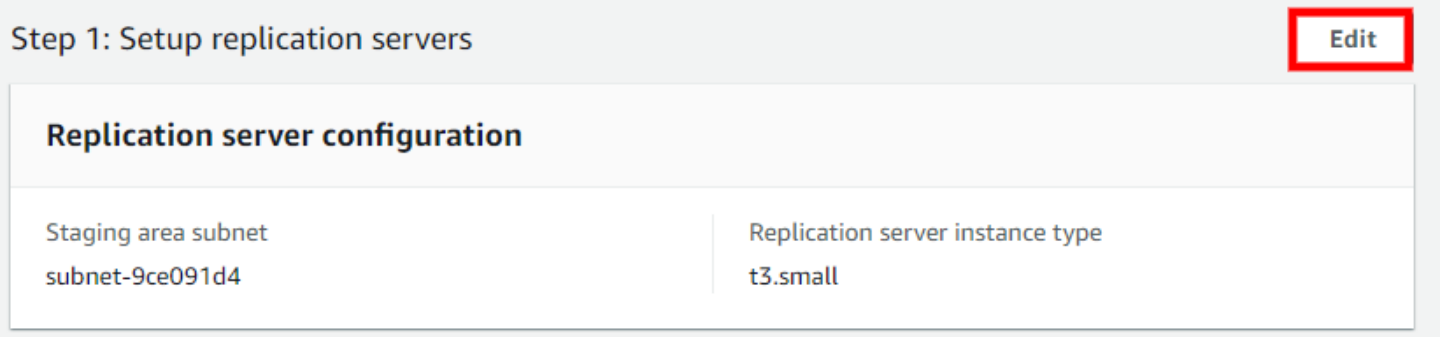
Advance configurable options only need to change in specific operational scenarios. They include:

- IAM instance profile
- Tenancy

Choose **Next** to proceed to the sixth and final page of the wizard, where you will be asked to **Review and initialize**.



Review the settings you configured. To change a specific setting, click **Edit**.



> **Note**
>
> Choosing **Edit** will redirect you to the page in the wizard on which the setting appears. You will then need to go through the remaining pages to return to the **Review and create** page.

Once you have reviewed all of the settings you chose, click **Configure and initialize**.

The default template will be created and you will be redirected to the AWS Elastic Disaster Recovery console.

> **Note**
>
> You can always edit the default replication settings by choosing **Settings** from the left-hand navigation menu. Remember that any new settings changes made will only be applied to newly added servers and not to existing servers.

Adding source servers

Add source servers to AWS Elastic Disaster Recovery by installing the AWS Replication Agent (also referred to as "the Agent") on them. The Agent can be installed on both Linux and Windows servers. Learn more about adding source servers.

Prior to adding your source servers, ensure that you meet all of the Network requirements.

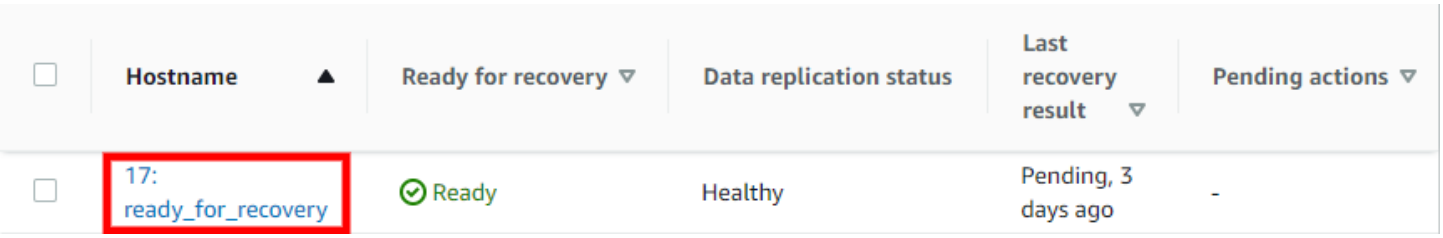
> **Note**
>
> DRS agents can only be installed on instances that are in AWS Regions that are supported by Elastic Disaster Recovery.

Configuring launch settings

After you have added your source servers to the AWS Elastic Disaster Recovery console, you will need to configure the launch settings for each server. The launch settings are a set of instructions that determine how a recovery instance will be launched for each source server on AWS. You must configure the launch settings prior to launching test or recovery instances. You can use the default settings or configure the settings to fit your requirements.

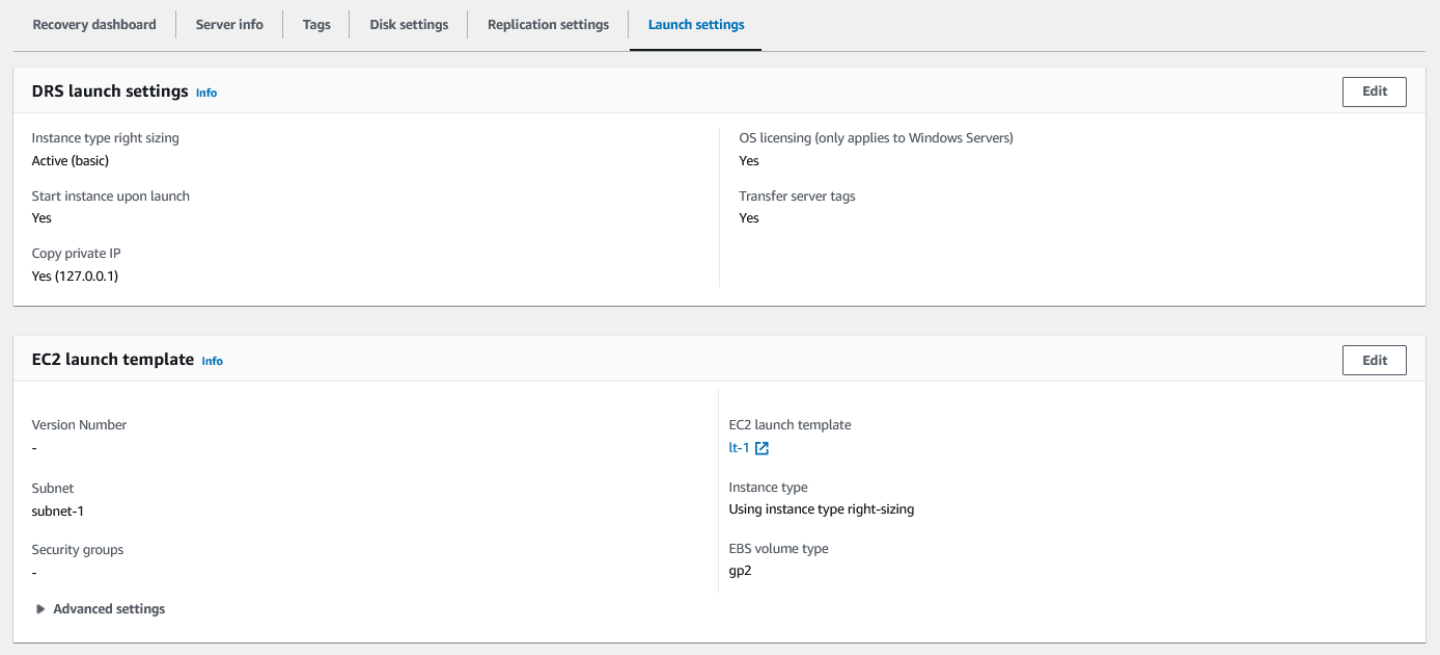
> **Note**
>
> You can change the launch settings after a drill or recovery instance has been launched. You will need to launch a new Drill or Recovery instance for the new settings to take effect.

You can access the launch settings by clicking on the hostname of a source server on the **Source servers** page.



Within the individual server view, navigate to the **Launch settings** tab.

Here you can see your **General launch settings** and your **EC2 launch template**. Click **Edit** to edit your launch settings or your EC2 launch template.



Launch settings are composed of the following:

- **Instance type right-sizing** – The Instance type right-sizing feature allows AWS Elastic Disaster Recovery to launch a drill or recovery instance type that best matches the hardware configuration of the source server. When activated, this feature overrides the instance type selected in the EC2 launch template.
- **Start instance upon launch** – Choose whether you want to start your Initiate recovery job instances automatically upon launch or whether you want to start them manually through the Amazon EC2 Console.

- **Copy private IP** – Choose whether you want AWS Elastic Disaster Recovery to verify that the private IP used by the drill or recovery instance matches the private IP used by the source server.
- **Transfer server tags** – Choose whether you want AWS Elastic Disaster Recovery to transfer any user-configured custom tags from your source servers to your drill or recovery instance.

AWS Elastic Disaster Recovery automatically creates an **EC2 launch template** for each new source server. AWS Elastic Disaster Recovery bases the majority of the instance launch settings on this template. You can edit this template to fit your needs.

Learn more about Launch settings.

Launching a drill instance

After you have added all of your source servers and configured their launch settings, you are ready to launch a drill instance. It is crucial to drill the recovery of your source servers to AWS prior to initiating a recovery in order to verify that your source servers function properly within the AWS environment.

> ⚠ **Important**
>
> - When launching a drill, recovery, or an in-AWS failback, you can launch up to 100 source servers in a single operation. Additional source servers can be launched in subsequent operations.
> - It is a best practice to perform drills regularly. After launching drill instances, use either SSH (Linux) or RDP (Windows) to connect to your instance and ensure that everything is working correctly.

You can drill one source server at a time, or simultaneously drill multiple source servers. For each source server, you will be informed of the success or failure of the drill. You can drill your source server as many times as you want. Each new drill first deletes any previously launched drill or recovery instance and dependent resources. Then, a new Drill instance is launched, which reflects the chosen Point-in-time state of the source server. After the drill, data replication continues as before. The new and modified data on the source server is transferred to the Staging Area Subnet and not to the Recovery instances that were launched during the test.

Note

- Windows source servers need to have at least 2 GB of free space to successfully launch a recovery instance.
- Take into consideration that once a drill instance is launched, actual resources will be used in your AWS account and you will be billed for these resources. You can terminate the operation of launched Recovery instances once you verify that they are working properly without impact in order to data replication.

Learn more about launching drill instances as part of the overall failover and failback framework.

Launching a recovery instance

Once you have finalized the testing of all of your source servers, you are ready for recovery. You should perform the recovery at a set date and time. The recovery will migrate your source servers to the recovery instances on AWS.

You can recover one source server at a time, or simultaneously recover multiple source servers. For each source server, you will be informed of the success or failure of the Recovery. For each new recovery, AWS Elastic Disaster Recovery first deletes any previously launched recovery instance and dependent resources. Then, it launches a new Recovery instance which reflects the most up-to-date state of the source server. After the Recovery, data replication continues as before. The new and modified data on the source server is transferred to the Staging Area Subnet, and not to the recovery instances that were launched during the recovery.

Learn more about launching Recovery instances as part of the overall failover and failback framework.

Performing a failback

Once the disaster is over, you can perform a failback to your original source server or to any otherAWS Elastic Disaster Recovery Failback Client on the server. In order to use the Failback Client, you need to generate Elastic Disaster Recovery-specific credentials. Once the failback is complete, you can opt to either terminate, delete, or disconnect the Recovery instance.

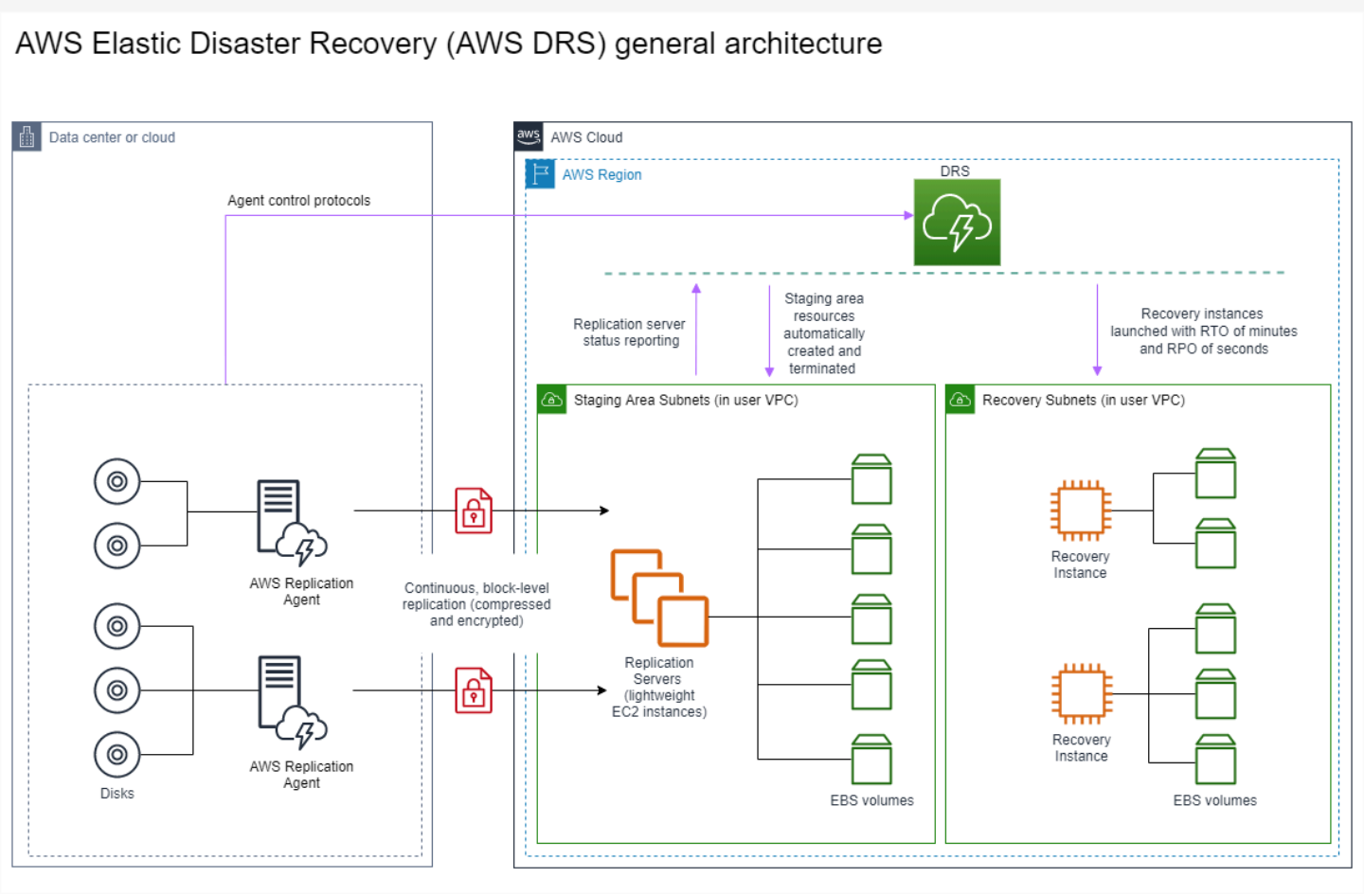
Learn more about performing a failback.

Replication network requirements

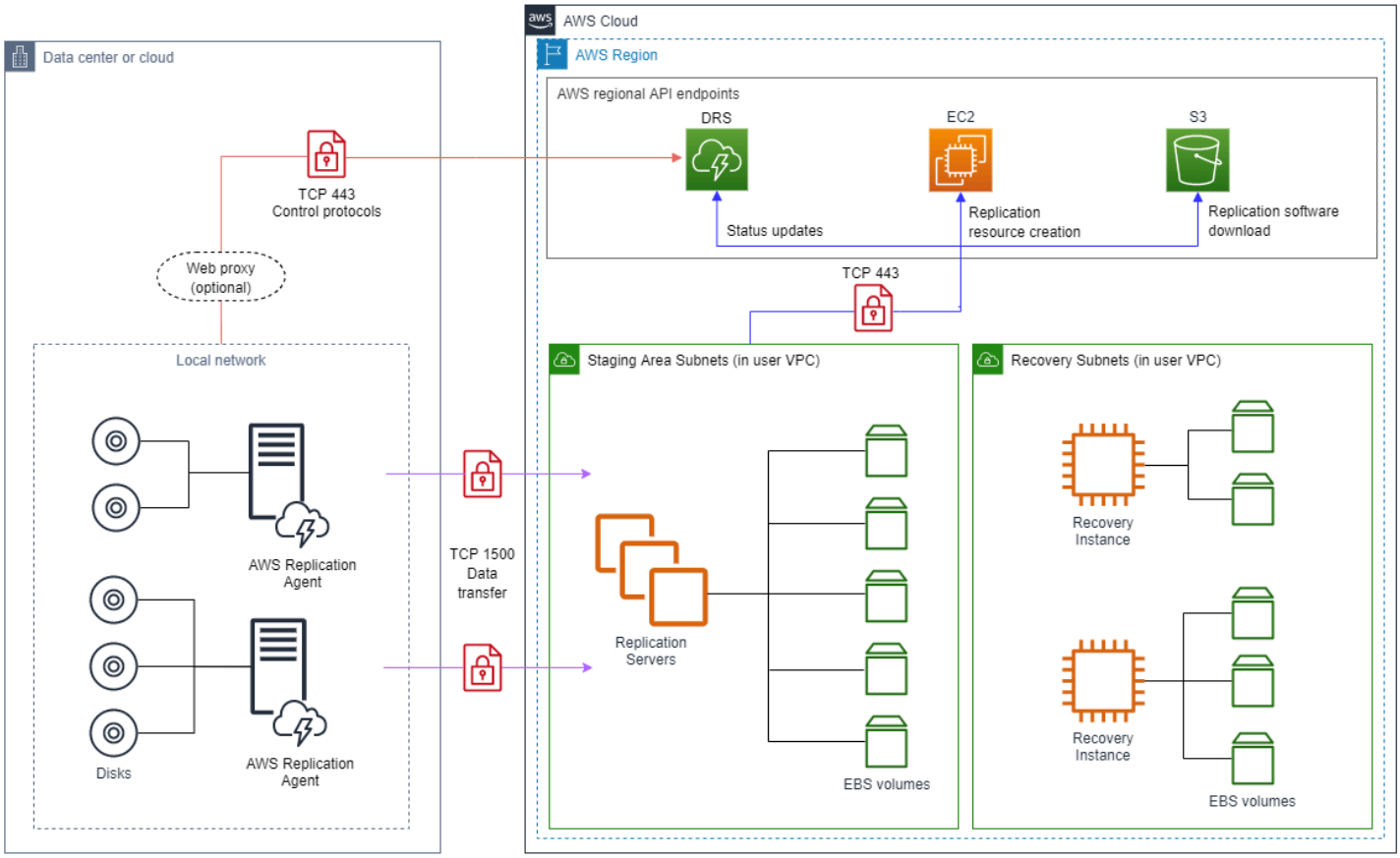
Topics

Network diagrams

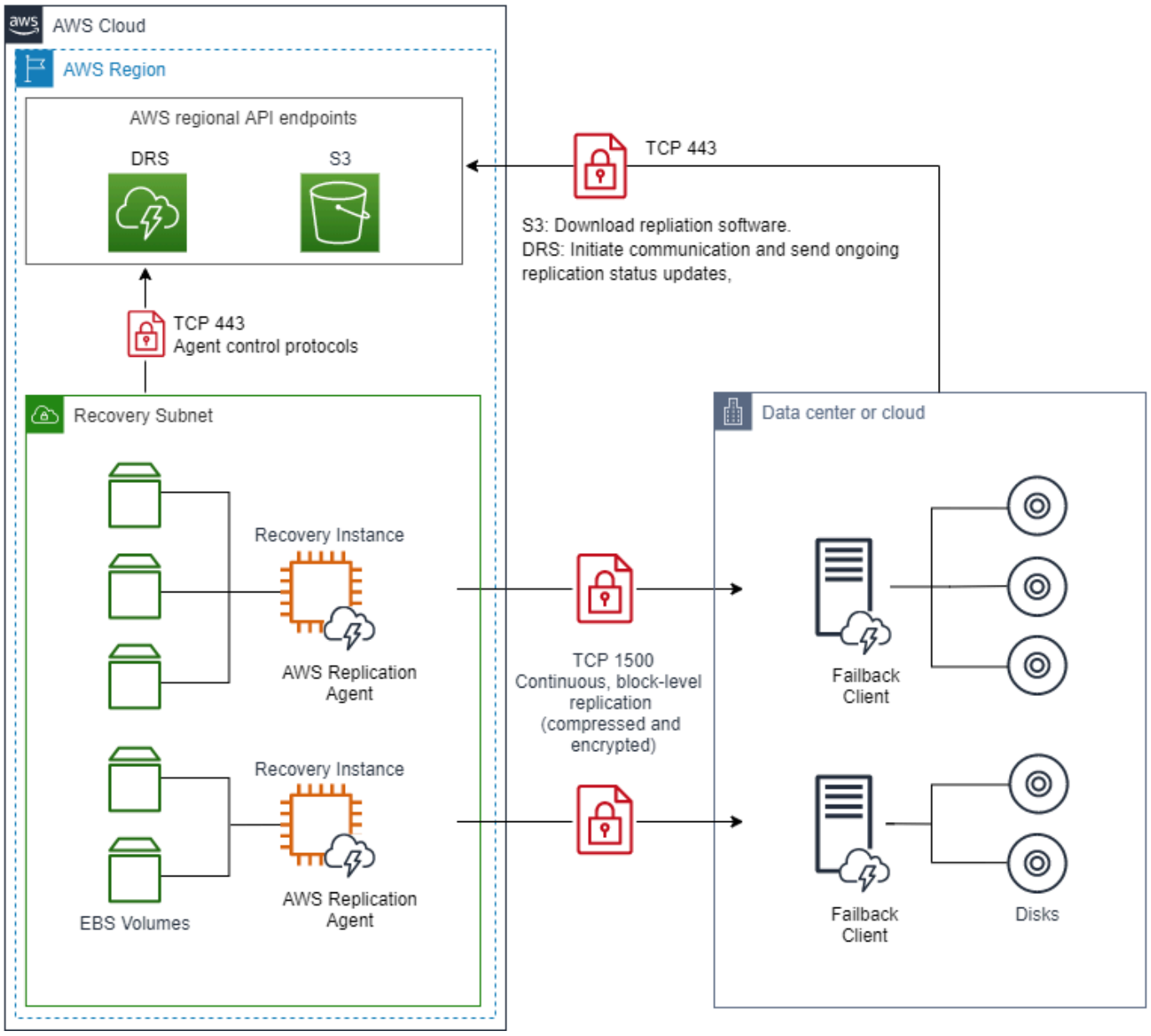
The following are the network diagrams for AWS Elastic Disaster Recovery:



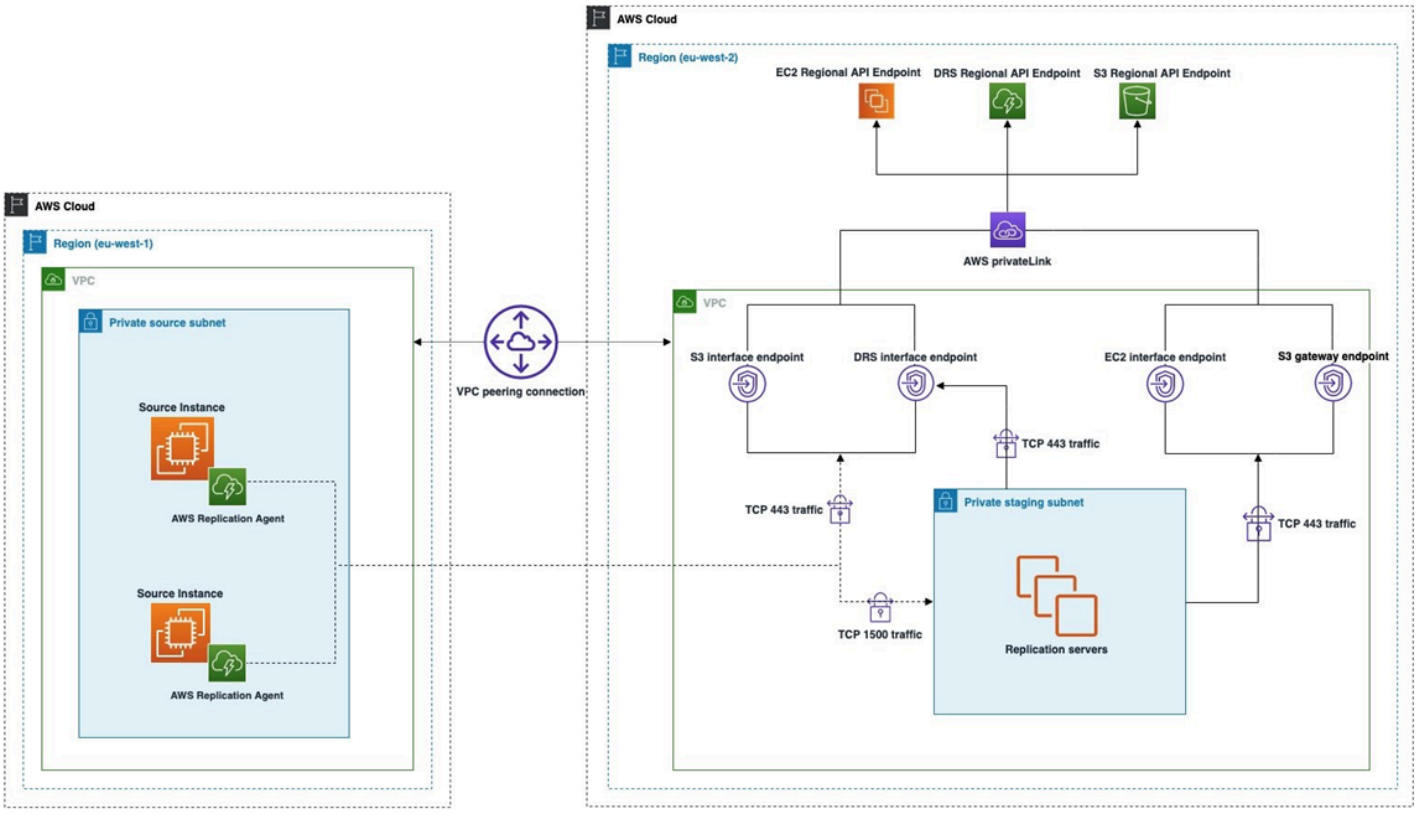
AWS Elastic Disaster Recovery (AWS DRS) network architecture



AWS Elastic Disaster Recovery (AWS DRS) failback replication – architecture and networking



AWS Elastic Disaster Recovery (AWS DRS) communication using a VPN connection



Network setting preparations

Topics

- Staging area subnet
- Network requirements
- Operational subnets

Staging area subnet

Before setting up AWS Elastic Disaster Recovery (AWS DRS), you should create a subnet which will be used by AWS DRS as a staging area for data replicated from your source servers to AWS. You must specify this subnet in the replication template. You can override this subnet for specific source servers in the replication settings. While you can use an existing subnet in your AWS

account, the best practice is to create a new dedicated subnet for this purpose. Learn more about replication settings.

Network requirements

The replication servers launched by AWS Elastic Disaster Recovery in your staging area subnet need to be able to send data over TCP port 443 to the AWS Elastic Disaster Recovery API endpoint at https://drs.{region}.amazonaws.com/. Replace “{region}” with the AWS Region code you are replicating to, for example “us-east-1” .

The source servers on which the AWS Replication Agent is installed need be able to send data over TCP port 1500 to the Replication Servers in the staging area subnet. They also need to be able to send data to AWS DRS's API endpoint at https://drs.{region}.amazonaws.com/. Replace “{region}” with the AWS Region code you are replicating to, for example “us-east-1” .

Operational subnets

Drill and recovery instances are launched in a subnet you specify in the Amazon EC2 launch template associated with each source server. The Amazon EC2 launch template is created automatically when you add a source server to AWS Elastic Disaster Recovery.

Network requirements

To prepare your network for running Elastic Disaster Recovery, you need to set the following connectivity settings:

> **Note**
>
> All communication is encrypted with TLS.

Communication over TCP Port 443:

Topics

- Communication between the source servers and the Staging Area Subnet over TCP port 1500

Communication over TCP Port 1500:

- Between the Source Machines and the staging area Subnet

Communication over TCP port 443

Add the following IP addresses and URLs to your firewall:

The Elastic Disaster Recovery AWS Region-specific Console address:

- (drs.<region>.amazonaws.com *example: drs.eu-west-1.amazonaws.com*)

Amazon S3 service URLs (required for downloading AWS Elastic Disaster Recovery software)

- The AWS Replication Agent installer should have access to the S3 bucket URL of the AWS Region you are using with Elastic Disaster Recovery.
- The staging area subnet should have access to S3.
- The following S3 buckets should be allowed:

```
                        https://aws-drs-clients-<REGION>.s3.<REGION>.amazonaws.com/
                        https://aws-drs-clients-hashes-
<REGION>.s3.<REGION>.amazonaws.com/
                        https://aws-drs-internal-<REGION>.s3.<REGION>.amazonaws.com/
                        https://aws-drs-internal-hashes-
<REGION>.s3.<REGION>.amazonaws.com/
                        https://aws-elastic-disaster-recovery-
<REGION>.s3.<REGION>.amazonaws.com/
                        https://aws-elastic-disaster-recovery-hashes-
<REGION>.s3.<REGION>.amazonaws.com/
```

Note

- Agent installation and replication server components require Amazon S3 bucket for service functionality.
- Ensure the relevant VPC endpoint policy includes access to all the required Amazon S3 buckets.

When using an S3 VPC Endpoint, you must provide sufficient permissions for service functionality. See example policy for replicating to us-east-1:

```
{
 "Version": "2008-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
    "AWS": "*"
   },
   "Action": "s3:GetObject",
   "Resource": [
    "arn:aws:s3:::aws-drs-clients-us-east-1/*",
    "arn:aws:s3:::aws-drs-clients-hashes-us-east-1/*",
    "arn:aws:s3:::aws-drs-internal-us-east-1/*",
    "arn:aws:s3:::aws-drs-internal-hashes-us-east-1/*",
    "arn:aws:s3:::aws-elastic-disaster-recovery-us-east-1/*",
    "arn:aws:s3:::aws-elastic-disaster-recovery-hashes-us-east-1/*"
   ]
  }
 ]
}
```

AWS specific

The staging area subnet requires outbound access to the Amazon EC2 endpoint of its AWS Region.

TCP port 443 is used for two communication routes:

1. Between the source servers and Elastic Disaster Recovery.

2. Between the staging area subnet and AWS Elastic Disaster Recovery.

Communication between the source servers and Elastic Disaster Recovery over TCP port 443

Each source server that is added to AWS Elastic Disaster Recovery (AWS DRS) must continuously communicate with AWS DRS (DRS.<region>.amazonaws.com) over TCP port 443.

The following are the main operations performed through TCP port 443:

- Downloading the AWS Replication Agent on the source servers.
- Upgrading installed Agents.
- Connecting the source servers to the AWS DRS Console and displaying their replication status.
- Monitoring the source servers for internal troubleshooting and the use of resource consumption metrics (such as CPU, RAM).
- Reporting source server-related events (for example, a removal of resizing of a disk).
- Transmit source server-related information to the AWS DRS Console (including hardware information, running services, installed applications and packages, and more).
- Preparing the source servers for drill or recovery.

> **Important**
>
> Make sure that your corporate firewall allows connections over TCP port 443.

Solving communication problems over TCP port 443 between the source servers and AWS Elastic Disaster Recovery

If there is no connection between your source servers and AWS Elastic Disaster Recovery, make sure that your corporate firewall facilitates connectivity from the source servers to AWS Elastic Disaster Recovery over TCP Port 443. If the connectivity is blocked, activate it.

Enabling Windows Firewall for TCP port 443 connectivity

> **Important**
>
> The information provided in this section is for general security and firewall guidance only. The information is provided on "AS IS" basis, with no guarantee of completeness, accuracy

or timeliness, and without warranty or representations of any kind, expressed or implied. In no event will AWS Elastic Disaster Recovery and/or its subsidiaries and/or their employees or service providers be liable to you or anyone else for any decision made or action taken in reliance on the information provided here or for any direct, indirect, consequential, special or similar damages (including any kind of loss), even if advised of the possibility of such damages. AWS Elastic Disaster Recovery is not responsible for the update, validation or support of security and firewall information.

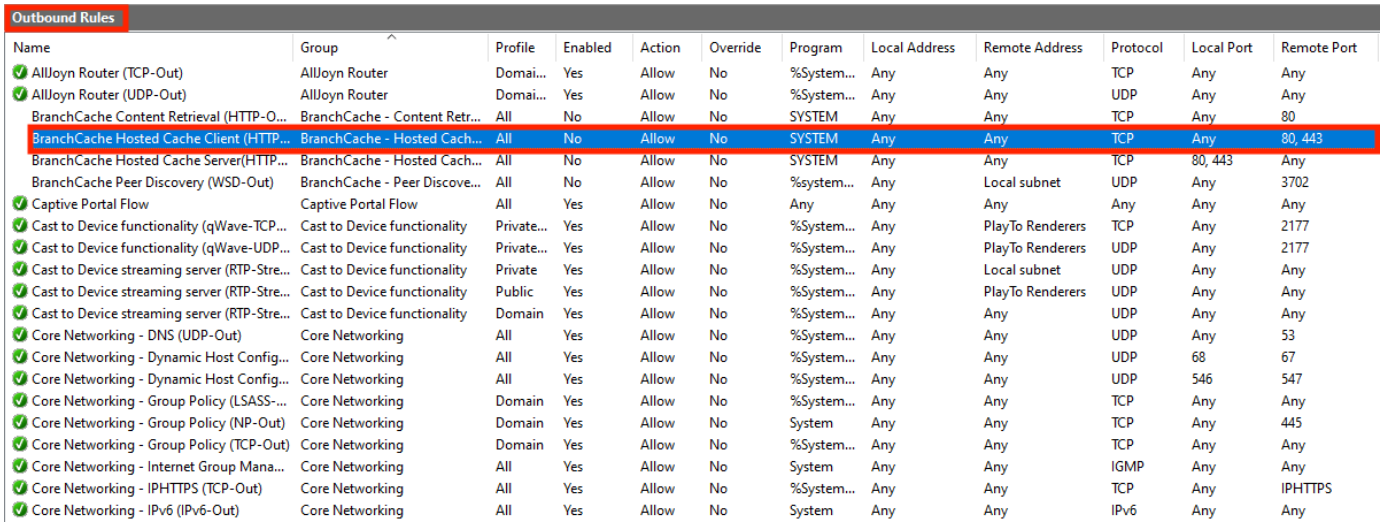
Note

Enabling Windows Firewall for TCP port 443 connectivity will allow your servers to achieve outbound connectivity. You may still need to adjust other external components, such as firewall blocking or incorrect routes, in order to achieve full connectivity.

Note

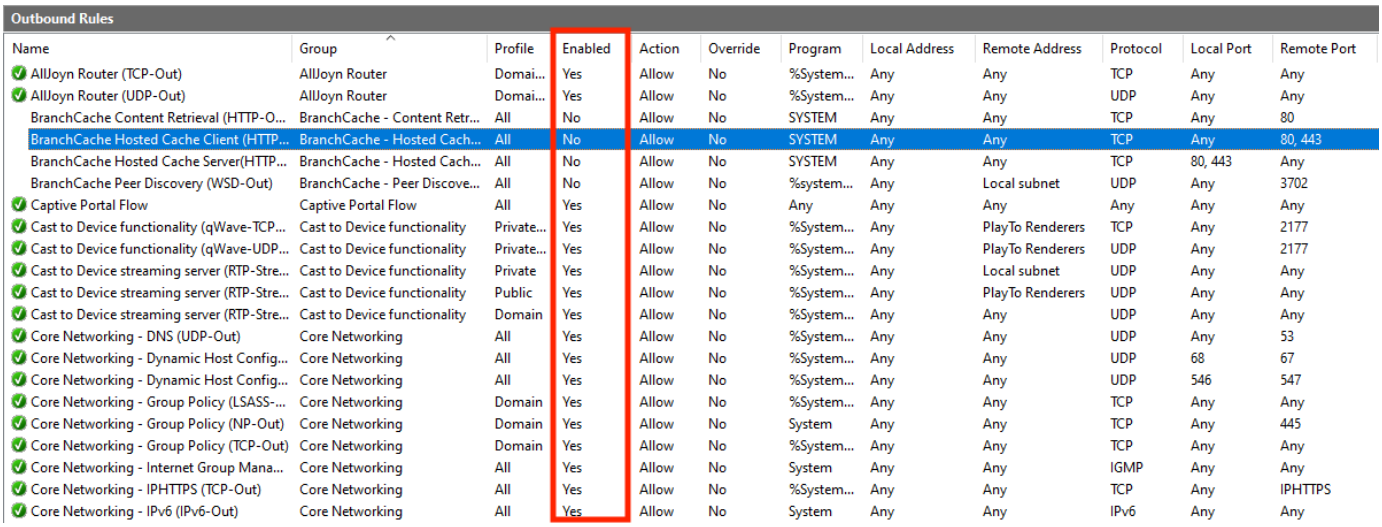
These instructions are intended for the default OS firewall. You will need to consult the documentation of any third-party local firewall you use to learn how to enable TCP port 443 connectivity.

1. On the source server, open the **Windows Firewall** console.
2. On the console, select the **Outbound Rules** option from the tree.



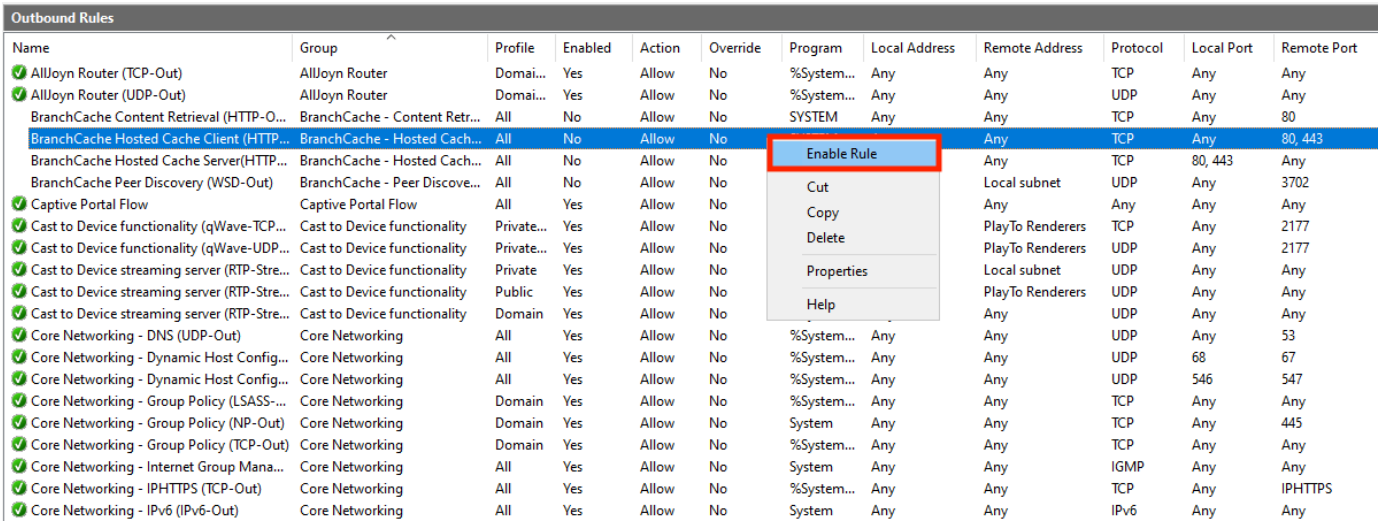
Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port
AllJoyn Router (TCP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
AllJoyn Router (UDP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	UDP	Any	Any
BranchCache Content Retrieval (HTTP-O...	BranchCache - Content Retr...	All	No	Allow	No	SYSTEM	Any	Any	TCP	Any	80
BranchCache Hosted Cache Client (HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No	SYSTEM	Any	Any	TCP	Any	80, 443
BranchCache Hosted Cache Server(HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No	SYSTEM	Any	Any	TCP	80, 443	Any
BranchCache Peer Discovery (WSD-Out)	BranchCache - Peer Discove...	All	No	Allow	No	%system...	Any	Local subnet	UDP	Any	3702
Captive Portal Flow	Captive Portal Flow	All	Yes	Allow	No	Any	Any	Any	Any	Any	Any
Cast to Device functionality (qWave-TCP...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	TCP	Any	2177
Cast to Device functionality (qWave-UDP...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP	Any	2177
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Private	Yes	Allow	No	%System...	Any	Local subnet	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Public	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Domain	Yes	Allow	No	%System...	Any	Any	UDP	Any	Any
Core Networking - DNS (UDP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	Any	53
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	68	67
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	546	547
Core Networking - Group Policy (LSASS-...	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Group Policy (NP-Out)	Core Networking	Domain	Yes	Allow	No	System	Any	Any	TCP	Any	445
Core Networking - Group Policy (TCP-Out)	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Internet Group Mana...	Core Networking	All	Yes	Allow	No	System	Any	Any	IGMP	Any	Any
Core Networking - IPHTTPS (TCP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	TCP	Any	IPHTTPS
Core Networking - IPv6 (IPv6-Out)	Core Networking	All	Yes	Allow	No	System	Any	Any	IPv6	Any	Any

3. On the **Outbound Rules**table, select the rule that relates to the connectivity to Remote Port - 443. Check if the **Enabled** status is **Yes**.



Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port
AllJoyn Router (TCP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
AllJoyn Router (UDP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	UDP	Any	Any
BranchCache Content Retrieval (HTTP-O...	BranchCache - Content Retr...	All	No	Allow	No	SYSTEM	Any	Any	TCP	Any	80
BranchCache Hosted Cache Client (HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No	SYSTEM	Any	Any	TCP	Any	80, 443
BranchCache Hosted Cache Server(HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No	SYSTEM	Any	Any	TCP	80, 443	Any
BranchCache Peer Discovery (WSD-Out)	BranchCache - Peer Discove...	All	No	Allow	No	%system...	Any	Local subnet	UDP	Any	3702
Captive Portal Flow	Captive Portal Flow	All	Yes	Allow	No	Any	Any	Any	Any	Any	Any
Cast to Device functionality (qWave-TCP...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	TCP	Any	2177
Cast to Device functionality (qWave-UDP...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP	Any	2177
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Private	Yes	Allow	No	%System...	Any	Local subnet	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Public	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Domain	Yes	Allow	No	%System...	Any	Any	UDP	Any	Any
Core Networking - DNS (UDP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	Any	53
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	68	67
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	546	547
Core Networking - Group Policy (LSASS-...	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Group Policy (NP-Out)	Core Networking	Domain	Yes	Allow	No	System	Any	Any	TCP	Any	445
Core Networking - Group Policy (TCP-Out)	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Internet Group Mana...	Core Networking	All	Yes	Allow	No	System	Any	Any	IGMP	Any	Any
Core Networking - IPHTTPS (TCP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	TCP	Any	IPHTTPS
Core Networking - IPv6 (IPv6-Out)	Core Networking	All	Yes	Allow	No	System	Any	Any	IPv6	Any	Any

4. If the Enabled status of the rule is **No**, right-click it and select **Enable Rule** from the pop-up menu.



Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port
AllJoyn Router (TCP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
AllJoyn Router (UDP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	UDP	Any	Any
BranchCache Content Retrieval (HTTP-O...	BranchCache - Content Retr...	All	No	Allow	No	SYSTEM	Any	Any	TCP	Any	80
BranchCache Hosted Cache Client (HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No			Any	TCP	Any	80, 443
BranchCache Hosted Cache Server(HTTP...	BranchCache - Hosted Cach...	All	No	Allow	No			Any	TCP	80, 443	Any
BranchCache Peer Discovery (WSD-Out)	BranchCache - Peer Discove...	All	No	Allow	No			Local subnet	UDP	Any	3702
Captive Portal Flow	Captive Portal Flow	All	Yes	Allow	No			Any	Any	Any	Any
Cast to Device functionality (qWave-TCP...	Cast to Device functionality	Private...	Yes	Allow	No			PlayTo Renderers	TCP	Any	2177
Cast to Device functionality (qWave-UDP...	Cast to Device functionality	Private...	Yes	Allow	No			PlayTo Renderers	UDP	Any	2177
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Private	Yes	Allow	No			Local subnet	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Public	Yes	Allow	No			PlayTo Renderers	UDP	Any	Any
Cast to Device streaming server (RTP-Stre...	Cast to Device functionality	Domain	Yes	Allow	No			Any	UDP	Any	Any
Core Networking - DNS (UDP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	Any	53
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	68	67
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow	No	%System...	Any	Any	UDP	546	547
Core Networking - Group Policy (LSASS-...	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Group Policy (NP-Out)	Core Networking	Domain	Yes	Allow	No	System	Any	Any	TCP	Any	445
Core Networking - Group Policy (TCP-Out)	Core Networking	Domain	Yes	Allow	No	%System...	Any	Any	TCP	Any	Any
Core Networking - Internet Group Mana...	Core Networking	All	Yes	Allow	No	System	Any	Any	IGMP	Any	Any
Core Networking - IPHTTPS (TCP-Out)	Core Networking	All	Yes	Allow	No	%System...	Any	Any	TCP	Any	IPHTTPS
Core Networking - IPv6 (IPv6-Out)	Core Networking	All	Yes	Allow	No	System	Any	Any	IPv6	Any	Any

Enabling Linux Firewall for TCP port 443 connectivity

1. Enter the following command to add the required Firewall rule:

   sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT

2. To verify the creation of the Firewall rule, enter the following commands:

   sudo iptables -L

Chain INPUT (policy ACCEPT)

target prot opt source destination

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

ACCEPT tcp -- anywhere anywhere tcp dpt:443

Communication between the staging area subnet and AWS Elastic Disaster Recovery over TCP port 443

The replication servers in the staging area subnet must continuously communicate with Elastic Disaster Recovery over TCP port 443. The main operations that are performed through this route are:

- Downloading the replication software by the replication servers.
- Connecting the replication servers to AWS Elastic Disaster Recovery, and displaying their replication status.
- Monitoring the replication servers for internal troubleshooting use and resource consumption metrics (such as CPU, RAM).
- Reporting replication-related events.

> **Note**
>
> The staging area subnet requires S3 access.

Configuring communication over TCP port 443 between the staging area subnet and AWS Elastic Disaster Recovery

You can establish communication between the staging area subnet and AWS Elastic Disaster Recovery over TCP port 443 directly.

There are two ways to establish direct connectivity to the Internet for the VPC of the staging area, as described in the VPC FAQ.

1. Public IP address + Internet gateway

2. Private IP address + NAT instance

Communication between the source servers and the Staging Area Subnet over TCP port 1500

Each source server with an installed AWS Replication Agent continuously communicates with the AWS Elastic Disaster Recovery replication servers in the staging area subnet over TCP port 1500. TCP port 1500 is needed for the transfer of replicated data from the source servers to the staging area subnet.

The replicated data is encrypted and compressed when transferred over TCP port 1500. Prior to being moved into the Staging Area Subnet, the data is encrypted on the source infrastructure. The data is decrypted after it arrives at the staging area subnet and before it is written to the volumes.

TCP port 1500 is primarily used for the replication server data replication stream.

Elastic Disaster Recovery uses TLS 1.2 end to end from the agent installed on the source server to the replication server. Each replication server gets assigned a specific TLS server certificate, which is distributed to the corresponding agent and validated against on the agent side.

Establishing communication over TCP port 1500

> **Important**
>
> To allow traffic over TCP port 1500, make sure that your corporate firewall enables this connectivity.

Required bandwidth between the source servers and the staging area subnet

Replicated data is transferred from the source servers to the staging area over the network. For replication to succeed, your average network bandwidth must be higher than the write rate on the source servers. If you attempt to conduct a replication of a write intensive source server under low bandwidth conditions, it will likely lag.

Settings

AWS Elastic Disaster Recovery (AWS DRS) manages default settings that apply to newly added source servers. These settings define the overall behavior of source servers. For example, default replication settings define how data will be replicated from newly added source servers to AWS.

You must configure the default replication and launch settings upon first use of AWS DRS. The replication settings that are first configured determine how your servers will be replicated to AWS through a variety of settings, including replication server instance type, Amazon EBS volume type, Amazon EBS encryption, security groups, data routing, and tags. The settings configured in the replication settings are automatically passed down to every new server you add to AWS DRS. The launch settings that are configured determine how servers will be launched and include multiple parameters both for the DRS launch settings and the EC2 template.

Once you have configured your default settings, you can make changes to individual servers or a group of servers from the source servers page. Learn more about editing the launch settings for single or multiple servers.

Topics

Replication settings

Replication settings determine how data will be replicated from your source servers to AWS. Your replication settings are governed by the default replication settings, which you must configure before adding your source servers to AWS Elastic Disaster Recovery. Once configured, those settings will be used for every newly added server. You can edit the default settings at any point.

You can always choose to edit the replication settings for each server or group of servers after they've been added to AWS Elastic Disaster Recovery.

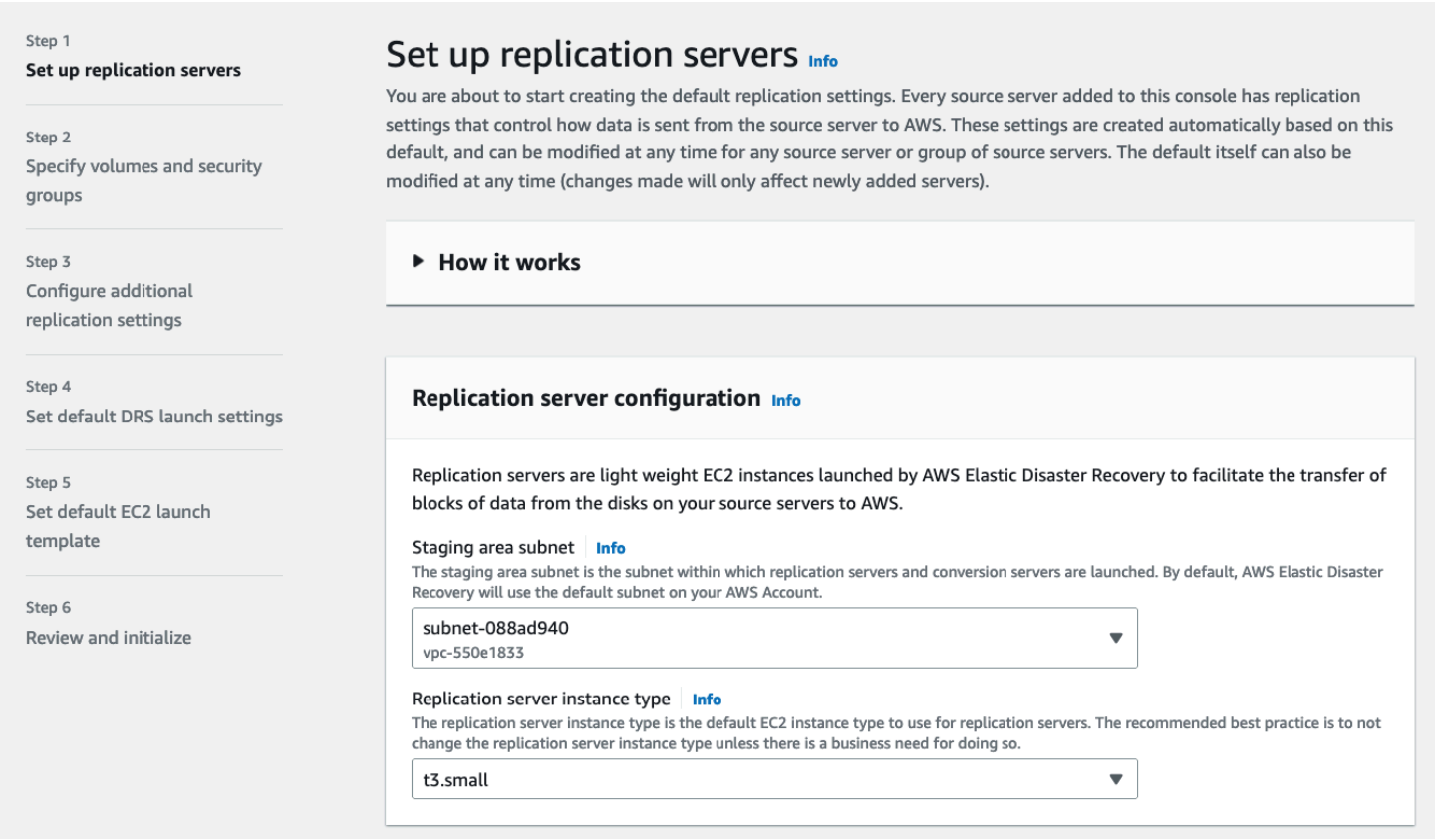
In addition, you can control a variety of other source server settings through the **Settings** tab, including **Tags**.

After you edit your settings, be sure to click **Save replication settings** to finalized your changes,.

Topics

Default replication settings vs individual server replication settings

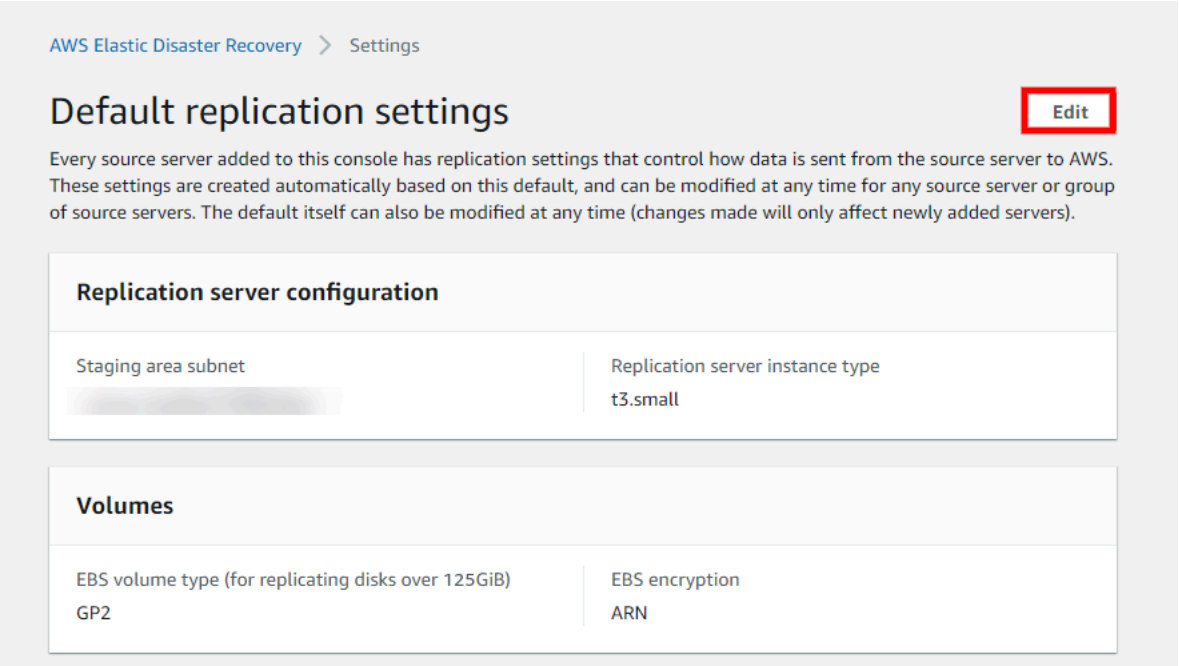
The default replication settings determine how data replication will work for each new server you add to AWS Elastic Disaster Recovery. These settings will be applied to each newly added source server. You will be prompted to configure your default replication settings upon your first use of AWS Elastic Disaster Recovery.



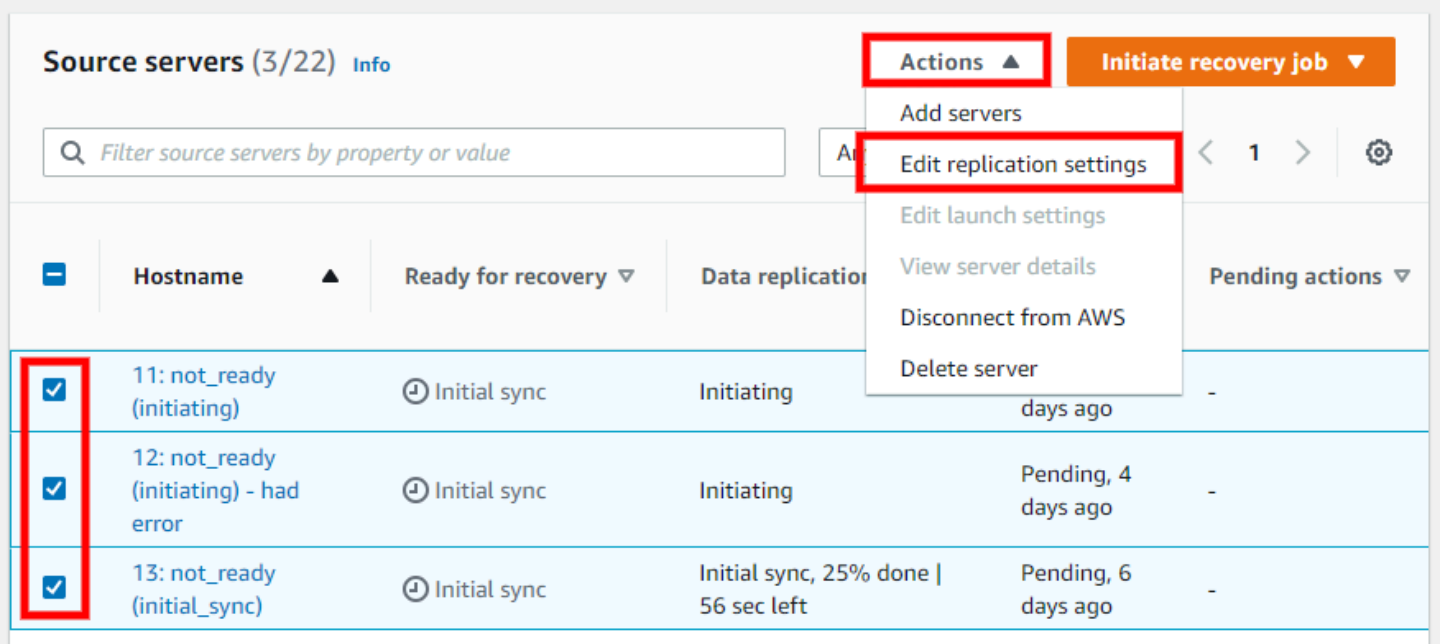
The configured replication settings can later be changed at any time, for individual source servers or for a group of source servers. The changes made will only affect the server or group of servers selected and will not affect the default replication settings. Learn more about configuring your default replication settings.

To edit the replication settings for your entire account, you will need to edit your default replication settings. Choose **Default replication** from the from the left-hand navigation menu (under **Settings**).

This will open the **Settings: default replication** view. Choose **Edit** to edit your account-wide replication settings. These settings changes will be applied to each newly added server but will not affect servers that have already been added to AWS Elastic Disaster Recovery.



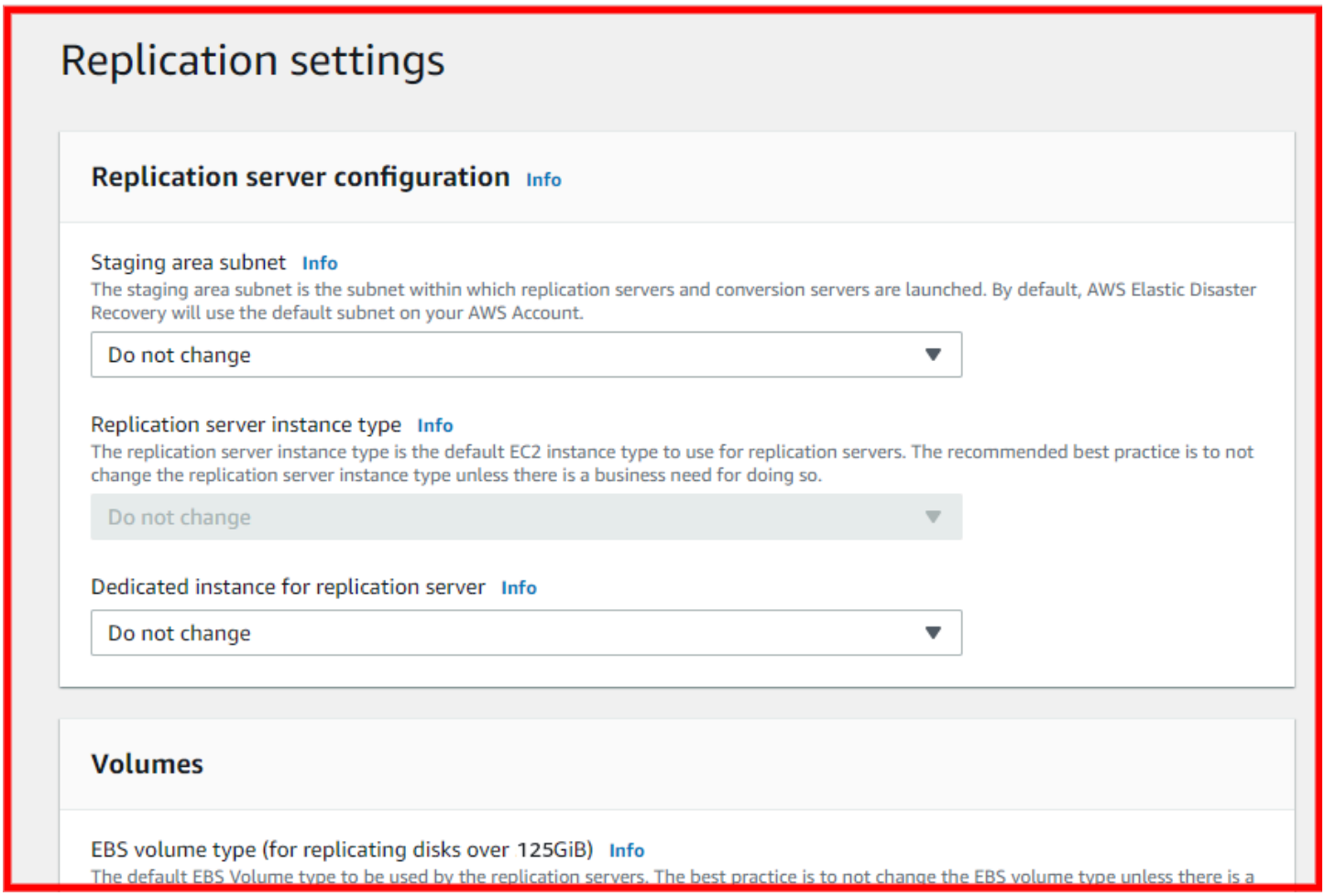
To edit the settings for an individual server or group of servers, select the box to the left of each server name on the **Source servers** page. Open the **Actions** menu and choose **Edit replication settings**.



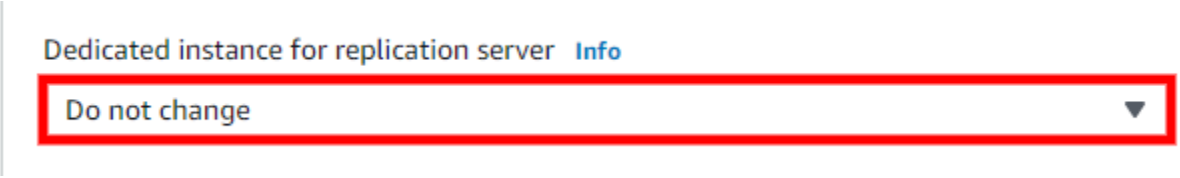
You will be redirected to the **Edit replication settings** tab.

The names of the servers for which you are editing the replication settings will appear under the **Selected servers** drop-down menu.

You can edit individual replication settings under the **Replication settings** category.



If you want to choose different settings for selected servers than those set in the default replication settings, edit these settings individually. Any setting that has not been changed is labeled with the **Do not change** option.



For any setting that you want to change, choose the setting option from the drop-down menu under each setting category.

Click **Save replication settings** to save your changes.

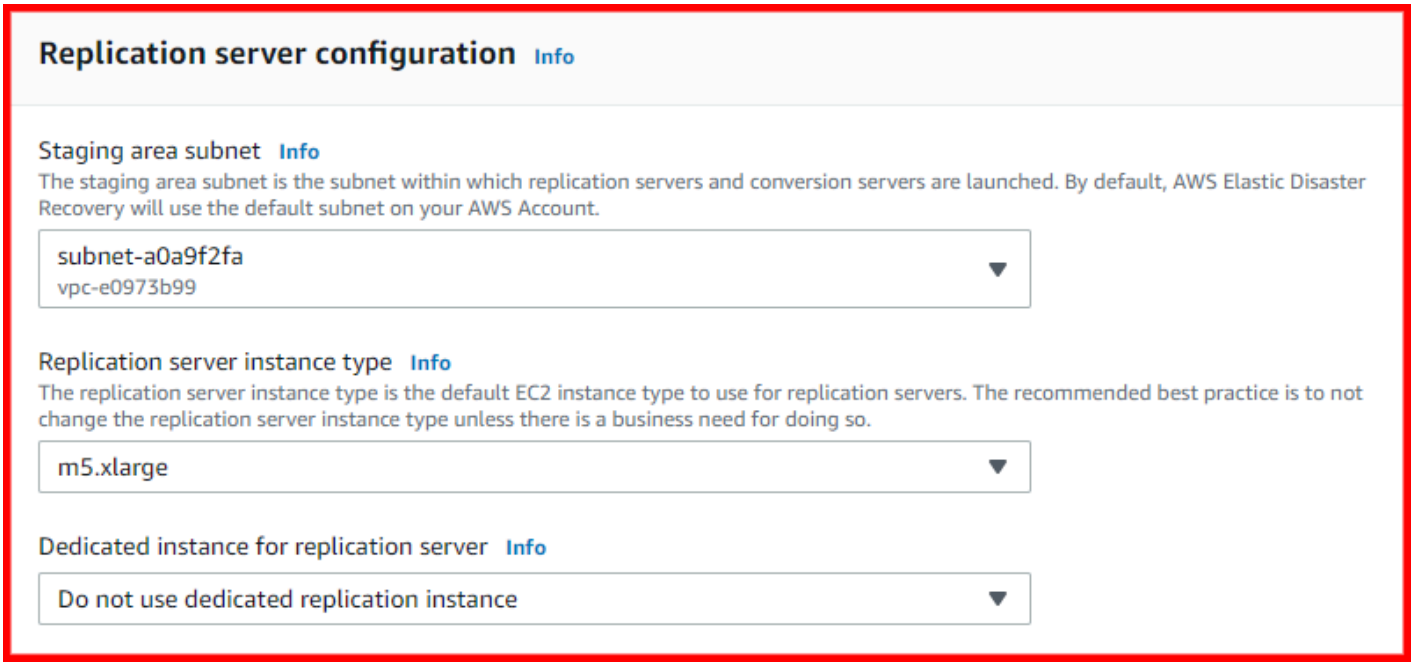
The individual replication settings categories are explained in the following sections.

Replication server configuration

Replication servers are lightweight Amazon EC2 instances that are used to replicate data between your source servers and AWS. Replication servers are automatically launched and terminated as needed. You can modify the behavior of the replication servers by modifying the settings for a single source server or multiple source servers. Alternatively, you can run AWS Elastic Disaster Recovery with the default replication server settings.

You can configure a variety of replication server options, including:

- The subnet within which the replication server will be launched
- Replication server instance type
- Whether a dedicated instance is used for the replication server



> **Note**
>
> Replication servers are only supported on x86_64 CPU architecture instance types.

Staging area subnet

Choose the **Staging area subnet** that you want to allocate as the staging area subnet for all of your replication servers.

The best practice is to create a single dedicated, separate subnet for all of your recovery waves using your AWS Account. Learn more about creating subnets in this AWS VPC article.

The staging area subnet is the subnet within which replication servers subnet in this AWS VPC article.

If a default subnet does not exist, select a specific subnet. The drop-down menu contains a list of all subnets that are available in the current AWS Region.

> **Note**
>
> Changing the subnet does not significantly interfere with ongoing data replication, although there may be a minor delay of several minutes while the servers are moved from one subnet to another.

Using multiple subnets

The best practice is to use a single staging area subnet for all of your recovery waves within a single AWS Account. You may want to use multiple subnets in certain cases, such as the recovery of thousands of servers.

> **Note**
>
> Using more than one staging area subnet might result in higher compute consumption as more replication servers will be needed.

Launching replication servers in Availability Zones

If you want your replication servers to be launched in a specific Availability Zone, then select or create a subnet in that specific Availability Zone. Learn more about using Availability Zones in this Amazon EC2 article.

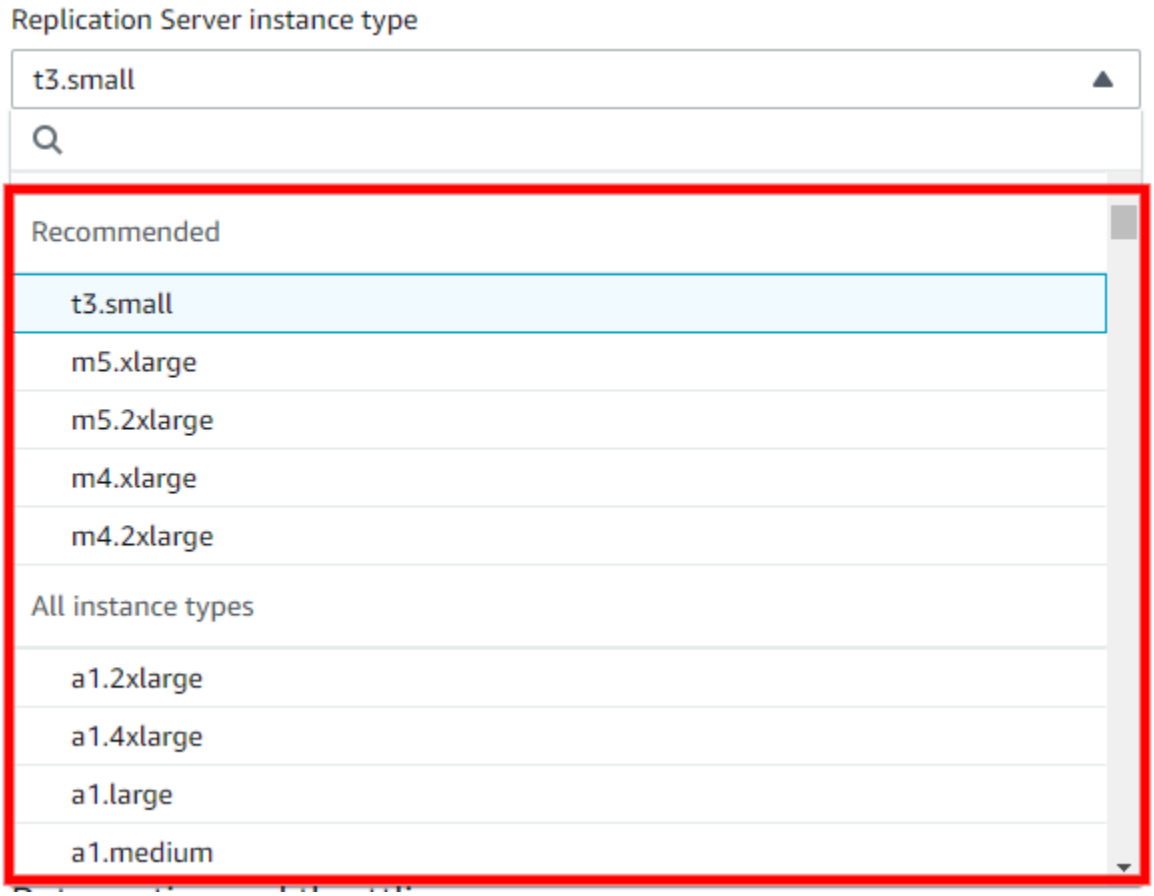
Replication server instance type

Choose the **Replication server instance type**. This will determine the instance type and size that will be used for the launch of each replication server.

The best practice is to not change the default replication server instance type unless there is a business need for doing so.

By default, AWS Elastic Disaster Recovery utilizes the t3.small instance type. This is the most cost effective instance type and should work well for most common workloads. You can change the replication server instance type to speed up the initial sync of data from your source servers to AWS. Changing the instance type will likely lead to increased compute costs.

You can change the **Replication server instance** type to any type you wish. The drop-down menu contains all available instance types. Recommended and commonly used instance types are displayed first.



You can search for a specific instance type within the search box.

The replication server instance type can be changed for servers that are replicating too slowly or servers that are constantly busy or experience frequent spikes. These are the most common instance type changes that are made:

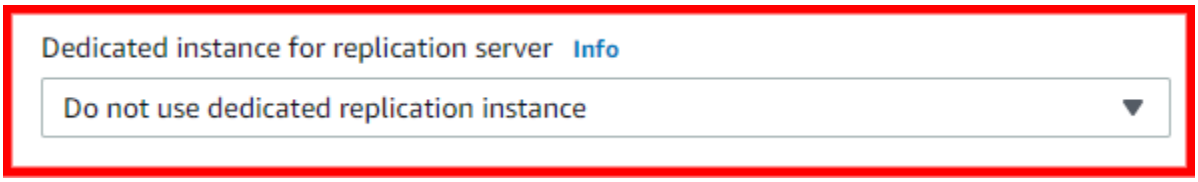
- Servers with less than 26 disks – Change the instance type to m5.large. Increase the instance type to m5.xl or higher as needed.
- Servers with more than 26 disks (or servers in AWS Regions that do not support m5 instance types) – Change the instance type to m4.large. Increase to m4.xlarge or higher, as needed.

Note

- Changing the replication server instance type will not affect data replication. Data replication will automatically continue from where it left off, using the new instance type you selected.
- By default, replication servers are automatically assigned a public IP address from Amazon's public IP space.

Dedicated instance for replication server

Choose whether you would like to use a **Dedicated instance for replication server**.



When an external server is very write-intensive, the replication of data from its disks to a shared replication server can interfere with the data replication of other servers. In these cases you should choose the **Use dedicated replication server** option (and also consider changing Replication server instance type).

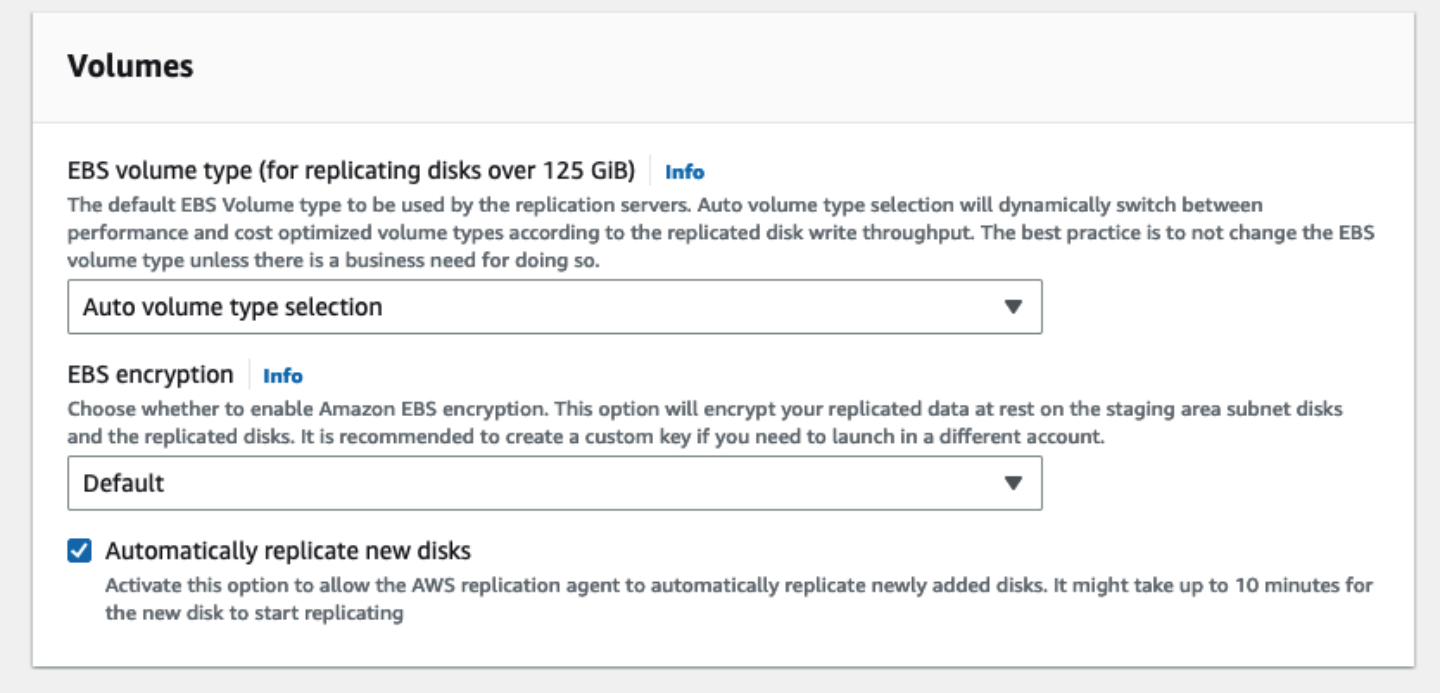
Otherwise, choose the **Do not use dedicated replication server** option.

Note

Using a dedicated replication server may increase the EC2 cost you incur during replication.

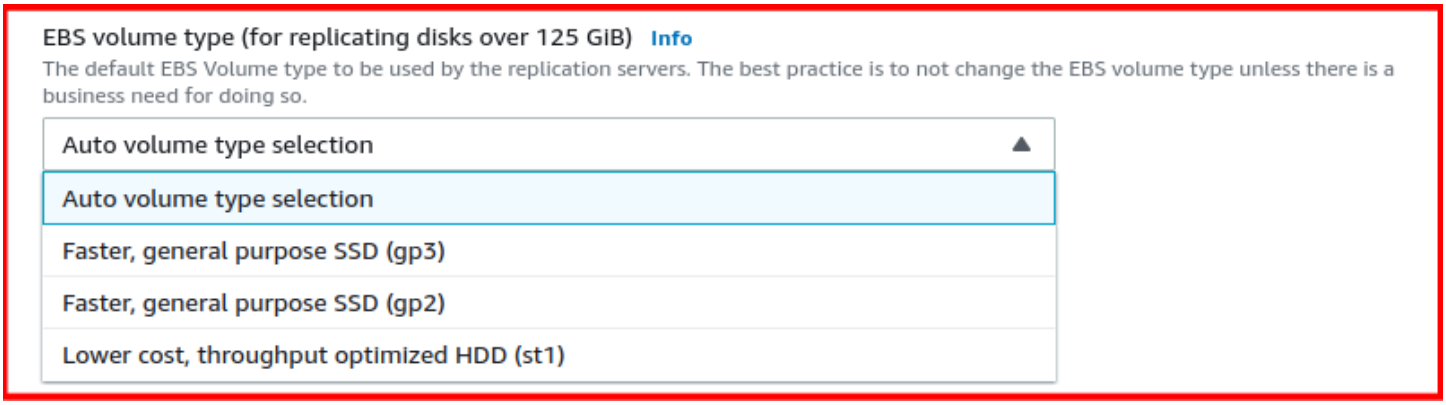
Volumes

Determine the default Amazon EBS volume type to be used by the replication servers and whether to use Amazon EBS encryption.



Amazon EBS volume type

Choose the default **Amazon EBS volume type** to be used by the replication servers for large disks.



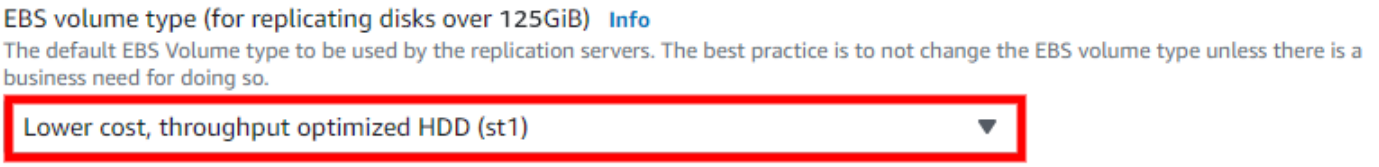
When choosing **Auto volume type selection** the service will dynamically switch between performance/cost optimized volume type according to the replicated disk write throughput.

Each disk has minimum and maximum sizes and varying performance metrics and pricing. Learn more about Amazon EBS volume types in this Amazon EBS article.

The best practice is to not change the default **Auto volume type selection** volume type, unless there is a business need for doing so.

> **Note**
>
> This option only affects disks over 125 GiB (by default, smaller disks always use Magnetic HDD volumes).

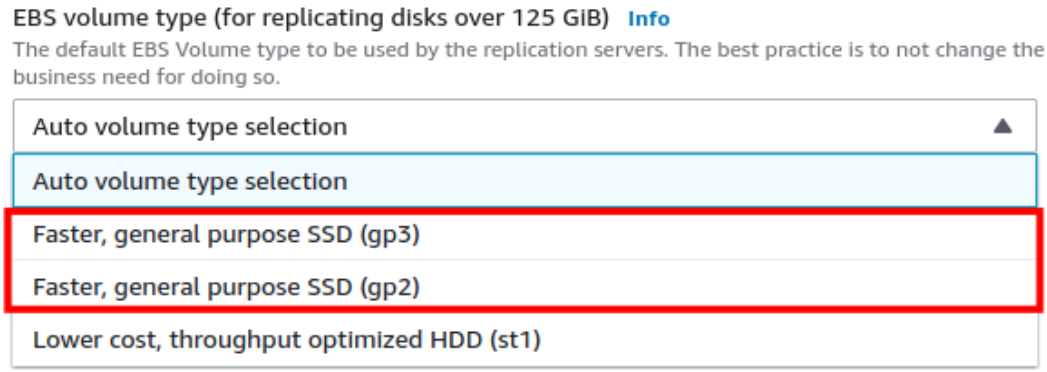
The default **Lower cost, Throughput Optimized HDD (st1)** option utilizes slower, less expensive disks.



You may want to use this option if:

- You want to keep costs low
- Your large disks do not change frequently
- You are not concerned with how long the Initial Sync process will take

The **Faster, General Purpose SSD (gp2)** and Faster, **General Purpose SSD (gp3)** options utilizes faster, but more expensive disks.



You may want to use this option if:

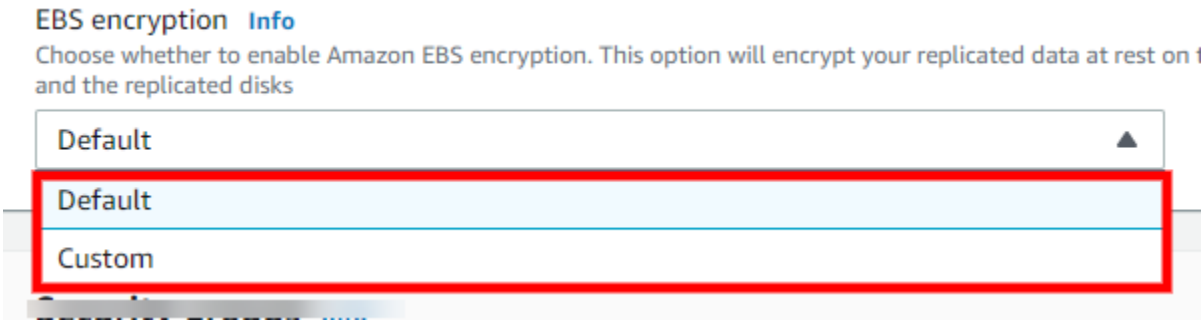
- Your source server has disks with a high write rate or if you want faster performance in general
- You want to speed up the initial sync process
- You are willing to pay more for speed

> **Note**
>
> You can customize the Amazon EBS volume type used by each disk within each source server in that source server's settings. Learn more about changing individual source server volume types.

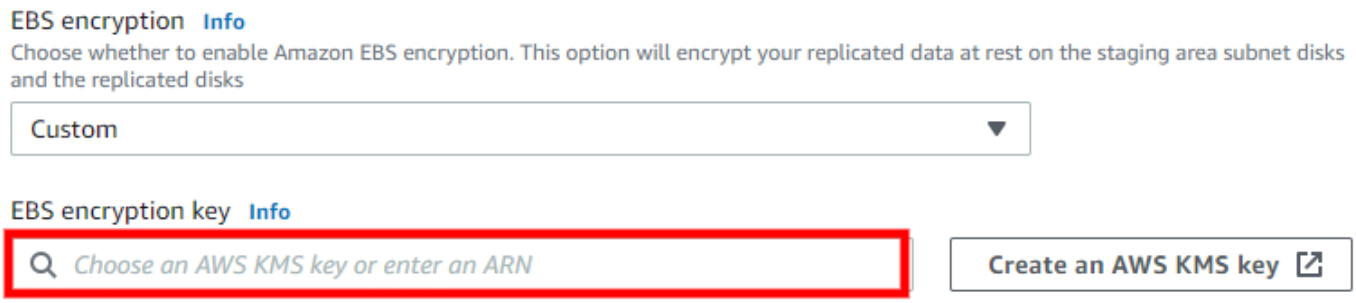
Amazon EBS encryption

Choose whether to use the default or custom Amazon **EBS encryption**, or select none to have no encryption. The option none, will not encrypt the replicated data at rest on staging area subnet disks and replicated disks. The options default or custom will encrypt your replicated data at rest on the staging area subnet disks and the replicated disks.

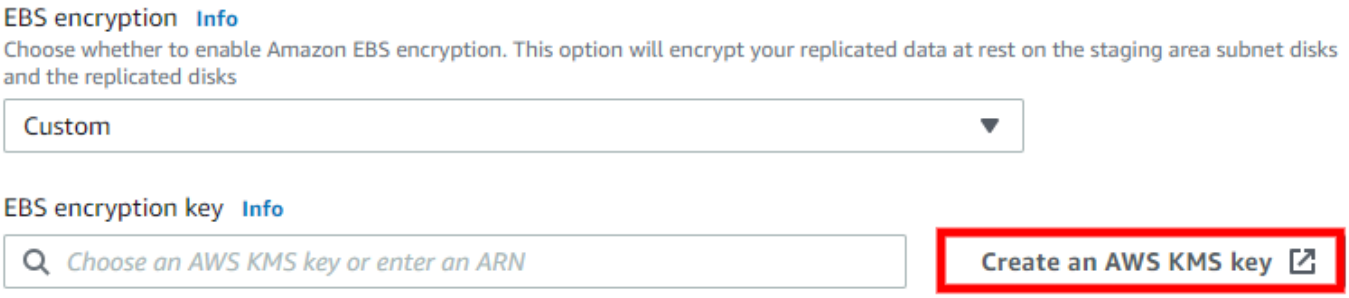
Choose whether to use the default Amazon EBS encryption Volume Encryption Key, or enter a custom customer-managed key in the regular key ID format. If you choose the **Default** option, the default key is used (which can be an EBS-managed key or a customer-managed key).



If the **Custom** option is chosen, the **EBS encryption key** box will appear. Enter the ARN or key ID of a customer-managed key from your account or another AWS account. Enter the encryption key (such as a cross-account KMS key) in the regular key ID format (KMS key example: 123abcd-12ab-34cd-56ef-1234567890ab).



To create a new AWS KMS key, click **Create an AWS KMS key**. You will be redirected to the Key Management Service (KMS) Console where you can create a new key to use.



Learn more about EBS Volume Encryption in this Amazon EBS article.

> **Important**
>
> Reversing the encryption option after data replication has started will cause data replication to start from the beginning.

Automatic replication of new disks

AWS Elastic Disaster Recovery (AWS DRS) allows you to automatically replicate newly added disks. When new disks are added to your source environment, AWS DRS initiates data replication to the staging area subnet in your AWS account.

Automating replication of new disks assists you in maintaining continuous data replication, saves time and resources, and reduces the risk of data loss in the event of a disruption.

This feature is activated automatically for newly added servers.

To deactivate or reactivate this feature for newly added servers:

- Under **Settings** on the left-hand navigation menu, choose **Default replication settings**.
- Click **Edit**.
- Under **Volumes**, uncheck the **Automatically replicate new disks** checkbox.

To activate or deactivate or reactivate this feature for a specific server:

- Go to the replication settings.

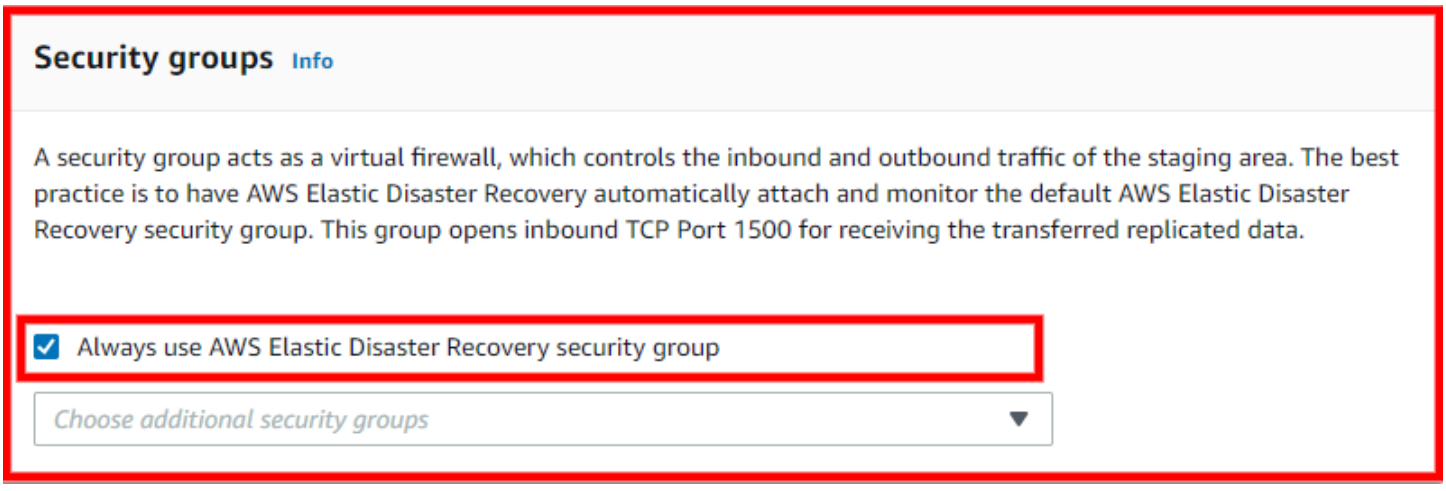
- Click **Edit**.
- Under **Volumes**, uncheck the **Automatically replicate new disks** checkbox.

> **Note**
>
> - This feature is only supported for new agent versions (version 4.6 or higher). For older versions, you must reinstall your agent to activate automatic replication of new disks.
> - Auto replication of new disks is not supported with --force-volumes.
> - It might take up to 10 minutes for new disks to start replicating.
> - New disks will only be replicated once the feature is activated and will not be replicated retroactively.

Security groups

A security group acts as a virtual firewall, which controls the inbound and outbound traffic of the staging area. The best practice is to have AWS Elastic Disaster Recovery automatically attach and monitor the default AWS Elastic Disaster Recovery security group. This group opens inbound TCP Port 1500 for receiving the transferred replicated data.

Choose whether you would like to **Always use the AWS Elastic Disaster Recovery security group**.



The best practice is to have AWS Elastic Disaster Recovery automatically attach and monitor the default AWS Elastic Disaster Recovery security group. This group opens inbound TCP Port 1500 for receiving the transferred replicated data. When the default AWS Elastic Disaster Recovery security group is used, AWS Elastic Disaster Recovery will constantly monitor whether the rules within this

security group are enforced, in order to maintain uninterrupted data replication. If these rules are altered, AWS Elastic Disaster Recovery will automatically fix the issue.

Choose the box next to the **Always use AWS Elastic Disaster Recovery security group** option to allow data to flow from your source servers to the replication servers, and that the replication servers can communicate their state to the AWS Elastic Disaster Recovery servers.

Otherwise, deselect the box next to the **Always use AWS Elastic Disaster Recovery security group** option. Doing this is not recommended.

Click the drop-down menu to select from additional security groups. The list of available security groups changes according to the **Staging area subnet** you selected.

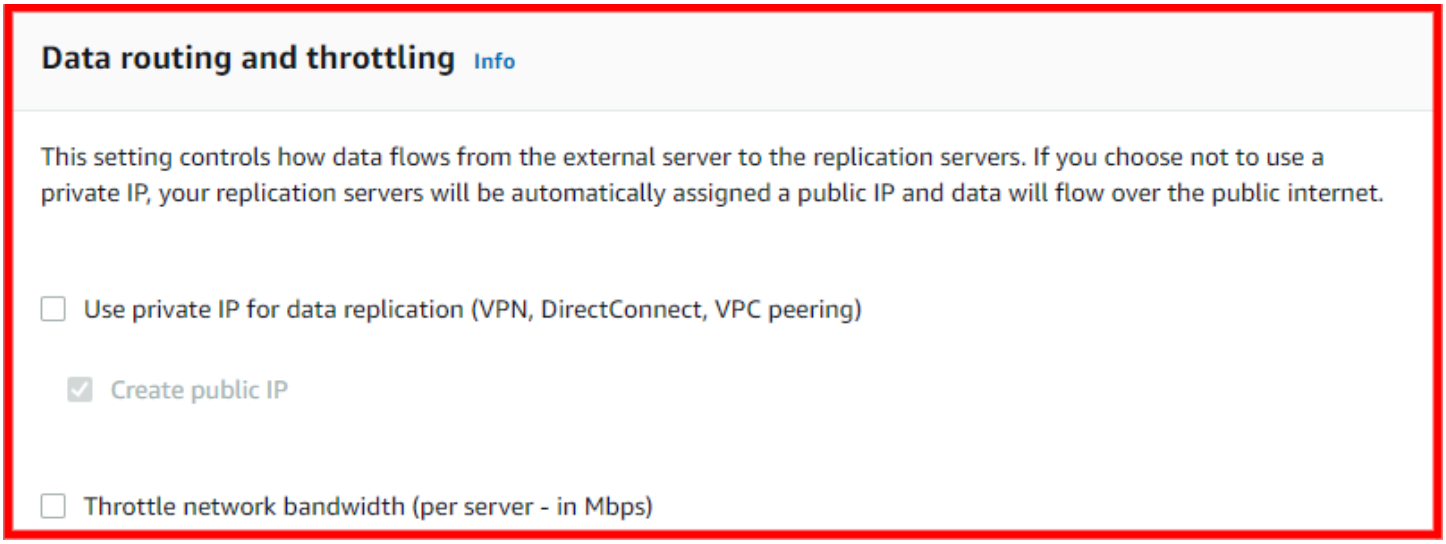
To search for a specific security group, use the search box.

If you add security groups via the AWS Console, they will appear on the Security group drop-down list in the AWS Elastic Disaster Recovery Console. Learn more about AWS security groups in this VPC article.

You can use the default AWS Elastic Disaster Recovery security group or select a different one. However, take into consideration that any selected security group that is not the AWS Elastic Disaster Recovery default will be added to the Default group, since the default security group is essential for the operation of AWS Elastic Disaster Recovery.

Data routing and throttling

AWS Elastic Disaster Recovery lets you control how data is routed from your source servers to the replication servers on AWS through the **Data routing and throttling** settings.



By default, data is sent from the source servers to the replication servers over the public internet, using the public IP that was automatically assigned to the replication servers. Transferred data is always encrypted in transit.

Use private IP for data replication

Choose the box to the left of the **Use private IP for data replication...** option if you want to route the replicated data from your source servers to the staging area subnet through a private network with a VPN, AWS Direct Connect, VPC peering, or another type of existing private connection.

Do not choose the box to the left of the **Use private IP for data replication...** if you do not want to route the replicated data through a private network.

> **Important**
>
> Data replication will not work unless you have already set up the VPN, AWS Direct Connect, or VPC peering in the AWS Console.

> **Note**
>
> - If you selected the Default subnet, it is highly unlikely that the Private IP is used for that Subnet. Ensure that Private IP (VPN, AWS Direct Connect, or VPC peering) is used for your chosen subnet if you wish to use this option.
> - You can safely switch between a private connection and a public connection for individual server settings by choosing or not choosing the box to the left of the **Use private IP for data replication....** option , even after data replication has begun. This switch will only cause a short pause in replication, and will not have any long-term effect on the replication.
> - Choosing the **Use Private IP for data replication...** option will not create a new private connection.

You should use this option if you want to:

- Allocate a dedicated bandwidth for replication;
- Use another level of encryption;

- Add another layer of security by transferring the replicated data from one private IP address (source) to another private IP address (on AWS).

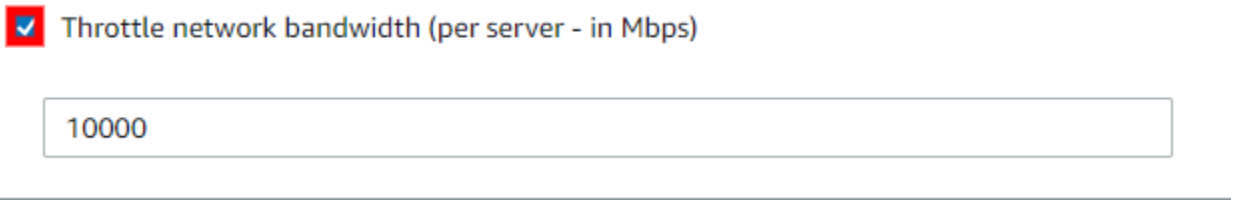
Create public IP

When you select the **Use private IP** option, you choose to **Create public IP**. Public IPs are used by default.

Throttle network bandwidth

You can control the amount of network bandwidth used for data replication per server. By default, AWS Elastic Disaster Recovery will use all available network bandwidth utilizing five concurrent connections.

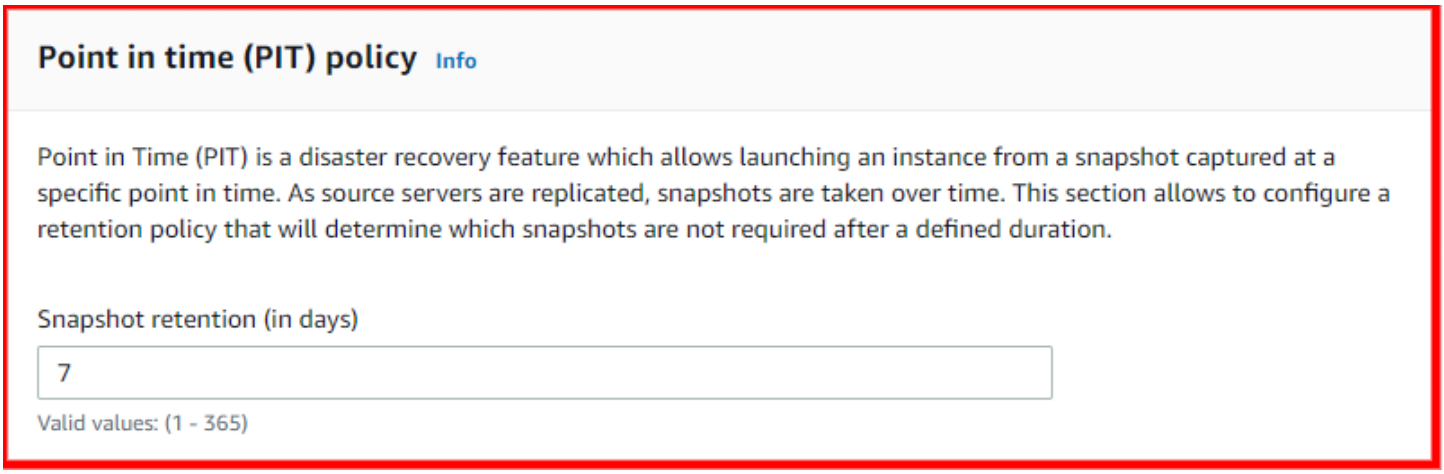
Choose the box to the left of the **Throttle network bandwidth...** option if you want to control the transfer rate of data sent from your source servers to the replication servers over TCP Port 1500.



If you activate the **Throttle network bandwidth** option, then the bandwidth field will appear. Enter your desired bandwidth in Mbps.

Point in time (PIT) policy

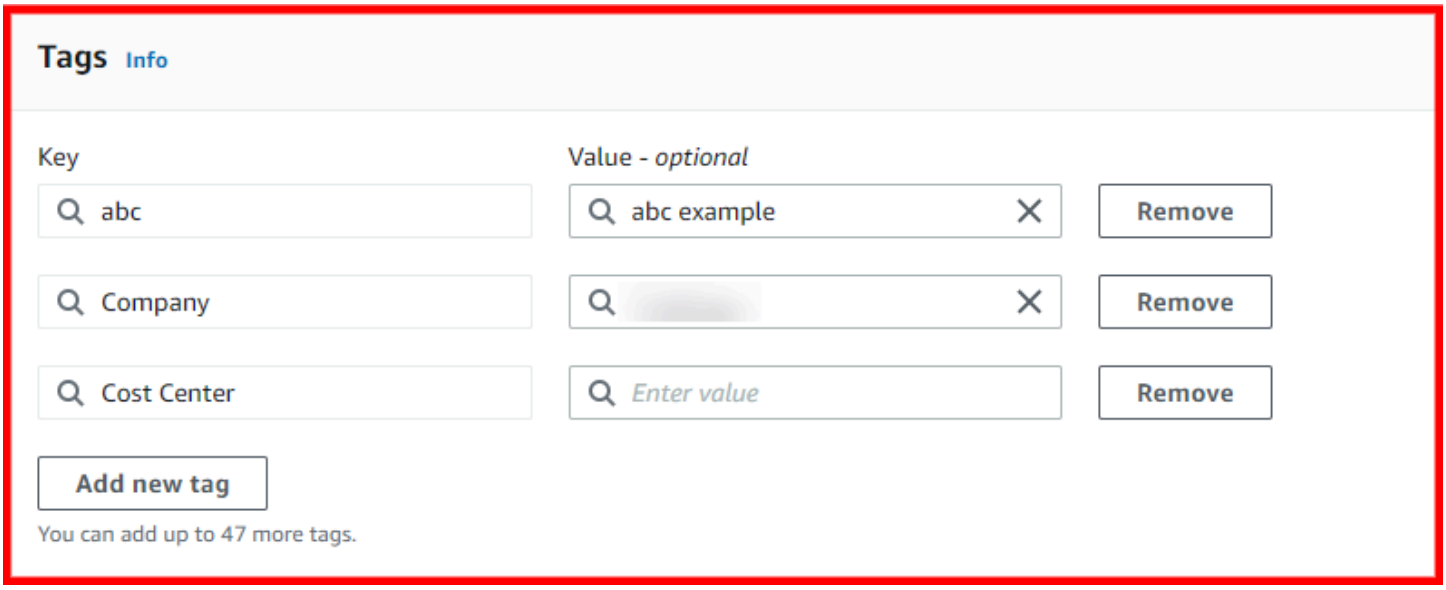
AWS Elastic Disaster Recovery allows you to select the number of days for which point in time snapshots will be retained through the **Point in time (PIT) policy** field.



You can select to save PIT snapshots for 1 up to 365 days. Saving PIT snapshots for more days will allow you more recovery options, but will also result in increased costs. Learn more about Point in time.

Tags

Add custom **tags** to resources created by AWS Elastic Disaster Recovery in your AWS account.



These are resources required to facilitate data replication, drilling and recovery. Each tag consists of a key and an optional value. You can add a custom tag to all of the AWS resources that are created on your AWS account during the normal operation of AWS Elastic Disaster Recovery.

To add new tags, take the following steps:

- Click **Add new tag**.
- Enter a **custom tag key** and an optional tag value.

Note

- You can add up to 50 tags.
- AWS Elastic Disaster Recovery already adds tags to every resource it creates, including service tags and user tags.

  These resources include:

  - Amazon EC2 instances

- Amazon EC2 launch templates
- Amazon EBS volumes
- Snapshots

Learn more about AWS tags in this EC2 article.

MAP program tagging

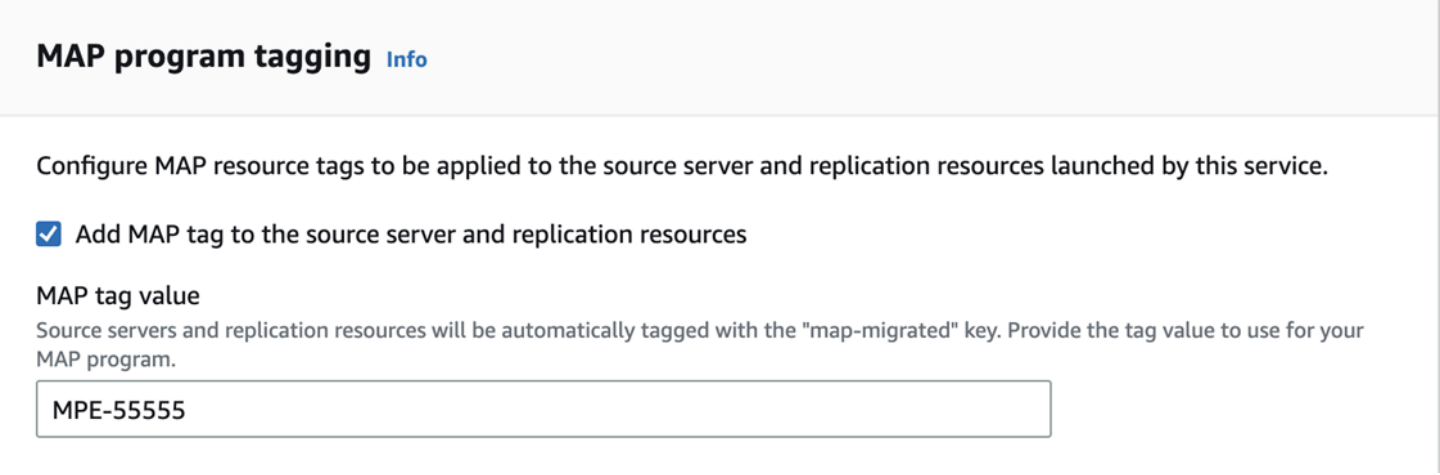
The AWS Migration Acceleration Program (MAP) provides tools that are designed to reduce costs, boost productivity, improve operational resilience and increase business agility.

DRS MAP program tagging is a feature that allows you to apply MAP program tags to your source servers and replication resources in order to offset the ongoing cost of protecting your servers.

Learn more about the AWS Migration Acceleration Program (MAP).

You can choose to add tags to:

- One or more existing source servers and replication resources
- All newly added source servers and replication resources



Adding tags to existing source servers and replication sources

To add tags to one or more existing source servers and replication sources:

- Select the relevant source servers.

- Select **Edit replication settings** from the replication drop-down menu
- Check the box to the left of **Add MAP tag to the source servers and replication resources**.
- Specify the MAP tag value that will be used in your MAP tagging.

DRS will automatically tag your source servers and replication resources with the tag key "map-migrated” and the value of the tag that you provide.

Adding tags to newly added source servers and replication sources

To add tags to all newly added source servers and replication sources:

- Select **Settings** from the left-hand menu.
- Click **Edit** to change the default replication settings.
- Check the box to the left of **Add MAP tag to the source servers and replication resources** option.
- Specify the MAP tag value that will be used in your MAP tagging.
- Click **Save changes**.

AWS Elastic Disaster Recovery will automatically tag every newly-added source server and replication resources with the tag key “map-migrated” and the value of the tag, that you provide.

For more details about the tag value that should be used here, please refer to the MAP tagging guide provided in your MAP term.

Launch settings

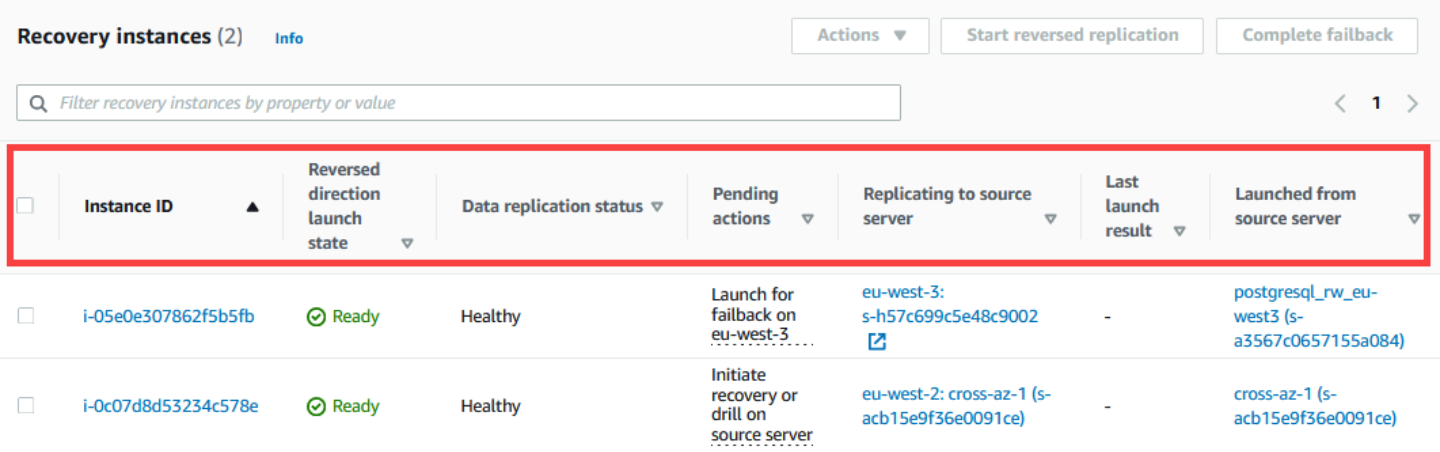
The launch settings are a set of instructions that determine how drill or recovery instance are launched in AWS. They include two sections: general launch settings and the EC2 launch template. These settings are created automatically every time you add a source server to AWS Elastic Disaster Recovery. The launch settings can be modified at any time, even before a specific source server has completed its initial sync.

Default DRS launch settings

AWS Elastic Disaster Recovery (AWS DRS) allows you to configure the default launch settings and change them at any time.

The default launch settings apply to any new source server added to AWS DRS. You will be prompted to configure your default launch settings upon your first use of AWS DRS.

Launch settings can also be edited manually for individual servers.



Editing the default DRS launch settings

The default launch settings will be applied to every newly launched source server in AWS Elastic Disaster Recovery (AWS DRS). You can change these settings for a single or multiple servers whenever you choose.

To edit these settings, take the following steps:

1. Select **Default launch** from the left-hand navigation menu (under **Settings**).
2. Click **Edit** in the **Default DRS launch settings** section.
3. Change the settings according to your preferences.
4. Click **Save**.

Launch settings parameters

AWS Elastic Disaster Recovery (AWS DRS) launch settings include:

- **Instance type right sizing –** Allow the service to automatically update the instance type on the EC2 launch template, based on the CPU and RAM of the source server. If this setting is active (default), any modification you make to the instance type in the EC2 launch template will be overwritten by the service.
- **Start instance upon launch –** Configure how the EC2 recovery instance should be launched – running or in a stopped state.

- **Copy private IP –** Define whether the private IP should be copied from the source server's primary network interface to EC2 launch template. If this setting is on, make sure that the subnet defined in the EC2 launch template includes that IP in its range.
- **Transfer server tags –** Define if the launched EC2 instance should have the same tags as the source server resource.
- **Launch into source instance -** Define whether DRS will automatically assign the ID of the source instances to the **Launch into instance ID** field in the Launch Settings of newly added source servers in this region. A source instance is the EC2 instance in this region that was the source of the data before replication was reversed to this region or availability zone. The EC2 instance to launch into must have a tag with key *AWSDRS* and value *AllowLaunchingIntoThisInstance*, and it must be stopped before launching into it. If **Launch into instance ID** is automatically set for a source server, the **Transfer server tags**, and **Copy private IP** settings will need to be deactivated for that server, as they cannot apply to an already launched instance.

  > **Note**
  >
  > Note that for the instance to appear as a recovery instance in DRS, it needs to have an instance profile that includes the policy **AWSElasticDisasterRecoveryRecoveryInstancePolicy**. The role **AWSElasticDisasterRecoveryRecoveryInstanceRole**, which is added to an account when initializing the service, contains this policy and can be used as an instance profile.

  Learn more about Launch into source instance.
- **OS licensing –** Choose the launched instance's license type for Windows Servers – License-included or Bring Your Own License (BYOL). Linux servers and Windows Home are automatically launched as BYOL. If you launch a Windows Server or Windows Home as BYOL, you must select Dedicated host for the Tenancy setting in the advance settings of the EC2 launch template.

Launch into source instance

This setting is only valid when the replication and recovery are done in-AWS, between 2 AWS regions or availability zones. This default setting applies to newly added source servers. Such servers will have their **Launch into instance ID** field in the Launch Settings set to the EC2 instance ID of the source instance, that was the source of the data in the same region or availability zone. See the examples below for more details.

Pre-requisites

Start reversed replication or **Protect recovered instance** will fail to create a source server if this setting is active and one of the following conditions is not met:

1. The instance to launch into must have the required tag with key *AWSDRS* and value *AllowLaunchingIntoThisInstance*.
2. The instance to launch into must have the same operating systems platform (Linux or Windows) as that of the recovery instance the **Start reversed replication** or **Protect recovered instance** was called on.
3. If the instance to launch into is a Linux it must have the BIOS boot mode and if this Windows, it must have the same boot mode as that of the recovery instance the **Start reversed replication** or **Protect recovered instance** was called on.
4. The instance to launch into must have the x86_64 architecture, HVM virtualization and an EBS root device.
5. **OS licensing** in **Default DRS launch settings** can only be **Bring Your Own License (BYOL)** if the instance's platform is Linux or if the instance's **tenancy** is **dedicated host**.
6. **Transfer server tags** and **Copy private IP** must be deactivated in **Default DRS launch settings**.

Cross-region

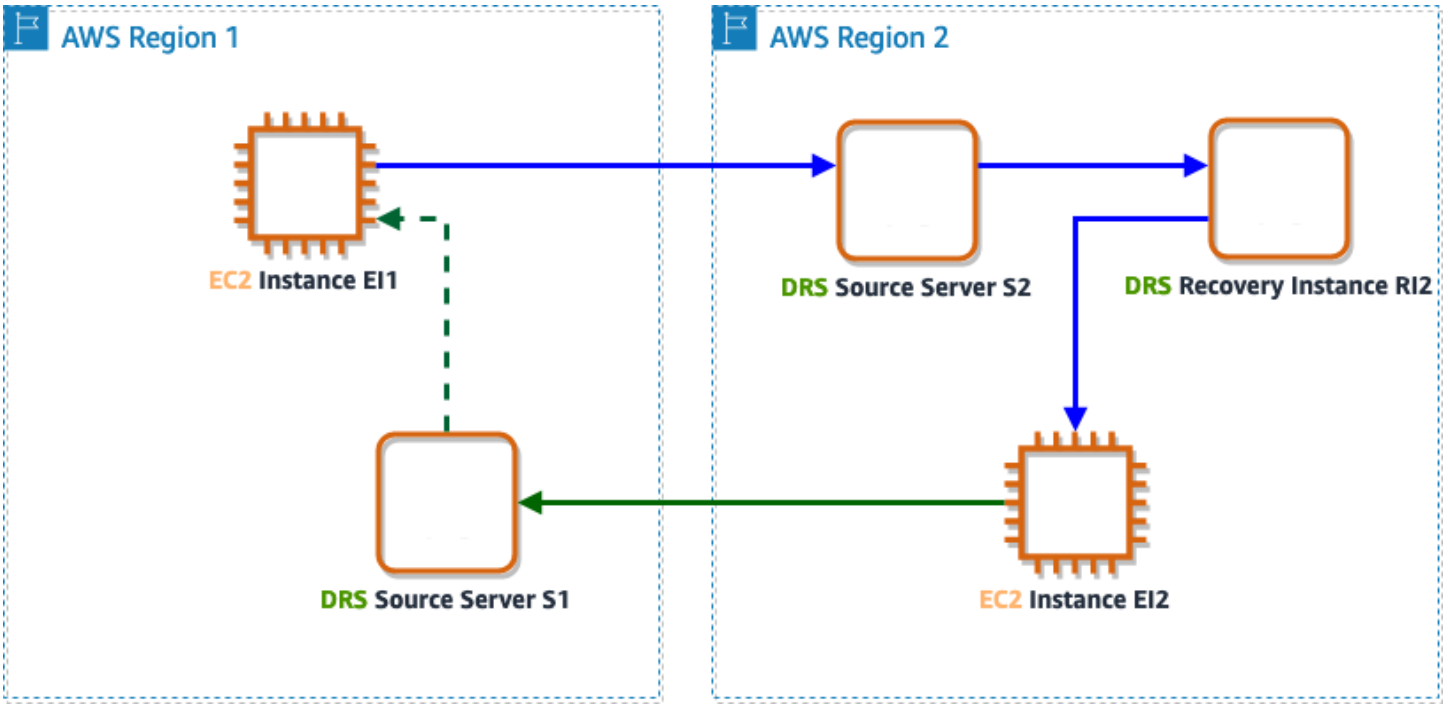
With this setting active, customers who replicate their EC2 instances between two AWS regions, launch in the second region from an instance in the first region, and call **Start reversed replication** to go back to the first region will have their source servers in the first region automatically set **Launch into instance ID** to the instance ID of the instance in the first region they initially launched from.

Using the diagram below as an example, the setting applies to source servers such as Source Server **S1** and automatically set the **Launch into instance ID** to the instance ID of EC2 instance **EI1** (marked by the dotted green arrow in the diagram).

This will only happen if:

1. **Launch into source instance** was set to be active in the **Default DRS launch settings** on region 1.

2. EC2 instance **EI1** (on AWS region 1) replicated into region 2 (replication handled through source server **S2** on AWS region 2), and was launched in AWS region 2 (launch handled through source server **S2** on AWS region 2), creating recovery instance **RI2**.
3. Source server **S1** was created by calling **Start reversed replication** in region 2 on recovery instance **RI2** (marked by the solid green arrow), replicating the data of EC2 instance **EI2**.



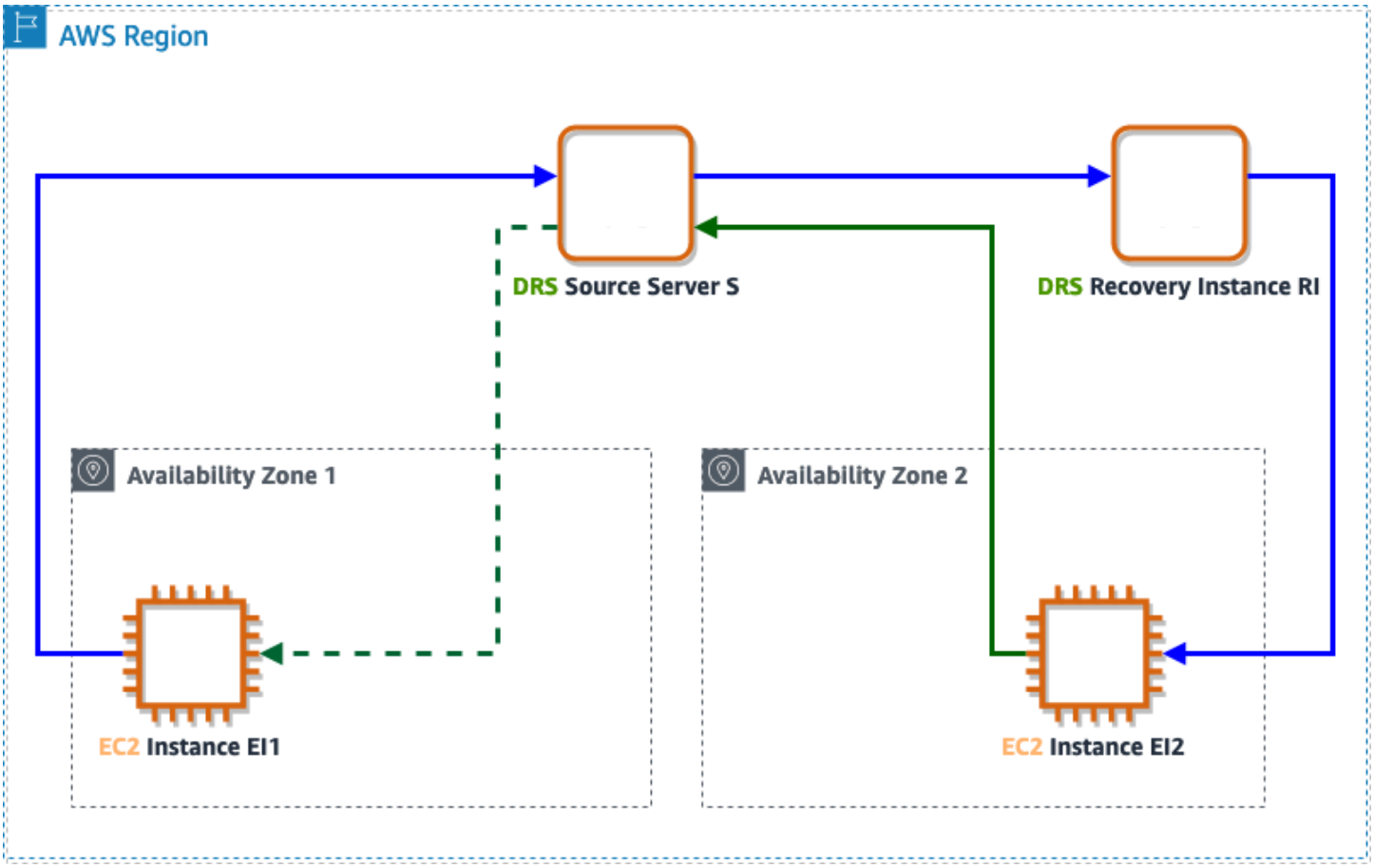
Cross availability zone

With this setting active, customers who replicate their EC2 instances into the same region (and if following our recommendation, into a different availability zone within that region), launch in the region (recommended to launch into the other availability zone) from an instance in the first availability zone, and perform **Protect recovered instance** on the source server after this launch, will have the source server automatically set **Launch into instance ID** to the instance ID of the instance in the first availability zone.

Using the diagram below as an example, the setting applies to source servers such as Source Server **S** and automatically set the **Launch into instance ID** to the instance ID of EC2 instance **EI1** (marked by the dotted green arrow in the diagram).

This will only happen if:

1. **Launch into source instance** was set to be active in the **Default DRS launch settings** on this region.
2. EC2 instance **EI1** (on AWS region 1) replicated into availability zone 2 (replication handled through source server **S**), and was launched in availability zone 2 (launch handled through source server **S**), creating recovery instance **RI**.
3. Source server **S** was then updated to protect recovery instance **RI** (marked by the solid green arrow), by calling Protect recovered instance, replicating the data of EC2 instance EI2.



Default EC2 launch template

The default EC2 launch template sets the default values that will be copied to EC2 templates created for newly added source servers. This template defines how drill, recovery, or failback instances are launched. If you didn't create any default EC2 template, AWS DRS will copy the default values for each setting to EC2 launch templates for newly added servers.

You can usually launch a drill instance without modifying the automatically created EC2 launch template (unless you have removed the default VPC/subnet from your AWS account).

Editing the default EC2 launch template

To edit the default EC2 launch template, take the following steps:

1. Select **Default launch** from the left-hand navigation menu (under **Settings**).
2. Click **Edit** in the **Default EC2 launch template** section.
3. Change the settings according to your preferences.
4. Click **Save**.

EC2 launch template parameters

AWS Elastic Disaster Recovery (AWS DRS) EC2 launch settings are divided into basic and advanced settings.

The basic settings include:

- **Subnet –** When you specify a subnet, this field defines where the instance will be launched. When selecting a subnet, only the default network interface will be updated. If you do not include a subnet, the launched instance will use the Region's default subnet.
- **Security groups –** The selected security groups to assign to the instance, applied to the subnet selected for the default network interface. If no security group is selected, there is no default value and no group will be used. Security groups can only be selected if a subnet is included.
- **Instance type –** The default instance type to use when launching. If instance type right-sizing is active, the system will disregard this setting. If no instance type is included, a default value will be used.
- **EBS volume type –** Applies to all volumes for which this type is relevant. If an unmatching type exists, the default type (GP3) will be used instead. Some volume types require setting additional values such as IOPs or throughput.

Advanced settings include additional parameters that add specific features to the EC2 template. If you choose not to include these parameters in the template, the specific capabilities will not be added.

The advanced settings include:

- **IAM instance profile –** Attach a specific profile to the instance that will be launched. Make sure the instance profile has the AWSElasticDisasterRecoveryRecoveryInstancePolicy IAM policy attached in addition to any other policy.
- **Auto assign public IP –** Automatically assign a public IP to the launched instance.
- **Termination protection –** Protect the launched instance from accidental termination using the EC2 console.
- **Tenancy –** Set tenancy information, such as dedicated host needed in conjunction with setting BYOL for Windows servers and Windows Home.
- **Capacity reservation –** Apply reservation consideration to the launched instances.
- **Key pair –** Associate a key pair with launched instances that are based on EC2 instances.

Note

AWS DRS only supports major EC2 template parameters. If you want to change values that are not supported by this feature, you can still do so by editing the EC2 launch template via the Amazon EC2 console:

- Create a new EC2 template version with the required changes.
- Mark it as default.

Important

Every time you modify an EC2 launch template on the Amazon EC2 console, a new version is created. AWS DRS uses the version that is marked as the default. if you prefer to use the EC2 launch template you just modified, make sure to mark it as the default. Changes made through the AWS DRS console are automatically set as the default version.

EC2 launch template tags – In addition to the basic and advanced settings, you can also add up to 50 tags. These will be transferred to EC2 launch template created by AWS Elastic Disaster Recovery (AWS DRS) service.

Learn more about EC2 launch template settings and configuration options in this EC2 article.

EC2 template considerations

1. **Revert to previous version –** The right-sizing mechanism can fix issues such as an incorrect instance type, but other issues may still occur. If you encounter any issues with the launch template, you can quickly address them by choosing the original default launch template that was created by AWS DRS when the agent was installed. Alternatively, you can edit the relevant fields from the AWS DRS console.
2. **IOPS –** If needed, set the number of I/O operations per second that the volume can support via the Amazon EC2 console. You can select any number as long as it matches the Amazon EBS guidelines.
   - Provisioned IOPS SSD (io1): 50 IOPS per GiB of storage
   - Provisioned IOPS SSD (io2): 500 IOPS per GiB of storage
   - General Purpose SSD (gp3): 500 IOPS per GiB of storage

Configuring the default post-launch actions

After finishing the AWS DRS initialization process, you can configure your default post-launch actions settings. The default post-launch actions settings apply to newly added source servers and controls which post-launch actions will run when launching new instances. These settings are created automatically for each server based on the default settings and can be modified at any time for any individual source server.

You can also use this settings to install the IAM roles required for post-launch actions to work, if the roles were not already installed in your account during the first initialization of AWS DRS. The IAM roles need to be installed once per AWS account, regardless of the region used.

Post launch actions can be of two different types: command and automation.
Command post-launch actions run on the launched instance using the instance profile attached to the instance. Note that if no instance profile is defined on the EC2 launch template, AWS Elastic Disaster Recovery (AWS DRS) places the **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole** instance profile by default if post-launch actions is active for the source server.

Automation actions run with the credentials of the same IAM entity that started the drill, recovery or failback. In addition some automation actions accept a parameter that is sent to the assumeRole key in the SSM document if provided, the action will assume that role for that action execution.

Note

In order to use post-launch actions, you should make sure you have the required permissions. To get these permissions, you can attach the **AWSElasticDisasterRecoveryLaunchActionsPolicy** or **AWSElasticDisasterRecoveryConsoleFullAccess_v2** policies to your IAM identity. These policies contain the permissions needed to run SSM Command and Automation documents that are owned by Amazon or by the account as post-launch actions.

Note

In order to run an SSM command or Automation document owned by a different account as a post-launch action you should provide the permission to use `ssm:SendCommand` and `ssm:StartAutomation` on the relevant document.
For example, if you have shared the SSM documents MyCommand (command) and MyAutomation (automation) from account 111111111111, you should attach the following permissions to you your IAM entities:

Example

```
                {
                    "Effect": "Allow",
                    "Action": [
                        "ssm:SendCommand",
                    ],
                    "Resource": "arn:aws:ssm:*:111111111111:document/
MyAutomation",
                    "Condition": {
                        "ForAnyValue:StringEquals": {
                            "aws:CalledVia": [
                                "drs.amazonaws.com"
                            ]
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
```

```
                        "ssm:StartAutomationExecution"
                    ],
                    "Resource": "arn:aws:ssm:*:111111111111:automation-
definition/MyAutomation",
                    "Condition": {
                        "ForAnyValue:StringEquals": {
                            "aws:CalledVia": [
                                "drs.amazonaws.com"
                            ]
                        }
                    }
                }
```

Topics

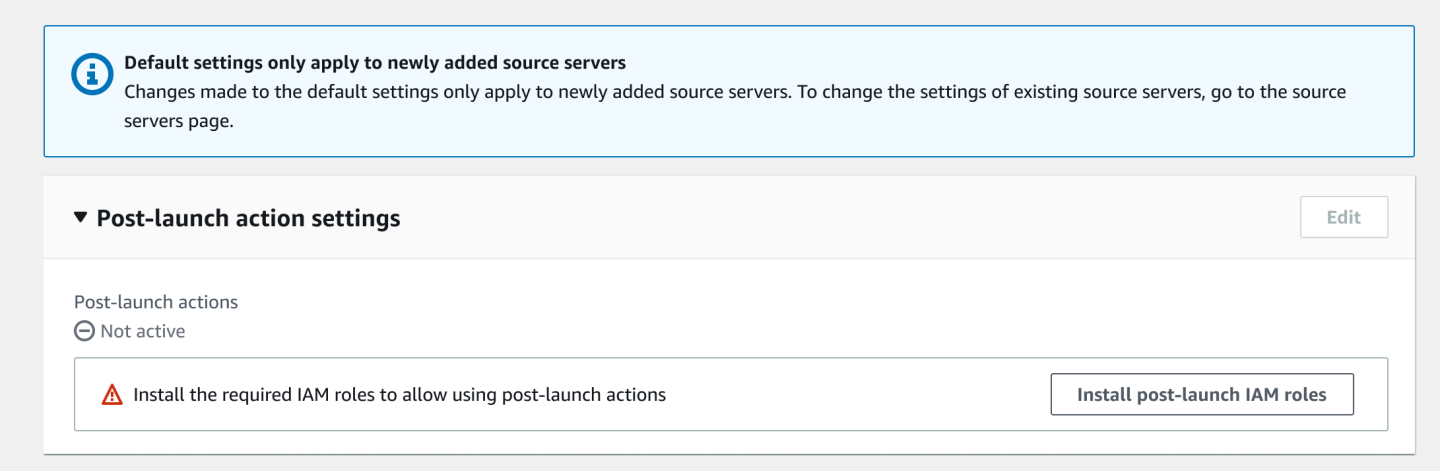
- Install the required IAM roles if needed
- Activating post-launch actions default settings
- Adding custom actions
- Activating, deactivating and editing predefined or custom actions
- Deleting custom actions
- Predefined post-launch actions
- Validate disk space
- EC2 connectivity checks
- Verify HTTP/HTTPS response
- Verify Tags

Install the required IAM roles if needed

To operate post-launch actions and allow to run SSM documents on launched instances, certain IAM roles must be installed. Usually these roles are installed into an AWS account when AWS DRS is initialized in the account for the first time in any region.

If you have already initialized Elastic Disaster Recovery in your account before September 13, 2023, it may happen that the required IAM roles were not installed in your account.

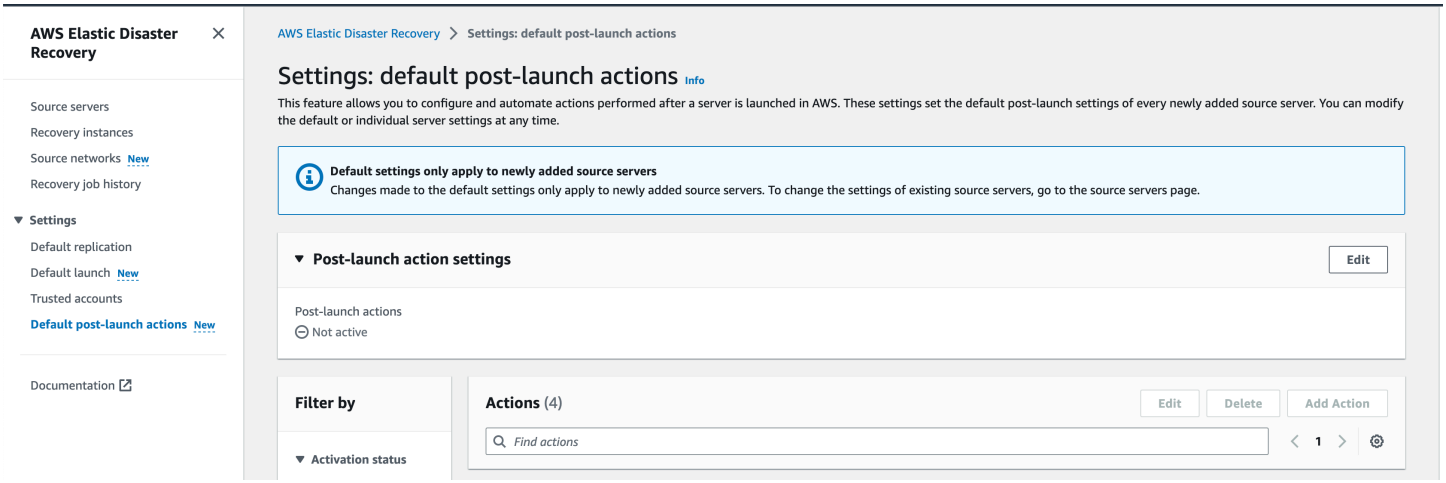
To verify the IAM roles are installed or install them if not installed (a one time operation, go to **Settings** → **Default post-launch actions** and check **Post-launch actions settings** to see if there is a message labelled: **Install the required IAM roles to allow using post-launch actions**. If this message appears, click the button labelled **Install post-launched IAM roles**. If the roles were installed successfully, the message to install the roles will not be present in **Post-launch actions settings**.



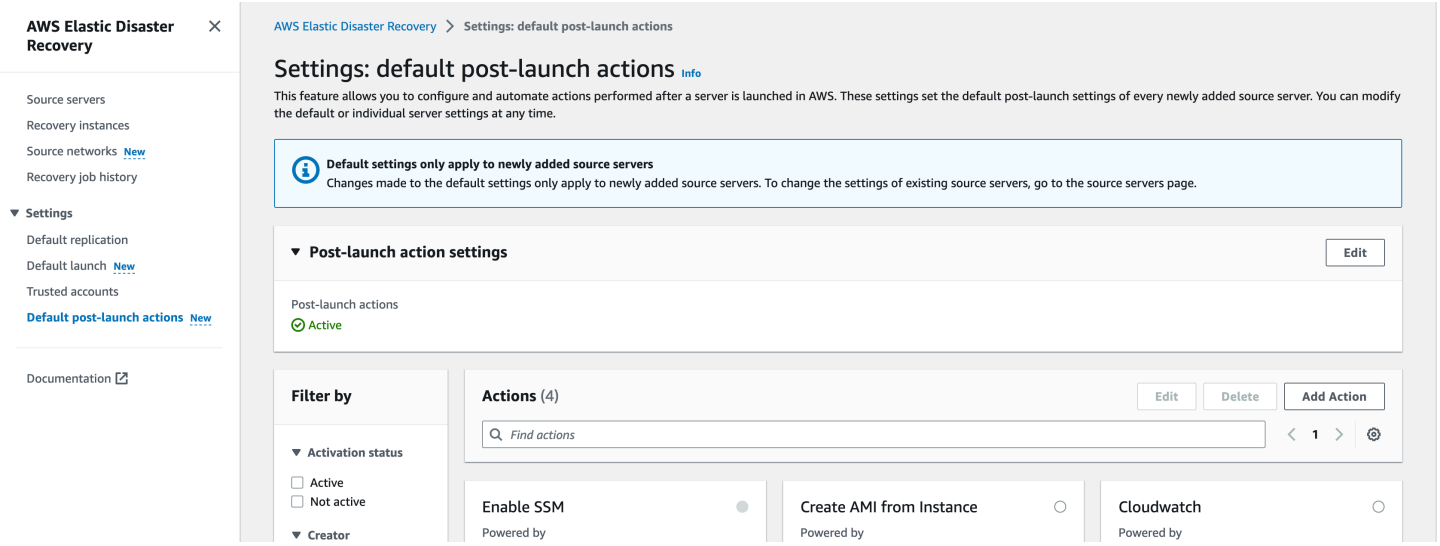
Activating post-launch actions default settings

Activate post-launch actions in the default settings, to make it active by default for newly added source servers, and to allow to update the default list of actions. Activating and deactivating is only possible if the required IAM roles have been installed.

To activate, make sure the required IAM role is installed by following this guide. After that, go to **Settings** → **Default post-launch actions** and check **Post-launch actions settings** to see if **Post-launch actions** is set to **Active**. In case it is not, click **Edit** and make sure **Post-launch actions activated** is checked. Then click **Save** to store these settings.



To deactivate go to **Settings** → **Default post-launch actions** and check **Post-launch actions settings** to see if **Post-launch actions** is set to **Not active**. In case it is not, click **Edit** and make sure **Post-launch actions activated** is not checked. Then click **Save** to store these settings.

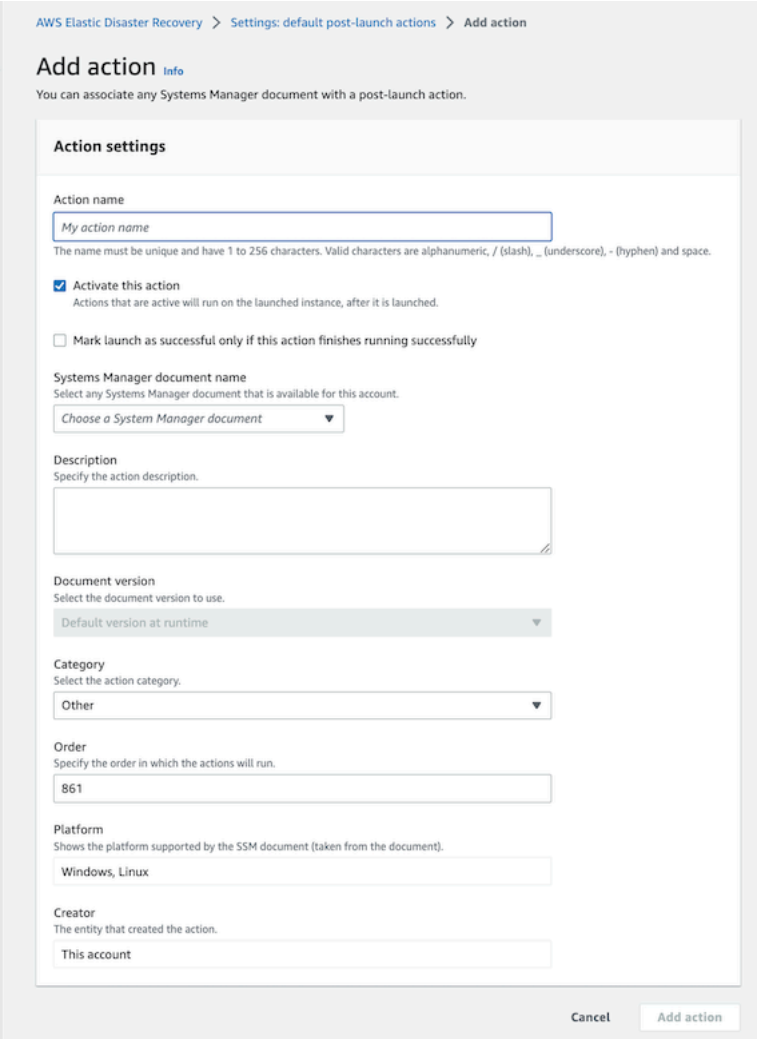


Adding custom actions

AWS Elastic Disaster Recovery (AWS DRS) allows you to run any SSM document that you like – public SSM documents or ones you created and uploaded to your account. You can configure a custom action to run any SSM document that is available in your account. To be able to create, edit or delete a default custom action, make sure the post-launch actions are activated in the default settings.

Create a custom action

Adding a custom action through the default settings, adds it to newly added servers. To add a custom action to an existing source server, do so using the **Post-launch settings** tab in the source server details page. To add a new custom action to the default post-launch action settings, go to **Settings** → **Default post-launch actions**. If the default post-launch actions settings is **Active**, you can create new custom actions by clicking on the **Add action** button.



The **Add action** page includes the following parameters:

Action name – The name of the action in AWS DRS, which should be intuitive, meaningful and unique in this AWS account and region.

Activate this action – Use this checkbox to activate or deactivate the action by default. Newly added source servers will have the action set to active or not active according to the value this field had when the source server was added.

Mark launch as successful only if this action finishes running successfully – This checkbox will dictate whether or not the launch will be marked as successful, based on the successful run of this action. Instance launches will still progress normally regardless of the success of the action.

System Manager document name – Select any Systems Manager document that is available to be used in this account.

System Manager document name – Select any Systems Manager document that is available to be used in this account.

View in Systems Manager – Click to open **System Managers** and view additional information about the document.

Description – Add a description or keep the default.

Document version – Select which SSM document version to run. AWS DRS can run a default version, the latest version, or a specific version, according to your preferences.

Category – Select from various available categories including monitoring, validation, security and more.

Order – Specify the order in which the actions will be executed. The lower the number, the earlier the action will be executed. Values allowed are between 2 and 10,000. The numbers must be unique but don't need to be consecutive.

Platform – Taken from the SSM document and reports which Operating System platform (Windows/Linux) is supported by the action.

Creator – Who created the action. For custom actions, the default is always **This account**.

The **Action parameters** change according to the specific SSM document that is selected. Note that for the instance ID parameter, you can choose to use the launch instance ID, in which case, AWS DRS will dynamically populate the value.

> **Note**
>
> AWS Elastic Disaster Recovery (AWS DRS) places **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole** instance profile on the launch instance if post-launch actions is active for the source server. If you add an SSM command action that requires additional permissions in the launch instance, you must ensure that the instance profile has the right policies or the right permissions. In order to do so, create a role that has the required permissions as per the policies above or has a policy or policies with those permissions attached to it. Go to **Launch settings** > **EC2 launch template** > **Modify** > **Advance** > **IAM instance profile**. Use an existing profile or create a new one using the **Create new IAM profile** link.

> **Note**
>
> Only trusted, authorized users should have access to the parameter store. For enhanced security, ensure that users who do not have permissions to execute SSM documents / commands, do not have access to parameter store. Learn more about restricting access to Systems Manager parameters. Action parameters are stored in the SSM parameter store as regular strings. Changing parameters in the SSM Parameter store may impact the post launch action run on target instances. We recommend to consider security implications, when choosing to use parameters that contain scripts or sensitive information, such as API keys and database passwords.

Activating, deactivating and editing predefined or custom actions

You can activate, deactivate and edit actions available in the default post-launch actions settings. Activating an action in the default settings, adds the action as activated to newly added servers. Likewise, deactivating it, adds it as a non-active action to newly added servers. Source servers already created with this action are not affected by changes in the default settings.

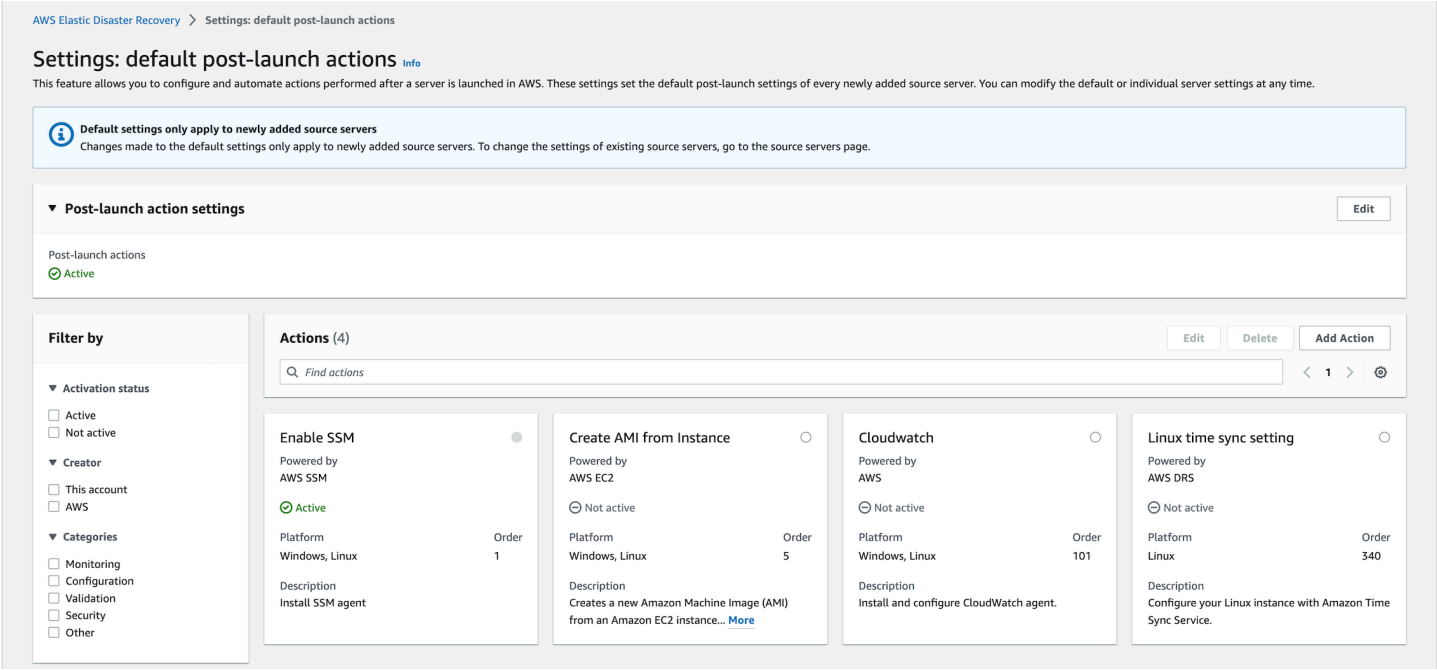
Editing an action in the default settings, adds the edited action to newly added servers. Servers already created with the action before the edit, are not updated with the changes present in any edit to the default settings that was made after their creation. Changes to actions on an existing source server can be made from the **Source server details** page, by going to the **Post-launch settings** tab and performing the change there.

To be able to activate, create, deactivate, edit, or delete a custom action and to activate, deactivate or edit predefined actions, make sure the post-launch actions are activated in the default settings.

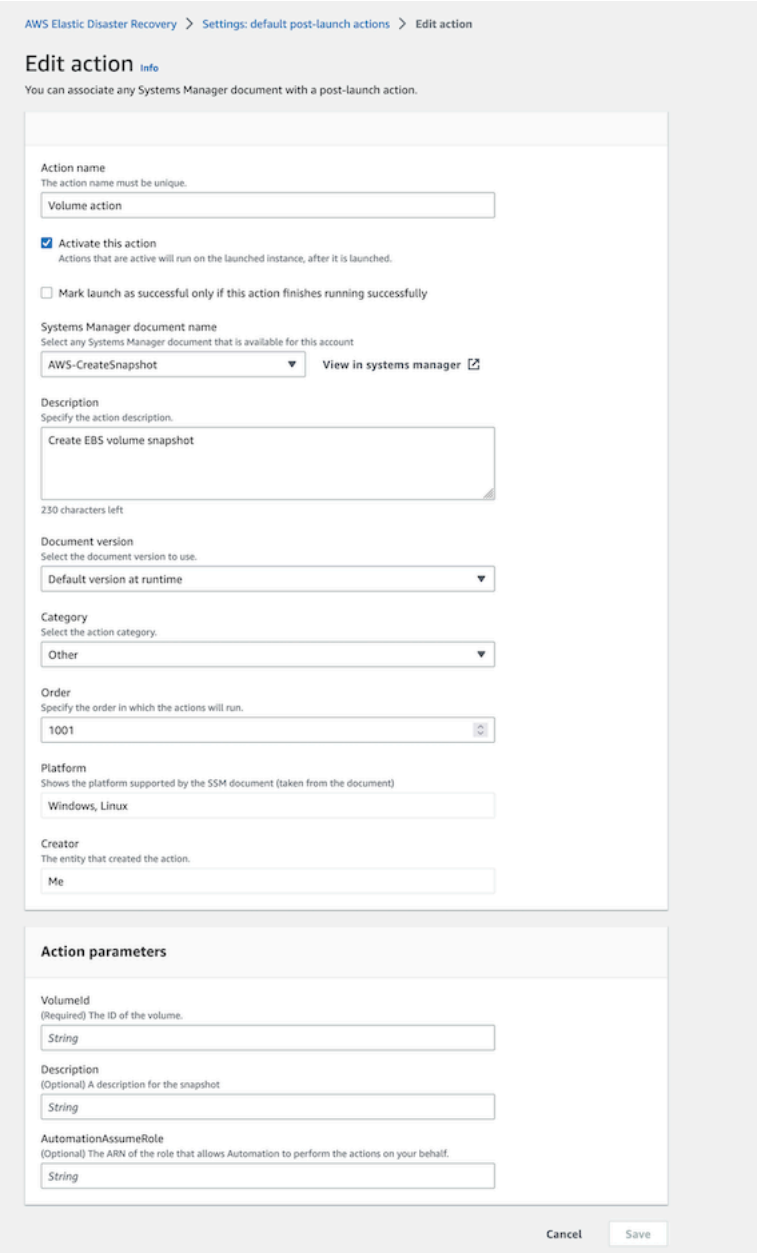
Activate, deactivate or edit a post-launch action

To activate, deactivate or edit a post launch action in the default post-launch actions settings, go to **Settings** → **Default post-launch actions**. If **Post-launch actions settings** shows **Post-launch actions** to be **Active**, you can edit any action defined in the default settings.

Locate the action you want to edit in the **Actions** card view, or use the search field to filter the actions by name.



Click on the action's card to select it, and then click on the **Edit** button.



To activate the action, make sure the **Activate this action** setting is checked and click the **Save** button. To deactivate, make sure the **Activate this action** setting is un-checked and click the **Save** button.

The edit page allows to change the value of some of the parameters for both pre-defined actions and custom actions. Some parameters can only be edited if the action is a custom action. See below for specific information.

The parameters that appear on the edit page:

Action name – Editable for custom actions. The name of the action in AWS DRS, which should be intuitive, meaningful and unique in this AWS account and region.

Activate this action – Use this checkbox to activate or deactivate the action by default. Newly added source servers will have the action set to active or not active according to the value this field had when the source server was added.

Mark launch as successful only if this action finishes running successfully – This checkbox will dictate whether or not the launch will be marked as successful, based on the successful run of this action. Instances launches will still progress normally regardless of the success of the action.

System Manager document name – Editable for custom actions. Select any Systems Manager document that is available to be used in this account.

View in Systems Manager – Click to open **System Managers** and view additional information about the document.

Description – Editable for custom actions. Add a description or keep the default.

Document version – Editable for custom actions. Select which SSM document version to run. AWS DRS can run a default version, the latest version, or a specific version, according to your preferences.

Category – Editable for custom actions. Select from various available categories including monitoring, validation, security and more.

Order – Specify the order in which the actions will run. The lower the number, the earlier the action will run. Values allowed are between 2 and 10,000. The numbers must be unique but don't need to be consecutive.

Platform – Not editable. Taken from the SSM document and reports which Operating System platform (Windows/Linux) is supported by the action.

Creator – Not editable. Who created the action. For custom actions, the default is always **This account**.

The **Action parameters** change according to the specific SSM document that is selected. Note that for the instance ID parameter, you can choose to use the launch instance ID, in which case, AWS DRS will dynamically populate the value. Some predefined actions, where applicable allow to use a dynamically populated value for the volumes. This value will be dynamically populated by AWS DRS with the volumes of the instance being launched.

After making the required changes, click **Save**, to save the changes and **Cancel** to abort them.

Deleting custom actions

Custom actions created in default settings can also be deleted. Deleting a custom action in the default settings removes it from the default settings and means the action will no longer be added to newly added servers. Deleting the action in the default settings does not remove it from existing source servers that have it. To delete a custom action from existing servers, go to the **Source server details** page, select the **Post-launch settings** tab and delete the action from there. Pre-defined actions cannot be deleted through AWS console. If a pre-defined action is not required, it can be deactivated or deleted via API.

Locate the action you want to delete in the **Actions** card view, or use the search field to filter the actions by name. Select the action, and click the **Delete** button. To confirm, press **Delete**.

Predefined post-launch actions

AWS Elastic Disaster Recovery allows you to run various predefined post-launch actions on your EC2 launched instance. Use these out-of-the-box actions to validate your launch or improve your launch flexibility.

Choose from a variety of predefined post-launch actions

- Enable SSM
- Install a CloudWatch Agent
- Create AMI from instance
- Configure Time Sync
- Volume integrity validation
- Process status validation
- Validate disk space
- EC2 connectivity check
- HTTP/HTTPS response validation
- Verify tags

Note

Currently, only the following predefined post-launch actions are supported in the Middle East (UAE) Region:

- Enable SSM
- CloudWatch agent installation
- Create AMI from instance
- Volume integrity validation
- Process status validation
- Validate disk space
- EC2 connectivity check
- HTTP/HTTPS response validation
- Verify tags

Enable SSM

The AWS Systems Manager (AWS SSM) allows AWS Elastic Disaster Recovery to run post-launch actions on your recovery instances after they are launched. When you activate the post-launch actions, AWS Elastic Disaster Recovery will install the **AWS SSM agent**.

The AWS SSM agent must be installed for any other post-launch action to run. Therefore, this is the only post-launch action that is activated by default and cannot be deactivated. Learn more about SSM.

Install a CloudWatch Agent

Use the **CloudWatch agent installation** feature to install and configure the CloudWatch Agent and Application Insights. The launched instance will require the following policies:

CloudWatchAgentServerPolicy – The permissions required to use AmazonCloudWatchAgent on servers

AmazonSSMManagedInstanceCore – The policy for Amazon EC2 Role to enable AWS Systems Manager service core functionality

To ensure that the launch instance has the right policies, create a role that has the required permissions as per the policies above or have access to a role with those permissions. Go to **Launch settings > EC2 launch template > Modify > Advance > IAM instance profile**. Use an existing profile or create a new one using the **Create new IAM profile** link.

> **Note**
>
> You must attach both policies to the template for the CloudWatch agent to operate. Without the **CloudWatchAgentServerPolicy**, the action will still be marked as successful but the CloudWatch Agent will not be active. Configuring the Application Insights is optional. You can choose to skip the Application Insights agent configuration and only install the CloudWatch agent. To do so, simply provide the required parameterStoreName parameter and leave the other parameters empty.

Learn more about the CloudWatch Agent.

Create AMI from instance

Use the Create AMI from Instance feature to create a new Amazon Machine Image (AMI) from your AWS DRS launched instance.

The action uses the following APIs:

- CreateImages
- DescribeImages

To allow the SSM document to run these APIs, you will need to have the required permissions or have access to a role with those permissions and then provide the role's ARN as an input parameter to the SSM automation document. Learn more about creating AMI from instance.

Configure Time Sync

Use the **Time Sync** feature to set the time for your Linux instance using ATSS.

Learn more about Amazon Time Sync.

Volume integrity validation

Use the **Volume integrity validation** to ensure that EBS volumes on the launched instance are:

- The same size as the source (rounded up)
- Properly mounted on the Amazon EC2 instance
- Accessible

This feature allows you to conduct the required validations automatically and saves the time of manual validations.

> **Note**
>
> Up to 50 volumes can be checked in a single action.

Process status validation

Use the **Process status validation** feature to ensure that processes are in running state following instance launch. You will need to provide a list of processes that you want to verify, and define how long the service should wait before testing begins.

To check a specific process that should run multiple times, include it several times in the list.

Validate disk space

Use the **Disk space validation** feature to obtain visibility into the disc space that you have at your disposal, as well as logs with actionable insights.

EC2 connectivity checks

Use the **EC2 connectivity checks** feature to conduct network connectivity checks to a predefined list of ports and hosts.

> **Note**
>
> Up to 5 Port:IP couples can be checked in a single action.

Verify HTTP/HTTPS response

Use the **Verify HTTP/HTTPS response** feature to conduct HTTP/HTTPS connectivity checks to a predefined list of URLs. The feature will verify that HTTP/HTTPS requests (for example, https://localhost) receive the correct response.

Verify Tags

Use the **Verify Tags** feature to validate that tags which have been defined in the launch template and on the source server are copied to the migrated server.

Source servers

You must add your source servers to the AWS Elastic Disaster Recovery console in order to replicate them into AWS. Source servers are added by installing the AWS Replication Agent on each individual server. The following documentation provides installation paths for both Linux and Windows servers. Ensure that your servers are supported by AWS Elastic Disaster Recovery by reviewing the Supported Operating Systems documentation.

Once your source servers have been added to AWS Elastic Disaster Recovery, you can monitor and interact with them from the Source Servers page. The source servers page is the default view in the AWS Elastic Disaster Recovery Console, and will be the page that you interact with the most. On the Source Servers page, you can view all of your source servers, monitor their recovery readiness and data replication state, see the last recovery result, see any pending actions, and sort your servers by a variety of categories. You can also perform a variety of commands from the Source Servers page through the command menus. These menus allow you to fully control your servers by launching drill and recovery instances and performing a variety of actions, such as adding servers, editing settings, disconnecting, and deleting servers.

You can choose the hostname of any individual source server on the source servers page in order to access the server details view. This view will allow you to see the details for individual servers. Here you will be able to see an in-depth overview of the server's recovery state, view the server's technical details, manage tags, manage disks, and most importantly, configure the individual replications settings and launch settings for the server.

Topics

Adding source servers

Add source servers to AWS Elastic Disaster Recovery by installing the AWS Replication Agent (also referred to as "the Agent") on them. The Agent can be installed on both Linux and Windows servers.

Linux installation instructions

Windows installation instructions

Topics

Supported operating systems

AWS Elastic Disaster Recovery allows replication of any physical, virtual or cloud-based source server to the AWS Cloud for a large variety of operating systems.

There may be a number of additional considerations to take into account when determining if your source operating system will be supported by AWS Elastic Disaster Recovery. Ensure that you check the Additional Considerations section in conjunction with the below lists of supported operating systems.

> **Note**
>
> - AWS Elastic Disaster Recovery does not support paravirtualized source servers.
> - AWS Elastic Disaster Recovery only supports 64-bit operating systems built for the x86 system architecture.

Windows

The following Windows operating systems are supported:

- Microsoft Windows Server 2022 64-bit
- Microsoft Windows Server 2019 64-bit
- Microsoft Windows Server 2016 64-bit
- Microsoft Windows Server 2012 R2 64-bit
- Microsoft Windows Server 2012 64-bit
- Microsoft Windows 10 64-bit

The following End of Life Windows operating systems are supported:

- Microsoft Windows Server 2008 R2 64-bit
- Microsoft Windows Server 2008 64-bit
- Microsoft Windows Server 2003 64-bit
- Microsoft Windows 7 64-bit

Linux

The following Linux operating systems are supported:

- Amazon Linux (AL) 1, 2, and 2023
- CentOS 5.6 to 7.9
- Debian Linux 8 to 11
- Oracle Linux (OL) 6.0 to 7.0 (running Unbreakable Enterprise Kernel Release 3 or higher or Red Hat Compatible Kernel only)
- Oracle Linux (OL) 8.5 to 8.9 (running Unbreakable Enterprise Kernel Release 6 or Red Hat Compatible Kernel only) – the following UEK kernels were tested:
  - 5.15.0-200.131.27.el8uek.x86_64
  - 5.15.0-101.103.2.1.el8uek.x86_64
  - 5.15.0-3.60.5.1.el8uek.x86_64
  - 5.4.17–2136.314.6.3.el8uek.x86_64
  - 5.4.17–2136.307.3.1.el8uek.x86_64
  - 5.4.17–2136.300.7.el8uek.x86_64
- Red Hat Enterprise Linux (RHEL) 5.0 to 9.0
- Rocky Linux 8 and 9
- SuSE Linux Enterprise Server 11 SP4 to 15 SP5
- Ubuntu LTS 12.04 to 22.04

Additional Considerations

There may be a number of additional considerations to take into account when determining if your source operating system will be supported by AWS Elastic Disaster Recovery. Ensure that you check the below considerations in conjunction with the above lists of supported operating systems.

Windows

- It is recommended to install all available Windows updates on the server.
- Windows source servers need to have at least 4 GB of free disk space in order to launch a drill or recovery instance successfully.
- When performing a recovery, you must boot the Failback Client with the same boot mode (BIOS or UEFI) as the Windows source server.
- A shutdown (from the OS menu or Windows CLI) of a Windows source server no longer triggers a rescan in AWS DRS once the source server is restarted. Hard reboots, disk changes, and crashes will still trigger a rescan.
- The WMI service must be activated to install the AWS Replication Agent.
- Microsoft Windows Server versions 2012 64-bit and above require .Net Framework version 4.5 or above to be installed by the end user.
- Ensure that the auto sleep function in Windows 10 is disabled. Data replication may be interrupted if the feature is activated.

End of Life Windows

- It is recommended to install all available Windows updates on the server.
- Windows source servers need to have at least 4 GB of free disk space in order to launch a drill or recovery instance successfully.
- The Nitro instance family can only be used with Windows Server 2008 R2 and upwards. Earlier versions are not supported.
- A shutdown (from the OS menu or Windows CLI) of a Windows source server will trigger a rescan in AWS DRS once the source server is restarted.
- The WMI service must be activated to install the AWS Replication Agent.
- Microsoft Windows Server versions 2008 R2 requires .Net Framework version 4.5 or above to be installed by the end user.
- Microsoft Windows Server 2003, 2008, and 2008 R2 have reached their end of life. We recommend that customers upgrade to more modern operating system versions.
- Windows 2003 does not support TLS 1.2, as such, you cannot download the AWS Replication Agent installer directly by using the default browser. The file needs to be copied to the server using another transfer method.

- The AWS Replication Agent and agent installer uses a separate installer file (AwsReplicationWindowsLegacyInstaller.exe) for Microsoft Windows 7, Microsoft Windows Server versions 2003, Microsoft Windows Server 2008, and Microsoft Windows Server 2008 R2 because these OSs are using older versions of software components that cannot be upgraded due to their end-of-life status.
- Windows 2008 x64 requires SP2 and other Microsoft updates to support the SHA-2 signature of the AWS Replication Agent driver.
- Windows 2008 with GPT partitioned system drives are not supported.

Linux

- Ensure that you have Python installed on the source server (version 2.4+, version 3.0+) for Agent installation.
- Only servers using the GRUB bootloader are supported.
- Kernel versions earlier than 2.6.18-164 are not supported by AWS and AWS Elastic Disaster Recovery. Therefore, servers that run these kernel versions cannot be replicated by AWS Elastic Disaster Recovery.
- Recovery of servers using the Oracle ASM Filter Driver is fully supported for Oracle, CentOS, and Red Hat Enterprise Linux (RHEL) 6.0 to 8.7.
- Secure Boot is not supported in Linux.
- When performing a recovery for a Linux server, you must boot the Failback Client with BIOS boot mode.
- A reboot of supported Linux servers no longer triggers a rescan in AWS DRS once the source server is restarted. Hard reboots, disk changes, and crashes will still trigger a rescan. Supported OSs include:
  - RHEL/CentOS/Oracle Linux 6+ (kernel versions 2.6.32–431 and above)
  - SUSE 12+
  - Ubuntu 16+ LTS
  - AL2
  - Rocky 8+
  - Debian 9+
- AWS requires that servers running Red Hat Enterprise Linux (RHEL) must have Cloud Access (BYOL) licenses in order to be recovered to AWS. Note that servers running RHEL Cloud Access

Gold Images allow you to access AWS Red Hat Update Infrastructure (RHUI), Red Hat Satellite, or Red Hat Subscription Manager (RHSM). If you are using RHEL Cloud Access Gold Images, you will not be able to access RHUI upon failover to AWS unless you link your AWS account to your Red Hat account via the Red Hat portal, and select the Gold image AMI in the launch template.

- You must select an AWS provided RHEL AMI in the Launch Template for servers running Red Hat Enterprise Linux (RHEL) Pay as You Go (PAYG) images. This will allow access to RHUI after failover. Note that usage of these images will incur EC2 charges for software and infrastructure per AWS Marketplace rates.
- Only Kernel 3.x or above are supported for Debian/Ubuntu on AWS.
- Kernel versions 2.6.32–71 are not supported in RHEL 6.0/CentOS 6.0/Oracle Linux 6.0 on AWS.
- Nitro instance types will work with RHEL 7.4+/CentOS 7.4+/Oracle Linux 7.4+. This specific limitation does not apply to other instance type families.
- A pre-requirement for installing the AWS Replication Agent on RHEL 8/CentOS 8/OL 8 is first running the following:

```
sudo yum install elfutils-libelf-devel
```

- Amazon Linux 1 is only supported for AWS to AWS recovery.
- For SUSE Linux (SLES) 11 to work, you must install the xen drivers before installing the AWS Replication Agent. You must reboot the server after installing the xen drivers (before installing the AWS Replication Agent). Use the following command to install the drivers:

```
zypper install -y xen-kmp-default
```

- Machines that boot off a disk configured with GPT partitioning need to have the package 'grub2-pc-modules' installed.
- Linux kernel version upto 6.5 is supported.

Installation requirements

Before installing the AWS Replication Agent on your source servers, ensure that they meet the following requirements:

Topics

- Linux installation requirements
- Windows installation requirements

General requirements

- Ensure that the source server operating system is supported by AWS. Learn more about supported operating systems.
- Ensure that your setup meets all replication networking requirements. Learn more about network requirements.
- Ensure MAC address stability – ensure that the MAC addresses of the source servers do not change upon a reboot or any other common changes in your network environment. AWS Elastic Disaster Recovery calculates the unique ID of the source server from the MAC address. When a MAC address changes, AWS Elastic Disaster Recovery is no longer able to correctly identify the source server. Consequently, replication will stop. If this happens, you will need to reinstall the AWS Replication Agent and start replication from the beginning.

> **Note**
>
> Elastic Disaster Recovery Agents can only be installed on instances that are in AWS Regions that are supported by Elastic Disaster Recovery. In case of AWS-AWS disaster recovery (in-AWS), Elastic Disaster Recovery should be initialized in both source and target region (done by going through the initialization wizard).

Source server requirements

The following are universal requirements for both Linux and Windows source servers:

- Root directory – Verify that your source server has at least 4 GB of free disk space on the root directory (/) .
- RAM – Verify that your source server has at least 300 MB of free RAM to run the AWS Replication Agent.

Note

AWS Elastic Disaster Recovery does not support paravirtualized source servers.

Note

The AWS Replication Agent installer supports multipath.

Linux installation requirements

Ensure that your Linux source server meets the following installation requirements prior to installing the AWS Replication Agent:

- Python is installed on the server – Python 2 (2.4 or above) or Python 3 (3.0 or above).
- Verify that you have at least 4 GB of free disk space on the root directory (/) of your source server for the installation. To check the available disk space on the root directory, run the following command: *df -h /*
- Free disk space on the */tmp* directory – for the duration of the installation process only, verify that you have at least 500 MB of free disk space on the */tmp* directory. To check the available disk space on the /tmp directory run the following command: *df -h /tmp*

  After you have entered the above commands for checking the available disk space, the results will be displayed as follows:

```
ubuntu@Linux-1:~$ df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.4G  6.0G  19% /
ubuntu@Linux-1:~$ df -h /tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.4G  6.0G  19% /tmp
```

- The active bootloader software is GRUB 1 or 2.
- Ensure that /tmp is mounted as read+write.
- Ensure that /tmp is mounted with the exec option. Verify that the /tmp directory is mounted in a way that allows you to run scripts and applications from it.

  To verify that the /tmp directory is mounted without the noexec option, run the following command: sudo mount | grep '/tmp'

If the result is similar to the following example, it means that the issue exists in your OS: /dev/xvda1 on /tmp type ext4 (rw,noexec)

|To fix and remove the noexec option from the mounted /tmp directory, run the following command: sudo mount -o remount,exec /tmp

The following example illustrates the troubleshooting procedure:

```
ubuntu@Linux-1:~$ sudo mount | grep '/tmp'
/dev/xvda1 on /tmp type ext4 (rw,noexec)
ubuntu@Linux-1:~$ sudo mount -o remount,exec /tmp
ubuntu@Linux-1:~$ sudo mount | grep '/tmp'
/dev/xvda1 on /tmp type ext4 (rw)
```

- The AWS Elastic Disaster Recovery user needs to be a user in the sudoers list (a user who can perform sudo).
- Ensure that the dhclient package is installed. If not, please install the package. (run yum install dhclient in CMD).
- Verify that you have kernel-devel/linux-headers installed that are exactly the same version as the kernel you are running.

  The version number of the kernel headers should be completely identical to the version number of the kernel. To handle this issue, follow these steps:

  1. Identify the version of your running kernel.

     To identify the version of your running kernel, run the following command:

     uname -r

     ```
     [root@ip-192-168-20-156 ~]# uname -r
     4.14.177-107.254.amzn1.x86_64
     [root@ip-192-168-20-156 ~]#
     ```

     The 'uname -r' output version should match the version of one of the installed kernel headers packages (kernel-devel-<version number> / linux-headers-<version number>).

  2. Identify the version of your kernel-devel/linux-headers.

     To identify the version of your running kernel, run the following command:

     On RHEL/CENTOS/Oracle/SUSE:

     rpm -qa | grep kernel

```
[root@ip-192-168-20-156 ~]# rpm -qa |grep kernel
kernel-4.14.177-107.254.amzn1.x86_64
kernel-headers-4.14.181-108.257.amzn1.x86_64
kernel-devel-4.14.177-107.254.amzn1.x86_64
kernel-tools-4.14.181-108.257.amzn1.x86_64
[root@ip-192-168-20-156 ~]#
```

> **Note**
>
> This command looks for kernel-devel.

On Debian/Ubuntu: apt-cache search linux-headers

```
ubuntu@Linux-1:~$ apt-cache search linux-headers
linux-headers-3.13.0-24 - Header files related to Linux kernel version
 3.13.0
linux-headers-3.13.0-24-generic - Linux kernel headers for version 3.1
3.0 on 64 bit x86 SMP
linux-headers-3.13.0-24-lowlatency - Linux kernel headers for version
3.13.0 on 64 bit x86 SMP
```

3. Verify that the folder that contains the kernel-devel/linux-headers is not a symbolic link.

   Sometimes, the content of the kernel-devel/linux-headers, which match the version of the kernel, is actually a symbolic link. In this case, you will need to remove the link before installing the required package.

   To verify that the folder that contains the kernel-devel/linux-headers is not a symbolic link, run the following command:

   On RHEL/CENTOS/Oracle:

   ls -l /usr/src/kernels

   On Debian/Ubuntu/SUSE:

   ls -l /usr/src

```
ubuntu@Linux-1:~$  ls -l /usr/src
total 8
lrwxrwxrwx  1 root root   41 May 29 15:40 3.13.0-116-generic -> /usr/src/linux-
headers-3.13.0-116-generic
drwxr-xr-x 24 root root 4096 Apr  5 20:43 linux-headers-3.13.0-116
drwxr-xr-x  7 root root 4096 Apr  5 20:43 linux-headers-3.13.0-116-generic
ubuntu@Linux-1:~$
```

In the above example, the results show that the linux-headers are not a symbolic link.

4. [If a symbolic link exists] Delete the symbolic link.

   If you found that the content of the kernel-devel/linux-headers, which match the version of the kernel, is a symbolic link, you need to delete the link. Run the following command: rm /usr/src/<LINK NAME>

   For example: rm /usr/src/linux-headers-4.4.1

5. Install the correct kernel-devel/linux-headers from the repositories.

   If none of the already installed kernel-devel/linux-headers packages match your running kernel version, you need to install the matching package.

   > **Note**
   >
   > You can have several kernel headers versions simultaneously on your OS, and you can therefore safely install new kernel headers packages in addition to your existing ones (without uninstalling the other versions of the package.) A new kernel headers package does not impact the kernel, and does not overwrite older versions of the kernel headers.

   > **Note**
   >
   > For everything to work, you need to install a kernel headers package with the exact same version number of the running kernel.
   > To install the correct kernel-devel/linux-headers, run the following command:
   > On RHEL/CENTOS/Oracle/SUSE:
   > sudo yum install kernel-devel-`uname -r`
   > On Oracle with Unbreakable Enterprise Kernel:
   > sudo yum install kernel-uek-devel-`uname -r`
   > On Debian/Ubuntu:
   > sudo apt-get install linux-headers-`uname -r`

6. [If no matching package was found] Download the matching kernel-devel/linux-headers package.

If no matching package was found on the repositories configured on your server, you can download it manually from the Internet and then install it.

To download the matching kernel-devel/linux-headers package, navigate to the following sites:

- RHEL, CENTOS, Oracle, and SUSE package directory
- Debian package directory
- Ubuntu package directory

Windows installation requirements

Note

Ensure that your source server operating system is supported. Learn more about supported operating systems.

Note

Ensure that your source server meets the Agent installation hardware requirements, including:

- At least 4 GB of free disk space on the root directory (/)
- At least 300 MB of free RAM

Learn more about AWS Replication Agent installation hardware requirements.

Installing the AWS Replication Agent

You must install the AWS Replication Agent on each source server that you want to add to AWS Elastic Disaster Recovery. Agent installation is composed of the following steps:

Topics

- Installing the AWS Replication Agent in AWS
- Installation instructions
- Installing the agent on a secured network
- Uninstalling the agent
- Reinstalling the agent
- Supporting marketplace licenses

Generating the required AWS credentials

In order to install the AWS Replication Agent, you must first generate the required AWS credentials. You can create temporary credentials with AWS STS.

Temporary credentials

Before you install the AWS Replication Agent, you need to generate temporary AWS security credentials. The temporary credentials provided by AWS Elastic Disaster Recovery utilize a similar mechanism to the one used by IAM Roles Anywhere.

To create temporary credentials, take the following steps:

1. Create a new IAM Role with the **AWSElasticDisasterRecoveryAgentInstallationPolicy** policy.
2. Request temporary security credentials via AWS STS using the AssumeRole API.

Learn more about how temporary credentials operate.

Installing the AWS Replication Agent in AWS

When installing an AWS Replication Agent on an AWS EC2 instance (when the source and recovery servers are both in AWS Regions), you don't need to generate credentials. Instead, you can use an instance profile with the required IAM policy:

- Go to the EC2 console and select your EC2 instance.
- From the top right-hand menu, select **Actions > Security > Modify IAM role**.
- Use a role that contains the AWSElasticDisasterRecoveryEc2InstancePolicy policy.

  If none exists, click **Create new IAM role**, attach the policy and return to the EC2 console window.

- Select your new role from the drop-down list and click **Update**.

Installation instructions

Once you have generated the required AWS credentials, you can install the AWS Replication Agent on your source servers. There are separate installation instructions for Linux and for Windows. Each operating system has its own installer.

Topics

Linux

To install the agent on a Linux source server, you should ensure that your source meets all the requirements list in the Supported Operating Systems documentation.

Before installing, please ensure that you are aware of the following:

- You need root privileges to run the Agent installer file on a Linux server. Alternatively, you can run the Agent Installer file with sudo permissions.
- The Linux installer creates the **"aws-replication"** group and **"aws-replication"** user within that group. The Agent will run within the context of the newly created user. Agent installation will attempt to add the user to **"sudoers"**. Installation will fail if the Agent is unable to add the newly created **"aws-replication"** user to **"sudoers"**.

1. Download the agent installer `aws-replication-installer-init` onto your Linux source server.

   The Agent installer download location follows this format:

   ```
   https://aws-elastic-disaster-recovery-<REGION>.s3.<REGION>.amazonaws.com/latest/linux/aws-replication-installer-init
   ```

Note

Replace <REGION> with the AWS Region into which you are replicating.

The following is an example for downloading the installer file from the us-east-1 region:

wget

```
wget -O ./aws-replication-installer-init https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/aws-replication-installer-init
```

curl

```
curl -o aws-replication-installer-init https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/aws-replication-installer-init
```

Note

If you are using a legacy Linux OS that does not support TLS 1.2, you need to download the installer on a different server with an OS that supports TLS 1.2 and copy it to the legacy servers you intend to install the agent on.

The command line will indicate when the installer has been successfully downloaded.

Important

If you need to validate the installer hash, the correct hash can be found here:
`https://aws-elastic-disaster-recovery-hashes-<REGION>.s3.<REGION>.amazonaws.com/latest/linux/aws-replication-installer-init.sha512`
Replace <REGION> with the AWS Region into which you are replicating
For example, when using the **us-east-1** Region
`https://aws-elastic-disaster-recovery-hashes-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/aws-replication-installer-init.sha512`

> **Note**
>
> AWS Regions that are not opt-in also support the shorter installer path: `https://aws-elastic-disaster-recovery-<REGION>.s3.amazonaws.com/latest/linux/aws-replication-installer-init`. Replace <REGION> with the AWS Region into which you are replicating.

> **Note**
>
> If you are using a Windows Servers of versions 2016 or older, and are using PowerShell to download the installer, you need to enable TLS 1.2: `[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'`

2. Use the following command on your source server in order to run the installation script.

```
chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
```

> **Note**
>
> To install the agent on a secured network, learn about the additional required configurations.

If you require additional customization, you can add a variety of parameters to the installer script in order to manipulate the way the Agent is installed on your server. See the AWS Replication Agent Installer Parameters for more information.

The installer will confirm that the installation of the AWS Replication Agent has started.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
The installation of the AWS Replication Agent has started.
```

3. The installer will prompt you to enter your **AWS Region Name**, the **AWS Access Key ID** and **AWS Secret Access Key** that you previously generated. Enter the complete AWS Region name (for example, eu-central-1), the full AWS Access Key ID and the full AWS Secret Access Key.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
The installation of the AWS Replication Agent has started.
AWS Region name: us-east-1
AWS Access Key ID: AKIAI0SF0DNN71EXAMPLE
AWS Secret Access Key: wJalrXUtnFEMI/K71MDENG/bPxRfiCYEXAMPLEKEY
```

> **Note**
>
> You can also enter these values as part of the installation script command parameters. If you do not enter these parameters as part of the installation script, you will be prompted to enter them one by one as described above. (for example: `chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init --region regionname --aws-access-key-id AKIAIOSFODNN71EXAMPLE --aws-secret-access-key wJalrXUtnFEMI/K71MDENG/bPxRfiCYEXAMPLEKEY`)

4. Once you have entered your credentials, the installer will identify volumes for replication. The installer will display the identified disks and prompt you to choose the disks you want to replicate.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
...
AWS Secret Access Key: wJalrXUtnFEMI/K71MDENG/bPxRfiCYEXAMPLEKEY
Identifying volumes for replication.
Choose the disks you want to replication. Your disks are: /dev/sda,/dev/xvda
To replication some of the disks, type the path of the disks, separated with a comma
 (for example, /dev/sda,/dev/sdb).
To replication all disks, press Enter:
```

To replicate some of the disks, type the path of the disks, separated by a comma, as illustrated in the installer (such as: /dev/sda, /dev/sdb, etc). To replicate all of the disks, press Enter. The installer will identify the selected disks and print their size.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
...
To replication some of the disks, type the path of the disks, separated with a comma
 (for example, /dev/sda,/dev/sdb).
To replication all disks, press Enter:
Identified volume for replication: /dev/xvda of size 8 GiB
```

The installer will confirm that all disks were successfully identified.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
...
Identified volume for replication: /dev/xvda of size 8 GiB
All volumes for replication were successfully identified.
```

Note

When identifying specific disks for replication, do not use apostrophes, brackets, or disk paths that do not exist. Type only existing disk paths. Each disk you selected for replication is displayed with the caption **Disk to replicate identified**. However, the displayed list of identified disks for replication may differ from the data you entered. This difference can due to several reasons:

- The root disk of the source server is always replicated, whether you select it or not. Therefore, it always appears on the list of identified disks for replication.
- AWS Elastic Disaster Recovery replicates whole disks. Therefore, if you choose to replicate a partition, its entire disk will appear on the list and will later be replicated. If several partitions on the same disk are selected, then the disk encompassing all of them will appear only once on the list.
- Incorrect disks may be chosen by accident. Ensure that the correct disks have been chosen.

Important

If disks are disconnected from a server, AWS Elastic Disaster Recovery can no longer replicate them, so they are removed from the list of replicated disks. When they are reconnected, the AWS Replication Agent cannot know that these were the same disks that were disconnected and therefore does not add them automatically. To add the disks after they are reconnected, rerun the AWS Replication Agent installer on the server. Note that the returned disks will need be replicated from the beginning. Any disk size changes will be automatically identified, but this will also cause a resync. Perform a test after installing the Agent to ensure that the correct disks have been added.

5. After all of the disks that will be replicated have been successfully identified, the installer will download and install the AWS Replication Agent on the source server.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
...
Identified volume for replication: /dev/xvda of size 8 GiB
All volumes for replication were successfully identified.
Downloading the AWS Replication Agent onto the source server... Finished
Installing the AWS Replication Agent onto the source server... Finished
```

6. Once the AWS Replication Agent is installed, the server will be added to the AWS Elastic Disaster Recovery console and will undergo the initial sync process. The installer will provide you with the source server's ID.

```
$ chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init
...
Installing the AWS Replication Agent onto the source server... Finished
Syncing the source server with the AWS Elastic Disaster Recovery console... Finished
The following is the source server ID: s-3146f90b19example
The AWS Replication Agent was successfully
 installed.
$
```

   You can review this process in real time on the **Source servers** page.

Windows

To install the AWS Replication Agent on a Windows source server, you should ensure that your source meets all the requirements list in the Supported Operating Systems documentation.

Prior to installing the AWS Replication Agent, please ensure that you are aware of the following:

- You need to run the agent installer file as an Administrator on each Windows server.
- We recommend using Windows PowerShell, which supports the 'Ctrl+V' shortcut for pasting. Windows Command Prompt (cmd) does not support this functionality.

Before installing the AWS Replication Agent, `AWSReplicationWindowsInstaller.exe`, it needs to be downloaded. Copy or distribute the downloaded agent installer to each Windows source server that you want to add to AWS Elastic Disaster Recovery.

The agent installer follows the following format:

`https://aws-elastic-disaster-recovery-<REGION>.s3.<REGION>.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe`

> **Note**
>
> Replace `<REGION>` with the AWS Region into which you are replicating.

The following is an example URL for downloading the installer file from the us-east-1 region :

```
https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
```

> **Note**
>
> AWS Regions that are not opt-in also support the shorter installer path: `https://aws-elastic-disaster-recovery-<REGION>.s3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe` . Replace `<REGION>` with the AWS Region into which you are replicating.

> **Note**
>
> If you are using a Windows Servers of versions 2016 or older, and are using PowerShell to download the installer, you need to enable TLS 1.2: `[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'`

> **Note**
>
> Microsoft Windows Server versions 2003, 2003 R2, 2008, and 2008 R2 use a unique version of the AWS Replication Agent that is only valid for legacy Windows OSs (AwsReplicationWindowsLegacyInstaller.exe). DO NOT use this installer file to install

the agent on any other OS types. You can download it from `https://aws-elastic-disaster-recovery-<REGION>.s3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe` . Replace `<REGION>` with the AWS Region into which you are replicating.

AWS Replication Agent for Windows download URL, for each supported AWS Region

Region name	Region identity	Download Link
US East (N. Virginia)	us-east-1	https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
US East (Ohio)	us-east-2	https://aws-elastic-disaster-recovery-us-east-2.s3.us-east-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
US West (N. California)	us-west-1	https://aws-elastic-disaster-recovery-us-west-1.s3.us-west-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
US West (Oregon)	us-west-2	https://aws-elastic-disaster-recovery-us-west-2.s3.us-west-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Hong Kong)	ap-east-1	https://aws-elastic-disaster-recovery-ap-east-1.s3.ap-east-1.amazonaws.com/

Region name	Region identity	Download Link
		latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Tokyo)	ap-northeast-1	https://aws-elastic-disaster-recovery-ap-northeast-1.s3.ap-northeast-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Seoul)	ap-northeast-2	https://aws-elastic-disaster-recovery-ap-northeast-2.s3.ap-northeast-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Osaka)	ap-northeast-3	https://aws-elastic-disaster-recovery-ap-northeast-3.s3.ap-northeast-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Singapore)	ap-southeast-1	https://aws-elastic-disaster-recovery-ap-southeast-1.s3.ap-southeast-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe

Region name	Region identity	Download Link
Asia Pacific (Sydney)	ap-southeast-2	https://aws-elastic-disaster-recovery-ap-southeast-2.s3.ap-southeast-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Jakarta)	ap-southeast-3	https://aws-elastic-disaster-recovery-ap-southeast-3.s3.ap-southeast-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Melbourne)	ap-southeast-4	https://aws-elastic-disaster-recovery-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Mumbai)	ap-south-1	https://aws-elastic-disaster-recovery-ap-south-1.s3.ap-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Asia Pacific (Hyderabad)	ap-south-2	https://aws-elastic-disaster-recovery-ap-south-2.s3.ap-south-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe

Region name	Region identity	Download Link
Europe (Frankfurt)	eu-central-1	https://aws-elastic-disaster-recovery-eu-central-1.s3.eu-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Zurich)	eu-central-2	https://aws-elastic-disaster-recovery-eu-central-2.s3.eu-central-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Stockholm)	eu-north-1	https://aws-elastic-disaster-recovery-eu-north-1.s3.eu-north-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Milan)	eu-south-1	https://aws-elastic-disaster-recovery-eu-south-1.s3.eu-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Spain)	eu-south-2	https://aws-elastic-disaster-recovery-eu-south-2.s3.eu-south-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Ireland)	eu-west-1	https://aws-elastic-disaster-recovery-eu-west-1.s3.eu-west-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe

Region name	Region identity	Download Link
Europe (London)	eu-west-2	https://aws-elastic-disaster-recovery-eu-west-2.s3.eu-west-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Europe (Paris)	eu-west-3	https://aws-elastic-disaster-recovery-eu-west-3.s3.eu-west-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Canada (Central)	ca-central-1	https://aws-elastic-disaster-recovery-ca-central-1.s3.ca-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Middle East (UAE)	me-central-1	https://aws-elastic-disaster-recovery-me-central-1.s3.me-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Middle East (Bahrain)	me-south-1	https://aws-elastic-disaster-recovery-me-south-1.s3.me-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Israel (Tel Aviv)	il-central-1	https://aws-elastic-disaster-recovery-il-central-1.s3.il-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe

Region name	Region identity	Download Link
South America (São Paulo)	sa-east-1	https://aws-elastic-disaster-recovery-sa-east-1.s3.sa-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe
Africa (Cape Town)	af-south-1	https://aws-elastic-disaster-recovery-af-south-1.s3.af-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe

Validating the downloaded AWS Replication Agent installer for Windows.

Important

If you need to validate the installer hash, the correct hash can be found here: `https://aws-elastic-disaster-recovery-hashes-<REGION>.s3.<REGION>.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512`
Replace <REGION> with the AWS Region into which you are replicating, for example: us-east-1: `https://aws-elastic-disaster-recovery-hashes-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512`

Region name	Region identity	SHA512 Hash Download Link
US East (N. Virginia)	us-east-1	https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/AwsReplic

Region name	Region identity	SHA512 Hash Download Link
		ationWindowsInstaller.exe.sha512
US East (Ohio)	us-east-2	https://aws-elastic-disaster-recovery-us-east-2.s3.us-east-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
US West (N. California)	us-west-1	https://aws-elastic-disaster-recovery-us-west-1.s3.us-west-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
US West (Oregon)	us-west-2	https://aws-elastic-disaster-recovery-us-west-2.s3.us-west-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Hong Kong)	ap-east-1	https://aws-elastic-disaster-recovery-ap-east-1.s3.ap-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Asia Pacific (Tokyo)	ap-northeast-1	https://aws-elastic-disaster-recovery-ap-northeast-1.s3.ap-northeast-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Seoul)	ap-northeast-2	https://aws-elastic-disaster-recovery-ap-northeast-2.s3.ap-northeast-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Osaka)	ap-northeast-3	https://aws-elastic-disaster-recovery-ap-northeast-3.s3.ap-northeast-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Singapore)	ap-southeast-1	https://aws-elastic-disaster-recovery-ap-southeast-1.s3.ap-southeast-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Sydney)	ap-southeast-2	https://aws-elastic-disaster-recovery-ap-southeast-2.s3.ap-southeast-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Asia Pacific (Jakarta)	ap-southeast-3	https://aws-elastic-disaster-recovery-ap-southeast-3.s3.ap-southeast-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Melbourne)	ap-southeast-4	https://aws-elastic-disaster-recovery-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Mumbai)	ap-south-1	https://aws-elastic-disaster-recovery-ap-south-1.s3.ap-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Asia Pacific (Hyderabad)	ap-south-2	https://aws-elastic-disaster-recovery-ap-south-2.s3.ap-south-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Frankfurt)	eu-central-1	https://aws-elastic-disaster-recovery-eu-central-1.s3.eu-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Europe (Zurich)	eu-central-2	https://aws-elastic-disaster-recovery-eu-central-2.s3.eu-central-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Stockholm)	eu-north-1	https://aws-elastic-disaster-recovery-eu-north-1.s3.eu-north-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Milan)	eu-south-1	https://aws-elastic-disaster-recovery-eu-south-1.s3.eu-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Spain)	eu-south-2	https://aws-elastic-disaster-recovery-eu-south-2.s3.eu-south-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Ireland)	eu-west-1	https://aws-elastic-disaster-recovery-eu-west-1.s3.eu-west-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Europe (London)	eu-west-2	https://aws-elastic-disaster-recovery-eu-west-2.s3.eu-west-2.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Europe (Paris)	eu-west-3	https://aws-elastic-disaster-recovery-eu-west-3.s3.eu-west-3.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Canada (Central)	ca-central-1	https://aws-elastic-disaster-recovery-ca-central-1.s3.ca-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Middle East (UAE)	me-central-1	https://aws-elastic-disaster-recovery-me-central-1.s3.me-central-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Middle East (Bahrain)	me-south-1	https://aws-elastic-disaster-recovery-me-south-1.s3.me-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
South America (São Paulo)	sa-east-1	https://aws-elastic-disaster-recovery-sa-east-1.s3.sa-east-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512
Africa (Cape Town)	af-south-1	https://aws-elastic-disaster-recovery-af-south-1.s3.af-south-1.amazonaws.com/latest/windows/AwsReplicationWindowsInstaller.exe.sha512

AWS Replication Agent for End-of-Life Windows download URL, for each supported AWS Region

Region name	Region identity	Download Link
US East (N. Virginia)	us-east-1	https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
US East (Ohio)	us-east-2	https://aws-elastic-disaster-recovery-us-east-2.s3.us-east-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
US West (N. California)	us-west-1	https://aws-elastic-disaster-recovery-us-west-1.s3.us-

Region name	Region identity	Download Link
		west-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
US West (Oregon)	us-west-2	https://aws-elastic-disaster-recovery-us-west-2.s3.us-west-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Hong Kong)	ap-east-1	https://aws-elastic-disaster-recovery-ap-east-1.s3.ap-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Tokyo)	ap-northeast-1	https://aws-elastic-disaster-recovery-ap-northeast-1.s3.ap-northeast-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Seoul)	ap-northeast-2	https://aws-elastic-disaster-recovery-ap-northeast-2.s3.ap-northeast-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe

Region name	Region identity	Download Link
Asia Pacific (Osaka)	ap-northeast-3	https://aws-elastic-disaster-recovery-ap-northeast-3.s3.ap-northeast-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Singapore)	ap-southeast-1	https://aws-elastic-disaster-recovery-ap-southeast-1.s3.ap-southeast-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Sydney)	ap-southeast-2	https://aws-elastic-disaster-recovery-ap-southeast-2.s3.ap-southeast-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Jakarta)	ap-southeast-3	https://aws-elastic-disaster-recovery-ap-southeast-3.s3.ap-southeast-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Melbourne)	ap-southeast-4	https://aws-elastic-disaster-recovery-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe

Region name	Region identity	Download Link
Asia Pacific (Mumbai)	ap-south-1	https://aws-elastic-disaster-recovery-ap-south-1.s3.ap-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Asia Pacific (Hyderabad)	ap-south-2	https://aws-elastic-disaster-recovery-ap-south-2.s3.ap-south-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Frankfurt)	eu-central-1	https://aws-elastic-disaster-recovery-eu-central-1.s3.eu-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Zurich)	eu-central-2	https://aws-elastic-disaster-recovery-eu-central-2.s3.eu-central-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Stockholm)	eu-north-1	https://aws-elastic-disaster-recovery-eu-north-1.s3.eu-north-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe

Region name	Region identity	Download Link
Europe (Milan)	eu-south-1	https://aws-elastic-disaster-recovery-eu-south-1.s3.eu-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Spain)	eu-south-2	https://aws-elastic-disaster-recovery-eu-south-2.s3.eu-south-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Ireland)	eu-west-1	https://aws-elastic-disaster-recovery-eu-west-1.s3.eu-west-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (London)	eu-west-2	https://aws-elastic-disaster-recovery-eu-west-2.s3.eu-west-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Europe (Paris)	eu-west-3	https://aws-elastic-disaster-recovery-eu-west-3.s3.eu-west-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe

Region name	Region identity	Download Link
Canada (Central)	ca-central-1	https://aws-elastic-disaster-recovery-ca-central-1.s3.ca-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Middle East (UAE)	me-central-1	https://aws-elastic-disaster-recovery-me-central-1.s3.me-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Middle East (Bahrain)	me-south-1	https://aws-elastic-disaster-recovery-me-south-1.s3.me-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
South America (São Paulo)	sa-east-1	https://aws-elastic-disaster-recovery-sa-east-1.s3.sa-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe
Africa (Cape Town)	af-south-1	https://aws-elastic-disaster-recovery-af-south-1.s3.af-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe

Validating the downloaded AWS Replication Agent installer for End of Life Windows.

Important

If you need to validate the installer hash, the correct hash can be found here: `https://aws-elastic-disaster-recovery-hashes-<REGION>.s3.<REGION>.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512`
Replace `<REGION>` with the AWS Region into which you are replicating, for example: us-east-1:
`https://aws-elastic-disaster-recovery-hashes-us-east-1.s3.us-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512`

Region name	Region identity	SHA512 Hash Download Link
US East (N. Virginia)	us-east-1	https://aws-elastic-disaster-recovery-us-east-1.s3.us-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
US East (Ohio)	us-east-2	https://aws-elastic-disaster-recovery-us-east-2.s3.us-east-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
US West (N. California)	us-west-1	https://aws-elastic-disaster-recovery-us-west-1.s3.us-west-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
US West (Oregon)	us-west-2	https://aws-elastic-disaster-recovery-us-west-2.s3.us-west-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Hong Kong)	ap-east-1	https://aws-elastic-disaster-recovery-ap-east-1.s3.ap-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Tokyo)	ap-northeast-1	https://aws-elastic-disaster-recovery-ap-northeast-1.s3.ap-northeast-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Seoul)	ap-northeast-2	https://aws-elastic-disaster-recovery-ap-northeast-2.s3.ap-northeast-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Osaka)	ap-northeast-3	https://aws-elastic-disaster-recovery-ap-northeast-3.s3.ap-northeast-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Asia Pacific (Singapore)	ap-southeast-1	https://aws-elastic-disaster-recovery-ap-southeast-1.s3.ap-southeast-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Sydney)	ap-southeast-2	https://aws-elastic-disaster-recovery-ap-southeast-2.s3.ap-southeast-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Jakarta)	ap-southeast-3	https://aws-elastic-disaster-recovery-ap-southeast-3.s3.ap-southeast-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Melbourne)	ap-southeast-4	https://aws-elastic-disaster-recovery-ap-southeast-4.s3.ap-southeast-4.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Asia Pacific (Mumbai)	ap-south-1	https://aws-elastic-disaster-recovery-ap-south-1.s3.ap-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Asia Pacific (Hyderabad)	ap-south-2	https://aws-elastic-disaster-recovery-ap-south-2.s3.ap-south-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Frankfurt)	eu-central-1	https://aws-elastic-disaster-recovery-eu-central-1.s3.eu-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Zurich)	eu-central-2	https://aws-elastic-disaster-recovery-eu-central-2.s3.eu-central-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Stockholm)	eu-north-1	https://aws-elastic-disaster-recovery-eu-north-1.s3.eu-north-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Milan)	eu-south-1	https://aws-elastic-disaster-recovery-eu-south-1.s3.eu-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Europe (Spain)	eu-south-2	https://aws-elastic-disaster-recovery-eu-south-2.s3.eu-south-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Ireland)	eu-west-1	https://aws-elastic-disaster-recovery-eu-west-1.s3.eu-west-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (London)	eu-west-2	https://aws-elastic-disaster-recovery-eu-west-2.s3.eu-west-2.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Europe (Paris)	eu-west-3	https://aws-elastic-disaster-recovery-eu-west-3.s3.eu-west-3.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Canada (Central)	ca-central-1	https://aws-elastic-disaster-recovery-ca-central-1.s3.ca-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

Region name	Region identity	SHA512 Hash Download Link
Middle East (UAE)	me-central-1	https://aws-elastic-disaster-recovery-me-central-1.s3.me-central-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Middle East (Bahrain)	me-south-1	https://aws-elastic-disaster-recovery-me-south-1.s3.me-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
South America (São Paulo)	sa-east-1	https://aws-elastic-disaster-recovery-sa-east-1.s3.sa-east-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512
Africa (Cape Town)	af-south-1	https://aws-elastic-disaster-recovery-af-south-1.s3.af-south-1.amazonaws.com/latest/windows_legacy/AwsReplicationWindowsLegacyInstaller.exe.sha512

1. Run the agent installer file `AWSReplicationWindowsInstaller.exe` as an Administrator.

   The installer will confirm that the installation of the AWS Replication Agent has started.

   ```
   The installation of the AWS Replication Agent has started.
   ```

2. The installer will prompt you to enter your **AWS Region Name**, the **AWS Access Key ID** and the **AWS Secret Access Key** that you previously generated. Enter the complete AWS Region name (for example: eu-central-1), and the full AWS Access Key ID and AWS Secret Access Key.

```
The installation of the AWS Replication Agent has started.
AWS Region name: us-east-1
AWS Access Key ID: AKIAI0SF0DNN71EXAMPLE
AWS Secret Access Key: wJalrXUtnFEMI/K71MDENG/bPxRfiCYEXAMPLEKEY
```

> **Note**
>
> You can also enter these values as part of the installation script command parameters. If you do not enter these parameters as part of the installation script, you will be prompted to enter them one by one as described above. (for example: `AwsReplicationWindowsInstaller.exe --region regionname --aws-access-key-id AKIAIOSFODNN7EXAMPLE --aws-secret-access-key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`)

If you require additional customization, you can add a variety of parameters to the installation script in order to manipulate the way the Agent is installed on your server. See the Installer Parameters for more information.

3. Once you have entered your credentials, the installer will verify that the source server has enough free disk space for Agent installation and identify volumes for replication. The installer will display the identified disks and prompt you to choose the disks you want to replicate.

```
...
AWS Secret Access Key: wJalrXUtnFEMI/K71MDENG/bPxRfiCYEXAMPLEKEY
Verifying that the source server has enough free disk space to install the AWS
 Replication Agent.
(a minimum of 2GB of free disk space is required)
Identifying volumes for replication.
Choose the disks you want to replication. Your disks are: c:
To replication some of the disks, type the path of the disks, separated with a
 comma (for example, C:,D:).
To replication all disks, press Enter:
```

To replicate some of the disks, type the path of the disks, separated by a comma, as illustrated in the installer (for example: C:, D:, etc). To replicate all of the disks, press **Enter**. The installer will identify the selected disks and print their size.

```
...
Identifying volumes for replication.
Choose the disks you want to replication. Your disks are: c:
To replication some of the disks, type the path of the disks, separated with a
 comma (for example, C:,D:).
To replication all disks, press Enter:
Disk to replciate identified: c:0 of size 30GiB
```

The installer will confirm that all of the disks were successfully identified.

```
...
Identifying volumes for replication.
Choose the disks you want to replication. Your disks are: c:
To replication some of the disks, type the path of the disks, separated with a
 comma (for example, C:,D:).
To replication all disks, press Enter:
Disk to replciate identified: c:0 of size 30GiB
All volumes for replication were successfully identified
```

Note

When identifying specific disks for replication, do not use apostrophes, brackets, or disk paths that do not exist. Type only existing disk paths. Each disk that you selected for replication is displayed with the caption **Disk to replicate identified**. However, the displayed list of identified disks for replication may differ from the data you entered. This difference can due to several reasons:

- The root disk of the source server is always replicated, whether you select it or not. Therefore, it always appears on the list of identified disks for replication.
- AWS Elastic Disaster Recovery replicates whole disks. Therefore, if you choose to replicate a partition, its entire disk will appear on the list and will later be replicated. If several partitions on the same disk are selected, then the disk encompassing all of them will only appear once on the list.

- Incorrect disks may be chosen by accident. Ensure that the correct disks have been chosen.

Important

If disks are disconnected from a server, AWS Elastic Disaster Recovery can no longer replicate them, so they are removed from the list of replicated disks. When they are re-connected, the AWS Replication Agent cannot know that these were the same disks that were disconnected and therefore does not add them automatically. To add the disks after they are reconnected, rerun the AWS Replication Agent installer on the server.

Note that the returned disks will need be replicated from the beginning. Any disk size changes will be automatically identified, but will also cause a resync. Perform a test after installing the Agent to ensure that the correct disks have been added.

4. After all of the disks that will be replicated have been successfully identified, the installer will download and install the AWS Replication Agent on the source server.

```
...
All volumes for replication were successfully identified
Downloading the AWS Replication Agent onto the source server... Finished
Installing the AWS Replication Agent onto the source server... Finished
```

5. Once the AWS Replication Agent is installed, the server will be added to the Elastic Disaster Recovery Console and will undergo the initial sync process. The installer will provide you with the source server's ID.

```
...
All volumes for replication were successfully identified
Downloading the AWS Replication Agent onto the source server... Finished
Installing the AWS Replication Agent onto the source server... Finished
Syncing the source server with the Elastic Disaster Recovery Console... Finished
The following is the source server ID: s-3146f90b19example
The AWS Replication Agent was successfully installed.
Press Enter to close...
```

You can review this process in real time on the **Source servers** page.

AWS Replication Agent Installer parameters

The AWS Replication Agent Installer supports the following command line parameters.

--region

The region into which the installer will register the source server.

--aws-access-key-id

The AWS IAM Access Key used for authenticating the installing user. If this parameter is not provided, the installer will prompt for it.

--aws-secret-access-key

The AWS IAM Secret Access Key tied to the AWS IAM Access Key used for authenticating the installing user. If this parameter is not provided, the installer will prompt for it.

--aws-session-token

The session token is generated when using temporary credentials generated using AWS STS.

--account-id

Use this parameter to install the DRS agent on an EC2 instance to replicate to another AWS account without any additional access key or temporary credentials. Specify the 12 digit ID of the account into which you want to replicate your source server. This action requires an EC2 instance profile with the AWSElasticDisasterRecoveryEc2InstancePolicy policy, to define the account to replicate into as a Trusted Account and select the roles in **Failback and in-AWS right-sizing roles**.

--no-prompt

Run the installation without prompting the user.

--devices

Specify exactly which disks to replicate.

--force-volumes

This parameter must be used with the *--no-prompt* parameter. This parameter will cancel the automatic detection of physical disks to replicate. You will need to specify the exact disks to replicate using the *--devices* parameter (including the root disk, failure to specify the root disk will cause replication to fail). This parameter should only be used as a troubleshooting tool if the *--devices* parameter fails to identify the disks correctly.

--tags

Use this parameter to add resource tags to the source server. Use a space to separate each tag.

```
--tags ["Key1"="Value1" "Key2"="Value2"]
```

> **Note**
>
> This flag may only be used when adding new source servers to AWS DRS. You cannot use the --tags flag to modify tags of source servers that have already been added to AWS DRS.

--s3-endpoint

Use this parameter to specify a VPC endpoint you created for S3 if you do not wish to open your firewall ports to access the default S3 endpoint. Learn more about installing the Agent on a blocked network.

--endpoint

Use this parameter to specify the Private Link endpoint you created for Elastic Disaster Recovery if you do not wish to open your firewall ports to access the default AWS Elastic Disaster Recovery endpoint. Learn more about installing the agent on a blocked network.

> **Note**
>
> We do not recommend using this flag when installing the AWS Elastic Disaster Recovery Agent on an EC2 Instance, as it can prevent successful failback from occuring. We recommend ensuring DNS automatically resolves the `{region}.drs.amazonaws.com` entry to the Private Link endpoint rather than leveraging this parameter.

--install-as-recovery-instance

Use this parameter to add an existing AWS instance to AWS Elastic Disaster Recovery as a recovery instance. You may opt to add recovery instances if you have added additional EC2 instances to AWS and now want to recover them into source servers. You will be asked to pair the newly added recovery instance with a source server during AWS Replication Agent installation.

--proxy-address

Linux Installer only.

Use this parameter to configure the agent to use a specific proxy server.

```
--proxy-address https://PROXY:PORT/
```

Ensure the proxy configuration has the trailing forward slash (/).

Installing the agent on a secured network

The AWS DRS AWS Replication Agent installer needs network access to AWS Elastic Disaster Recovery and S3 endpoints. If your on premise network is not open to Elastic Disaster Recovery and S3 endpoints, then you can install the Agent with the aid of PrivateLink.

You can connect your on premise network to the subnet in your staging area VPC using AWS VPN or DirectConnect. To use the AWS VPN or DirectConnect, you must activate private IP in the replication settings

> **Note**
>
> This feature is not supported in the Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Asia Pacific (Melbourne), Asia Pacific (Osaka), Europe (Spain), Europe (Zurich), and Middle East (UAE) Regions.

Create a VPC Endpoint for AWS Elastic Disaster Recovery

To allow the AWS Replication Agent installer to communicate with AWS Elastic Disaster Recovery, create an interface VPC endpoint for AWS Elastic Disaster Recovery in your staging area subnet. For more information, see Creating an Interface Endpoint in the Amazon VPC User Guide.

If the AWS replication agents are installed with a principal using AWSElasticDisasterRecoveryAgentInstallationPolicy and a VPCE policy is used (to scope down access), add the following statement to your policy:

```
{
          "Effect": "Allow",
```

```
        "Principal": "*",
        "Action": "execute-api:Invoke",
        "Resource": "arn:aws:execute-api:<region>::*/POST/CreateSessionForDrs"
        }
```

Use the created VPC Endpoint for AWS Elastic Disaster Recovery

Once you have created the VPC Endpoint, the AWS Replication Agent can connect to Elastic Disaster Recovery via VPN/DirectConnect by using the --endpoint installation parameter. Learn more about Private DNS for interface endpoints in the *Amazon VPC User Guide*.

Run the AWS Replication Agent installer with the --endpoint parameter. Enter your endpoint-specific DNS hostname within the parameter. The installer will then be able to connect to AWS Elastic Disaster Recovery via the endpoint over your VPN/DirectConnect connection.

Example of an interface endpoint DNS name: vpce-0123456789-abcdef.drs.<REGION>.vpce.amazonaws.com

Create a S3 Endpoint for AWS Elastic Disaster Recovery

To allow the AWS Replication Agent installer to communicate with S3, create an interface S3 endpoint for AWS Elastic Disaster Recovery in your staging area subnet. For more information, see Endpoints for Amazon S3 in the *Amazon VPC User Guide*.

Use the created S3 Endpoint for AWS Elastic Disaster Recovery

Once you have created the interface VPC Endpoint, the AWS Replication Agent can connect to S3 via VPN/DirectConnect by using the --s3-endpoint installation parameter. Learn more about Private DNS for interface endpoints in the *Amazon VPC User Guide*.

Run the AWS Replication Agent installer with the --s3-endpoint parameter. Enter your endpoint-specific DNS hostname. The installer will then be able to connect to Elastic Disaster Recovery via the endpoint over your VPN/DirectConnect connection.

Example of an interface endpoint DNS name: vpce-0123456789-abcdef.s3.<REGION>.vpce.amazonaws.com

Preparing the AWS VPC

To prepare the staging area subnet in a private subnet, two more endpoints have to be created to ensure the successful creation of the replication servers.

- EC2 Interface Endpoint: used to establish connectivity to EC2 endpoint from the staging area subnet
- S3 Gateway Endpoint: used by the replication servers to download the replication software from S3

For more information about setting up AWS Elastic Disaster Recovery with a site-to-site VPN connection, visit this blog post.

Uninstalling the agent

Uninstalling the AWS Replication Agent from a source server stops the replication of that server. Uninstalling the AWS Replication Agent will remove the source server from the Elastic Disaster Recovery Console.

Uninstalling the Agent through the AWS Elastic Disaster Recovery console

To uninstall the AWS Replication Agent though the AWS Elastic Disaster Recovery console.

Navigate to the **Source servers** page.

Check the box to the left of each server that you want to disconnect from Elastic Disaster Recovery (by uninstalling the AWS Replication Agent). Open the **Actions** menu, and choose the **Disconnect from AWS** option to disconnect the selected server from AWS Elastic Disaster Recovery and AWS.

When the **Disconnect X server/s from service** dialog appears, click **Disconnect**.

The AWS Replication Agent will be uninstalled from all of the selected source servers.

Uninstalling the Agent manually through the source server

To uninstall the AWS Replication Agent manually through the source server:

Windows

Copy the following folder to a new location:`C:\Program Files (x86)\AWS Replication Agent\dist`

From the new location, run in CMD as an administrator:

```
install_agent_windows.exe --remove
```

Linux

As root, cd to `/var/lib/aws-replication-agent`.

Run the following commands from that folder:

```
./stopAgent.sh
```

```
./uninstall_agent_linux.sh
```

Reinstalling the agent

To reinstall the AWS Replication Agent, download the latest version of the agent and follow the installation instructions. You do not need to remove any previous versions prior to reinstalling the agent.

- Linux
- Windows

> **Note**
>
> You must reinstall the agent to benefit from new features.

Reinstalling the agent on a recovery instance

If you are reinstalling an agent on a recovery instance:

1. Select your recovery instance and choose **Disconnect from AWS** from the **Actions** drop-down menu.
2. When reinstalling the agent, include the **--install-as-recovery-instance** parameter.

Example:

```
chmod +x aws-replication-installer-init; sudo ./aws-replication-installer-init --
install-as-recovery-instance s-abcd01234567890
```

> **Note**
>
> In order to reinstall the agent on a recovery instance, you will need to provide the temporary credentials for a role that has the AWSElasticDisasterRecoveryAgentInstallationPolicy policy.

Supporting marketplace licenses

Installing the AWS replication agent on an EC2 instance on AWS that has one or more active subscriptions to a marketplace license requires taking the following points into consideration:

- Some marketplace products do not function with certain instance types or on certain regions. DRS does not verify if the marketplace license applies to the instance type and region defined. To see if the marketplace product applies to the current settings, visit the marketplace product page. It is also very recommended to do periodic drills as some of these incompatibilities are only identified upon launch.
- If an agent is to be installed on an EC2 instance existing on one account (source account) which is a different AWS account than the AWS account where DRS is operated (the target account), it is mandatory to provide permissions that allow getting the marketplace license information from the source account. Create a Failback and in-AWS right-sizing role for trusted account using the target account AWS account ID. This role must be created in the source account, or the agent installation will fail. If this role is removed or modified, launch operations might fail if new marketplace licenses are added.
- If an agent was installed on an EC2 instance existing on one account (source account), and DRS is operated on a different account (target account), and a new volume, that has a marketplace license associated with it, is connected to the instance with the **Automatically replicate new disks** setting active, the volume might fail to be added if permissions to allow getting the marketplace license information were removed or do not exist. Create a Failback and in-AWS right-sizing role for trusted account using the target account AWS account ID, and re-install the agent if a volume fails to be added due to this reason.
- In case of EC2 instances from one account that replicate to a staging account (see multi-account) and launch in one or more target accounts, only the staging account must have a Failback and in-AWS right-sizing role created for.

Adding instances from the EC2 Console

You can now add EC2 instances as source servers in DRS, starting from the EC2 console. New or existing instances can be added by selecting the appropriate action on the EC2 console, sending you to the AWS focused page allowing to install the AWS replication agent used by DRS on the selected instances.

Add instances

You can protect your EC2 instances using AWS Elastic Disaster Recovery (DRS) in the chosen AWS Region, by adding to them to AWS DRS as source servers. Utilize AWS Systems Manager (SSM) if present on your instance to install the AWS replication agent, a step needed to start replicating data from your instance to AWS. Only instances managed by AWS Systems Manager would be able to have the AWS replication agent installed on them.

> **Note**
>
> You will need an instance profile with the policies listed below in order to have your instances managed by SSM and for installing the AWS replication agent:
>
> 1. AmazonSSMManagedInstanceCore
> 2. AWSElasticDisasterRecoveryEC2InstancePolicy

Successfully installing the AWS replication agent adds the instance to AWS DRS (as a **source server**) in the chosen target region.

Supported EC2 instances

> **Note**
>
> Any additional EBS volumes added during the EC2 Instance creation that are offline, unmounted, or unformatted will not be replicated. Any volume that is later placed online or mounted with a valid file system will automatically be replicated if Automatically replicate new disks is enabled.

This section lists all the instances that were selected to be protected by AWS DRS. The list shows which instances are currently managed by SSM and which instances are currently not managed.

Only instances managed by SSM can have the AWS replication agent installed on them using this page. You can also install the agent using the installer as defined in Installing the AWS Replication Agent, without requiring the SSM agent to be present and active on the server to be protected.

To have an instance managed by SSM, requires the SSM agent to be installed on a compatible operating system (or preinstalled in the AMI), and the instance to have the correct permissions (as defined in the AmazonSSMManagedInstanceCore and the AWSElasticDisasterRecoveryEC2InstancePolicy policies). To update the instance profiles, the **Instance profile role installation** section allows to create the default instance profile (with the two policies mentioned above) if needed. The **Instance profiles** section allows to assign instance profiles to instances, and will automatically assign the default instance profile to all instances that do not have any instance profile attached to them. Use the **Attach profiles to all instances** button to attach the assigned instance profiles to the instances in case the default profile was created and automatically assigned to them or if you changed the assigned instance profile.

Target disaster recovery region

On this section, you can define the target disaster recovery region. This can be the same region where the instances are present in, or it can be a different region, for cross-region protection. AWS DRS must be initialized in the target region in order to protect the instances onto that region. The indicator next to the region's name will show if AWS DRS is already initialized in the target region, or not. If the region is not initialized, a button labelled Initialize and configure AWS Elastic Disaster Recovery will be visible and active. Clicking this button opens the AWS DRS initialization wizard for AWS DRS in the target region on another browser tab.

Instance profile role installation - optional

This section provides you with the option to create the default IAM role with the required permissions as an instance profile. The role **AWSElasticDisasterRecoveryAutomatedAgentInstallRole** includes the permissions defined in the policies AmazonSSMManagedInstanceCore and AWSElasticDisasterRecoveryEC2InstancePolicy. These permissions are required to allow the SSM agent to operate and to install the AWS replication agent, respectively. Clicking the **Install default IAM role** installs this role. This needs to be done only once per account. If the role was already installed in the account, this button is inactive. The default instance profile role will be automatically assigned to instances without an instance profile in the **Instance profiles** section. If you click the **Attach profiles to all instances** button, this role will be attached to all instances it was assigned to in the **Instance profiles** section. If this default IAM role is not installed, you will need to make sure you have an instance profile with

the AmazonSSMManagedInstanceCore and AWSElasticDisasterRecoveryEC2InstancePolicy policies (or the combined set of permissions within both of these policies).

Instance profiles

This section lists all the instances that were selected to be protected by adding them as source servers to AWS DRS and their current instance profiles. Instances without any instance profile will have the **AWSElasticDisasterRecoveryAutomatedAgentInstallRole** instance profile and IAM role assigned to them if it exists on this account. Using the default profile is not mandatory, as any instance profile in the account can be assigned to any instance, but care must be taken to verify each instance has an instance profile with the permissions defined in the AmazonSSMManagedInstanceCore and AWSElasticDisasterRecoveryEC2InstancePolicy policies.

> **Note**
>
> AWS DRS does not validate the instance profile has the required permissions to support working with the SSM agent or installing the AWS replication agent for DRS.

> **Note**
>
> Attaching an instance profile with the needed permissions is a mandatory step if you want to install AWS DRS on instances that have the SSM agent installed on them (manually, or preinstalled on AMI) but are not managed on SSM due to missing an instance profile with the AmazonSSMManagedInstanceCore policy.

Click the button labelled **Attach profiles to all instances** to attach the assigned instance profiles to their instances.

After attaching such a profile, allow AWS DRS a few minutes to identify the instance as managed by SSM. If SSM is present on the instance, and an instance profile with the needed permissions was attached to the instance, then within a few minutes, the marker near the instance ID will change to show that the instance is currently managed by SSM.

Attach profiles to all instances

Clicking this button attaches the instance profiles assigned in the **Instance profiles** section to their instances. After attaching appropriate instance profiles to instances, allow a few minutes for DRS to detect if these instances are managed by SSM.

Add instances

Click this button to install the AWS replication agent on all instances that are currently managed by SSM. If there are such instances, AWS DRS will list all these instances and the progress of installing the AWS replication agent on them. Successfully installing the AWS replication agent on these instances adds them as source servers to AWS DRS. If there are no instances that are currently managed by SSM, try installing the SSM agent on these instances, then attach an appropriate instance profile to them.

Add instances result page

On this page you can view the result of adding instances to AWS DRS by installing the AWS replication agent on them. The page shows the progress of this process if currently running, or the summary of the last run. In addition, for each instance that is currently managed by SSM, there is a table listing the following:

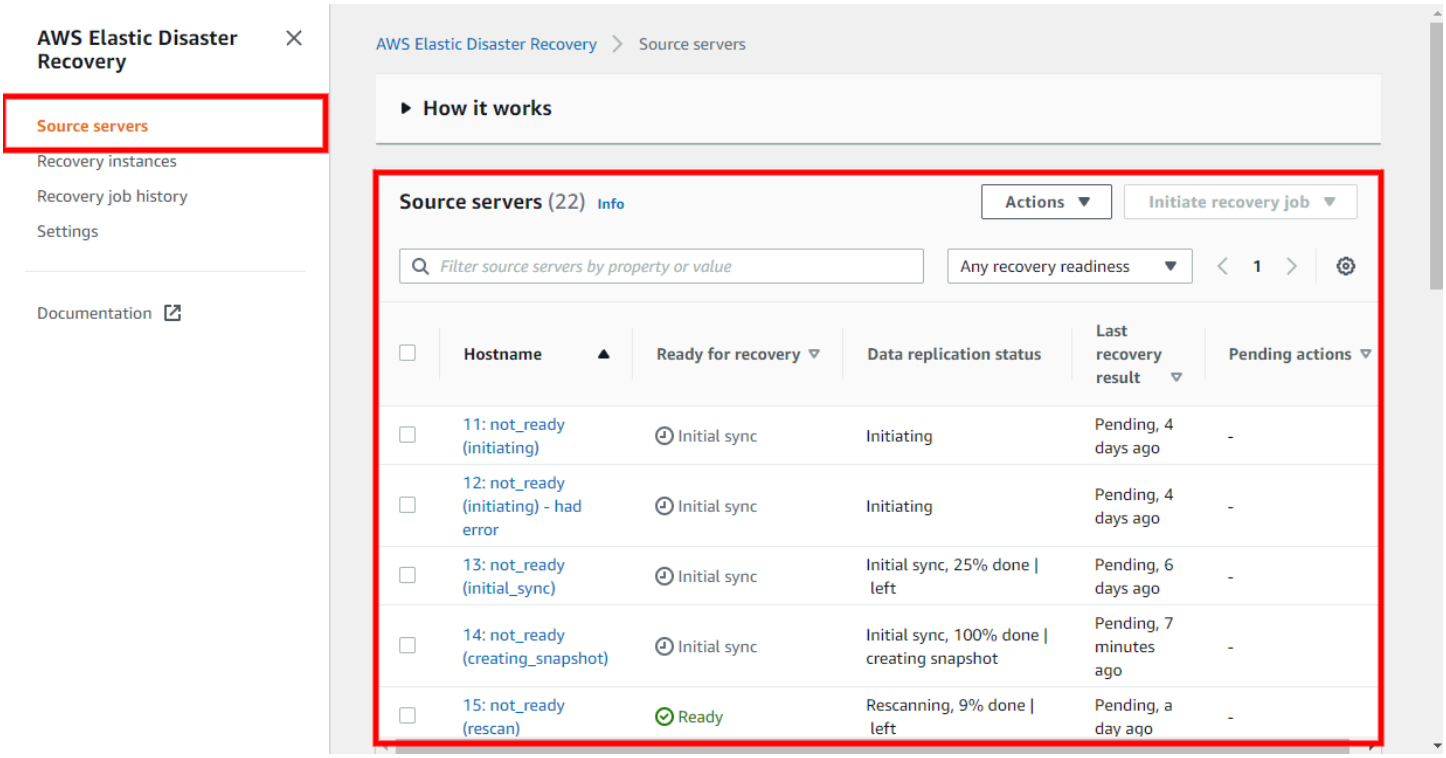
Instance ID - The ID of the instance. This also links to the instance on the EC2 console page (opens in a different browser tab).

Status - The current status of the installation, possible values include **Success**, **In Progress**, **Pending** and **Error**.

Details - holds a link to the source servers page on the target region for successful installations, or a link to the run log on the SSM console (opens in a new browser tab) for runs that have failed, are pending or are in progress.

Source servers page

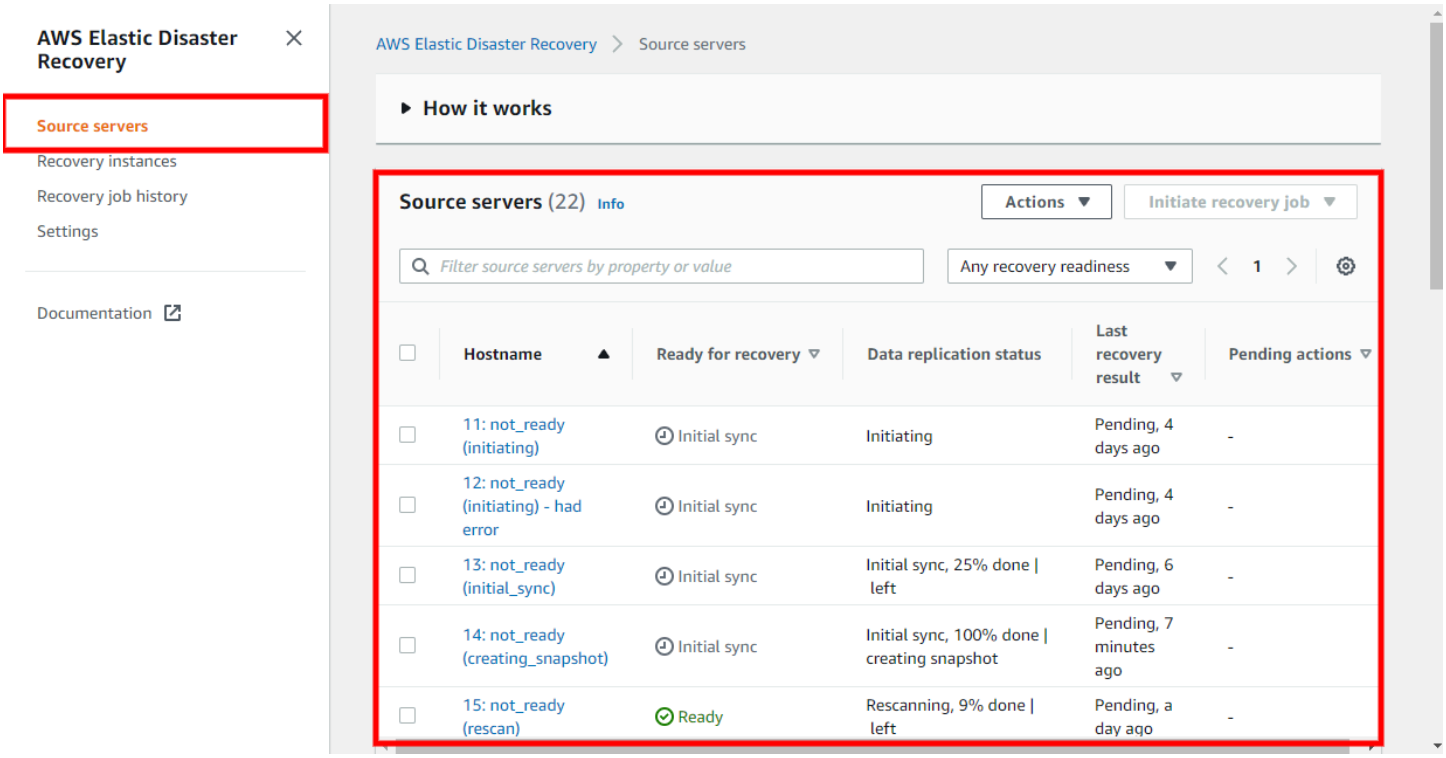
The **Source servers** page lists all of the source servers that have been added to AWS Elastic Disaster Recovery. The **Source servers** page allows you to manage your source servers and perform a variety of commands for one or more servers (such as controlling replication and launching Initiate recovery job instances). This page is the "main" page of AWS Elastic Disaster Recovery and you will most likely interact with AWS Elastic Disaster Recovery predominantly through this page.



Topics

Interacting with the Source Servers page

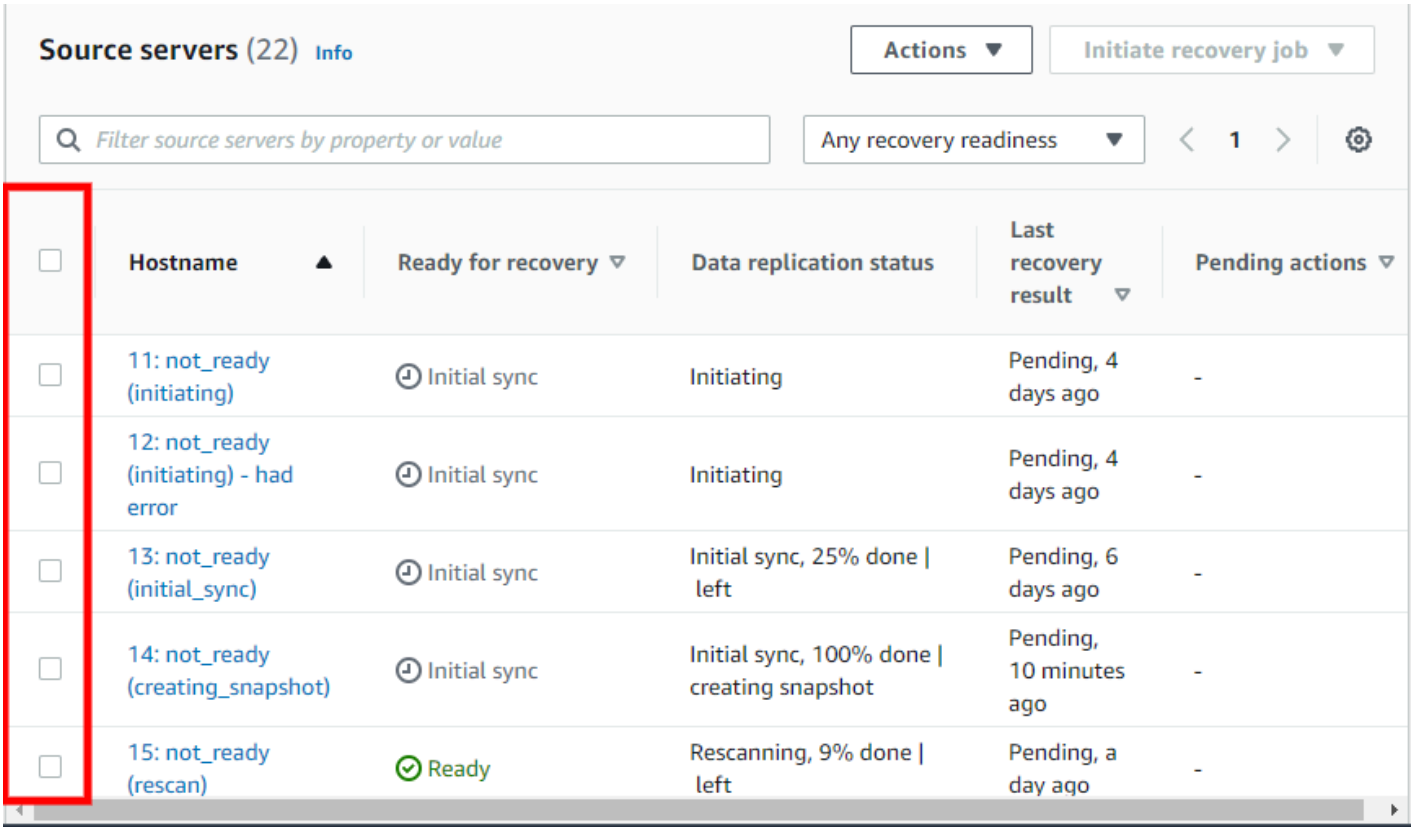
The **Source servers** page shows a list of source servers. Each row on the list represents a single server.



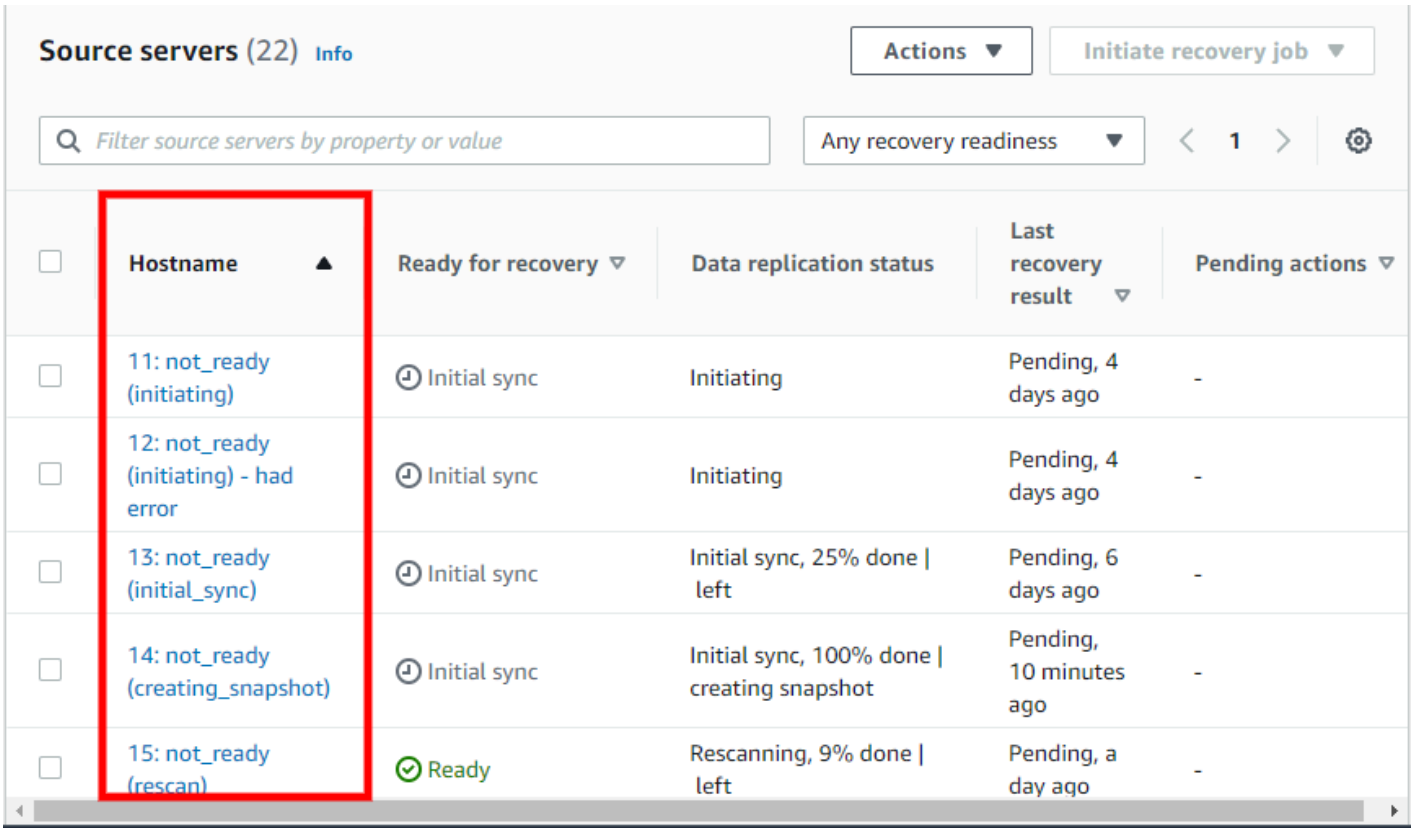
The **Source servers** page provides key information for each source server under each of the columns on the page.

The columns include:

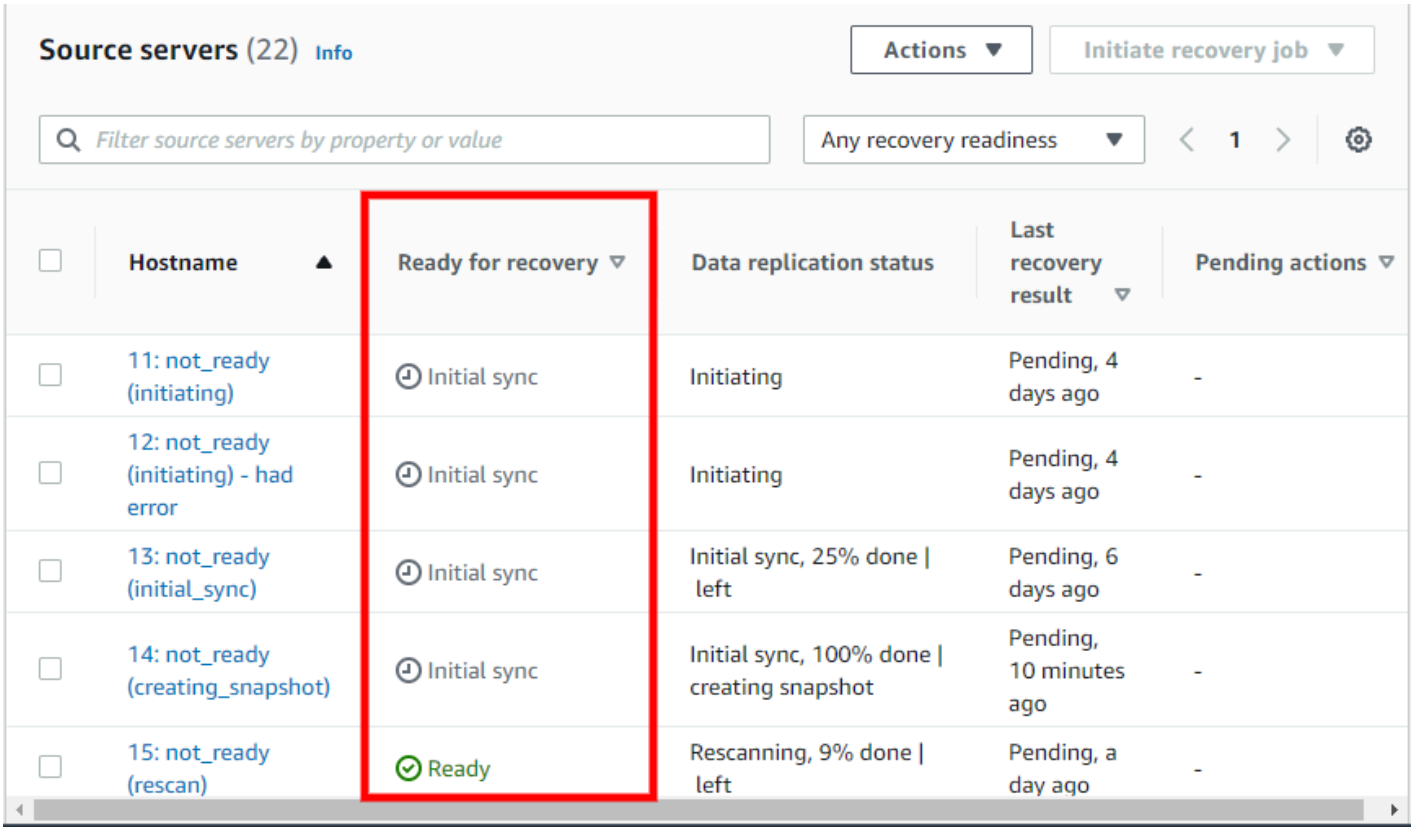
- **Selector column** – This blank checkbox selector column allows you to select one or more source servers. When a server is selected, you can interact with the server through the **Actions**, **Replication**, and **Initiate recovery job** menus. Selected servers are highlighted.



- **Hostname** – This column shows the unique server hostname for each source server.



- **Ready for recovery** – This column shows whether the server is ready for recovery. You can use this column to easily tell whether a server is ready or not and the server's exact status. You can learn more about the server's status by reviewing the **Data replication status** column.

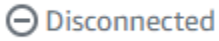


A server that is ready will show the green checkmark and will state **Ready**.

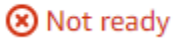
A server that is ready, but is experiencing a non-critical issue such as lag will show the blue warning sign and will state **Ready** and will display the lag duration to the right.

A server that is still undergoing initial sync will show a gray circle with three dots and will state **Initial sync**.

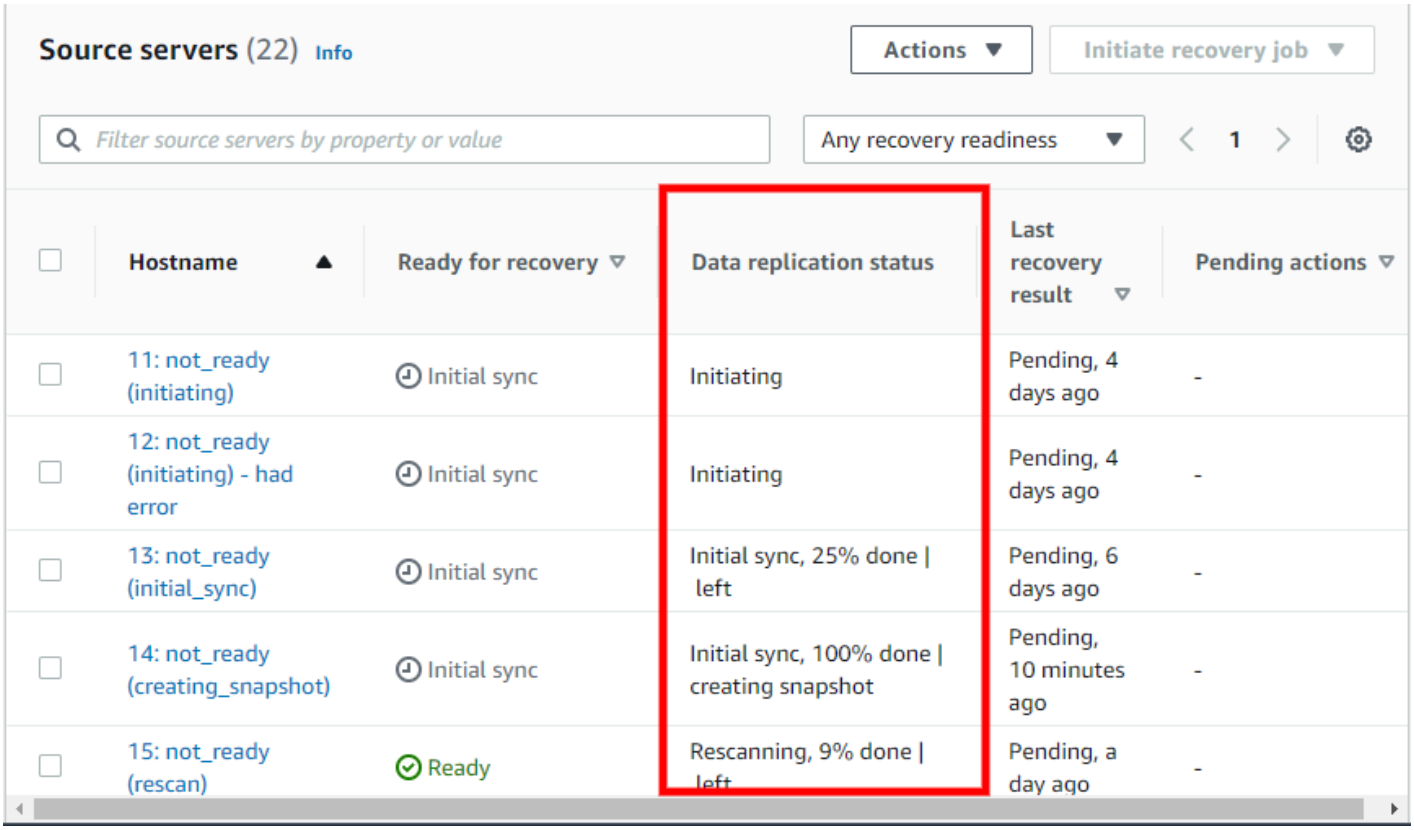
A server that is disconnected will show the gray warning sign and will state **Disconnected**.



A server that is not ready due to a significant error, such as a stall, will show a red **X** and will state **Not Ready**. Servers that have one or more marketplace licenses assigned to them may not be able to launch if there was an error reading their license information.



- **Data replication status** – This column shows the current status of data replication for the server.

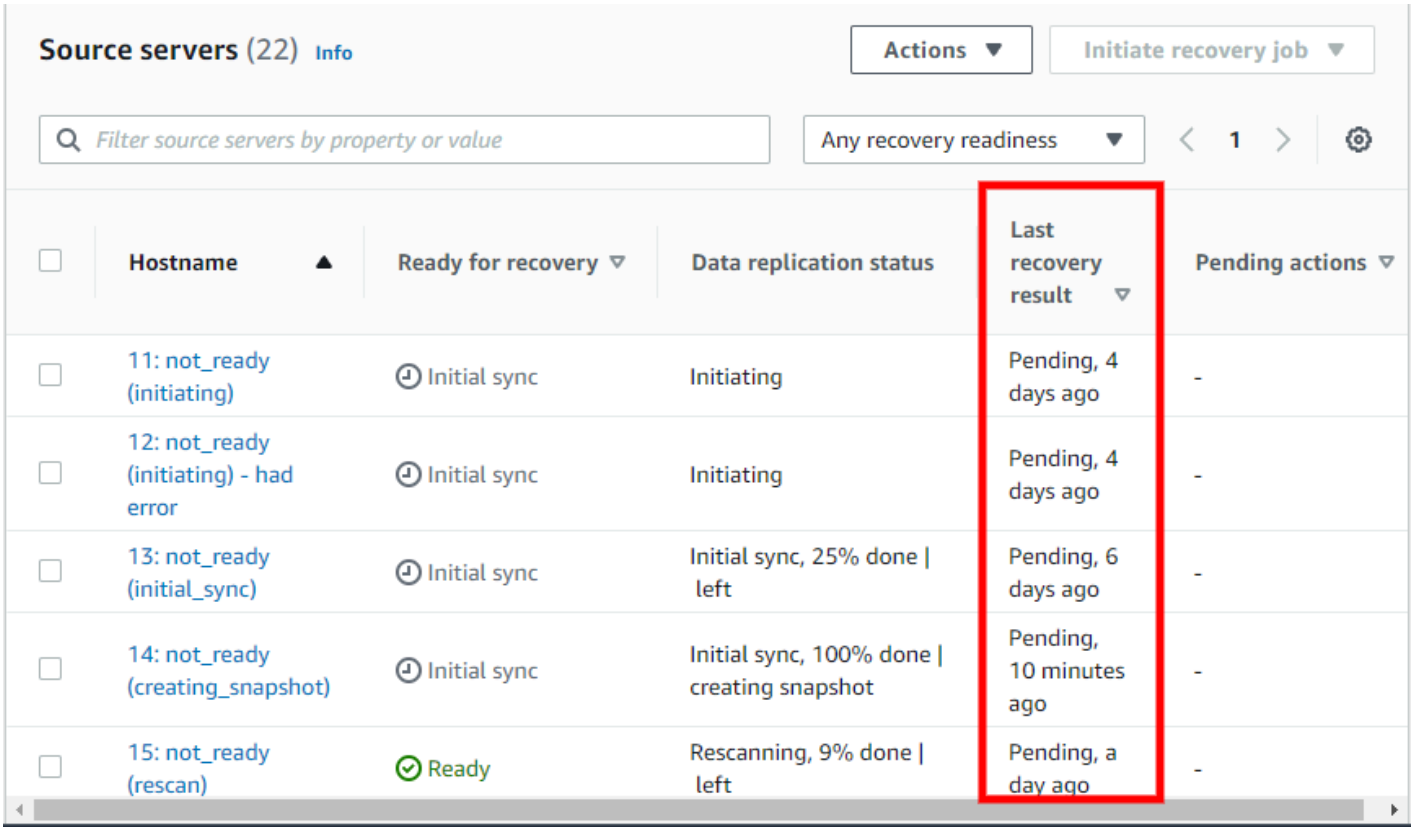


This column will show a variety of information, including:

- **Initiating** – The server has just been added to AWS Elastic Disaster Recovery and replication is being initiated.
- **Initial sync** – The server is undergoing the initial sync process. The console will show the percentage of the server that has been synced and the step the server is undergoing in the initial sync process. You can learn more about the exact state of the server in the server info view.
- **Rescanning** – The server is undergoing a rescan. The console will show the percentage of the server that has been rescanned successfully.

- **Healthy** – The server is healthy and is ready to initiate a recovery job.
- **Lag** – The server is experiencing lag. The console will show the amount of lag time. You can learn more about the exact state of the server in the server info view.
- **Stalled** – The server is stalled due to a replication error. You can learn more about the specific cause of the stall in the server info view.
- **Disconnected** – The server has been disconnected from AWS Elastic Disaster Recovery.

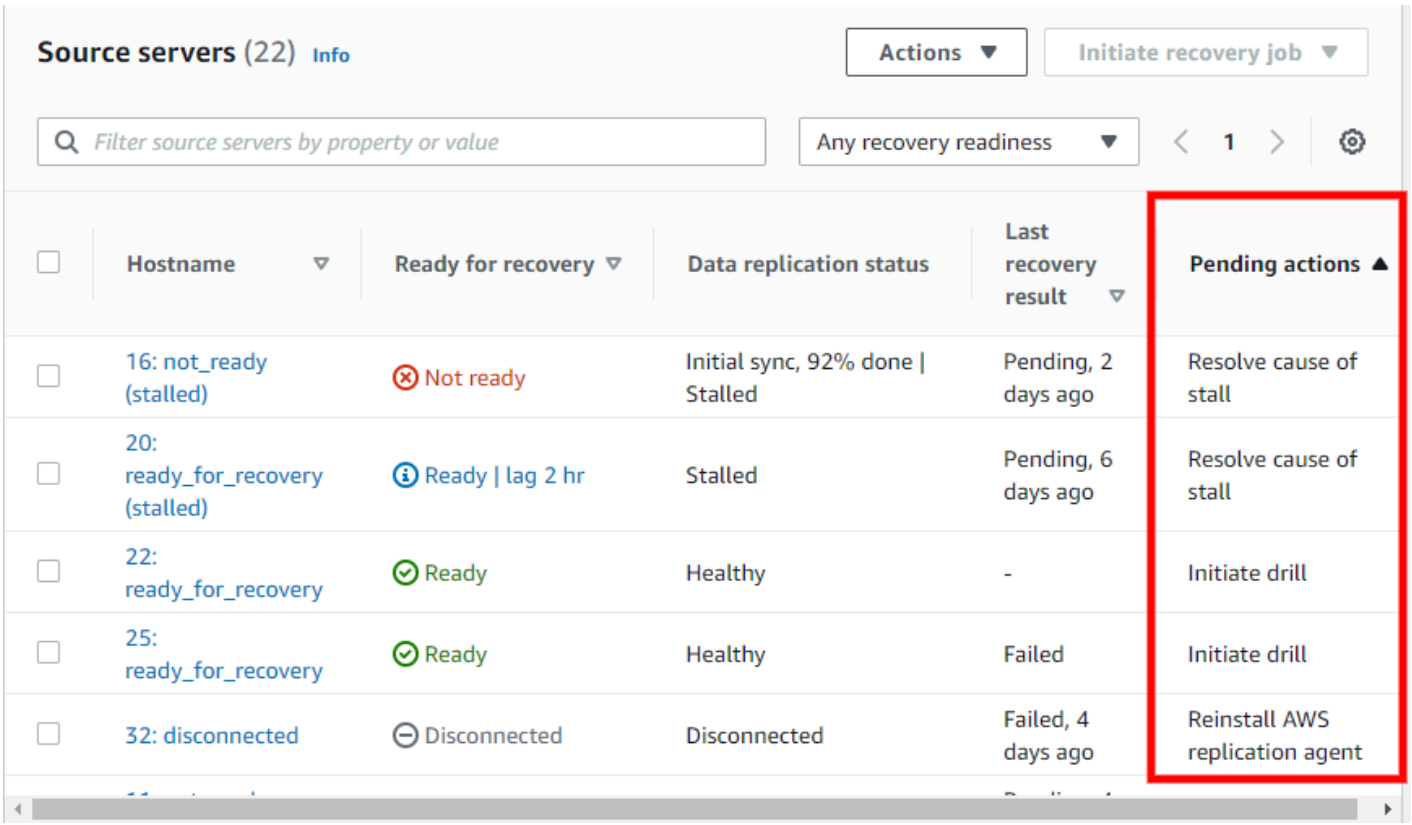
- **Last recovery result** – This column shows the result of the last recovery job launch. The column will be empty if no recovery job has ever been launched for the server.



This column can show the following:

- **Successful** - Recovery launch job was completed successfully. The console will indicate how long ago the job was completed.
- **Failed** – Recovery launch job failed. The console will indicate how long ago the job failed. You can learn more about why the job failed in the job history.
- **Pending** – Recovery launch job is pending. The console will indicate how long ago the job was initiated.

- **Pending actions** – This column shows any pending actions that need to be performed on the server. This column will appear empty unless there is an actionable pending action.

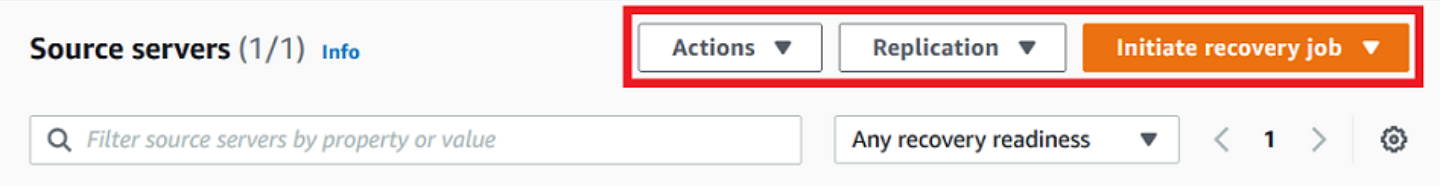


This column can show the following:

- **Initiate drill** – The source server is healthy, but no drill instances have been launched for the source server. Initiate a drill by launching a drill instance.
- **Resolve cause of stall** – The source server is stalled. Resolve the cause of the stall for the server to return to healthy function.
- **Reinstall AWS Replication Agent** – The AWS Replication Agent was removed from the source server. Reinstall the agent for replication to resume.
- **Error: Missing permissions to retrieve marketplace licenses from the source account, cannot launch this server** – The marketplace license belongs to a different AWS account, permissions to get information about this marketplace license are missing. Create a Failback and in-AWS right-sizing role for trusted account using the target account AWS account ID.
- **Warning: server uses marketplace product, drill recommended** – This source server uses one or multiple marketplace licenses. Doing a drill is strongly recommended as some marketplace incompatibilities can only be identified during launch. Learn more here.

Command menus

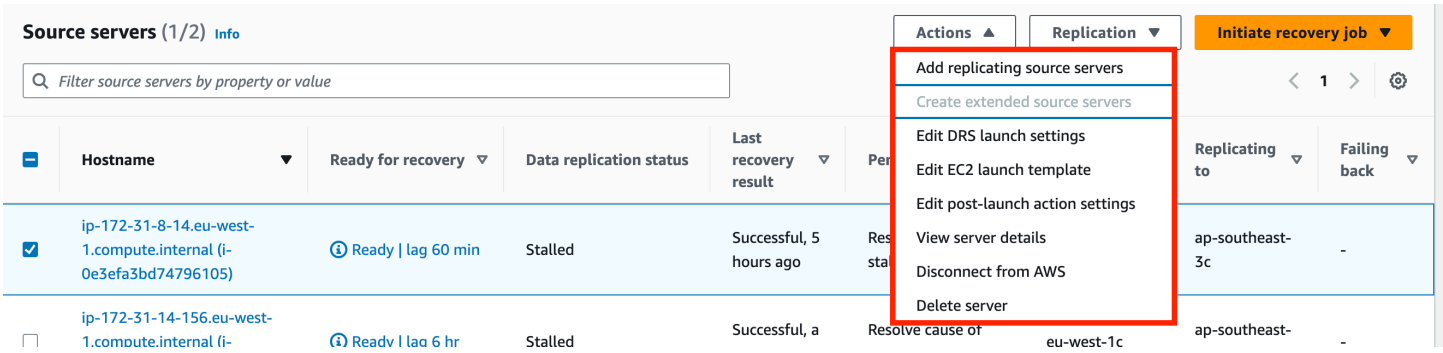
You can perform a variety of actions, control data replication, and manage your drill and recovery instances for one or more source servers through the command menu buttons. Select one or more servers on the **Source servers** page and choose the **Actions**, **Replication**, or **Initiate recovery job** menu to control your source servers.



Topics

Actions menu

The **Actions** menu allows you to perform the following actions:



- **Add servers** – Choosing this option will redirect you to the AWS Replication Agent installation instructions.
- **Create extended source servers** – Choose this to start a wizard to create extended source servers from source servers replicating into staging accounts, in multi-account setups.
- **Edit DRS launch settings** – Choose this option to edit a single or multiple selected source servers for their DRS launch settings.
- **Edit EC2 launch template** – Choose this option to enter edit a single or multiple selected source servers for their EC2 launch template.

- **Edit post-launch action settings** – Choose this option to activate or deactivate post-launch actions for a single or multiple selected source servers.
- **View server details** – Choose this option to enter the source server's **Server details view.**
- **Disconnect from AWS** – Choose this option to disconnect the selected server from AWS Elastic Disaster Recovery and AWS.

  When the **Disconnect X server/s from service** dialog appears, click **Disconnect**.

  > **Important**
  >
  > This will uninstall the AWS Replication Agent from the source server and data replication will stop for the source server. This action will not affect any Drill or Recovery instances that have been launched for this source server, but you will no longer be able to identify which source servers your Amazon EC2 instances correspond to.

- **Delete server** - Choose the **Delete server** option to permanently delete a source server from AWS Elastic Disaster Recovery. This will remove all information related to the server from the AWS Elastic Disaster Recovery service. You can only delete servers that have been disconnected from AWS. You will need to reinstall the AWS Replication Agent on a deleted source server to add it back to AWS Elastic Disaster Recovery.

  When the **Delete X servers** dialog appears, click **Permanently delete**.

Initiate recovery job menu

The **Initiate recovery job** menu allows you to start drills and recoveries by launching drill and recovery instances as part of the overall failback process. You can learn more about the entire failback and failover process with AWS Elastic Disaster Recovery in the Performing a failback and failover with AWS Elastic Disaster Recovery documentation.

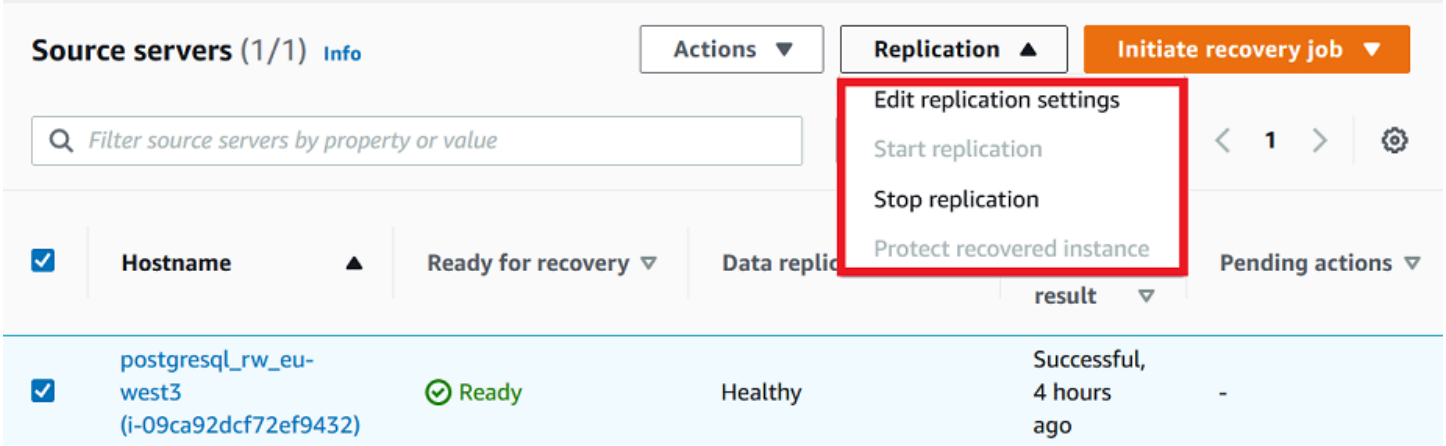
- **Initiate drill** – Choose this option to launch a drill instance for this server or group of servers for the purpose of testing your recovery solution. You should perform periodic drills in order to

ensure that you are ready for recovery. Learn more about launching Drill instances in AWS Elastic Disaster Recovery.

- **Initiate recovery** – Choose this option to launch a Recovery instances for this server or group of servers for the purpose of recovering the server in the event of a disaster. Learn more about launching Recovery instances in AWS Elastic Disaster Recovery.

Replication menu

The **Replication** menu allows you to perform the following actions:



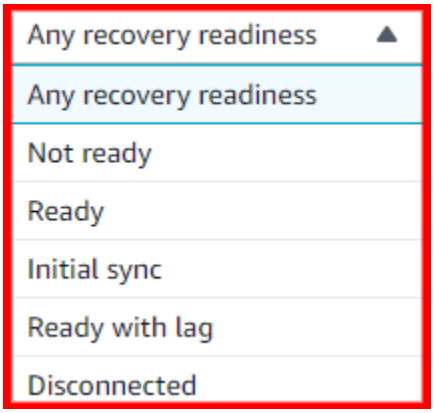
- **Stop replication** – You can stop replication of a source server at any time. After you stop the replication, you will no longer be charged for the ongoing replication and the staging area infrastructure. Changes will not be reported by the agent to the replication server, and all saved snapshots will be deleted, leaving this instance unprotected. The agent remains installed during this process. If you want to replicate this EC2 instance again, simply click the **Start replication** button. This will trigger an initial sync.
- **Start replication** – You can start replication of a previously stopped source server. After you start the replication, the agent replicates the selected instances.

Filtering

You can customize the **Source servers** page through filtering by recovery readiness.

Click within the **Filter source servers....** field and choose the filtering property from the menu.

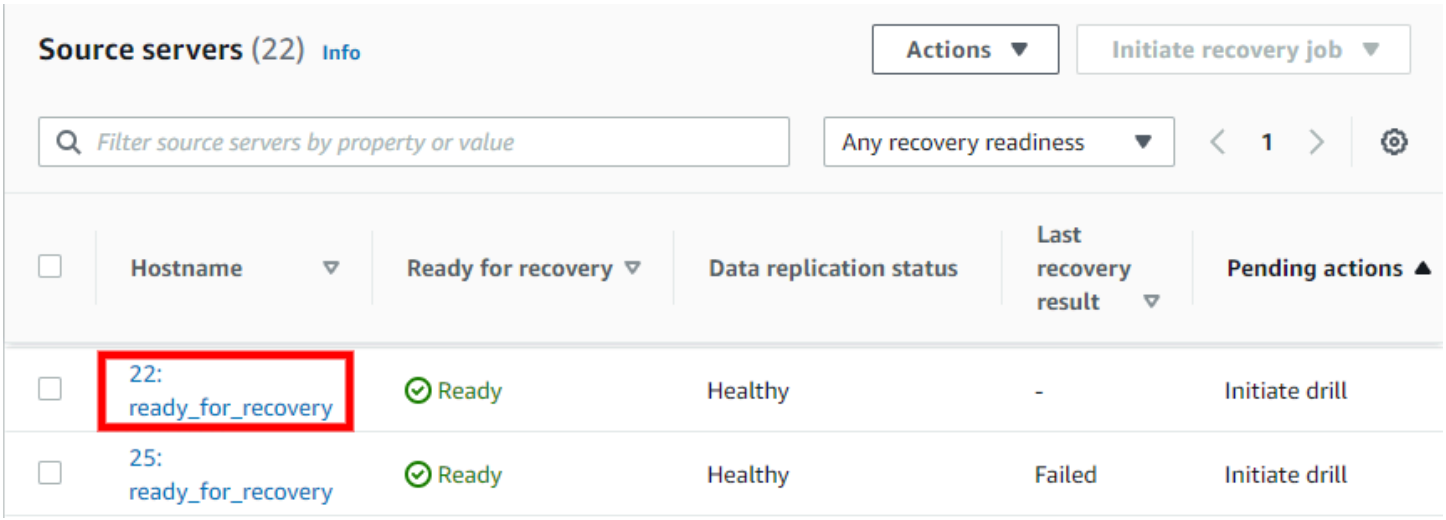
You can filter by a variety of properties, including:



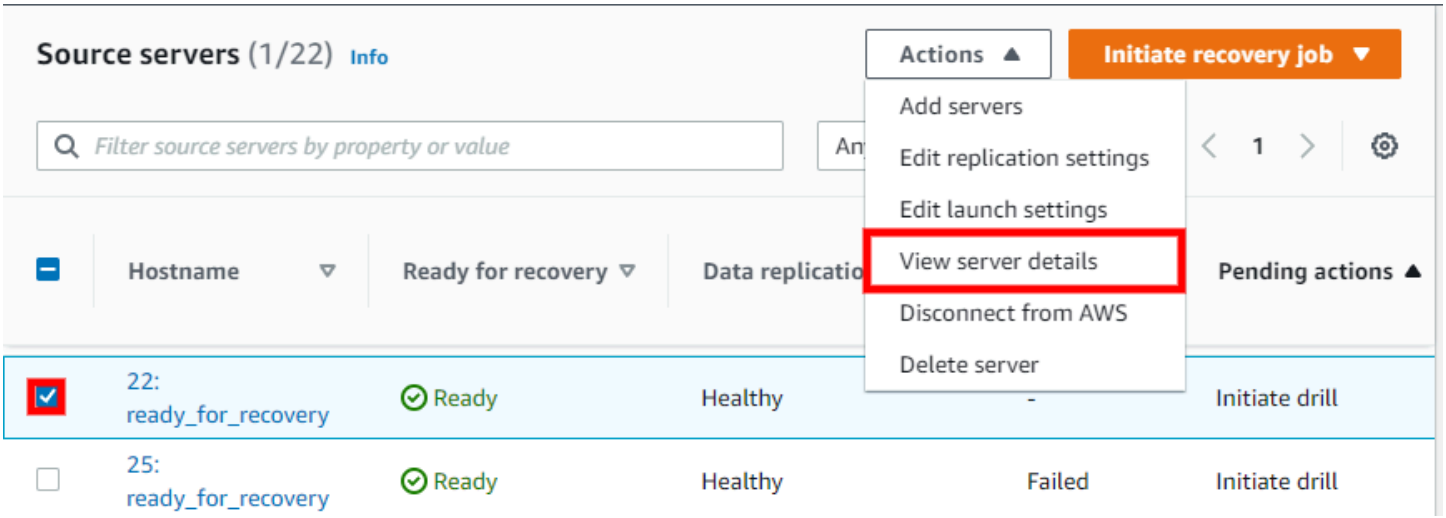
- Any recovery readiness – Filter by specific alert (lagging, stalled, launched)
- Not ready – Filter by a specific hostname or a specific string of characters
- Ready – Filter by the recovery lifecycle state
- Initial sync – Filter by the data replication status
- Ready with lag
- Disconnected

Server details

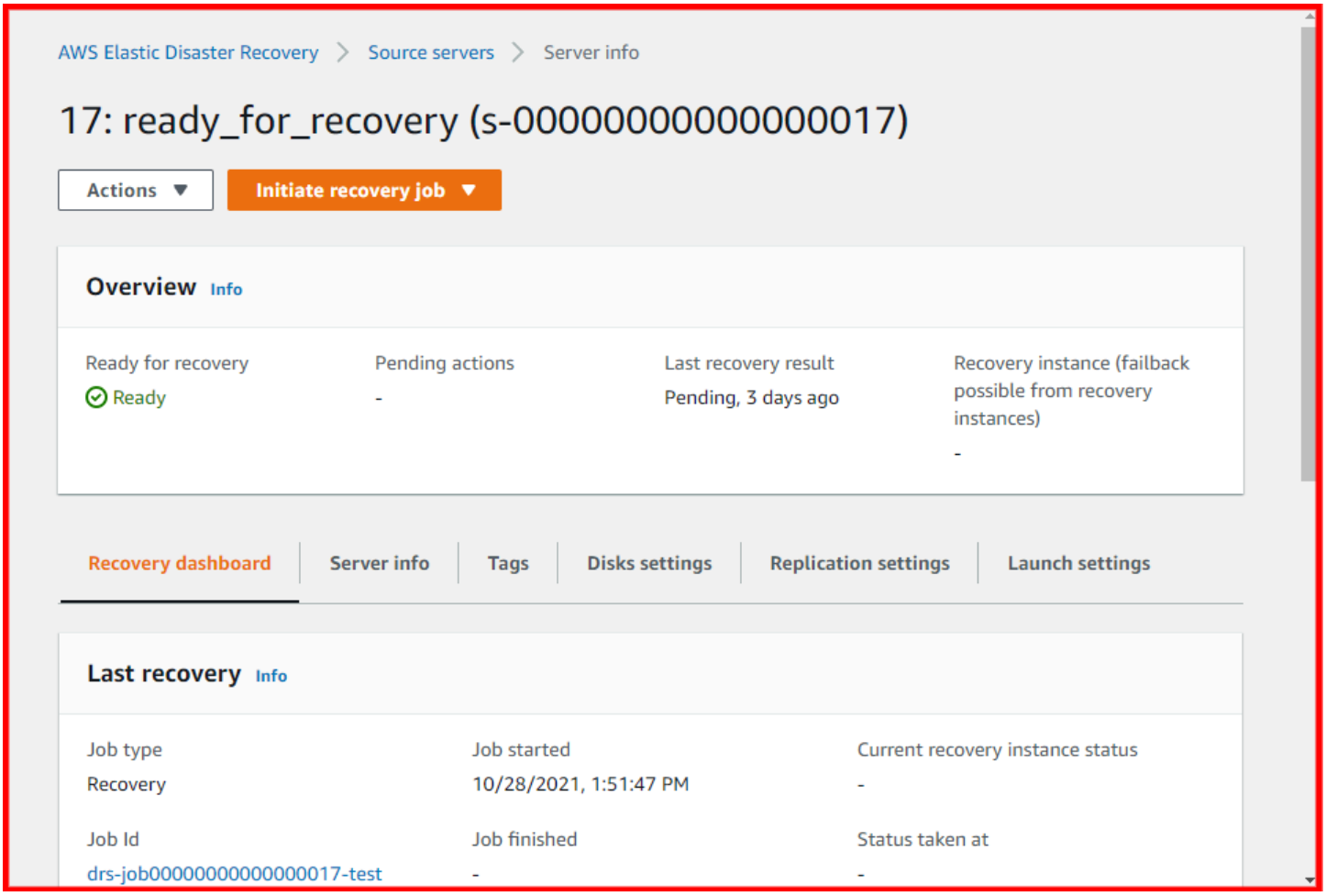
To access the server details view, click the **Hostname** of any server on the **Source servers** page.



You can also access the server details view by checking the box to the left of any single source server on the **Source servers** page and choosing **Actions > View server details**.

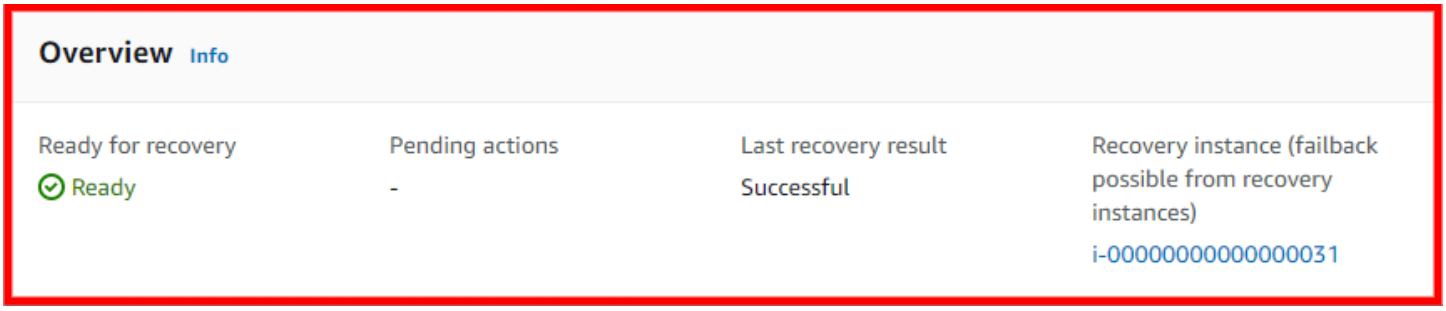


The server details view shows information and options for an individual server. Here, you can fully control and monitor the individual server.



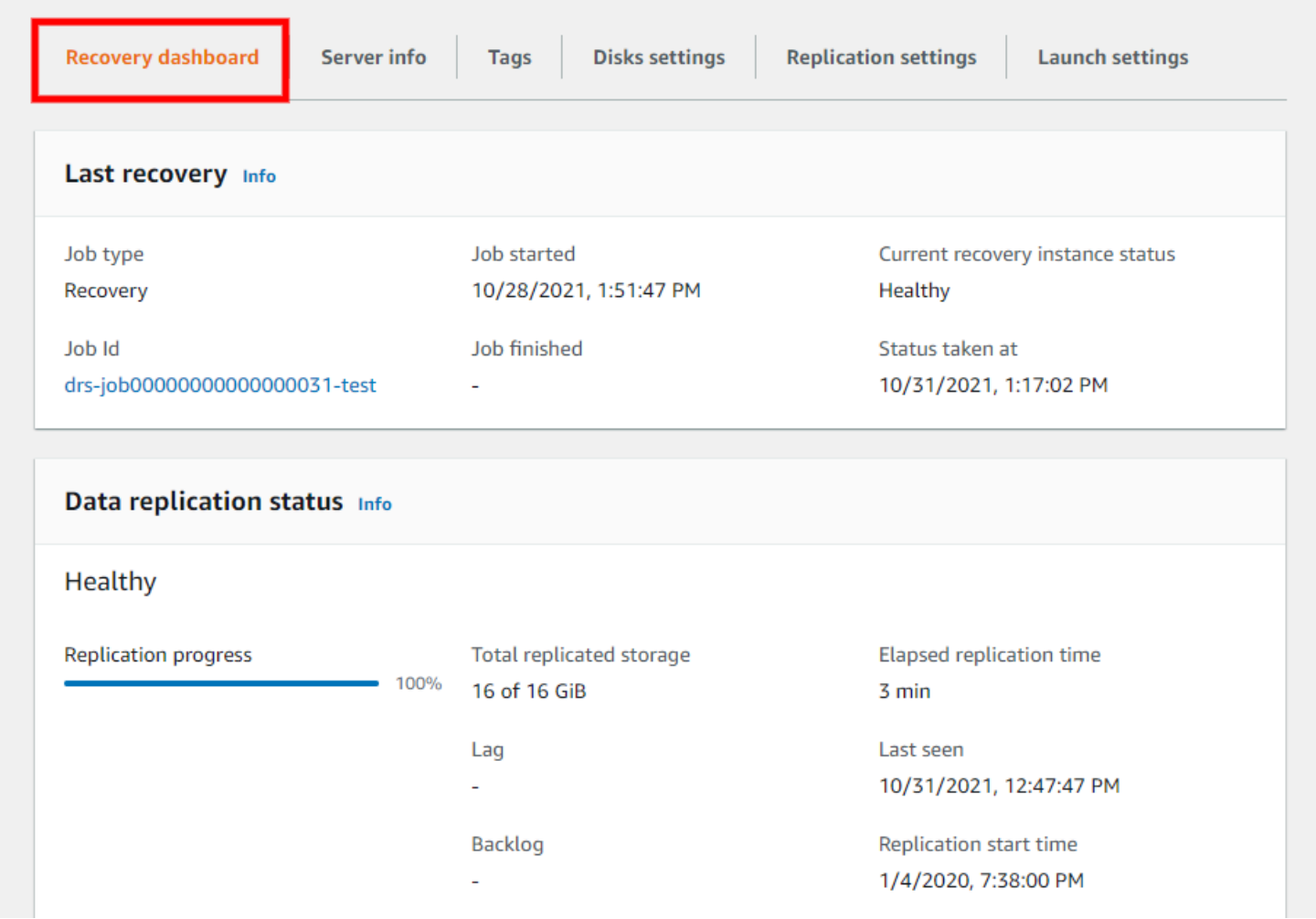
You can also perform a variety of actions, control replication, and launch Recovery instances for the individual server from the server details view.

The **Overview** box provides a basic overview of the server's status, including the whether the server is ready for recovery, any pending actions, the last recovery result (if any), and a link to the Recovery instance (if one was launched for the server).



Recovery dashboard

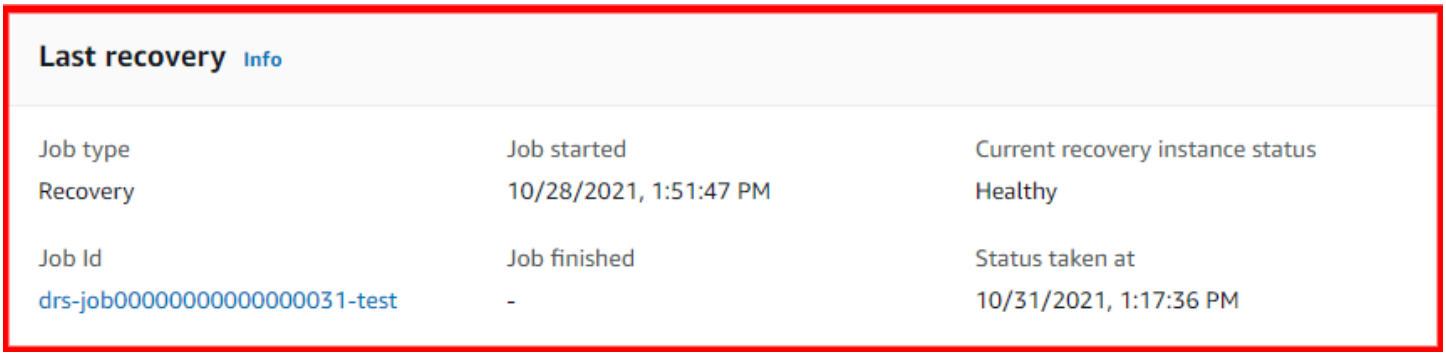
The **Recovery dashboard** tab allows you to monitor the server, its data replication status, and view events and metrics in CloudTrail.



Topics

Last recovery

The **Last recovery** box provides an overview of the recovery process for the server.

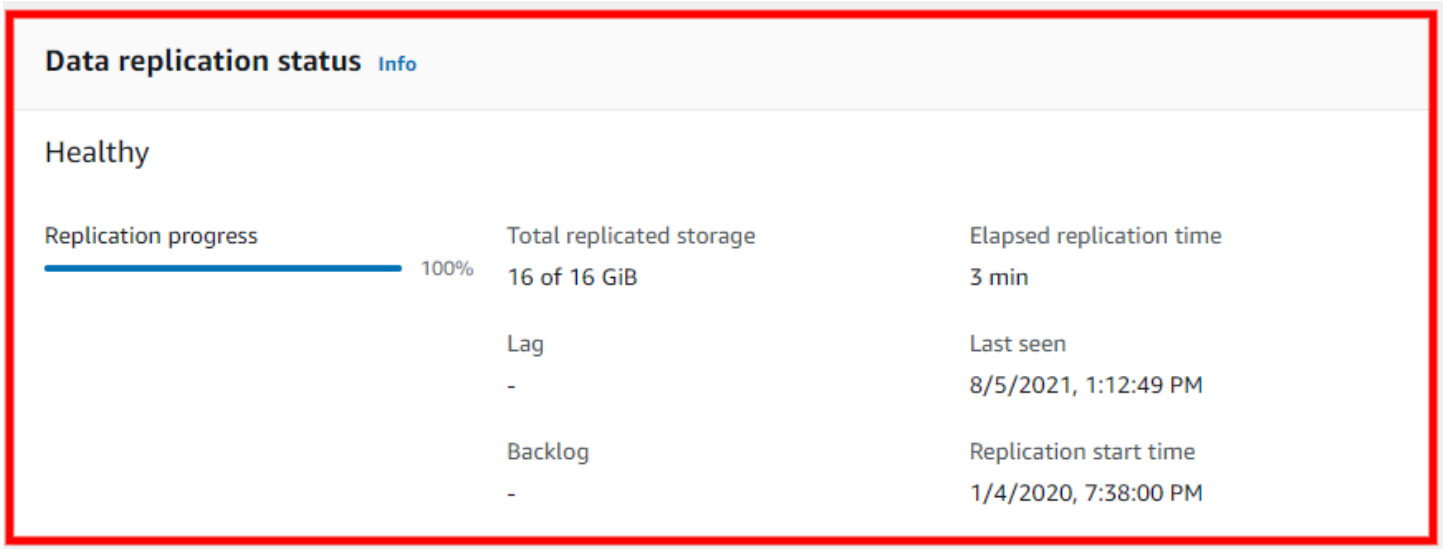


Here, you can see the following:

- **Job type** – The type of recovery job performed (drill or recovery)
- **Job ID** – The ID of the last recovery job. Choose the **Job Id** to be redirected to the **Job** page for that specific recovery launch within the **Recovery job history**.
- **Job started** – The date and time the last recovery job was started.
- **Job finished** – The date and time the last recovery job was finished. This field will be blank if the job is still ongoing.
- **Current recovery instance status** – The current status of the latest Recovery instance (if one has been launched).
- **Status taken at** – The last date and time the **current recovery instance status** was queried.

Data replication status

The **Data replication status** section provides an overview of the overall source server status, including:



- **Replication progress** – The percentage of the server's storage that was successfully replicated.
- **Rescan progress** – The percentage of the server's storage that was rescanned (in the event of a rescan)
- **Total replicated storage** – The total amount of storage replicated (in GiB).
- **Lag** – Whether the server is experiencing any lag. If it is - the lag time is indicated.
- **Backlog** – Whether there is any backlog on the server (in MiB)
- **Elapsed replication time** – Time elapsed since replication first began on the server.
- **Last seen** – The last time the server successfully connected to AWS Elastic Disaster Recovery.
- **Replication start time** – The date and time replication first began on the server.

Data replication can be in one of several states, as indicated in the panel title:

- **Initial sync**: initial copying of data from external servers is not done. Progress bar and **Total replicated storage** fields will indicate how far along the process is.
- **Healthy**: all data has been copied and any changes at source are continuously being replicated (data is flowing).
- **Rescan**: an event happened that forced the agent on the external server to rescan all blocks on all replicated disks (same as initial sync but faster because only changed blocks need to be copied; a rescan progress bar will also appear).
- **Stalled**: data is not flowing and user intervention is required (either initial sync will never complete, or state at source will become further and further the state at AWS). When the state is stalled, then the replication initiation checklist is also shown, indicating where the error occurred that caused the stalled state.

This panel also shows:

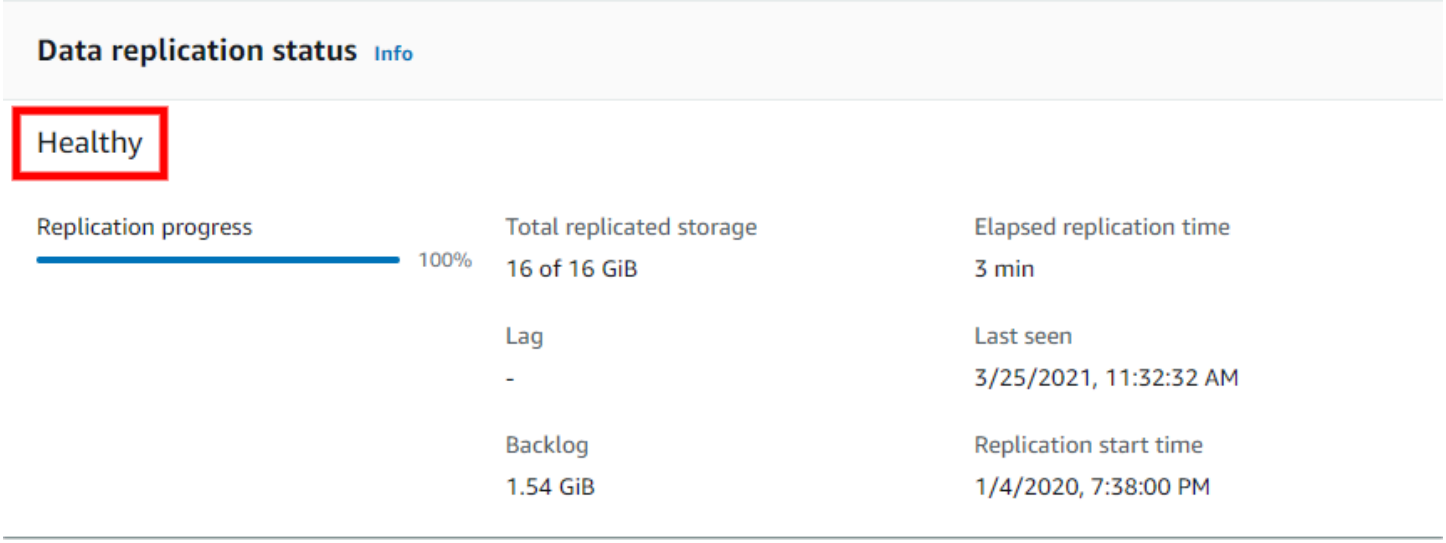
- **Total replicated storage:** size of all disks being replicated for this source server, and how much has been copied to AWS (once initial sync is complete)

  Lag: if you launch a recovery instance now, how far behind will it be from state at source. Normally this should be none.

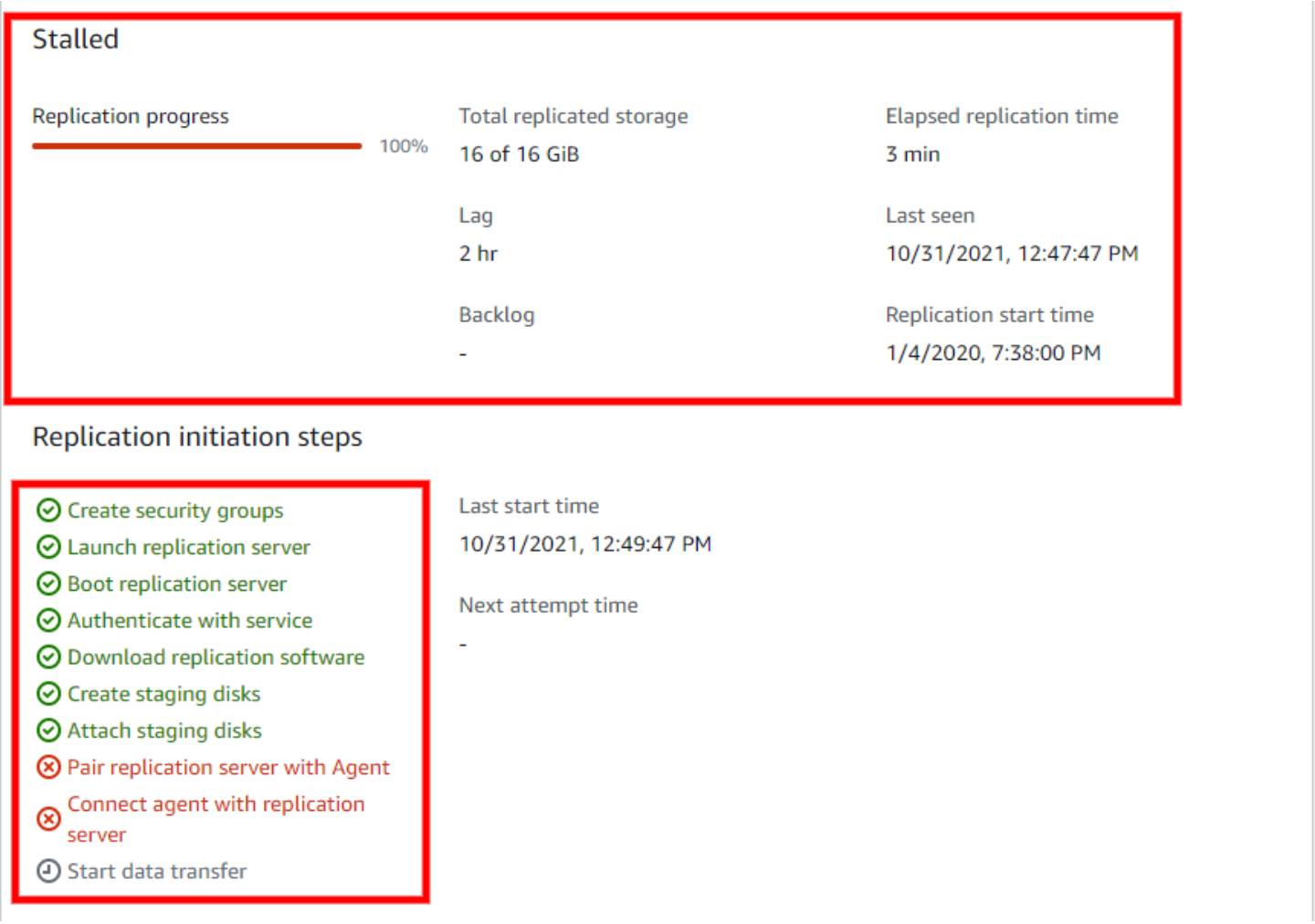
  Backlog: how much data has been written at source but has not yet been copied to AWS. Normally this should be none.

Last seen: when is the last time the AWS Replication Agent communicated with the AWS DRS service or the replication server.

If everything is working as it should and replication has finished initializing, the Data replication progress section will show a **Healthy** status.

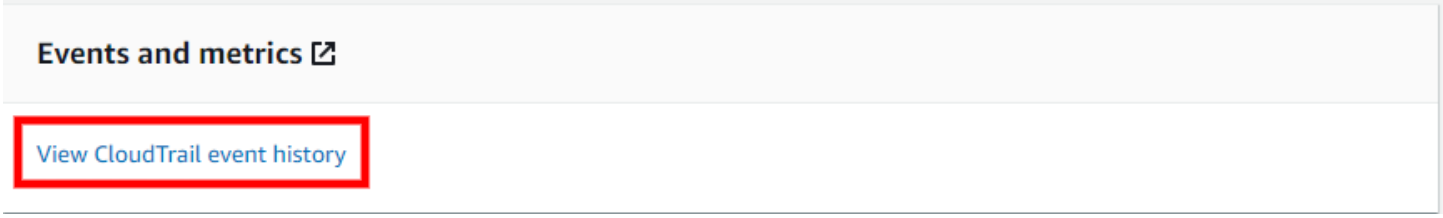


If there are initialization, replication, or connectivity errors, the **Data replication status** section will show the cause of the issue (for example, a stall). If the error occurred during the initialization process, then the exact step during which the error occurred will be marked with a red "x" under **Replication initiation steps.**



Events and metrics

You can review AWS Elastic Disaster Recovery events and metrics in AWS CloudTrail. Click on **View CloudTrail event history** to open AWS CloudTrail in a new tab.



Learn more about AWS CloudTrail events in the AWS CloudTrail user guide.

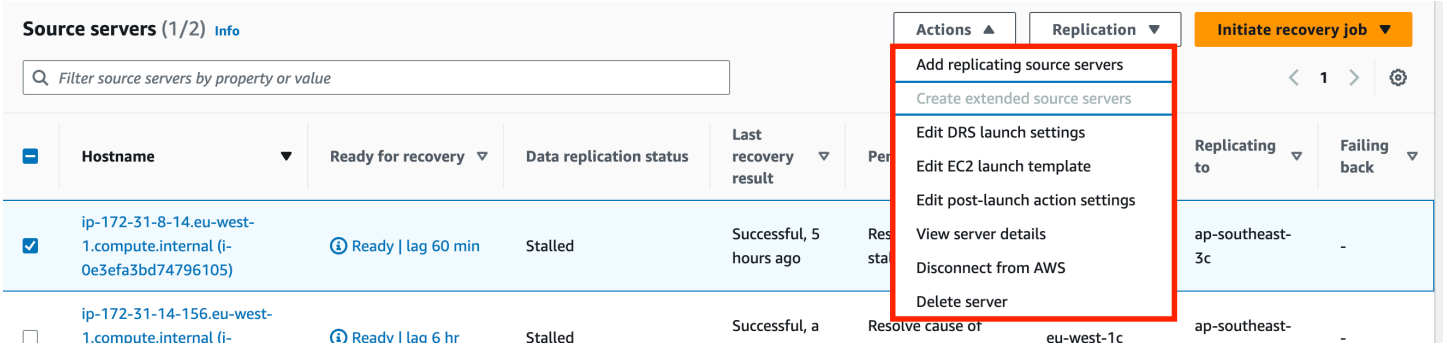
Server actions and replication control

You can perform a variety of actions, control data replication, and manage your recovery and drill instances for an individual server from the server details view.

Topics

Actions menu

The **Actions** menu allows you to perform the following actions:



- **Add servers** – Choosing this option will redirect you to the AWS Replication Agent installation instructions.
- **Edit replication settings** – Choose this option to edit the replication settings for the selected server or group of servers through on the **Edit replication settings** tab.
- **Edit launch settings** – Choose this option to enter the source server's **Server details view > Launch settings** tab.
- **View server details** – Choose this option to enter the source server's **Server details view**.
- **Disconnect from AWS** – Choose this option to disconnect the selected server from AWS Elastic Disaster Recovery and AWS.

  On the **Disconnect X server/s from service** dialog, choose **Disconnect**.

> **Important**
>
> This will uninstall the AWS Replication Agent from the source server, and data replication will stop for the source server. This action will not affect any drill or recovery instances that have been launched for this source server, but you will no longer be able to identify which source servers your Amazon EC2 instances correspond to.

- **Delete server** – Choose this option to permanently delete a source server from AWS Elastic Disaster Recovery. This will remove all information related to the server from the AWS Elastic Disaster Recovery service. You can only delete servers that have been disconnected from AWS. You will need to reinstall the AWS Replication Agent on a deleted source server to add it back to AWS Elastic Disaster Recovery.

  When the **Delete X servers** dialog appears, click **Permanently delete**.

Initiate recovery job menu

The Initiate recovery job menu allows you to start drills and recoveries by launching drill and recovery instances as part of the overall failback process. You can learn more about the entire failback and failover process with AWS Elastic Disaster Recovery in the Performing a failback and failover with AWS Elastic Disaster Recovery documentation.

- **Initiate drill** – Choose the **Initiate drill** option to launch a drill instance for this server or group of servers for the purpose of testing your recovery solution. You should perform periodic drills in order to ensure that you are ready for recovery. Learn more about launching drill instances in AWS Elastic Disaster Recovery.
- **Initiate recovery** – Choose the **Initiate recovery** option to launch a recovery instances for this server or group of servers for the purpose of recovering the server in the event of a disaster. Learn more about launching recovery instances in AWS Elastic Disaster Recovery.

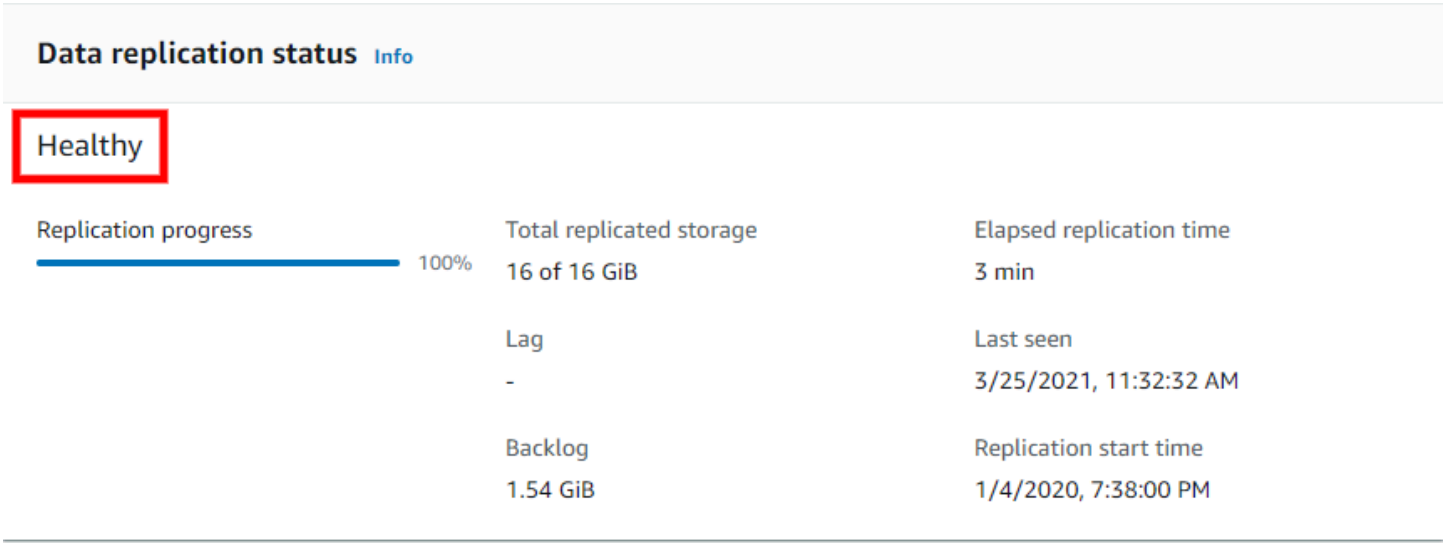
Alerts and errors

You can easily distinguish between healthy servers and servers that are experiencing issues on the **Recovery dashboard** in several ways.

The entire AWS Elastic Disaster Recovery Console is color-coded for ease of use.

Healthy servers with no errors are characterized by the color blue. The **Data replication status** boxes will display all steps and information in blue if the server is healthy.

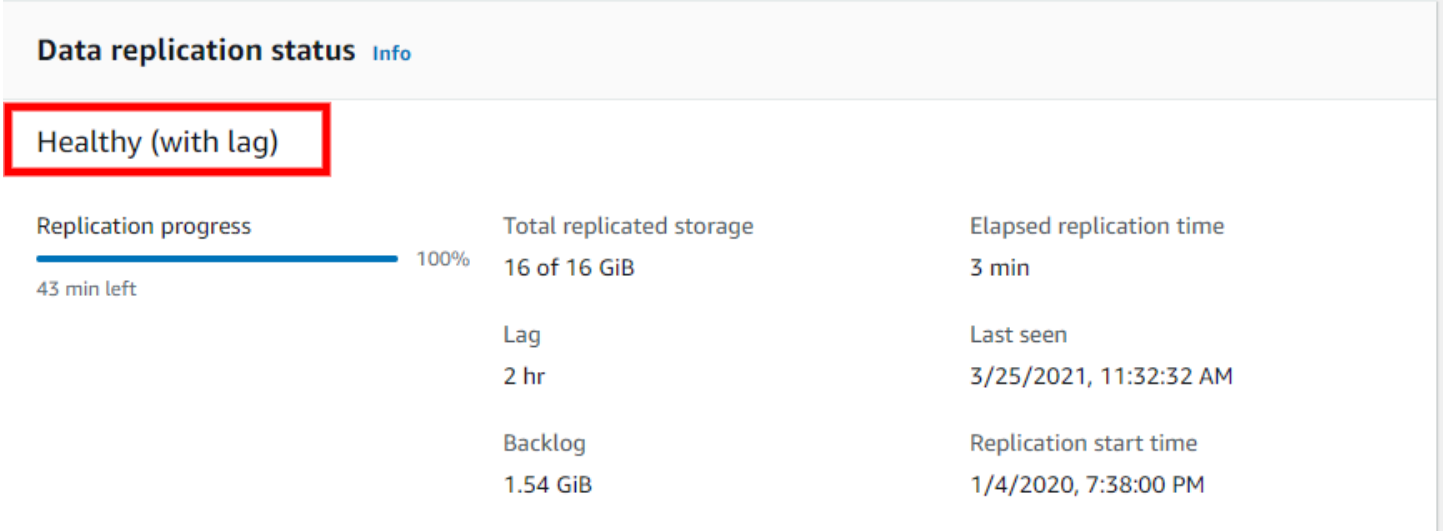
The following are examples of healthy servers:



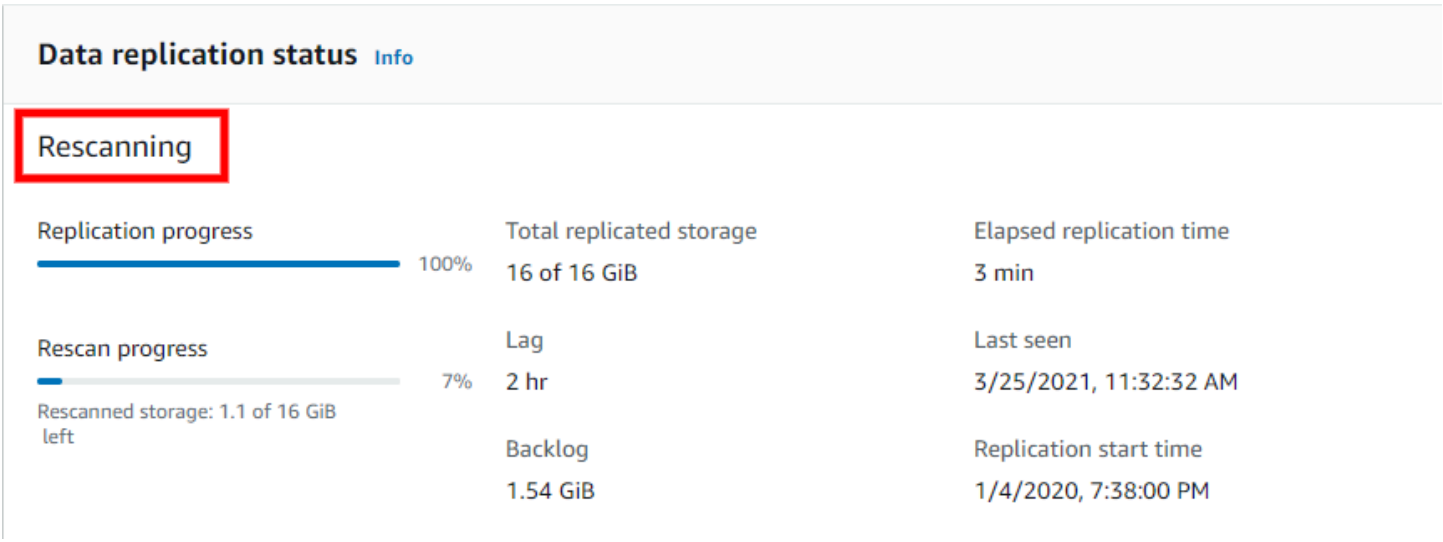
Servers that are experiencing temporary issues will be characterized by the color yellow. This can include issues such as lag or a rescan. These issues will not break replication, but may delay replication or indicate a bigger problem.

The following are examples of servers experiencing temporary issues:

Lagging server:

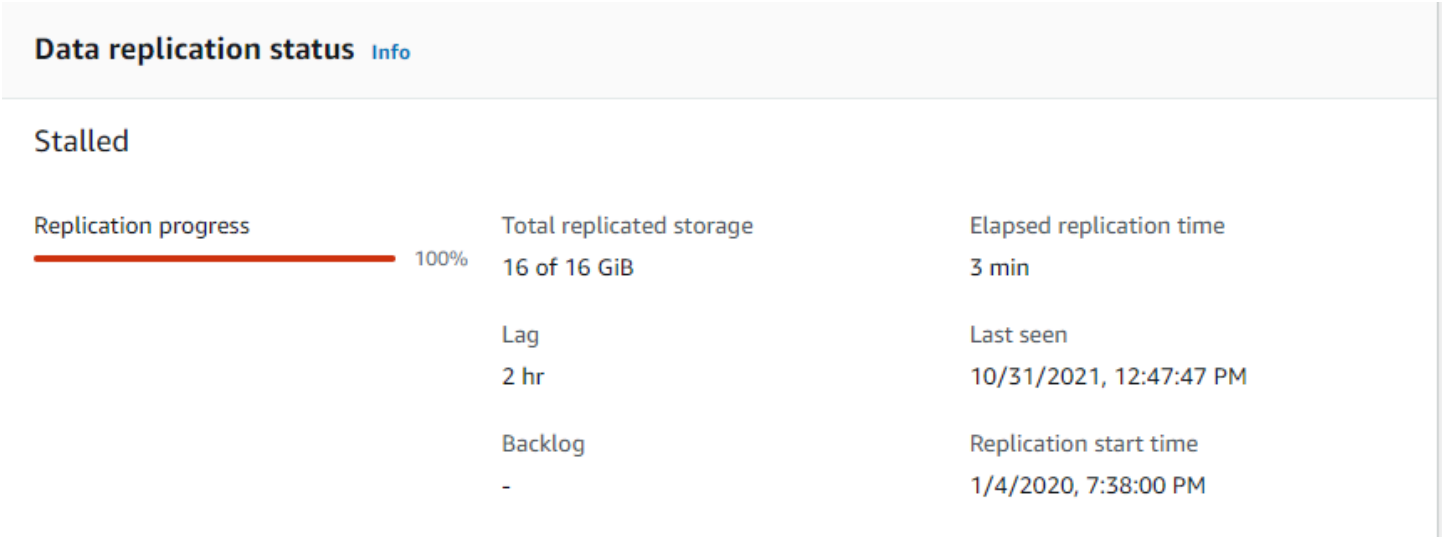


Rescanning server:

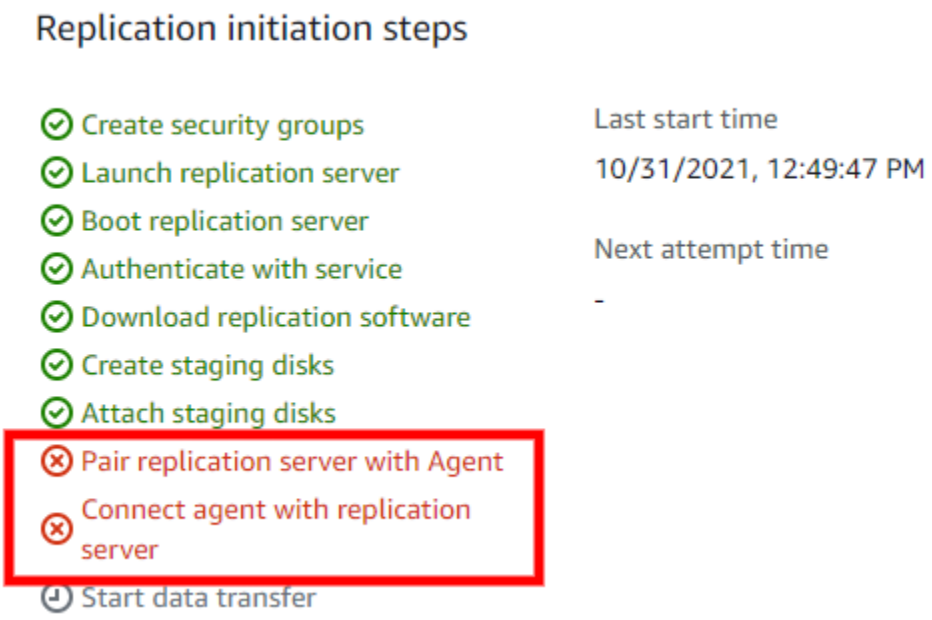


Servers that are experiencing serious issues will be characterized by the color red. These issues can include a loss of connection, a stall, or other issues. You will have to fix these issues in order for data replication to resume.

The **Data replication status** box will include details of the issue.

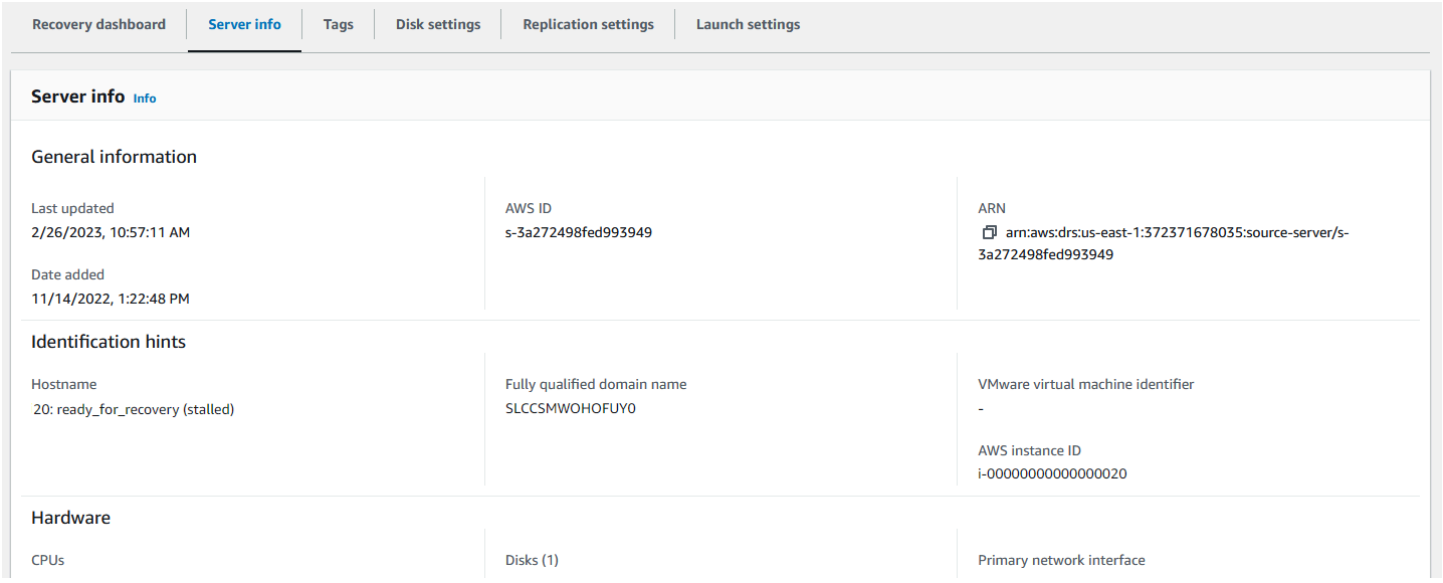


If the stall occurred during initiation, scroll down to **Replication initiation steps**. The exact step where the issue arose will be marked with a red "x".



Server info

The **Server info** tab shows a variety of general server information, hardware, and network information.



This tab shows you general information about the source server:

- **General information**
  - **Last updated**: when was the data in this tab updated.

  - **Date added**: when was this server added to the service.
  - **AWS ID**: the ID of this source server resource.
  - **arn**: the AWS Resource Name for this source server.
- **Identification hint**s: under most circumstances, the hostname is the best identifier, as it is what is used throughout the console as the name of the source server. If you need to validate which external server this is referring to in your data center, you can use one of the additional fields: Fully qualified domain name, VMware virtual machine identifier (only if source is VMWare), AWS instance ID (only is source is running on AWS).
- **Hardware and operating system**: the CPUs, RAM, disks, and network interfaces on the external server, as well as the type and full name of the operating system running on that server. The disks shown are all the disk on the source server, and may include disks not being replicated.
- **Recommended instance type**: this is the EC2 instance type the service is auto-recommending to use for the launched recovery instance. This is based only on the CPUs and RAM at the source (and not on utilization information). This is the instance type that will be launched for this server by default.

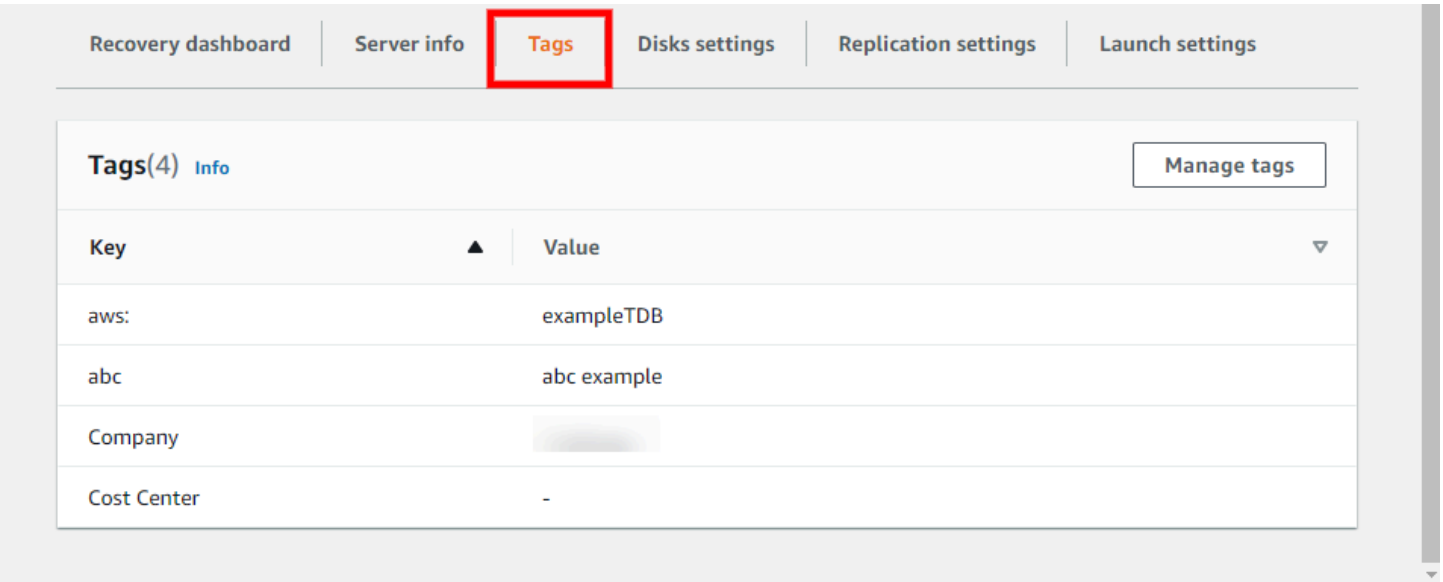
Information shown includes:

- **Last updated**
- **Date added**
- **AWS ID** (if relevant)
- **Hostname**
- **Fully qualified domain name**
- **VMware virtual machine identifier** (if relevant)
- **AWS instance ID**
- **AWS ID**
- **CPUs**
- **RAM**
- **Disks**
- **Network interfaces**
- **Operating system** information
- **Recommended instance type**

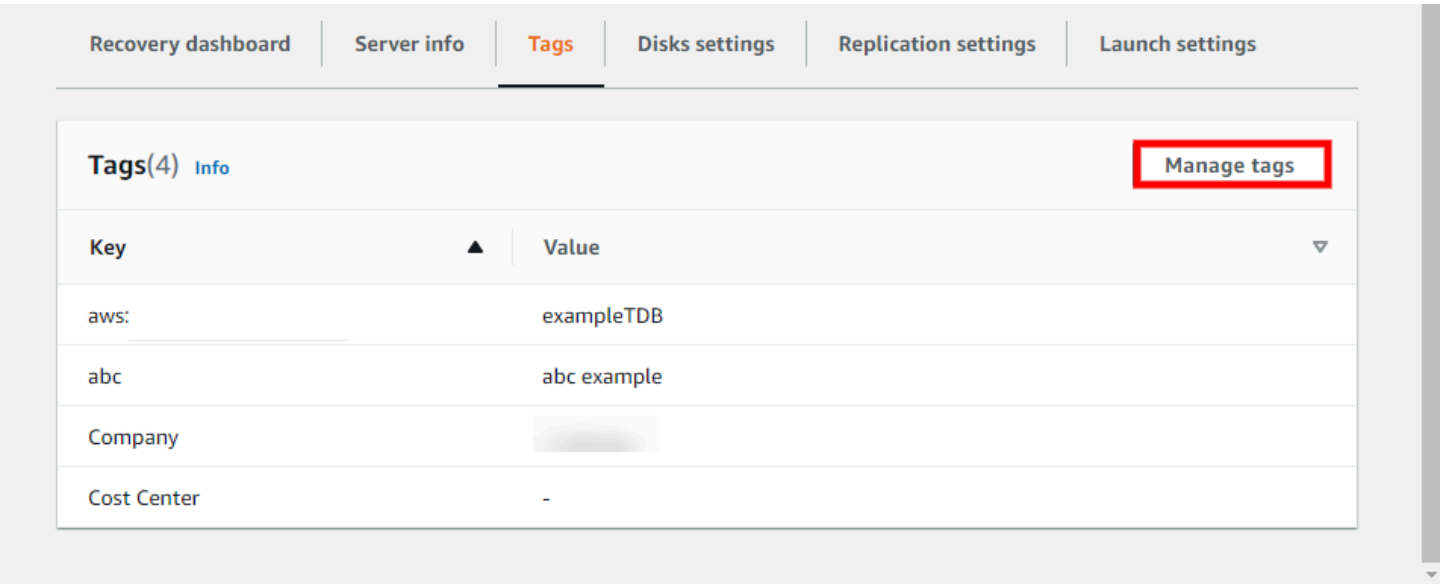
Tags

The Tags section shows any tags that have been assigned to the server. A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Learn more about AWS tags in this Amazon EC2 article.

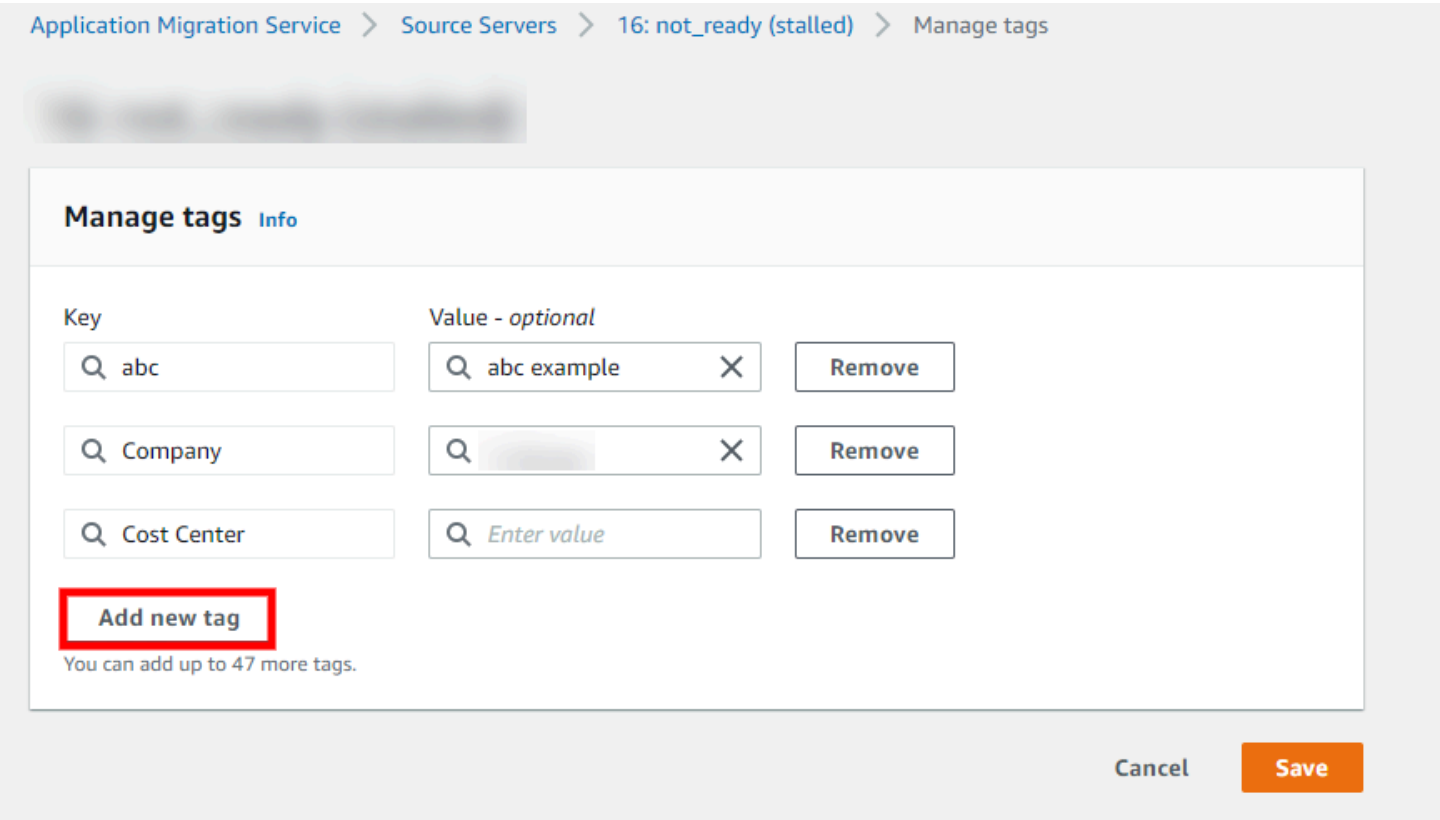
> **Important**
>
> **Do not** alter the **Name** tag of resources created by AWS DRS (replication servers, EBS volumes, EBS snapshots, Conversion servers).



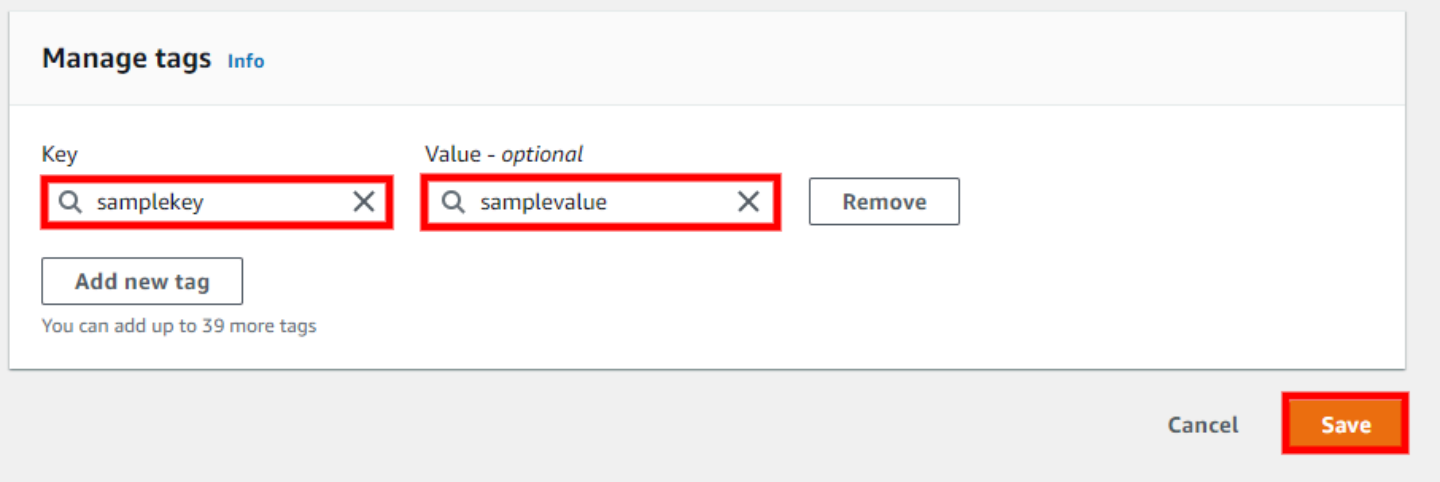
Choose **Manage tags** to add or remove tags.



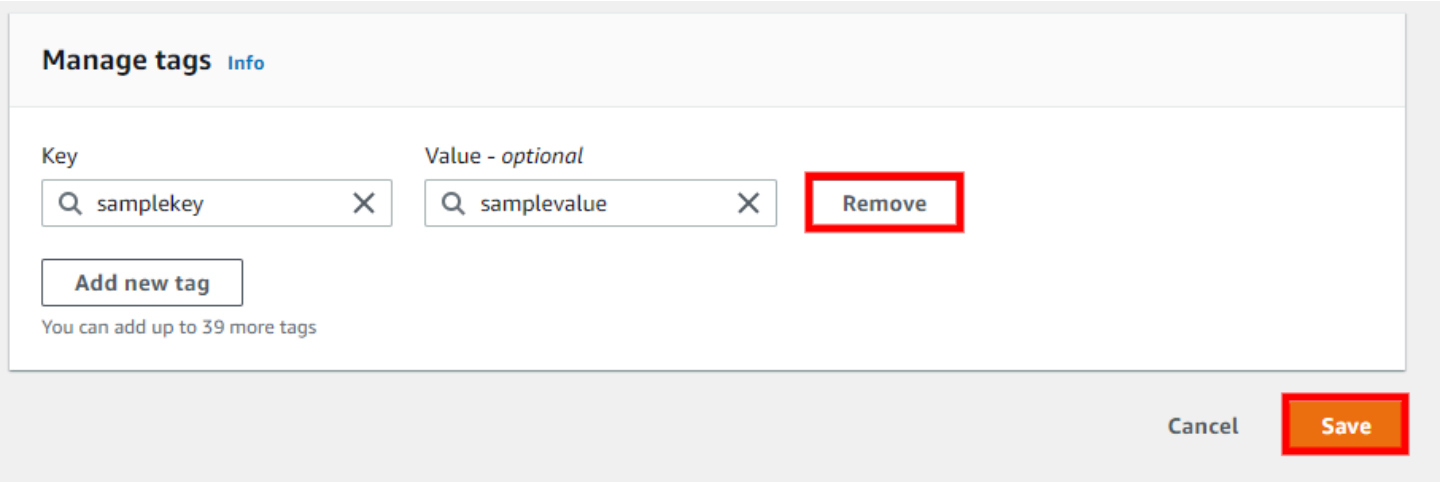
The **Manage tags** page will open. Choose **Add new tag** to add a new tag



Add a tag **Key** and an optional tag **Value**. Choose **Save** to save your added tags.

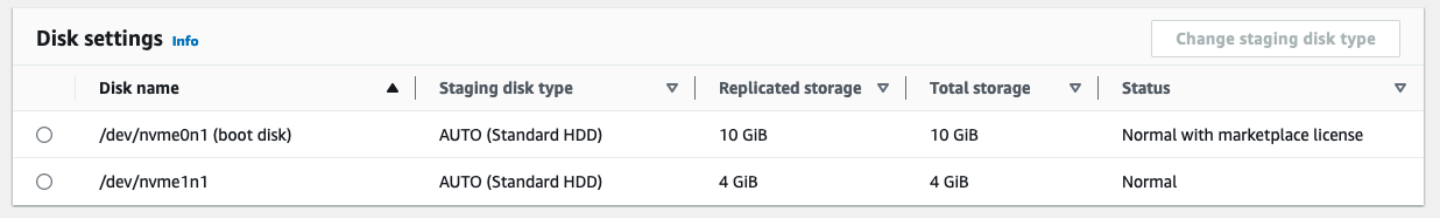


To remove a tag, choose **Remove** to the right of the tag you want to remove, and then choose **Save**.



Disk settings

The **Disk settings** tab shows a list of all of the disks on the source server and information for each disk.



Disk settings include:

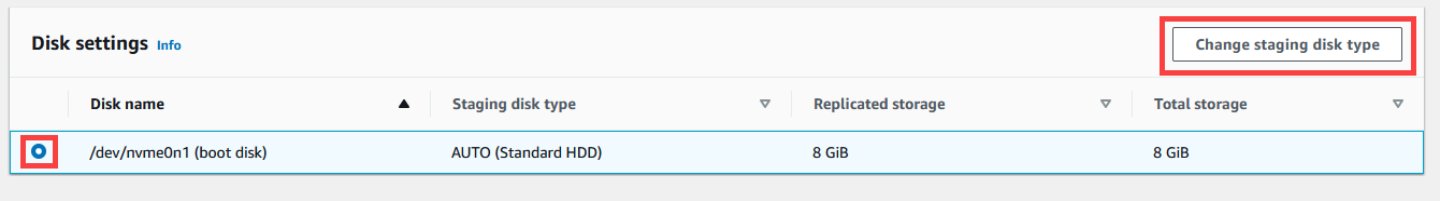
- **Disk name**

- **Staging disk type** – The corresponding Amazon EBS volume disk type that is being used for the disk.
- **Replicated storage** – The amount of storage that has been replicated from the disk to the Replication Server.
- **Total storage** – The total storage capacity of the disk.
- **Status** – shows the status of each disk, values can be either **Normal**, **Normal with marketplace license**, **Error** (with error description). Normal with marketplace license means that the server has at least one marketplace license associated with this volume. Volumes with marketplace licenses pose some limitations on launch: the target region and the selected instance type must support this license. If launching into a different account, the marketplace product must be subscribed to in that account as well or the launch will fail. The state is set to Error if there is a problem with the volume, such as not having permissions to read the marketplace license details if the server is owned by a different AWS account. The value can also be empty if the status is not known at this time.

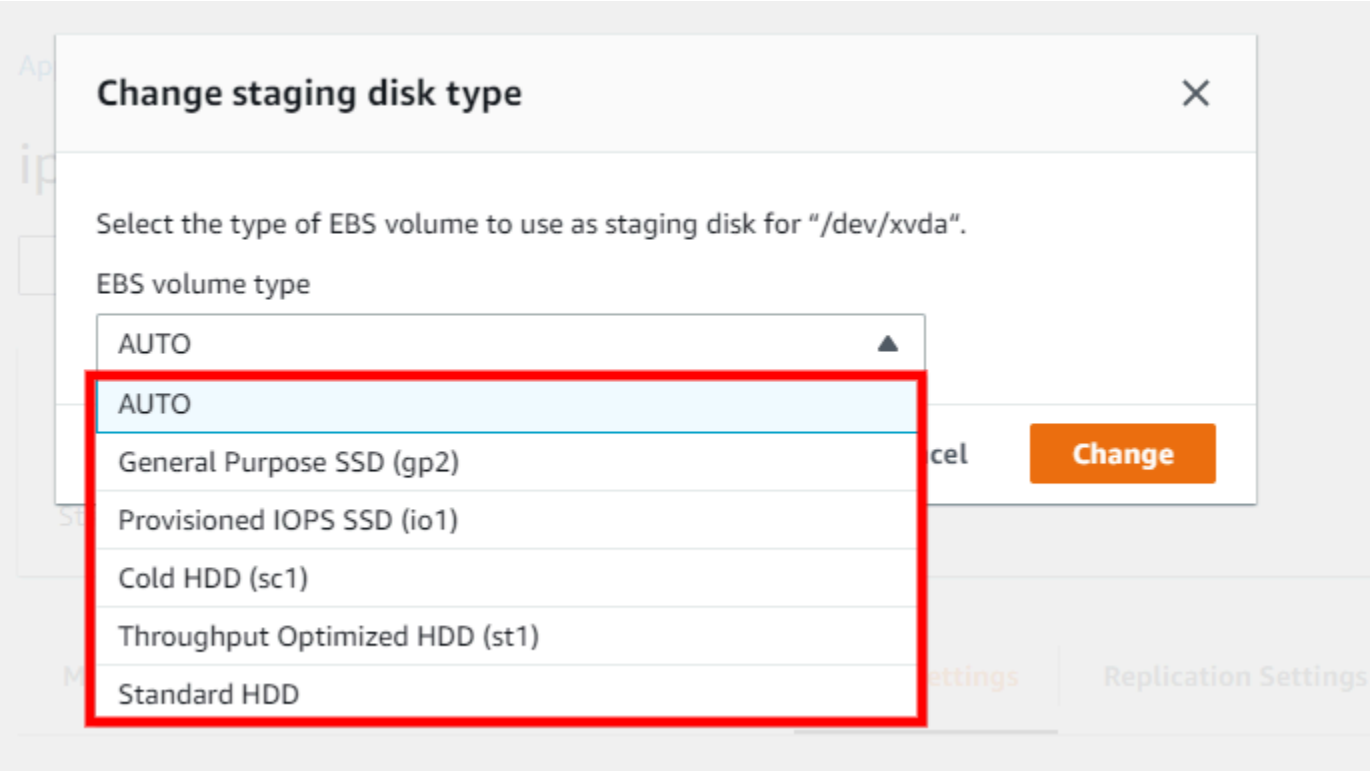
Change staging disk type

You can change the EBS volume disk type for each disk or for a group of disks.

To change the EBS volume disk type, select the circle to the left of each disk name and choose **Change staging disk type**.



On the **Change staging disk type** dialog, select the type of EBS volume to use for the disk or group of disks.



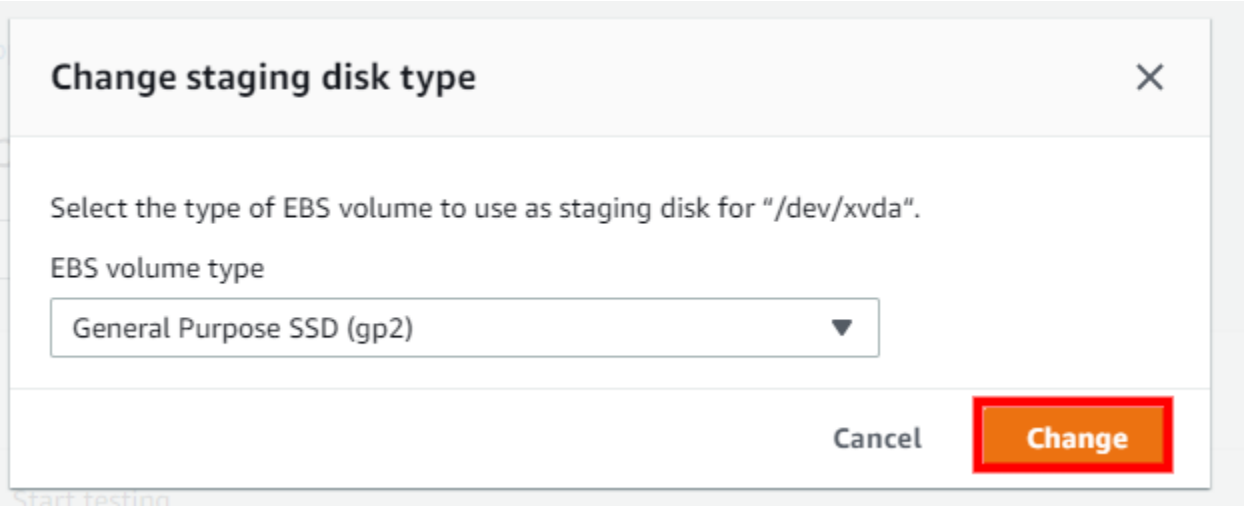
Select the **AUTO** option if you want AWS Elastic Disaster Recovery to automatically select the most cost-effective EBS volume disk type for each disk based on the disk size and type based on the option you defined in the **Replication settings** (either the default **Lower cost, Throughput Optimized HDD (st1)** option or the **Faster, General Purpose SSD (gp2) or (gp3)** s option).

AWS Elastic Disaster Recovery uses a single Replication Server per 15 source disks. Selecting the **Auto** option will ensure that the least amount of replication servers are used, resulting in increased cost savings.

> **Note**
>
> AWS Elastic Disaster Recovery will always use EBS magnetic volumes for disks that are under 125 GiB in size, no matter which option is selected.

If you do not want AWS Elastic Disaster Recovery to automatically select a disk, you can select a disk manually. Select the disk type from the **EBS volume type** menu.

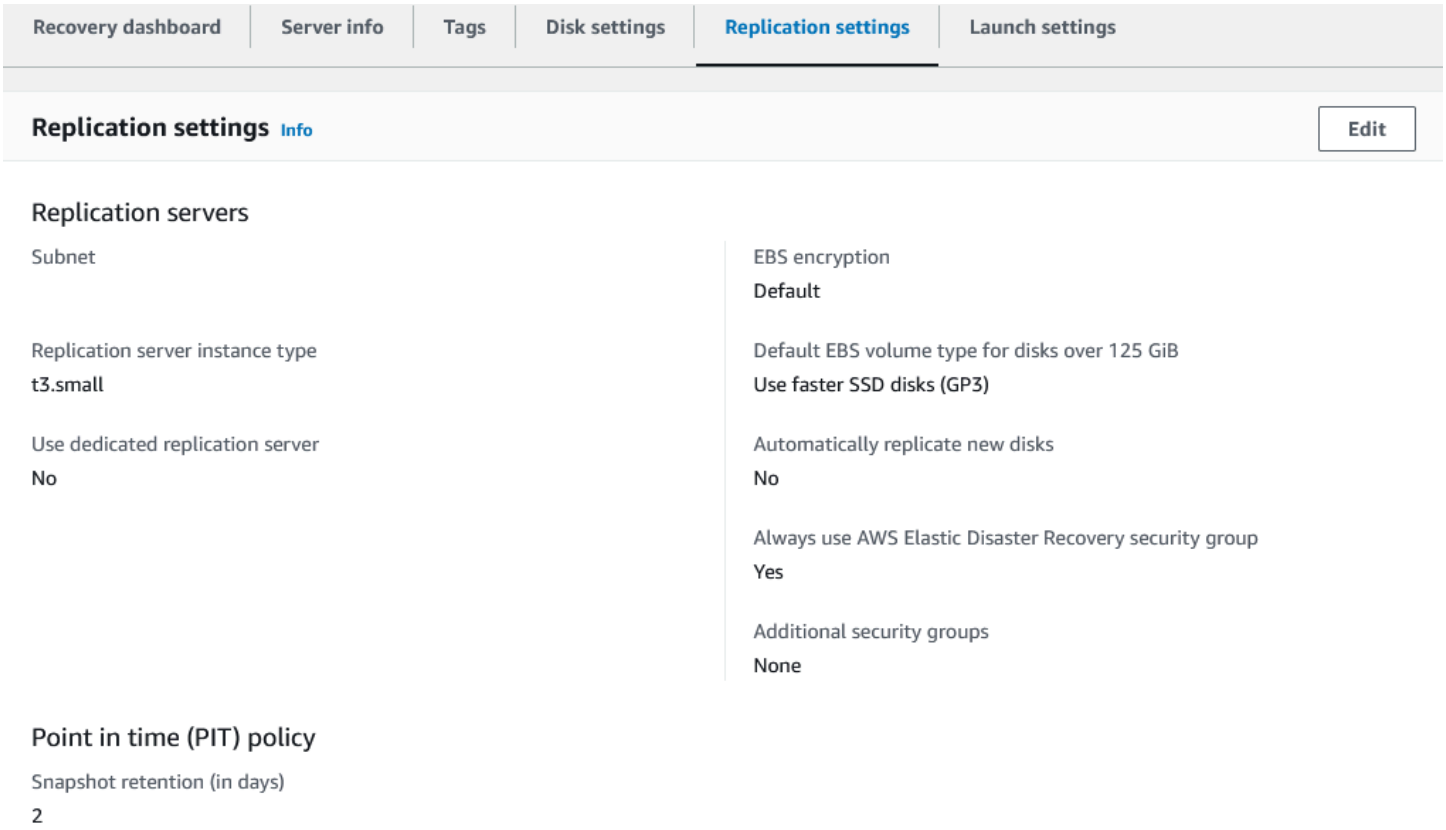
For certain disks, you can configure the amount of IOPS to be allocated per GB of disk space under **IOPS**. You can allocate up to 50 IOPS per GB. 64,000 IOPS are available for Nitro-based instances. Other instances are guaranteed up to 32,000 IOPS. The maximum IOPS per instance is 80,000.



Choose **Change** to confirm the change.

Replication settings

The **Replication settings** tab allows you to edit the replication settings for an individual source server.



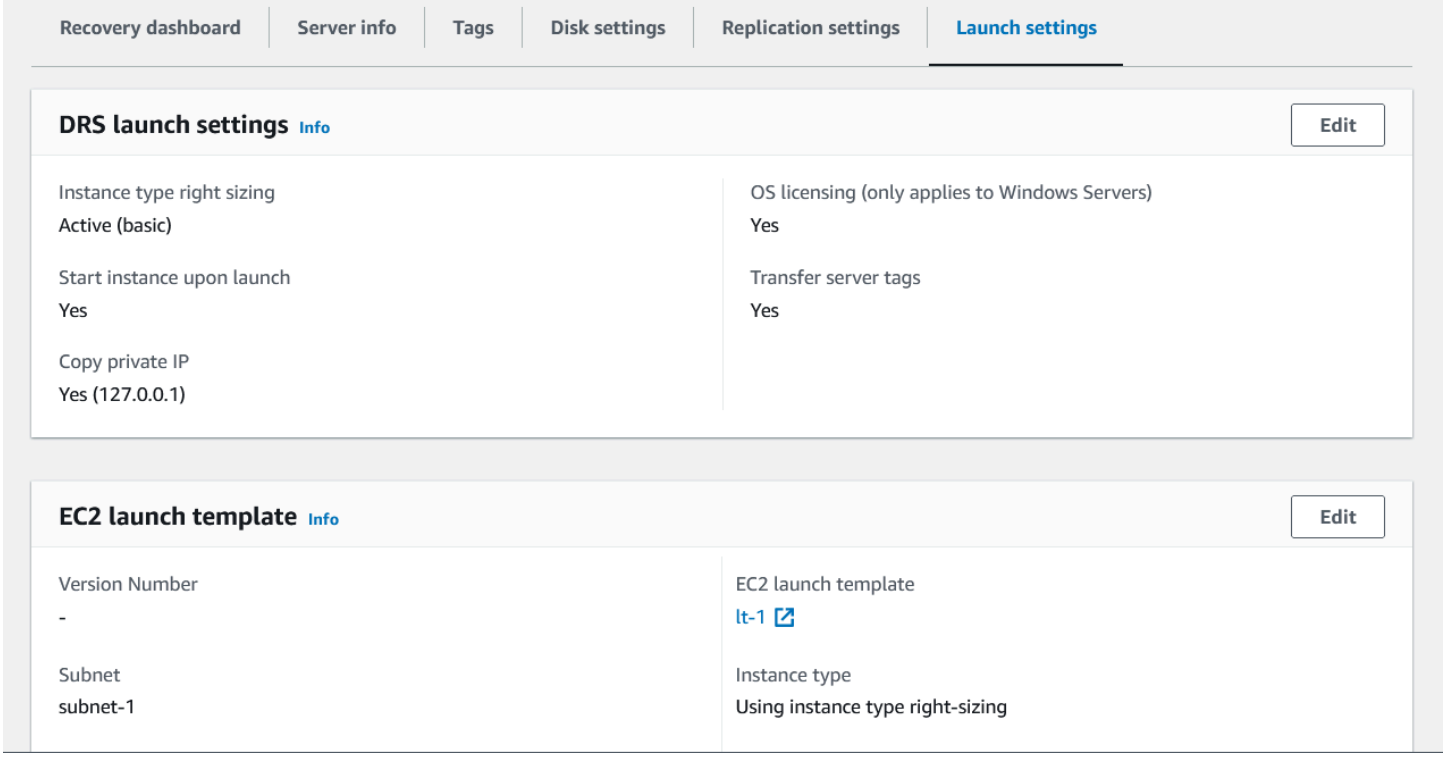
After the source server is added to AWS Elastic Disaster Recovery, the replication settings that are defined in the Replication Settings template are automatically applied to the server. You can later edit them for a single source server or multiple source servers through the **Replication settings** tab.

Edit each setting as required and then choose **Save replication settings**.

Learn more about replication settings.

Launch settings

The launch settings are a set of instructions that comprise an EC2 launch template and other settings, which determine how a recovery instance will be launched for each source server on AWS.



Launch settings, including the EC2 launch template, are automatically created every time you add a server to AWS Elastic Disaster Recovery.

The launch settings can be modified at any time, including before the source servers have even completed initial sync.

Learn more about individual launch settings.

Important

If the source server's instance type includes instance store, please consider the following:

- It is **not** recommended to change the instance type of an instance to a type that has no ephemeral volumes, or has a different number of ephemeral volumes, as such changes could lead to data inconsistencies and may even cause recovery, drill, or failback to fail.

Post-launch settings

Post-launch settings allow you to control and automate actions performed after a recovery instance has been launched for the source server in AWS. These settings are created automatically based on the **Default post-launch actions**.

Activating the post-launch actions for a specific source server:

- Navigate to the **Source servers** page and select a source server.
- Go to the **Post-launch settings** tab. If **Post launch action settings** has **Post launch actions** set to **Active**, click **Edit** for **Post launch action settings**.
- You will be redirected to the **Edit post-launch settings** screen. Make sure the **Post-launch actions active** option is not checked and click **Save**.

Alternatively, you can activate and deactivate post-launch actions for multiple servers by navigating to the **Source servers** page, selecting the servers you want to update and clicking **Actions > Edit post-launch action settings**. To activate, make sure the **Post-launch actions active** option is checked, and to deactivate, it should be unchecked. If you made a change, click **Save**.

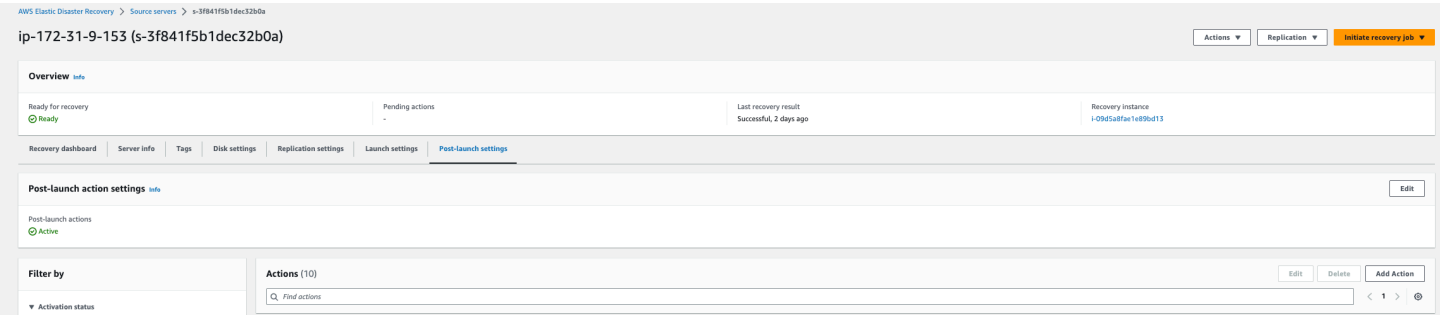
Topics

Adding custom actions

AWS Elastic Disaster Recovery (AWS DRS) allows you to run any SSM document that you like – public SSM documents, SSM documents that you created and uploaded to your account or SSM documents that are shared with you. You can configure a custom action to run any SSM document that is available in your account. To be able to create, edit or delete a custom action, make sure the post-launch actions are activated for this source server. Custom actions added to the default settings are automatically added to newly added source servers.

Create a custom action

Adding a custom action through source server's **Post-launch settings**, adds it to this source server. To add a custom action to all newly added source servers, do so using the **Settings** → **Default post-launch actions** page. To add a new custom action to the source server, go to **Source server details** → **Post-launch settings** tab. If the **Post-launch actions** post-launch actions settings is **Active**, you can create new custom actions by clicking on the **Add action** button.



The **Add action** page includes the following parameters:

Action name – The name of the action in AWS DRS, which should be intuitive, meaningful and unique in this AWS account and region.

Activate this action – Use this checkbox to activate or deactivate the custom action for this source server. Only active actions will run after the launch of a recovery instance.

Mark launch as successful only if this action finishes running successfully – This checkbox will dictate whether or not the launch will be marked as successful, based on the successful run of this action. Instance launches will still progress normally regardless of the success of the action.

System Manager document name – Select any Systems Manager document that is available to be used in this account.

View in Systems Manager – Click to open **System Managers** and view additional information about the document.

Description – Add a description or keep the default.

Document version – Select which SSM document version to run. AWS DRS can run a default version, the latest version, or a specific version, according to your preferences.

Category – Select from various available categories including monitoring, validation, security and more.

Order – Specify the order in which the actions will be executed. The lower the number, the earlier the action will be executed. Values allowed are between 2 and 10,000. The numbers must be unique but don't need to be consecutive.

Platform – Taken from the SSM document and reports which Operating System platform (Windows/Linux) is supported by the action.

Creator – Who created the action. For custom actions, the default is always **This account**.

The **Action parameters** change according to the specific SSM document that is selected. Note that for the instance ID parameter, you can choose to use the launch instance ID, in which case, AWS DRS will dynamically populate the value.

Note

AWS Elastic Disaster Recovery (AWS DRS) places **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole** instance profile on the launch instance if post-launch actions is active for the source server. If you add an SSM command action that requires additional permissions in the launch instance, you must ensure that the instance profile has the right policies or the right permissions. In order to do so, create a role that has the required permissions as per the policies above or has a policy or policies with those permissions attached to it. Go to **Launch settings** > **EC2 launch template** > **Modify** > **Advance** > **IAM instance profile**. Use an existing profile or create a new one using the **Create new IAM profile** link.

Note

Only trusted, authorized users should have access to the parameter store. For enhanced security, ensure that users who do not have permissions to execute SSM documents / commands, do not have access to parameter store. Learn more about restricting access to

Systems Manager parameters. Action parameters are stored in the SSM parameter store as regular strings. Changing parameters in the SSM Parameter store may impact the post launch action run on target instances. We recommend to consider security implications, when choosing to use parameters that contain scripts or sensitive information, such as API keys and database passwords.

Activating, deactivating and editing predefined or custom actions

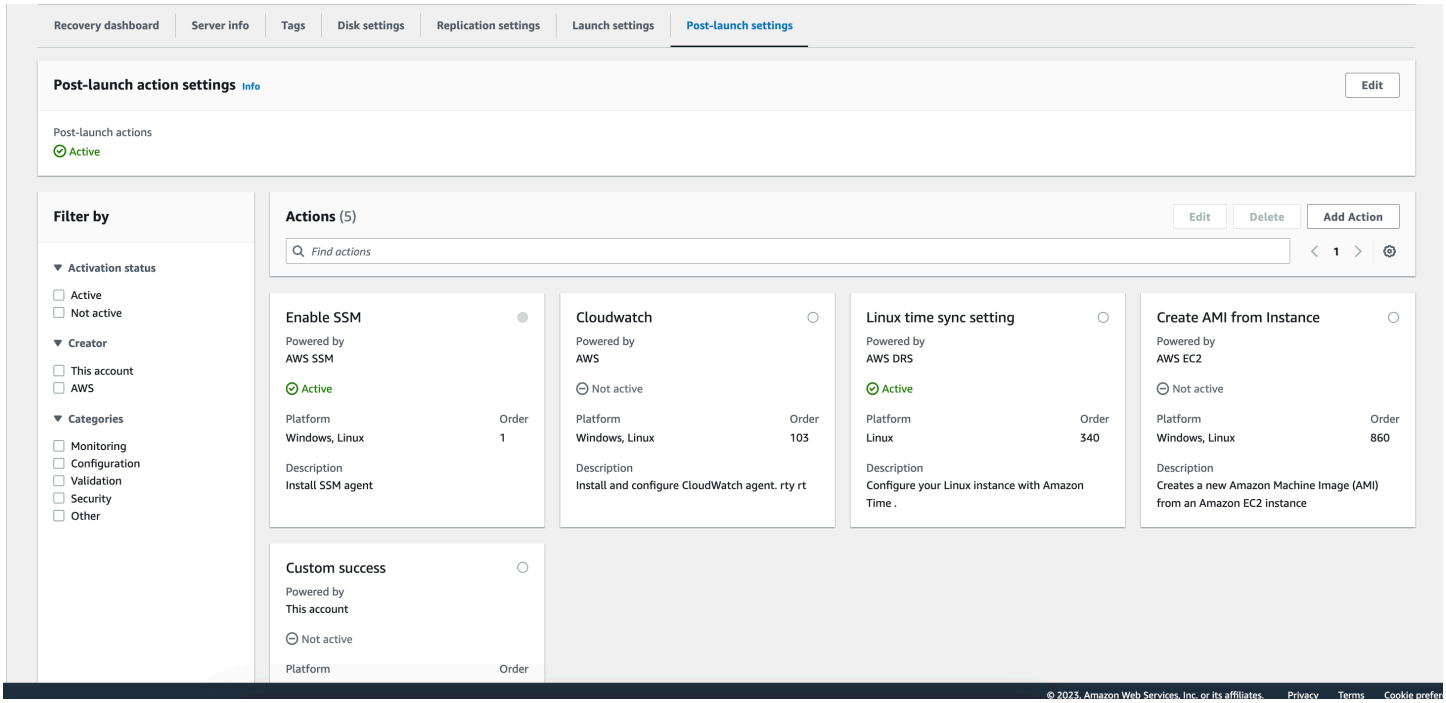
You can activate, deactivate and edit actions available for this source server. Activating an action will ensure it runs after launching a recovery instance. Likewise, deactivating it, prevents it from being run after launching a recovery instance. The default settings are not affected by activating, deactivating or editing an action for a source server. Editing an action for a source server updates it for that source server. These changes are not reflected on the action, if it exists in the default post-launch actions settings. Changes to actions in the default settings, as to apply to newly added source servers, can be done from the **Settings** → **Default post-launch actions** page.

To be able to activate, create, deactivate, edit, or delete a custom action and to activate, deactivate or edit predefined actions for a source server, make sure the post-launch actions are activated for that source server.

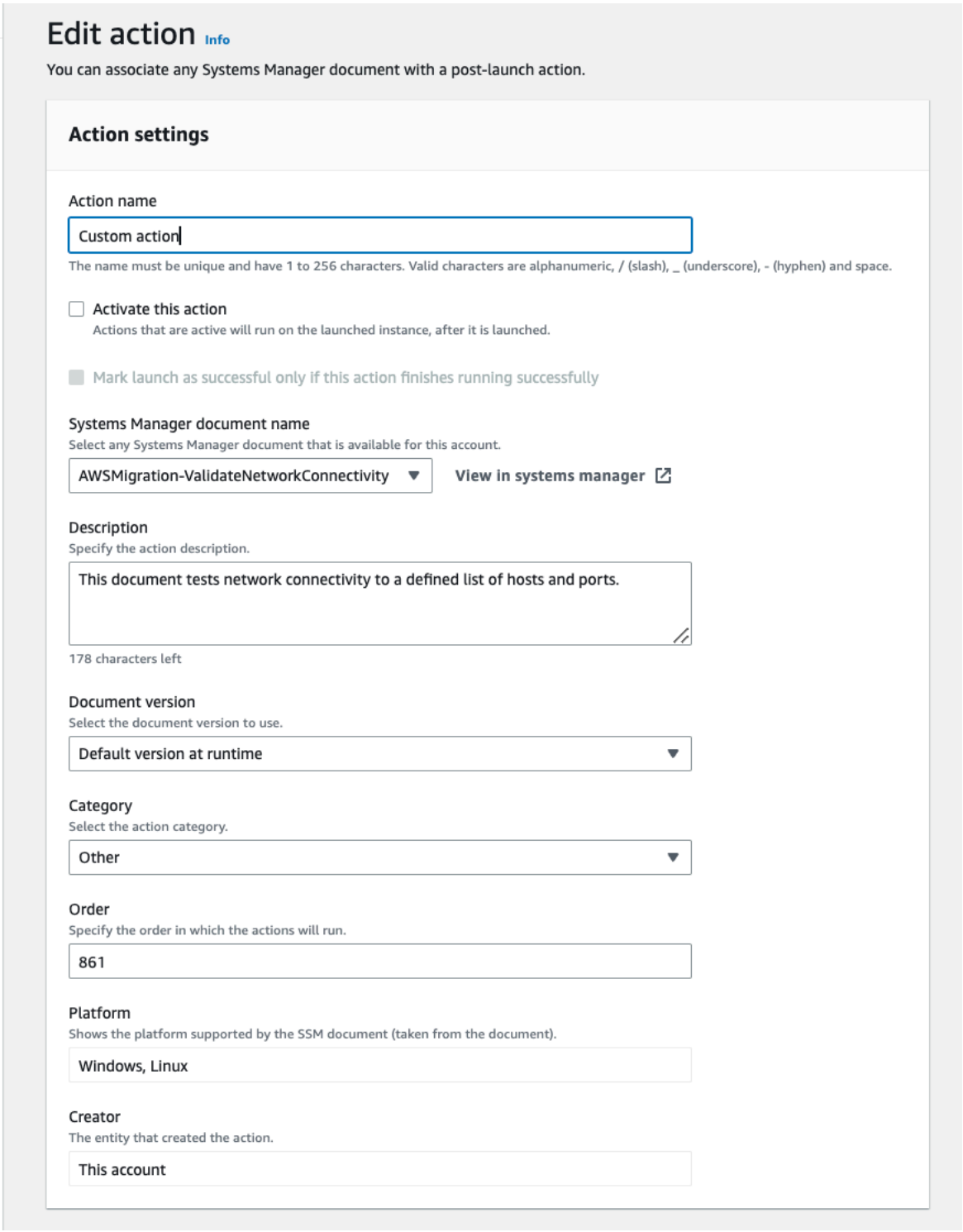
Activating, deactivating and editing predefined or custom actions

To activate, deactivate or edit a post launch action in the default post-launch actions settings, go to **Source server details** page, and visit the **Post-launch settings** tab. If **Post-launch actions settings** shows **Post-launch actions** to be **Active**, you can edit any action defined for the source server.

Locate the action you want to edit in the **Actions** card view, or use the search field to filter the actions by name.



Click on the action's card to select it, and then click on the **Edit** button.



To activate the action, make sure the **Activate this action setting** is checked and click the **Save** button. To deactivate, make sure the **Activate this action** setting is un-checked and click the **Save** button.

The edit page allows to change the value of some of the parameters for both pre-defined actions and custom actions. Some parameters can only be edited if the action is a custom action. See below for specific information.

The parameters that appear on the edit page:

Action name – Editable for custom actions. The name of the action in AWS DRS, which should be intuitive, meaningful and unique in this AWS account and region.

Activate this action – Use this checkbox to activate or deactivate the action for this source server. Only active actions will run after the launch of a recovery instance.

Mark launch as successful only if this action finishes running successfully – This checkbox will dictate whether or not the launch will be marked as successful, based on the successful run of this action. Instances launches will still progress normally regardless of the success of the action.

System Manager document name – Editable for custom actions. Select any Systems Manager document that is available to be used in this account.

View in Systems Manager – Click to open **System Managers** and view additional information about the document.

Description – Editable for custom actions. Add a description or keep the default.

Document version – Editable for custom actions. Select which SSM document version to run. AWS DRS can run a default version, the latest version, or a specific version, according to your preferences.

Category – Editable for custom actions. Select from various available categories including monitoring, validation, security and more.

Order – Specify the order in which the actions will run. The lower the number, the earlier the action will run. Values allowed are between 2 and 10,000. The numbers must be unique but don’t need to be consecutive.

Platform – Not editable. Taken from the SSM document and reports which Operating System platform (Windows/Linux) is supported by the action.

Creator – Not editable. Who created the action. For custom actions, the default is always **This account**.

The **Action parameters** change according to the specific SSM document that is selected. Note that for the instance ID parameter, you can choose to use the launch instance ID, in which case, AWS DRS will dynamically populate the value. Some predefined actions, where applicable allow to use a dynamically populated value for the volumes. This value will be dynamically populated by AWS DRS with the volumes of the instance being launched.

After making the required changes, click **Save**, to save the changes and **Cancel** to abort them.

Deleting custom actions

Custom actions added to a source server from the default settings on creation or created later for that source server can also be deleted. Deleting a custom action for a source server removes it from that source server and means the action will no longer be available to that source server. Deleting the action for a source server does not remove it from the default settings if the action was defined there as well. To delete a custom action from the default settings to avoid adding it to newly added source servers, go to the **Settings** → **Default post-launch actions** page, and delete the action from there. Pre-defined actions cannot be deleted. If a pre-defined action is not required, it can be deactivated.

Locate the action you want to delete in the **Actions** card view, or use the search field to filter the actions by name. Select the action, and click the **Delete** button. To confirm, press **Delete**.

Predefined post-launch actions

AWS Elastic Disaster Recovery allows you to run various predefined post-launch actions on your EC2 launched instance. Use these out-of-the-box actions to improve your launch flexibility.

These actions can be activated, edited or deactivated for a specific source servers.

List of available pre-defined actions

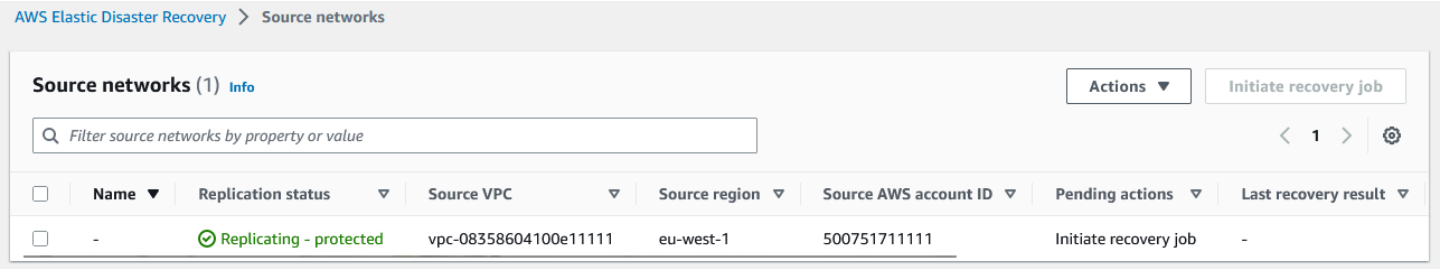
Source networks

The network replication feature allows you to keep track of network changes and perform quick updates. The feature helps prevent configuration mismatch during recovery, saves time and resources and provides enhanced security. For example, when a security group is updated, this change will be automatically replicated, ensuring compliance and preventing potential security risks. In addition, recovery instances will be launched within the recovered source networks automatically, preventing the need to configure each server manually.

> **Important**
>
> Only in-AWS networks can be replicated.

Source network page

The **Source networks** page automatically presents all of the available source networks. This page allows you to manage your source networks, view their specifications, and perform updates.



Each row represents a specific network. It includes various network parameters including:

- Name – the selected source network name
- Replication status – options include **Replicating - protected**, **Stopped**, **In progress**, and **Error**
- Source region – the AWS Region of the source network
- Source AWS account ID – the AWS account ID of the source network
- Pending actions – the next step in the source network replication workflow
- Last recovery result – **Not started**, **Pending**, **Successful**, **Failed**, and **Partial success** (meaning the network was deployed, but the source servers were not configured as part of the recovered network)

- Launched VPC –the recovered network
- CFN stack name – the name of the CloudFormation stack which was used to deploy the launched VPC
- Source network ID – the ID of the source network

Use the top navigation to select an S3 bucket, which is required to enable recovery or to initiate a recovery job.

Use the **Actions** menu to perform various actions including:

- Start replication – Use this option if you want to start replicating your network configuration.
- Stop replication – Use this option if you want to stop replicating your network configuration.
- Export CloudFormation (CFN) template – This option allows you to export the CloudFormation template to your selected S3 bucket. This allows you to verify that the configurations match your preferences and conduct security checks.

> **Note**
>
> If you choose to make changes to the CloudFormation template, it cannot be reuploaded to AWS Elastic Disaster Recovery.

- Manage tags – This option will open the **Manage tags** page which allows you to add or remove tags from your selected network resource.
- Select S3 bucket – This option allows you to save network CFN stacks in your account's Amazon S3 bucket. You must specify the S3 bucket before you initiate network replication. It is recommended that you employ security best practices for Amazon S3.

Adding source networks

Available source networks are presented automatically on the **Source networks** page, along with their details: replication status, pending action, CloudFormation stack name, and more.

When adding a source server to AWS Elastic Disaster Recovery, and after an agent is installed, the VPC network will be automatically identified and created.

To replicate and recover your network configurations, take the following steps:

1. Install the AWS Replication agent on your source servers. Alternatively, source networks can be added manually by calling the CreateSourceNetwork API.
2. Create the required role.
3. Select the relevant network.
4. Start replication.
5. Select an S3 bucket.

> **Important**
>
> You only need to configure your S3 bucket once. Configurations will apply to all existing and newly added source networks.

6. Test or recover your network configurations by initiating a recovery job. This will include creating or updating your CloudFormation stack.

Installing the AWS Replication Agent

In order to use the network replication feature, you must first install the AWS Replication Agent on each source server that you want to add to AWS Elastic Disaster Recovery.

Linux installation instructions

Windows installation instructions

Creating the required role

In order to replicate network configurations between different accounts, you need to go to the source account and create the **Network role** from the **Trusted accounts** page. This will automatically create the role and attached the required policies.

> **Note**
>
> This is only required if your target account is different from the source account.

To create the required role, take the following steps:

1. Go to your source account.

2. Go to the **Trusted accounts** page.
3. Click **Add trusted accounts and create roles**.
4. Click **Add new trusted account**.
5. Enter the target account ID and choose **Network role**.
6. Click **Add trusted accounts and roles**. A success message will appear at the top of the screen.

This action will create the DRSSourceNetworkRole role that is required to utilize the feature.

This role includes the AWSElasticDisasterRecoverySourceNetworkPolicy policy and the following trust policy permissions:

```
{
 "Version" :  "2012-10-17" ,
 "Statement" : [
    {
 "Effect" :  "Allow" ,
 "Principal" : {
 "Service" :  "drs.amazonaws.com"      },
 "Action" :  "sts:AssumeRole" ,
 "Condition" : {
 "StringLike" : {
 "aws:SourceArn" :  "arn:aws:drs:*:*:source-network/*" ,
 "aws:SourceAccount" :  "{{target_account}}"        }
      }
    }
  ]
}
```

After you install the agent and create the relevant role, you can start replicating your network configurations.

Replicating your network configurations

Once you install your agent and created the required role, go to the **Source networks** page and take the following steps:

1. Select the network you want to replicate from the list.
2. Click **Actions** and select **Start replication** from the drop-down menu.

3. Click **Select S3 bucket**. This will allow to save the CloudFormation stack in your account's S3 bucket. You must specify the S3 bucket before you initiate network recovery. It is recommended that you employ S3 bucket security and access management policies.

   You can choose between selecting an existing S3 bucket and creating a new bucket using the S3 bucket console.

   > **Note**
   >
   > You must enable S3 versioning.

4. To test or recover your network configurations, click **Initiate recovery job** and the **Initiate recovery job** prompt will appear.

   If this is the first time you are replicating network configurations, you will need to create a new stack.

   If you already created a stack, you can choose between 3 options:

   a. **Update a recommended stack** – The recommended stack is always the last stack you used.

      > **Note**
      >
      > If the update is not successful, simply create a new stack.

   b. **Create new stack**

   c. **Use a previously created stack** – if you want to choose a stack that you have previously used, select your preferred stack from the drop-down. This will only update the launch templates. The selected stack will then become the recommended stack, allowing you to update it.

Once the recovery job is marked as **Successful**, the network (VPC) is launched in the target Region. All the EC2 launch templates of the source servers in the relevant network will be automatically updated and will feature the new values. This means that when you perform a recovery, those source servers will be launched as part of the new network and the correct subnet.

Trusted accounts

Trusted accounts provide enhanced account management capabilities and visibility, including the ability to easily create multiple IAM roles for different users. Use this feature to quickly add the roles you need to use various AWS Elastic Disaster Recovery features and see the permissions of different accounts from a single screen.

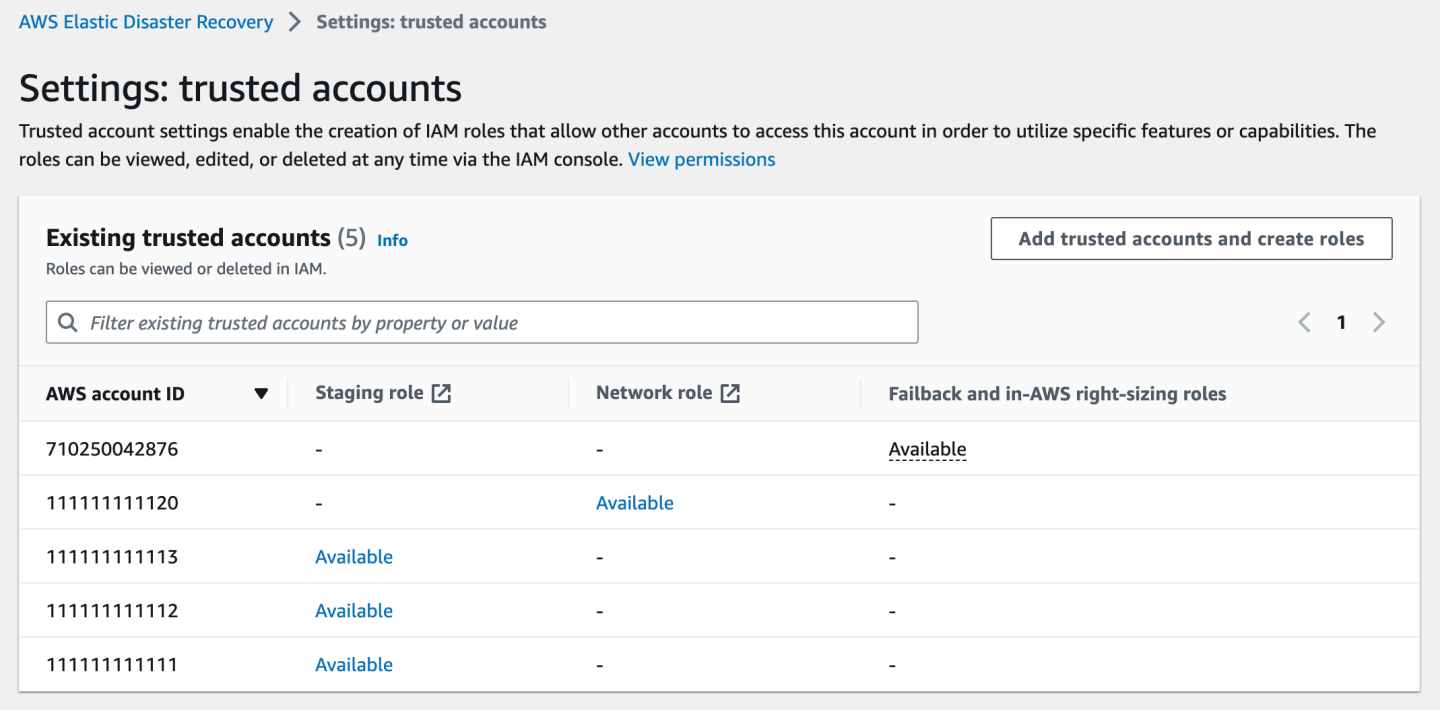
Roles created via CloudFormation (Failback and in-AWS right-sizing roles), should be deleted from the CloudFormation console.

Trusted account page

The **Trusted accounts** page allows you to automatically create IAM roles that are required in order to utilize specific features and capabilities.

This page provides visibility into the existing roles assigned to each trusted account.

To edit or delete these roles, go to the IAM console. Deleting the IAM role will automatically remove the trusted account from the AWS Elastic Disaster Recovery console.



> **Note**
>
> Commercial AWS accounts can only be trusted to other Commercial AWS accounts and GovCloud AWS accounts can only be trusted to other GovCloud AWS accounts.

Adding a trusted account

To add a trusted account, take the following steps:

1. Click **Add trusted accounts and create roles**.
2. Click **Add new trusted account**.
3. Enter an account ID and choose the relevant role or roles. There are 3 available options: Staging role, Network role, and Failback and in-AWS right-sizing roles.
4. Click **Add trusted accounts and roles**. A success message will appear at the top of the screen.

> **Note**
>
> Up to 10 accounts can be added in a single batch and up to 100 accounts for a single AWS DRS account.

Creating the Staging role

The **Staging role** is required to utilize various AWS Elastic Disaster Recovery capabilities, including the multi-account feature. To automatically create the role and the attached required policies, simply create it for a specific account via the **Trusted accounts** page.

This action will create the DRSStagingAccountRole role which includes the AWSElasticDisasterRecoveryStagingAccountPolicy_v2 policy and the following trust policy permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "drs.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": "{{target_account}}",
          "aws:SourceAccount": "{{target_account}}",
          "aws:SourceArn": "arn:aws:drs:*:*:source-server/*"
        }
      }
    }
  ]
}
```

Creating the Network role

The **Network role** is required to utilize various AWS Elastic Disaster Recovery capabilities, including the network replication feature. To automatically create the role and the attached required policies, simply create it for a specific account via the **Trusted accounts** page.

This action will create the DRSSourceNetworkRole role which includes the AWSElasticDisasterRecoverySourceNetworkPolicy policy and the following trust policy permissions:

```
{
 "Version" :  "2012-10-17" ,
 "Statement" : [
    {
      "Effect" :  "Allow" ,
      "Principal" : {
      "Service" :  "drs.amazonaws.com"      },
      "Action" :  "sts:AssumeRole" ,
      "Condition" : {
          "StringLike" : {
             "aws:SourceArn" :  "arn:aws:drs:*:*:source-network/*" ,
             "aws:SourceAccount" :  "{{target_account}}"        }
      }
    }
 ]
}
```

Creating the Failback and in-AWS right-sizing roles

The **Failback and in-AWS right-sizing roles** is required to utilize various AWS Elastic Disaster Recovery capabilities, including cross account failback and in-AWS features. To automatically create the role and the attached required policies, simply create it for a specific account via the **Trusted accounts** page.

This action will create 2 roles:

1. The DRSCrossAccountReplicationRole role which includes the AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy and the following trust policy permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "drs.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringLike": {
          "aws:SourceAccount": "{{target_account_id}}",
          "aws:SourceArn": "arn:aws:drs:*:{{target_account_id}}:recovery-instance/*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "drs.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringLike": {
          "aws:SourceAccount": "{{target_account_id}}",
          "aws:SourceArn": "arn:aws:drs:*:{{target_account_id}}:source-server/*"
```

```
        }
      }
    }
  ]
}
```

2. The DRSCrossAccountAgentAuthorizedRole role which includes the following trust policy permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Resource": "arn:aws:iam::{{target_account_id}}:role/service-role/
DRSCrossAccountAgentRole_{{role_account_id}}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sts:SetSourceIdentity"
      ],
      "Resource": "arn:aws:iam::{{target_account_id}}:role/service-role/
DRSCrossAccountAgentRole_{{role_account_id}}",
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": "i-*"
        }
      }
    }
  ]
}
```

3. The DRSCrossAccountAgentRole role which includes the AWSElasticDisasterRecoveryEc2InstancePolicy policy and the following trust policy permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "{{target_account_id}}"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::{{target_account_id}}:role/service-role/
DRSCrossAccountAgentAuthorizedRole_{{role_account_id}}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "{{target_account_id}}"
      },
      "Action": [
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": "i-*"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::{target_account_id}:role/service-role/
DRSCrossAccountAgentAuthorizedRole_{{role_account_id}}"
        }
      }
    }
  ]
}
```

Configuring launch settings

Launch settings determine how your drill and recovery instances are launched in AWS. They are composed of DRS launch settings and EC2 launch template, allowing you to fully customize your drill and recovery instances by configuring key metrics, such as the subnet within which the instance will be launched, the instance type to be used, licence transfers, replication status, and a variety of other settings. AWS Elastic Disaster Recovery ensures that your drill and recovery instances constantly abide by the latest AWS security, instance, and other updates by utilizing EC2 launch templates. EC2 launch templates always use the latest EC2 instance and technology. EC2 launch templates integrate with AWS Elastic Disaster Recovery in order to give you full control over every single setting within your drill and recovery instance.

Preparing for drill and recovery instance launch

Prior to launching your instances, make sure that your environment is set up properly to ensure successful launches. Check the following prior to continuing:

- Prepare your subnets for launch – Plan which subnets you will use to launch your drill and recovery instances. You will use these subnets in your EC2 launch template when you configure launch settings.
- Create security groups within the subnets – Create the security groups you want to use within your prepared subnets. You will set these security groups in your EC2 launch template when you configure launch settings.

> **Note**
>
> If you want to run a proof of concept, you can skip this step. AWS Elastic Disaster Recovery will automatically use the default subnet and security groups. Ensure that you have not deleted your default subnet.

Important

When launching a drill, recovery, or an in-AWS failback, you can launch up to 100 source servers in a single operation. Additional source servers can be launched in subsequent operations.

Launch settings

The launch settings are a set of instructions that are comprised of two sections: DRS launch settings and the EC2 launch template that determine how a drill or recovery instance will be launched for each source server in AWS.

Launch settings, including the EC2 launch template, are automatically created every time you add a source server to AWS Elastic Disaster Recovery.

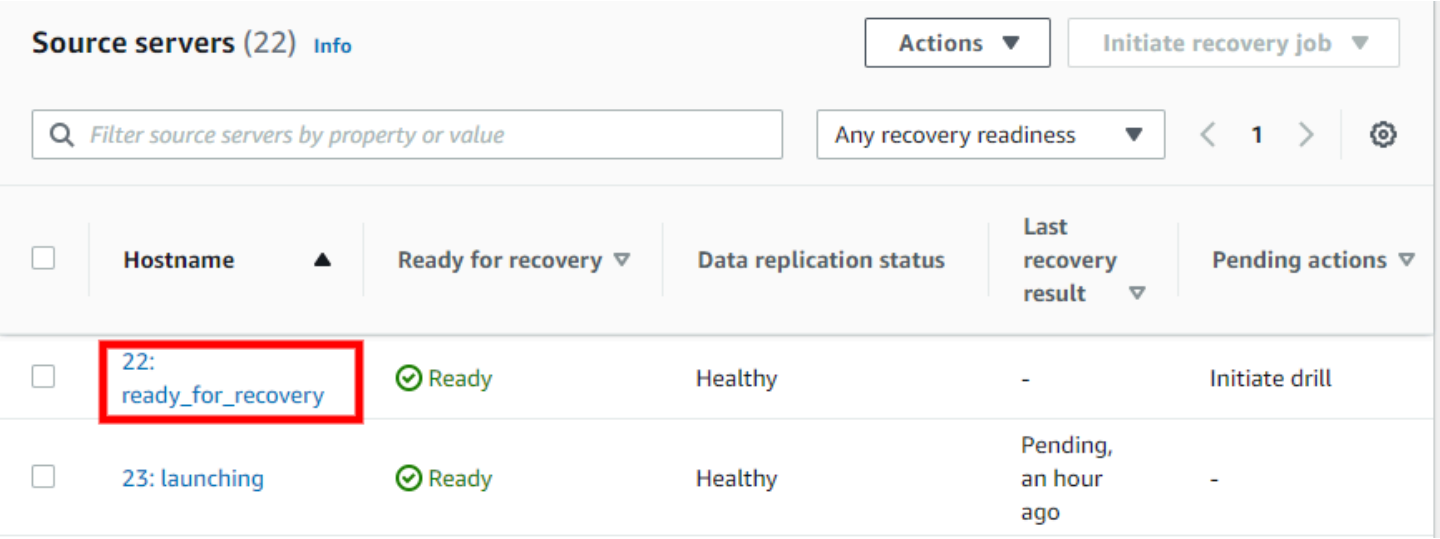
The launch settings can be modified at any time, including before the source server has completed its initial sync.

Note

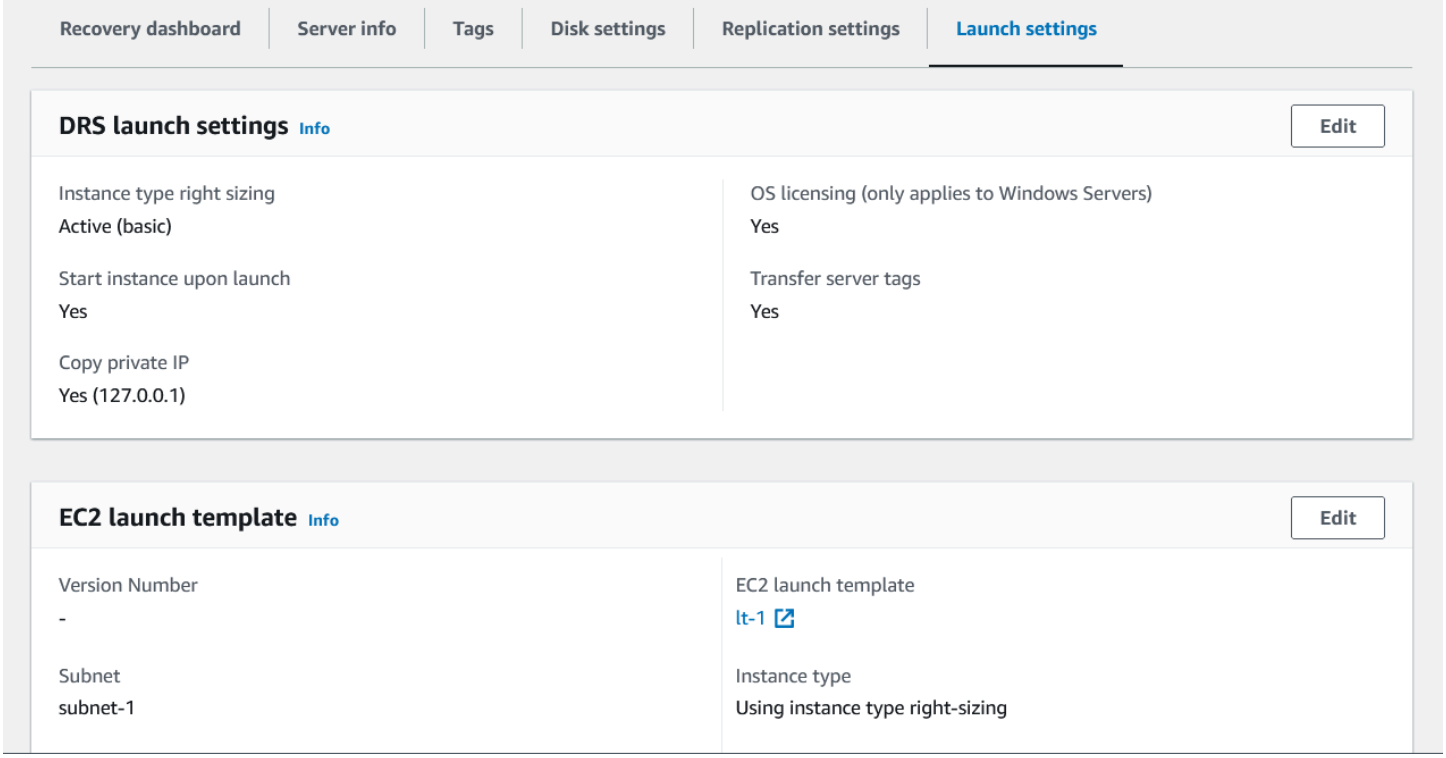
- Any changes made to the launch settings will only affect newly launched drill and recovery instances.
- For many customers, there is no need to modify the DRS launch settings or the EC2 launch template in order to launch drill or recovery instances.

Launch settings can be changed for a single server or for multiple servers via the AWS DRS console. This option allows you to quickly make changes to multiple servers at once. You can also modify launch settings for multiple servers via the AWS Elastic Disaster Recovery API.

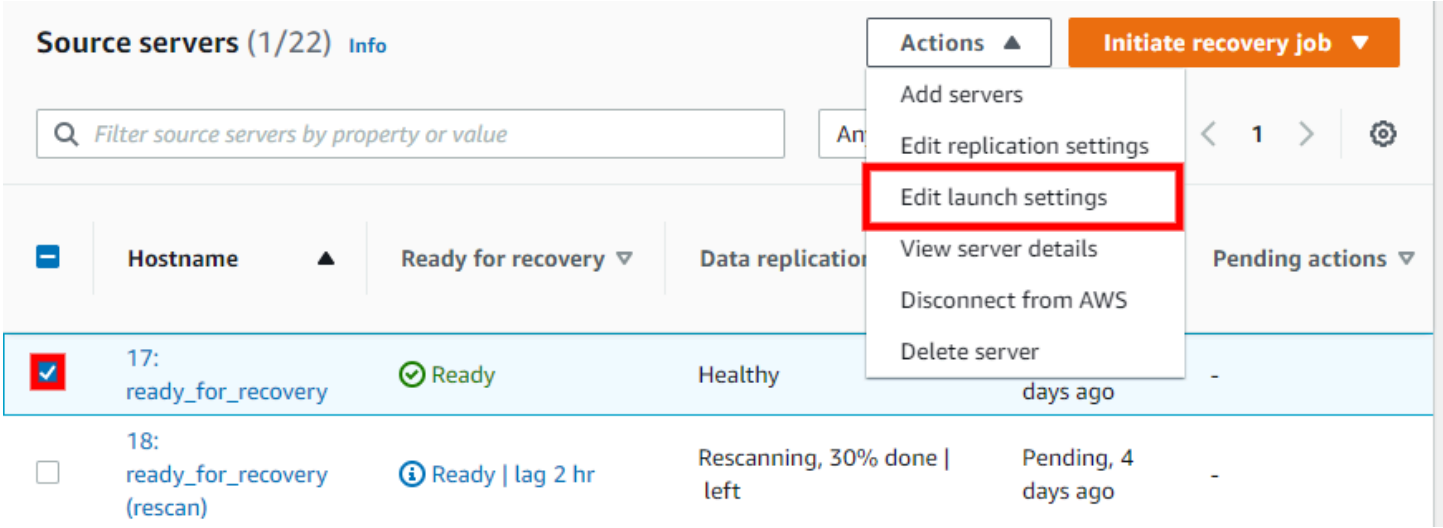
To access the launch settings of a specific source server, go to the **Source servers** page and click the server's hostname.



Within the individual server view, navigate to the **Launch settings** tab.



You can also access the launch settings of a single server by checking the box to the left a single source server on the **Source servers** page and choosing **Actions > Edit DRS launch settings** or **Actions > Edit EC2 launch template**.

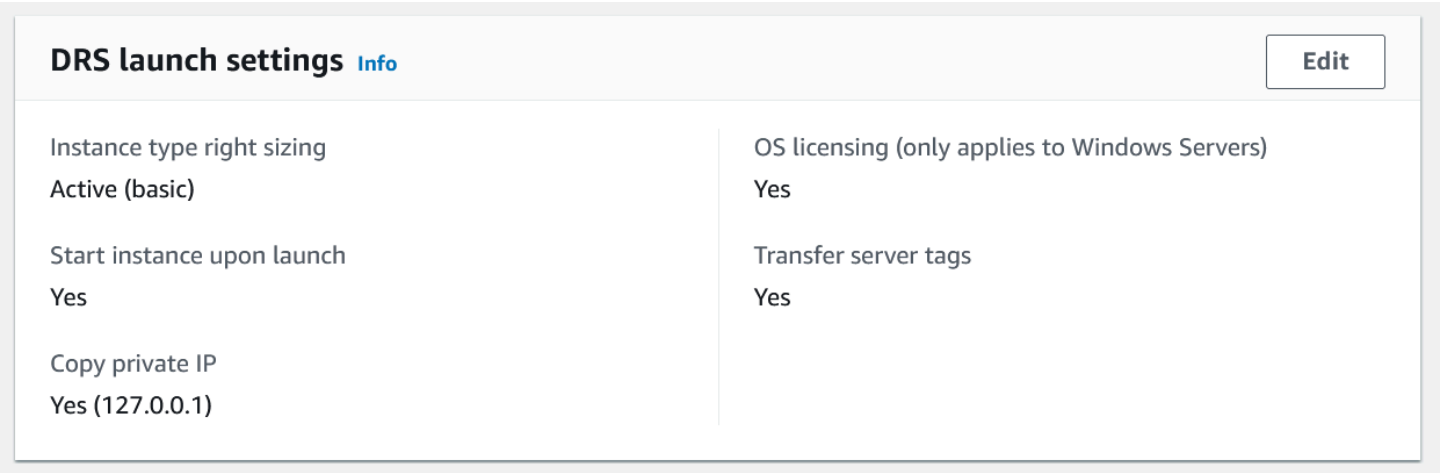


The **Launch settings** tab is divided into two sections:

- DRS launch settings
- EC2 launch template

DRS launch settings

The **DRS launch settings** section allows you to control a variety of server-specific settings.



To edit these settings for a single server, take the following steps:

1. Go to the **Source servers** page.
2. Select a source server to update.

3. Under the **Actions** menu, select **Edit DRS launch settings** and you will be navigated to the **Edit DRS launch template** page within the AWS DRS console.
4. Change the settings according to your preferences.
5. Click **Save settings**.

Alternatively:

- Go to the **Source servers** page.
- Select a specific source server.
- Go to the **Launch settings** tab.
- Click **Edit** in the DRS launch settings section.

DRS launch settings parameters

The DRS launch settings include the following parameters:

- **Instance type right-sizing –** choose whether to allow AWS Elastic Disaster Recovery to launch a drill, recovery, or failback instance type that best matches the hardware configuration of the source server. If you activate this feature, any modification you make to the instance type in the EC2 launch template will be overwritten by the service.
- • If you select the **Active (basic)** option, AWS Elastic Disaster Recovery will launch an AWS instance type that best matches the OS, CPU, and RAM of your source server. AWS Elastic Disaster Recovery will launch a new instance type after every change of configuration on the source server (for example, added/removed disks, added/removed RAM). Instance types are only chosen from the C5 family.
  - If you select the **Active (in-aws)** option, AWS Elastic Disaster Recovery will periodically update the EC2 launch template based on the hardware configuration of the EC2 instance source server.
  - If you select **Inactive**, AWS Elastic Disaster Recovery will launch the AWS instance type as configured in your EC2 launch template. Select this option if you want to determine the instance type that will be launched in AWS for all your drill or recovery servers.

> **Important**
>
> - The AWS instance type selected by AWS Elastic Disaster Recovery when this feature is activated will overwrite the instance type defined in your EC2 launch template.
> - Hardware changes and the resulting AWS instance type change may take up to 90 minutes to be processed by AWS Elastic Disaster Recovery.

The right-sizing instance type selected by AWS Elastic Disaster Recovery will be featured on the **Server details** tab.

- **Start instance upon launching –** Choose whether you want to start your drill and recovery instances automatically upon launch or whether you want to launch them in a stopped state.

  If you choose **No**, you will have to start the drill or recovery AWS instance manually from the EC2 Console.

- **Copy Private IP –** Choose whether you want AWS Elastic Disaster Recovery to ensure that the private IP used by the drill or recovery instance matches the private IP used by the source server. AWS Elastic Disaster Recovery will monitor the source server on an hourly basis to identify the private IP and will use the private IP of the primary network interface.

  The **No** option is chosen by default. Choose **No** if you do not want the private IP of the drill or recovery instance to match that of the source machine.

  Choose **Yes** if you want to use a private IP. The IP will be shown in brackets next to the option.

> **Note**
>
> - If you choose **Yes**, ensure that the IP range of the subnet you set in the EC2 launch template includes the private IP address.
> - If the both the source server and the drill or recovery instance share the same subnet though a VPN, then the source private IP is already in use, and the **Copy private IP** option should not be used.

Copy private IP will be deactivated if you set a value for **Launch into instance ID**, as this setting cannot affect an already launched instance.

- **Transfer server tags –** Choose whether you want AWS Elastic Disaster Recovery to transfer any user-configured custom tags from your source servers onto your drill or recovery instance. These tags are attached to all source servers, all launched drill and recovery instances, and all of the ephemeral resources that are created on your AWS Account during the normal operation of AWS Elastic Disaster Recovery. Transfer server tags only copies tags associated with the source servers in the AWS Elastic Disaster Recovery console, and does not copy the EC2 source server tags (in case of AWS to AWS DR implementation).

  These resources include:

  - EC2 instances
  - Conversion groups
  - Security groups
  - EBS volumes
  - Snapshots

  > **Note**
  >
  > - AWS Elastic Disaster Recovery automatically adds system tags to all resources.
  > - Tags that are added on the EC2 launch template will take precedence over tags that are transferred directly from the source server.

  You can always add tags from the Amazon EC2 console as described in this Amazon EC2 article.

  Transfer server tags will be deactivated if you set a value for **Launch into instance ID**, as this setting cannot affect an already launched instance.

- **Launch into instance ID** - Configure an existing instance ID to launch into, instead of creating a new instance. This field allows to select an EC2 instance from the list of EC2 instances available in this region. The EC2 instance to launch into must have a tag with key *AWSDRS* and value *AllowLaunchingIntoThisInstance* to appear in the list, and it must be stopped prior to launching into it. When this value is set, the **Transfer server tags** and **Copy private IP** settings will be deactivated, as they cannot apply to an already launched instance.

> **Note**
>
> For the instance to appear and perform as a recovery instance in DRS and allow to run post-launch actions on it, it needs to have an instance profile that includes the policies **AWSElasticDisasterRecoveryRecoveryInstancePolicy** and **AmazonSSMManagedInstanceCore**. The role **AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole**, installed from the Default post-launch actions settings page if not already present, contains these policies and can be used as an instance profile.

The launch into an instance will fail if the following pre-requisites are not met:

1. The instance to launch into must have the required tag with key *AWSDRS* and value *AllowLaunchingIntoThisInstance*.
2. The instance to launch into has been stopped.
3. The instance to launch into must have the same operating systems platform (Linux or Windows) as that of the server it is protecting.
4. If the instance to launch into is a Linux it must have the BIOS boot mode, and if Windows it must have the same boot mode as that of the server it is protecting.
5. The instance to launch into must have the x86_64 architecture, HVM virtualization and an EBS root device.
6. **OS licensing** can only be **Bring Your Own License (BYOL)** if the instance's platform is Linux or if the instance's **tenancy** is **dedicated host**.
7. **Transfer server tags** and **Copy private IP** must be deactivated (this is done automatically when **Launch into instance ID** is set via the console).

- **OS licensing –** Choose whether you want to Bring Your Own Licenses (BYOL) from the source server into the drill or recovery instance.

  The **Use default** option will use the default licensing mechanism for your operating system.

  Choose **BYOL** if:

  - You are migrating a Linux server. All Linux licenses are BYOL by default. Any RHEL, SUSE, or Debian licenses will be transferred in their current form to the recovered instance. Make sure that the terms of your licenses allow this license transfer.

- You want to BYOL your Windows licenses. This will set up a dedicated host through which all the licenses from the Windows source server will be automatically transferred to the drill or recovery instance.

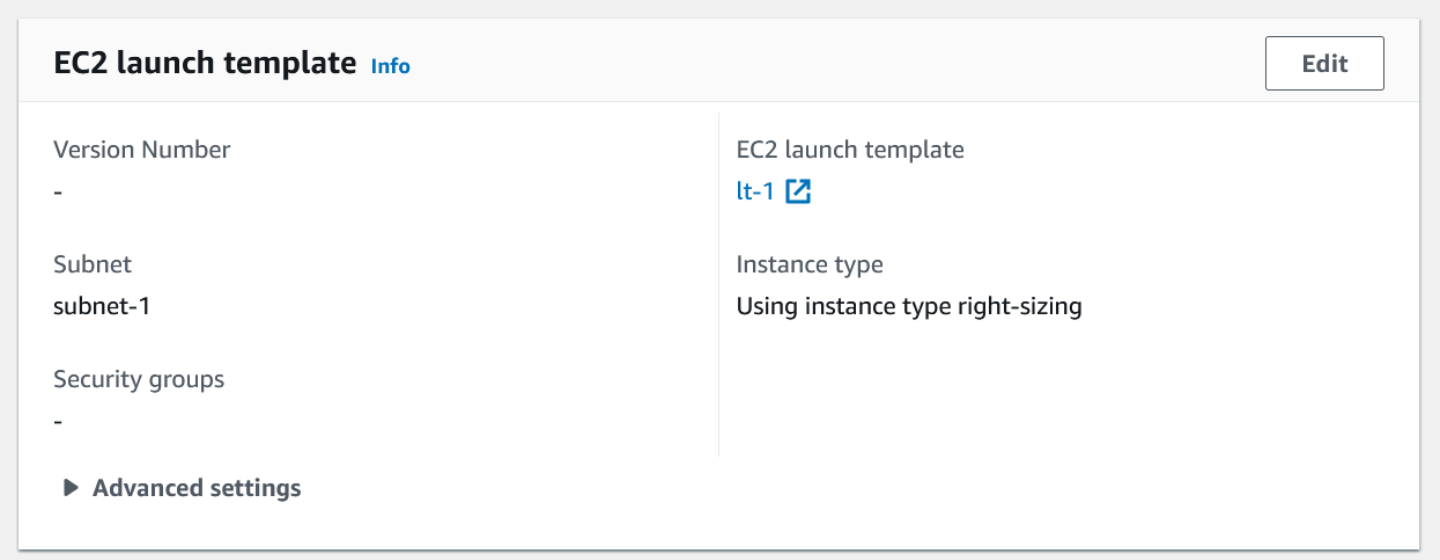
> **Important**
>
> If you activate BYOL licensing for Windows, you have to change the **Placement.tenancy** type in the EC2 launch template to **Host**. Otherwise, instance launch will fail.

> **Note**
>
> - Windows Desktop Editions require BYOL – note the specific restrictions for AWS Provided Licenses.
> - If you are using Windows Servers datacenter: Azure addition, note the specified restrictions for BYOL.

EC2 launch template

AWS Elastic Disaster Recovery (AWS DRS) utilizes EC2 launch templates to launch drill and recovery EC2 instances for each source server. You can edit those templates for each source server directly from the AWS DRS console.

The EC2 launch template is created automatically for each source server that is added to AWS DRS upon the installation of the AWS Replication Agent.



Topics

- EC2 launch template parameters
- EC2 template considerations

Note

- In most use cases, the EC2 launch template does not need to be edited.
- You cannot use the same template for multiple servers.
- Many EC2 launch template parameters can be changed, but some may not be used by the AWS DRS launch process and some may interfere with the AWS Elastic Disaster Recovery launch process.
- You must set the EC2 launch template you want to use with AWS DRS as the **default** launch template.

To edit the EC2 template for a single servers, take the following steps:

1. Go to the **Source servers** page.
2. Select a source servers to update.
3. Under the **Actions** menu, select **Edit EC2 launch settings** and you will be navigated to the **Edit EC2 launch template** page within the AWS DRS console.
4. Change the settings according to your preferences.

5. Click **Save settings**.

Alternatively:

- Go to the **Source servers** page.
- Select a specific source server.
- Go to the **Lunch settings** tab.
- Click **Edit** in the EC2 launch template section.

EC2 launch template parameters

AWS Elastic Disaster Recovery (AWS DRS) EC2 launch settings are divided into basic and advanced settings.

The basic settings include:

- **Subnet –** When you specify a subnet, this field defines where the instance will be launched. When selecting a subnet, only the default network interface will be updated. If you do not include a subnet, the launched instance will use the Region's default subnet located in the default VPC.

  > **Note**
  >
  > - If you have a default VPC, you must modify the EC2 launch template and explicitly define the subnet in which to launch. Failure to do so will result in errors when launching drill or recovery instances.
  > - For cross-AZ recovery, ensure that the staging area subnet and the subnets that you configure your recovery instances to launch in are not in the same AZ as your source EC2 instances. .

- **Security groups –** The selected security groups to assign to the instance, applied to the subnet selected for the default network interface. If no security group is selected, there is no default value and no group will be used. Security groups can only be selected if a subnet is included.
- **Instance type –** The default instance type to use when launching. If instance type right-sizing is active, the system will disregard this setting. If no instance type is included, a default value will be used.

- **Note**

  If you change your instance type and do not activate the instance right-sizing feature, then AWS Elastic Disaster Recovery will use the instance type determined by the **Instance right-sizing** feature and not the instance type you chose in the EC2 launch template. AWS Elastic Disaster Recovery verifies the instance type once per hour, as a result, if you did not activate the instance right-sizing feature, the first time instance launch may still utilize the instance type you set in the EC2 launch template, but any subsequent launches will utilize the right-sizing instance.

Advanced settings include additional parameters that add specific features to the EC2 template. If you choose not to include these parameters in the template, the specific capabilities will not be added.

The advanced settings include:

- **IAM instance profile –** Attach a specific profile to the instance that will be launched. Make sure the instance profile has the AWSElasticDisasterRecoveryRecoveryInstancePolicy IAM policy attached in addition to any other policy.
- **Auto assign public IP –** Automatically assign a public IP to the launched instance.
- **Termination protection –** Protect the launched instance from accidental termination using the EC2 console.
- **Tenancy –** Set tenancy information, such as dedicated host needed in conjunction with setting BYOL for Windows servers and Windows Home.
- **Capacity reservation –** Apply reservation consideration to the launched instances.
- **Key pair –** Associate a key pair with launched instances that are based on EC2 instances.

Note

AWS DRS only supports major EC2 template parameters. If you want to change values that are not supported by this feature, you can still do so by editing the EC2 launch template via the Amazon EC2 console:

- Create a new EC2 template version with the required changes.

- Mark it as default.

Important

Every time you modify an EC2 launch template on the Amazon EC2 console, a new version is created. AWS DRS uses the version that is marked as the default. if you prefer to use the EC2 launch template you just modified, make sure to mark it as the default. Changes made through the AWS DRS console are automatically set as the default version.

EC2 launch template tags – In addition to the basic and advanced settings, you can also add up to 50 tags. These will be transferred to your drill and recovery instances. Note that these tags may interfere with other tags that have already been added to the source server. Launch template tags always take precedence over tags set in the AWS DRS Console or tags manually added to the server.

Learn more about EC2 launch template settings and configuration options in this EC2 article.

EC2 template considerations

Revert to previous version – The right-sizing mechanism can fix issues such as an incorrect instance type, but other issues may still occur. If you encounter any issues with the launch template, you can quickly address them by choosing the original default launch template that was created by AWS DRS when the agent was installed. Alternatively, you can edit the relevant fields from the AWS DRS console.

Important

If you decide to create the EC2 template from the Amazon EC2 console, be sure not to change or edit the following fields:

- RAM disk ID
- Kernel
- Nitro Enclave
- Metadata accessible

These fields must remain unchanged for AWS Elastic Disaster Recovery to function properly.

Using Elastic Disaster Recovery for failover and failback

Topics

Failover and failback overview

In the event of a disaster, you will need to perform a failover to AWS with the help of AWS Elastic Disaster Recovery (AWS DRS). Once the disaster has been mitigated, you will then need to perform a failback to your original source infrastructure.

AWS Elastic Disaster Recovery ensures that your recovery systems are ready in the case of a disaster. The actual failover is a networking operation that is performed outside of AWS Elastic Disaster Recovery. You launch your recovery instances with AWS Elastic Disaster Recovery, up to the latest second, or to a certain PIT. Once you are ready to resume operations on your primary system, you will need to perform failback replication. Most likely while you are using your recovery system on AWS, new data has been written and this data needs to be copied back to your primary system.

AWS Elastic Disaster Recovery (AWS DRS) helps you be ready for a failover event by making the running of drills easy. AWS DRS allows you to perform frequent launching of you instances for test and drill purposes without redirecting the traffic.

In order to be prepared for a failover, you need to perform continuous drills by launching drill instances in AWS through Elastic Disaster Recovery and testing these instances.

Performing drills is a key aspect of being prepared for a disaster. Once the actual disaster strikes, you can immediately perform a failover by launch Recovery instances in AWS based on a chosen Point In Time snapshot.

Once the disaster is over, you can perform a failback to your original source server or to any other server that meets the prerequisites by installing the AWS Elastic Disaster Recovery Failback Client on the server. In order to use the Failback Client, you need to generate AWS Elastic Disaster Recovery-specific credentials.

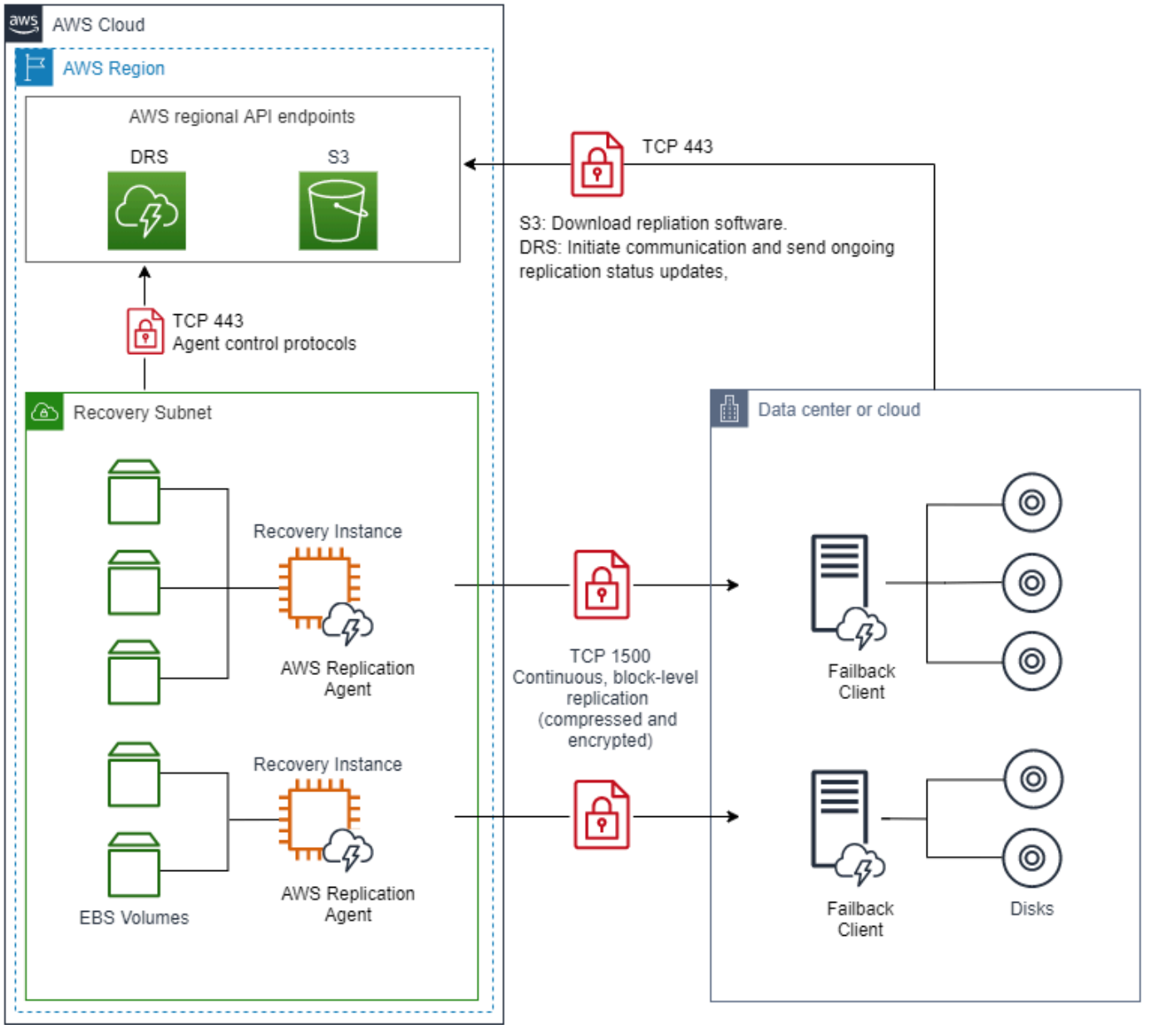
You can perform a cross-Region or cross-AZ failover and failback directly with the aid of the AWS DRS Console.

In addition, AWS DRS allows you to perform a scalable failback for vCenter with the DRS Mass Failback Automation client (DRSFA client).

Once your failback is complete, you can opt to either terminate, delete, or disconnect the recovery instance.

The following is the architectural diagram for DRS failback replication:

AWS Elastic Disaster Recovery (AWS DRS) failback replication – architecture and networking



Understanding drill and recovery instances

AWS Elastic Disaster Recovery allows you to launch drill and recovery instances for your source servers in AWS. Drill and recovery instances are launched in a similar fashion. You can launch a drill or recovery instance from the most up-to-date state, typically achieving an RPO of seconds, or from one of the point-in-time states that the system maintains.

Understanding Point In Time states

Point in Time (PIT) is a disaster recovery feature which allows launching an instance from a snapshot captured at a specific Point In Time. As source servers are replicated, Point in Time states are chronicled over time, while a retention policy will determine which Points in Time are not required after a defined duration.

Elastic Disaster Recovery has the following PIT state schedule:

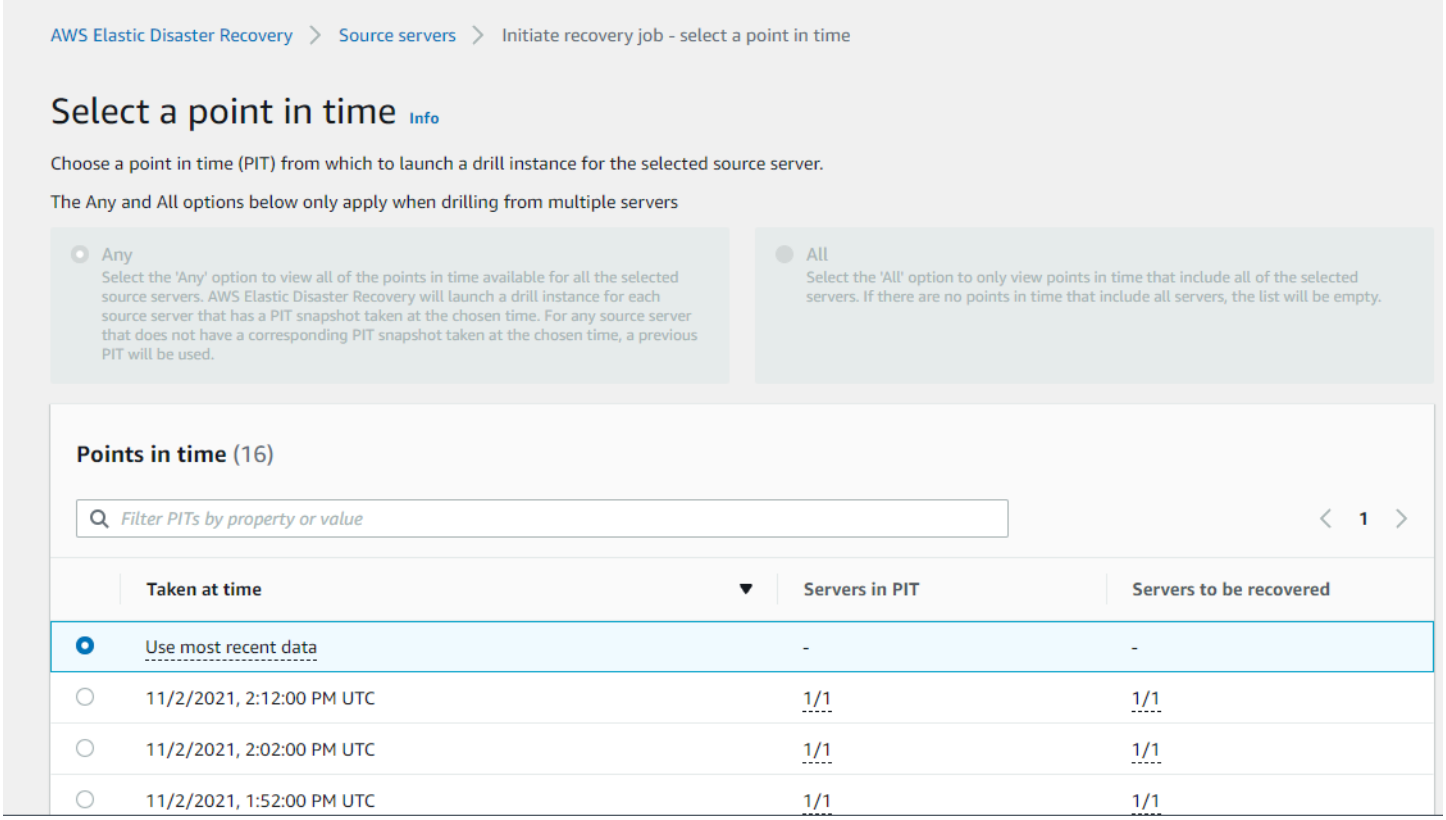
- Every 10 minutes for the last hour
- Once an hour for the last 24 hours
- Once a day for the last 7 days (or a different retention period, as configured)

You can increase or decrease the default 7 day snapshot retention rate from anywhere between 1 day and 365 days in the replication settings. Learn more about managing Point in Time retention.

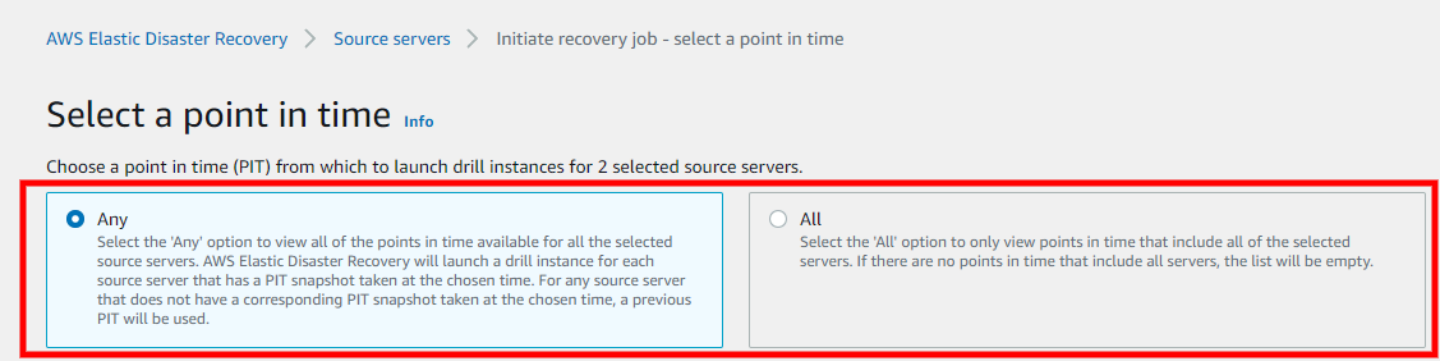
Note

Increasing the PIT retention rate will result in additional costs.

Upon launching drill instances and recovery instances, you will be prompted to select the Point in Time from which to launch the instances for the selected source servers.



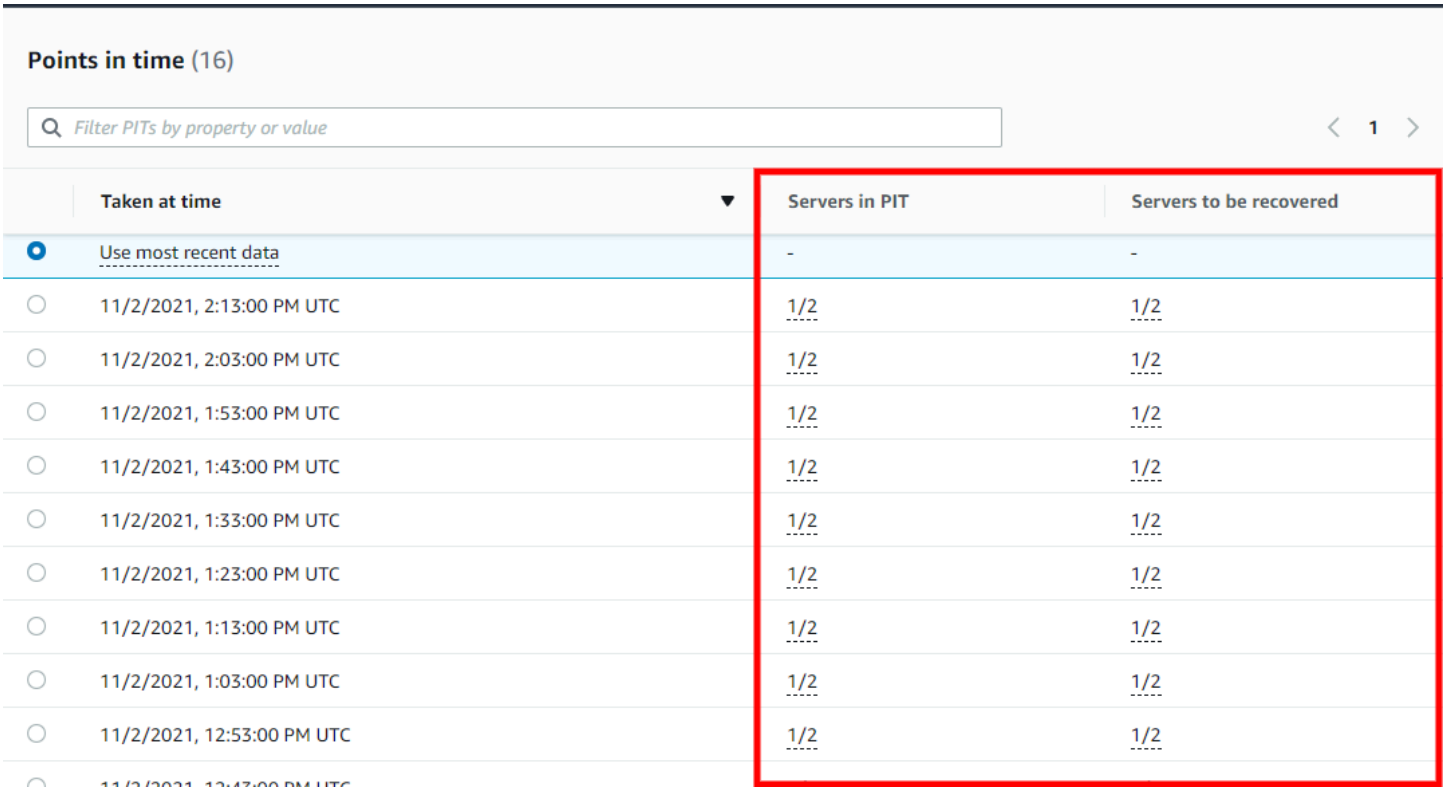
When launching two or more source servers simultaneously, you can select which specific Points in Time to view.



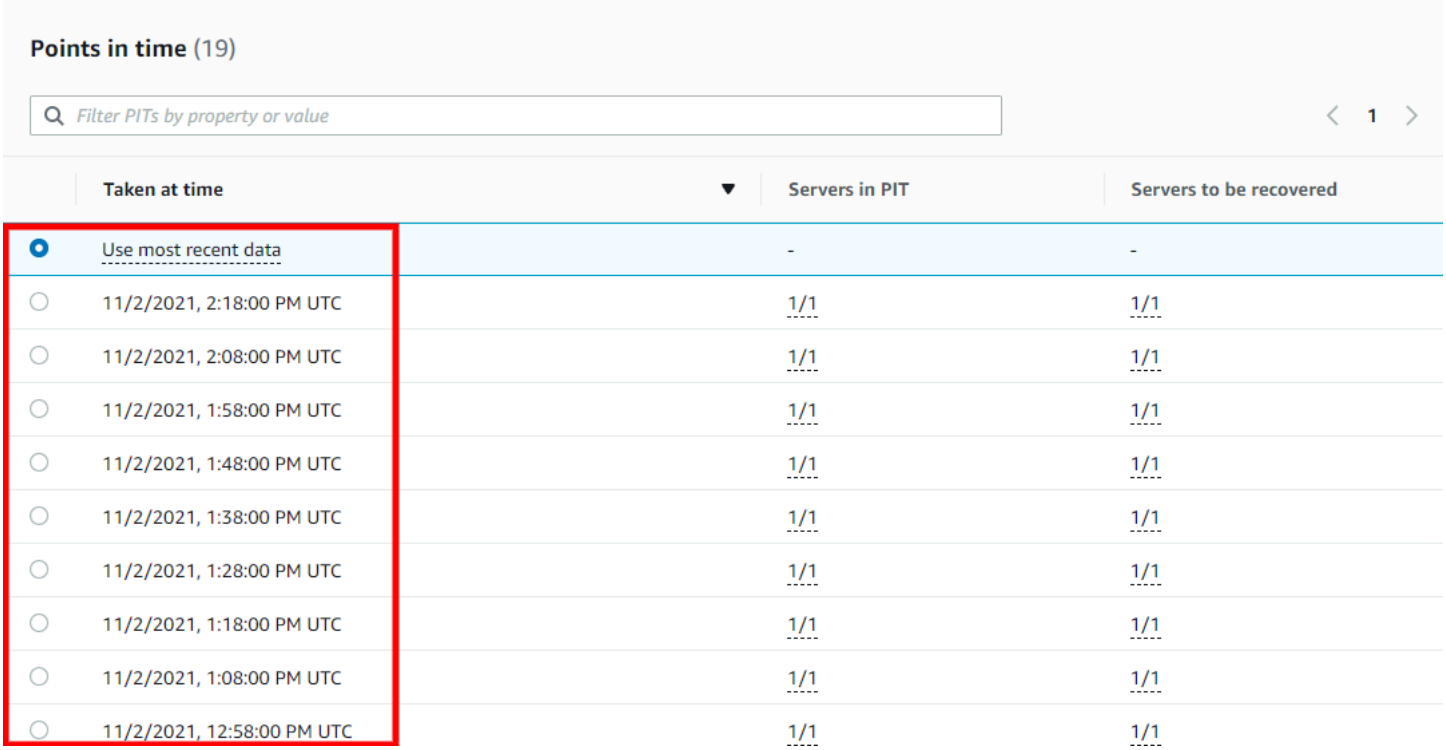
Choose the **Any** option to view all of the points in time available for all of the selected source servers. AWS Elastic Disaster Recovery will launch a drill instance for each source server that has a PIT snapshot taken at the chosen time. For any source server that does not have a corresponding PIT snapshot taken at the chosen time, a previous PIT will be used.

Choose the **All** option to only view points in time that include all of the selected servers. If there are no points in time that include all servers, the list will be empty.

The **Servers in PIT** and **Servers to be recovered**columns show the number of servers within the chosen PIT.



Under the **Taken at time (UTC)** column, you can either select the **Use most recent data option**, which will immediately create a new Point in Time and use that state, or you can select a previously taken snapshot from the list of available PIT states.



Note

If you selected the **Use most recent data option** but AWS Elastic Disaster Recovery is unable to take a new PIT snapshot of the source server (due to a disaster, connectivity issues, and more), then AWS Elastic Disaster Recovery will automatically use the last PIT taken.

Understanding Recovery Objectives

AWS Elastic Disaster Recovery (AWS DRS) provides continuous block-level replication, recovery orchestration, and automated server conversion capabilities. These allow customers to achieve a crash-consistent recovery point objective (RPO) of seconds, and a recovery time objective (RTO) typically ranging between 5–20 minutes. Below is an explanation of how RPO and RTO are measured, how AWS DRS supports these RPOs and RTOs, and what common environment conditions can impact RPO and RTO.

Recovery Point Objective (RPO)

How is RPO measured?

RPO is measured based on the latest point in time in which block data was written to the source server volume(s) and successfully copied in a crash-consistent state into the replication staging area located in the customer's target AWS account.

How does AWS DRS allow an RPO of seconds?

The AWS Replication Agent continuously monitors the blocks written to the source server volume(s), and immediately attempts to copy the blocks across the network and into the replication staging area subnet located in the customer's target AWS account. This continuous replication approach allows an RPO of seconds as long as the written data can be immediately copied across the network and into the replication Staging Area volumes.

> **Important**
>
> A crash-consistent recovery point allows the successful recovery of crash-consistent applications, such as databases. The recovery point will include any data that has been successfully written to the source server volume(s). Application data that is kept in memory is not replicated to the target replication Staging Area until it is written to the source server volume(s). Therefore, if a disruption occurs before in-memory application data is written to the volume(s), this data will not be available on the target server when launched for test or recovery purposes.

What environment conditions can impact the ability to achieve a typical RPO of seconds?

To achieve an RPO of seconds, AWS Elastic Disaster Recovery primarily requires that the outbound network, inbound network, and staging area resources must allow data to be copied across the network and written to the target environment faster than the rate at which it is written to the source volume(s). In the case that block writes burst at faster rates than these components can support, the RPO will temporarily increase until the data replication can catch up, at which point the RPO will return to seconds. Examples:

1. Outbound network: If a source server writes block data at a rate of 10 MB/second, the outbound network bandwidth must also support a rate of at least 10 MB/second in order to maintain a seconds RPO. If the source network contains 10 servers that each write at an average rate of 10

MB/second, the total bandwidth will need to support a rate of at least 100 MB/second in order to allow a seconds RPO.

2. Inbound network: Once the replicated data is sent from the source network, it must enter the target network at a rate greater to that at which the data is written to the source servers and sent from the source network in order to maintain a seconds RPO.
3. Staging area resources: When the data arrives to the target network, it is received by the AWS DRS replication server instance(s), which in turn writes the replicated data to attached EBS volumes. Both the replication server instance(s) and attached Amazon EBS volumes must allow the data to be written at a rate faster than that at which it is written to the source servers and sent by the source network in order to maintain an RPO of seconds.

What happens if the block data written to the source volume(s) cannot be sent immediately to the target replication Staging Area Subnet?

If the block data written on the source volume(s) cannot be sent immediately to the target replication Staging Area, the RPO will increase until the data can be flushed across the network. During this time, you will still be able to recover your server(s), but to a recovery point older than seconds, in accordance with the increase in RPO. The RPO represents the latest crash-consistent point in time during which data was replicated.

Recovery Time Objective (RTO)

How is RTO measured?

RTO is measured from the recovery job start time until the recovered target server is booted and has network access on AWS.

What environment conditions can impact the ability to achieve a typical RTO of 5–20 minutes?

A: When launching a recovery job, the AWS DRS orchestration process creates cloned volumes by using the replicated volumes in the replication staging area. During this process, AWS DRS also initiates a process that converts all volumes that originated outside of AWS into AWS-compatible volumes, which are attached to EC2 instances that can boot natively on AWS. The job and boot time depend on the following environment conditions:

1. OS type: The average recovered Linux server normally boots within 5 minutes, while the average recovered Windows server normally boots within 20 minutes because it is tied to the more resource-intensive Windows boot process.
2. OS configuration: The OS configuration and application components it runs can impact the boot time. For example, some servers run heavier workloads and start additional services when booted, which may increase their total boot time.
3. Target instance performance: AWS DRS sets a default instance type based on the CPU and RAM provisioned on the source server. Changing to a lower performance instance type will result in a slower boot time than that of a higher performance instance type.
4. Target volume performance: Using a lower performance volume type will result in a slower boot time than that of a higher performance volume type with more provisioned IOPS.

Preparing for failover

In order to be able to launch your recovery instances quickly, you should preconfigure how those instances are to be launched and perform drills in order to make sure that all of your network and application settings are properly configured. You can configure how your instances will be launched by editing the Launch settings for each source server. Launch settings can be configured immediately when a source server has been added to AWS DRS, there is no need to wait for the initial sync process to finalize. Performing frequent drills is key for failover preparedness. Elastic Disaster Recovery makes it easy for you to launch drill instances as frequently as you want. Drills are nondisruptive – they do not impact the source server or ongoing data replication. If you experience a disaster in the middle of a drill, you can launch a new recovery instance from the source server's current state.

Configuring your launch settings

Before you can launch drill and recovery instances, you must configure your launch settings. Learn more about configuring launch settings.

Performing drills

After you have added all of your source servers and configured their launch settings, you are ready to launch a Recovery drill. It is crucial to drill the recovery of your source servers to AWS prior to initiating a Recovery in order to verify that your source servers function properly within the AWS environment.

Important

It is crucial to perform a meaningful drill. For example, if you have a multi-server application, it's not enough to see that those servers launched as EC2 instances, it's crucial to see that they communicate, boot, and so on, so that in the case of a disaster they can achieve their desired RTO. Depending on what your system is, a meaningful test could be either ensuring that your server has booted, ranging to running your applications. At a minimum, ensure that your servers have their networking correctly configured. It is a best practice to perform drills regularly. After launching recovery drill instances, use either SSH (Linux) or RDP (Windows) to connect to your instance and ensure that everything is working correctly.

You can drill one source server at a time, or simultaneously drill multiple source servers. For each source server, you will be informed of the success or failure of the launch of your drill instances, including whether it has achieved the first boot. This is often not enough to determine whether you are prepared for a disaster. You can drill your source server as many times as you want. Each new drill first deletes any previously launched Recovery instance and dependent resources. After the drill, data replication continues as before. The new and modified data on the source server is transferred to the staging area subnet and not to the recovery instances that were launched during the test.

Note

Windows source servers need to have at least 2 GB of free disk space to successfully launch a recovery instance.

Note

Take into consideration that once a recovery instance is launched, other resources will be used in your AWS account and you will be billed for these resources. You should terminate launched recovery instances as soon as your drill is over in order to avoid unnecessary charges (primarily for Amazon EC2 and Amazon EBS).

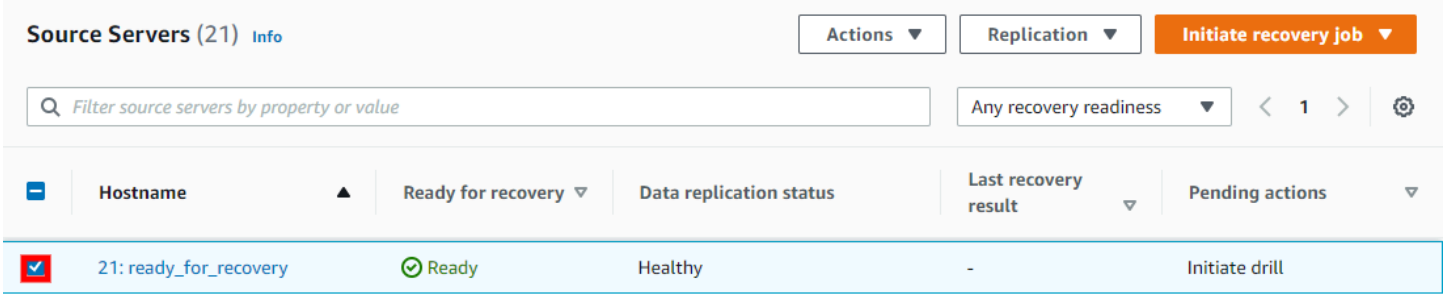
Indicators that your source server is ready for a drill

Prior to launching a drill instance, ensure that your source servers are ready for testing by looking for the following indicators on the **Source servers** page:

1. Under the **Ready for recovery** column, the server should show **Ready**. This means that initial sync has been completed and all data from the source server has been replicated to AWS.
2. Under the **Data replication status** column, the server should show the **Healthy** status, but you can also launch the source server if the system is undergoing **Lag** or even **Stall**, but in that case the data may not be up to date. You can still launch a drill instance from a previous Point In Time.
3. Under the **Pending actions** column, the server should show **Initiate drill** if no drill instances have ever been launched for the server. Otherwise, the column will be blank. This helps you identify whether the server has had a recent drill launch.

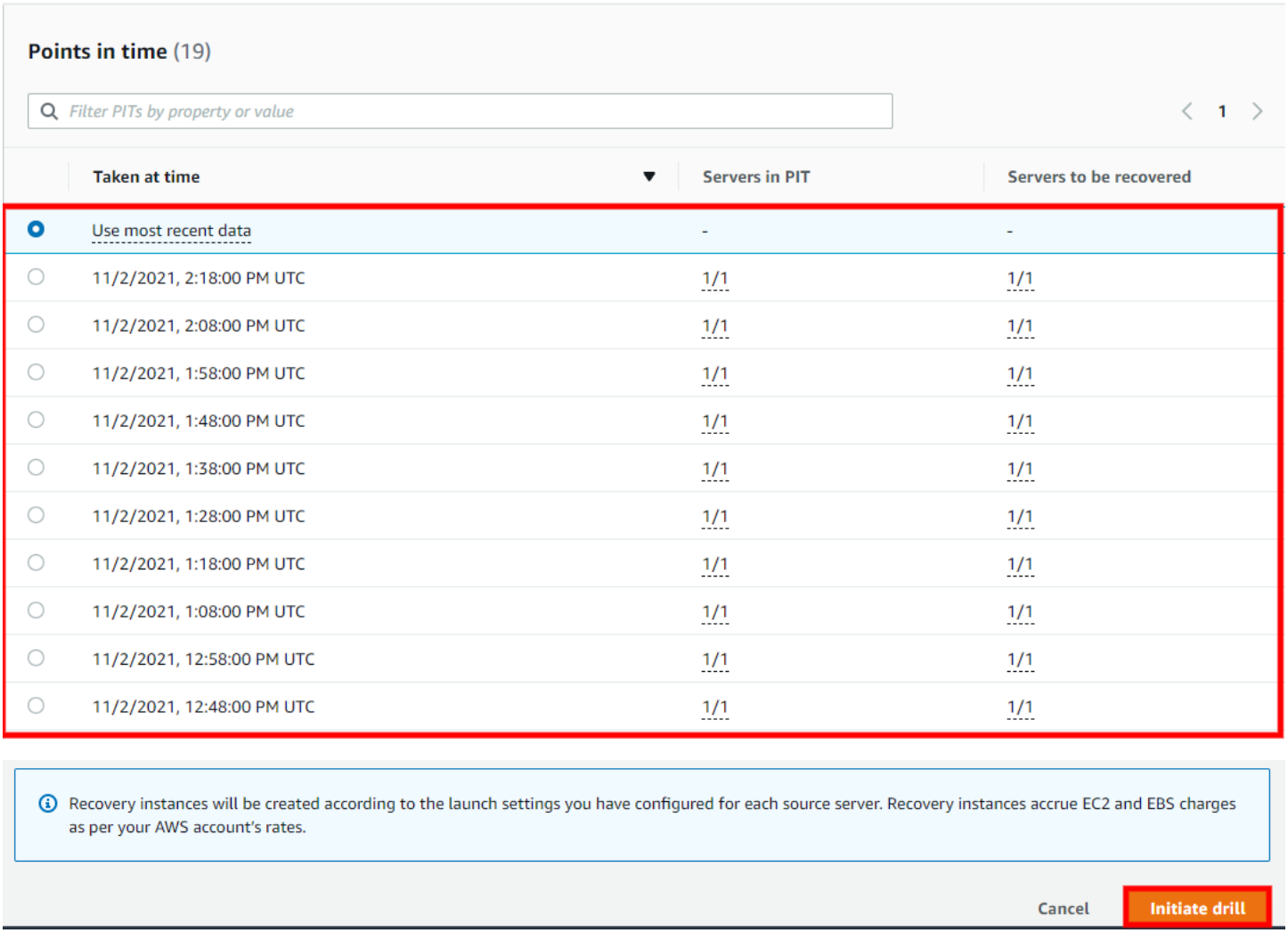
Launching drill instances

To launch a drill instance for a single source server or multiple source servers, go to the **Source servers** page and check the box to the left of each server for which you want to launch a drill instance.



Open the **Initiate recovery job** menu and select **Initiate drill**.

Select the Point in time snapshot from which to launch the drill instance for the selected source server. You can either select the **Use most recent data** option to use the latest snapshot available or select an earlier specific Point-in-time snapshot. You may opt to select an earlier snapshot in case you wish to return to a specific server configuration before a disaster occurred. After you have selected the Point in Time snapshot, click **Initiate drill**.



[Learn more about Point in Time snapshots](#).

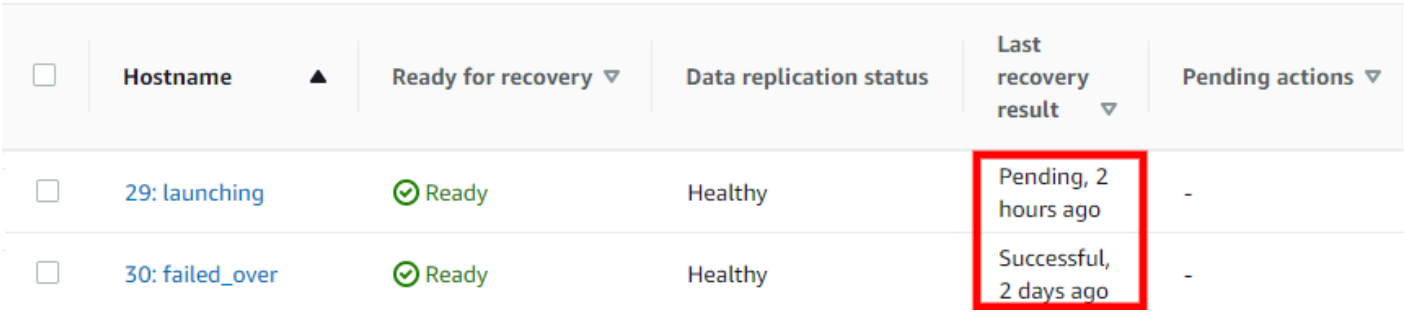
The AWS Elastic Disaster Recovery Console will indicate **Recovery job is creating drill instance for X source servers** when the drill has started.

Choose **View job details** on the dialog to view the specific Job for the test launch in the **Recovery job history** tab.

Successful drill instance launch indicators

You can tell that the Drill instance launch started successfully through several indicators on the **Source servers** page.

1. The **Last recovery result** column will show the status of the recovery launch and the time of the launch. A successful drill instance launch will show the **Successful** status. A launch that is still in progress will show the **Pending** status.



2. The launched Drill instance will also appear on the **Recovery instances** page.

Performing a failover

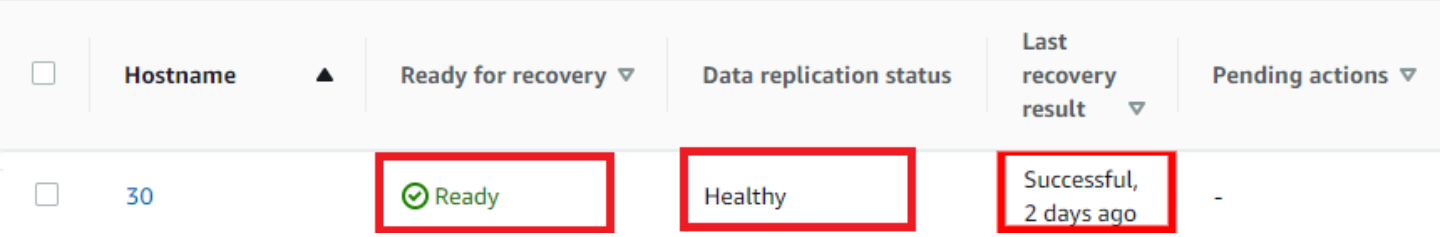
A failover is the redirection of traffic from a primary system to a secondary system. It's a network operation that's performed outside of AWS Elastic Disaster Recovery. AWS Elastic Disaster Recovery helps you perform a failover by launching recovery instances in AWS. Once the Recovery instances are launched, you will need to redirect the traffic from your primary systems to the launched recovery instances.

> **Note**
>
> These instructions also apply to the cross-Region or cross-AZ failover process.

Launching recovery instances

Ready for launch indicators

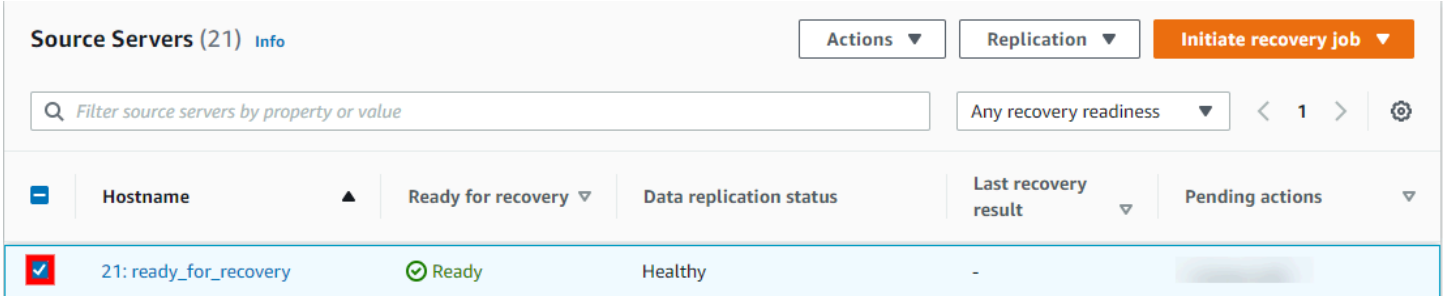
Prior to launching a Recovery instance, ensure that your source servers are ready for testing by looking for the following indicators on the **Source Servers** page:



1. Under the **Ready for recovery** column, the server should show **Ready**
2. Under the **Data replication status** column, the server should show the **Healthy** status.
3. Under the **Last recovery result** column, there should be an indication of a successful Drill instance launch sometime in the past. The column should state **Successful** and show when the last successful launch occurred. This column may be empty if a significant amount of time passed since your last drill instance launch.

Launching recovery instances

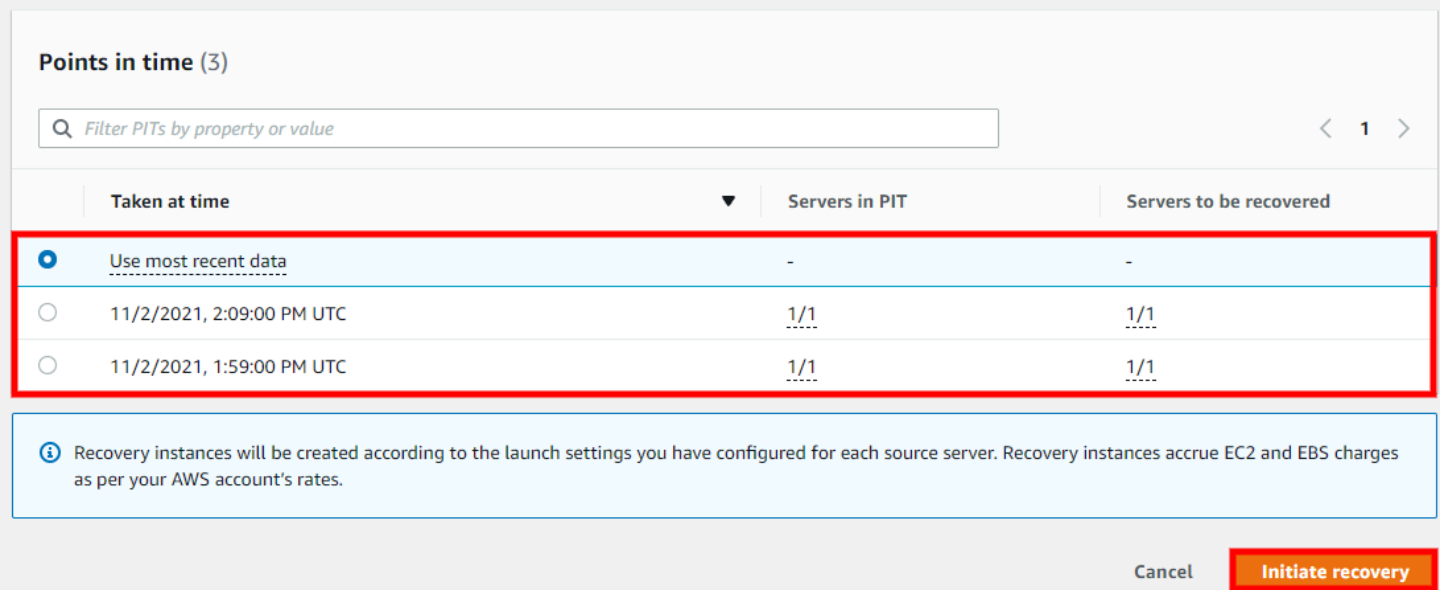
To launch a recovery instance for a single source server or multiple source servers, go to the **Source servers** page and check the box to the left of each server for which you want to launch a recovery instance.



Open the **Initiate recovery job** menu and select **Initiate recovery**.

Select the Point in time snapshot from which to launch the recovery instance for the selected source server. You can either select the **Use most recent data** option to use the latest snapshot available or select an earlier specific Point-in-time snapshot. You may opt to select an earlier

snapshot in case you wish to return to a specific server configuration before a disaster occurred. After you have selected the Point in Time snapshot, choose **Initiate recovery**.



Learn more about Point in Time snapshots.

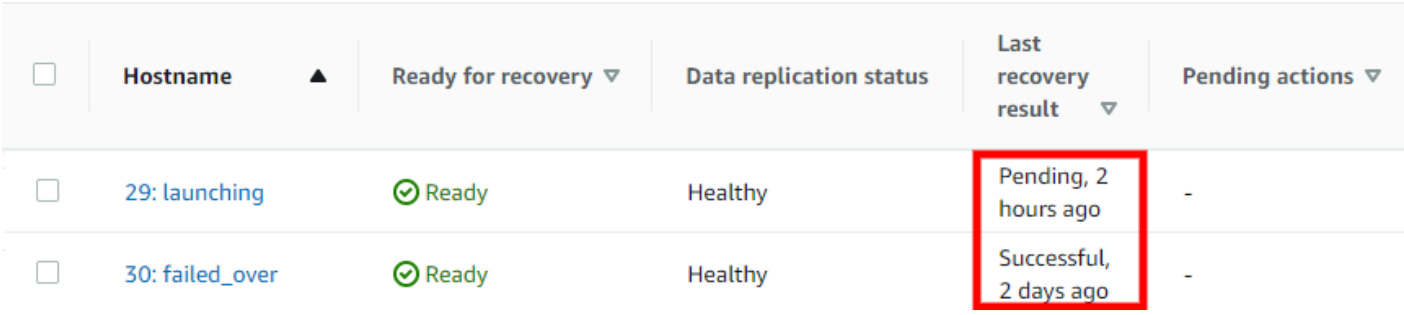
The AWS Elastic Disaster Recovery Console will indicate **Recovery job is creating drill instance for X source servers** when the drill has started.

Click **View job details** on the dialog to view the specific Job for the test launch in the **Recovery job history** tab.

Successful recovery instance launch indicators

You can tell that the recovery instance launch started successfully through several indicators on the **Source servers** page.

1. The **Last recovery result** column will show the status of the recovery launch and the time of the launch. A successful recovery instance launch will show the **Successful** status. A launch that is still in progress will show the **Pending** status.



2. The launched recovery instance will appear on the **Recovery instances** page. Learn more about the Recovery instances page.
3. You can now redirect traffic from your primary systems to the launched recovery instances.

Performing a failback

Failback is the act of redirecting traffic from your recovery system to your primary system. This is an operation that is performed outside of AWS Elastic Disaster Recovery. AWS Elastic Disaster Recovery assists you in performing the failback by ensuring that the state of your primary system is up to date with the state of your recovery system.

Failback is only supported to non-AWS environments that can boot up from an ISO. For non-AWS environments which do not support ISO boot, it is suggested to convert the ISO to a suitable format. Examples - Building DR on AWS with Microsoft Azure and Building DR on AWS with Google Cloud. These blog posts are not maintained or supported by AWS Premium Support and guidance for these are provided on a best effort basis.

Before performing a failback, you want to make sure that any data that was written to your failover systems during the failover is replicated back to your original systems before you perform the actual failback and redirecting users to your primary systems. AWS Elastic Disaster Recovery helps you prepare for failback by replicating the data from your Recovery instances on AWS back to your source servers with the aid of the Failback Client.

Failback to on-premises environment

Topics

Using the Failback Client

Failback replication is performed by booting the Failback Client on the source server into which you want to replicate your data from AWS. In order to use the Failback Client you must meet the failback prerequisites and generate failback AWS credentials as described below. The AWS DRS Console allows you to track the progress of your failback replication on the **Recovery instances** page. Learn more about the Recovery instances page.

Failback prerequisites

Prior to performing a failback, ensure that you meet all replication network requirements and the following failback-specific requirements:

- Ensure that the volumes on the server you are failing back to are the same size, or larger, than the Recovery instance.
- The Failback Client must be able to communicate with the Recovery instance on TCP 1500, this can be done either by via a private route (VPN/DX) or a public route (public IP assigned to the recovery instance)
- TCP Port 1500 inbound and TCP Port 443 outbound must be open on the recovery instance for the pairing to succeed.
- You must allow traffic to S3 from the server you are failing back to.
- The server on which the Failback Client is ran must have at least 4 GB of dedicated RAM.
- The recovery instance used as a source for failback must have permissions to access the DRS service via API calls. This is done using instance profile for the underlying EC2 instance. The instance profile must include the AWSElasticDisasterRecoveryRecoveryInstancePolicy in addition to any other policy you require the EC2 instance to have. By default, the launch settings that DRS creates for source servers already have an instance profile defined that includes that policy and that instance profile will be used when launching a Recovery Instance.
- Be sure to deactivate secure boot on the server on which the Failback Client is run.
- Ensure the hardware clock the on the server on which the Failback Client is run is set to UTC rather than Local Time.

Failback AWS credentials

In order to perform a failback with the Elastic Disaster Recovery Failback Client, you must first generate the required AWS credentials. You can create temporary credentials with AWS STS. These credentials are only used during Failback Client installation.

You will need to enter your credentials into the Failback Client when prompted.

Generating temporary failback credentials

In order to generate the temporary credentials required to install the AWS Elastic Disaster Recovery Failback Client, take the following steps:

1. Create a new IAM Role with the **AWSElasticDisasterRecoveryFailbackInstallationPolicy** policy.
2. Request temporary security credentials via AWS STS using the AssumeRole API.

Learn more about creating a role to delegate permissions to an AWS service in the IAM documentation. Attach the following policy to the role: **AWSElasticDisasterRecoveryFailbackInstallationPolicy**.

Failback Client detailed walkthrough

Once you are ready to perform a failback to your original source servers or to different servers, take the following steps:

> **Note**
>
> Replication from the source instance to the source server (in the target AWS Region) will continue when you perform failback on a test machine.

1. Complete the recovery as described above.
2. Configure your failback replication settings on the recovery instances you want to fail back. Learn more about failback replication settings.
3. Download the AWS Elastic Disaster Recovery Failback Client ISO (aws-failback-livecd-64bit.iso) from the S3 bucket that corresponds to the AWS Region in which your recovery instances are located.
   a. Direct download link: Failback Client ISO: `https://aws-elastic-disaster-recovery-{REGION}.s3.{REGION}.amazonaws.com/latest/failback_livecd/aws-failback-livecd-64bit.iso`
   b. Failback Client ISO hash link: `https://aws-elastic-disaster-recovery-hashes-{REGION}.s3.{REGION}.amazonaws.com/latest/failback_livecd/aws-failback-livecd-64bit.iso.sha512`
4. Boot the Failback Client ISO on the server you want fail back to. This can be the original source server that is paired with the recovery instance, or a different server.

Important

Ensure that the server you are failing back to has the same number of volumes or more than the Recovery Instance and that the volume sizes are equal to or larger than the ones on the recovery instance.

Note

- When performing a recovery **for a Linux server**, you must boot the Failback Client with BIOS boot mode.
- When performing a recovery **for a Windows server**, you must boot the Failback Client with the same boot mode (BIOS or UEFI) as the Windows source server.

5. If you plan on using a static IP for the Failback Client, run following once the Failback Client ISO boots:

   ```
   IPADDR="enter IPv4 address" NETMASK="subnet mask" GATEWAY="default
   gateway" DNS="DNS server IP address" CONFIG_NETWORK=1 /usr/bin/start.sh
   ```

   For example,

   ```
   IPADDR="192.168.10.20" NETMASK="255.255.255.0" GATEWAY="192.168.10.1"
   DNS="192.168.10.10" CONFIG_NETWORK=1 /usr/bin/start.sh
   ```

6. Enter your AWS credentials, including your **AWS Access Key ID** and **AWS Secret Access Key** that you created for Failback Client installation, the **AWS Session Token** (if you are using temporary credentials – users who are not using temporary credentials can leave this field blank), and the **AWS Region** in which your Recovery instance resides. You can attach the Elastic Disaster Recovery Failback Client credentials policy to a user or create a role and attach the policy to that role to obtain temporary credentials. Learn more about Elastic Disaster Recovery credentials.

7. Enter the custom endpoint or press Enter to use the default endpoint. You should enter a custom endpoint if you want to use a VPC Endpoint (PrivateLink).

8. If you are failing back to the original source machine, the Failback Client will automatically choose the correct corresponding recovery instance.

```
Failback Client automated instance detection...
Recovery instance i-02               is matched with Failback Client 4221a
```

9. If the Failback Client is unable to automatically map the instance, then you will be prompted to select the recovery instance to fail back from. The Failback Client will display a list with all recovery instances. Select the correct recovery instance by either entering the numerical choice from the list that corresponds to the correct recovery instance or by typing in the full recovery instance ID.

```
Failback Client automated instance detection...
Cannot automatically detect the Recovery Instance that matches the configuration of this server, asking for manual instance mapping.
Which Recovery Instance would you like to fail back from? Select a number from the list or enter the full Recovery Instance ID (e.g: i-xxxx).
1: i-08
Enter input: 1
Recovery instance i-08b                matched with Failback Client 4221c
```

> **Note**
>
> The Failback Client will only display recovery instances whose volume sizes are equal to or smaller than the volume sizes of the server you're failing back to. If the recovery instance has volume sizes that are larger than that of the server you are failing back to, then these Recovery instances will not be displayed.

10. If you are failing back to the original source server, then the Failback Client will attempt to automatically map the volumes of the instance.

```
Trying to get recovery volumes for i-02
Local devices:
       /dev/sda 8.0 GB
Remote devices:
       /dev/xvda 8.0 GB
```

11. If the Failback Client is unable to automatically map the volumes, you will need to manually enter a local block device (example /dev/sdg) to replicate to from the remote block device. Enter the EXCLUDE command to specifically Recovery Instance volumes from replication.

    Optionally, you can also enter the complete volume mapping in the same CSV or JSON format used by --device-mapping Failback Client argument. For example: `ALL="/dev/nvme2n1=/dev/sda,/dev/nvme0n1=EXCLUDE, . . ."`.

    The full volume mapping should be provided as single CSV or JSON line in the format of --device-mapping Failback Client argument.

    Learn more about using --device-mapping program argument

```
Cannot detect volume mapping automatically, asking for manual volume mapping
Enter local block device (e.g., /dev/sdg) to replicate from remote block device /dev/xvda or EXCLUDE to exclude: /dev/sda
Successfully mapped volumes
```

> **Important**
>
> The local volumes must be the same in size or larger than the recovery instance volumes. The valid special case is when original local volume has fractional GiB size (e.g. 9.75 GiB). Then the recovery instance volume size will be larger because of rounding to nearest GiB (e.g. 10 GiB).

12. The Failback Client will verify connectivity between the recovery instance and AWS Elastic Disaster Recovery.

```
Verifying Recovery Instance connectivity to the Dirrus service...
Connectivity established
```

13. The Failback Client will download the replication software from a public S3 bucket onto the source server.

```
Downloading AWS Replication Software [113 MB]. This may take a few minutes, please wait...
Downloaded the AWS Replication Software successfully
```

> **Important**
>
> You must allow traffic to S3 from the source server for this step to succeed.

14. The Failback Client will configure the replication software.

```
Configuring AWS Replication Software...
Finished AWS Replication Software configuration
```

15. The Failback Client will pair with the AWS Replication Agent running on the recovery instance and will establish a connection.

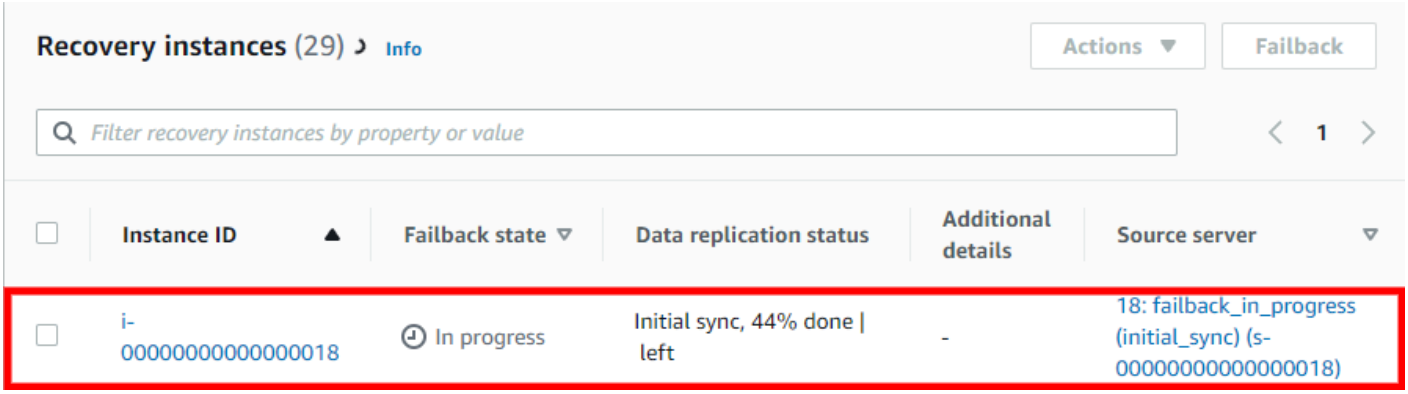
```
Pairing the AWS Replication Agent running on the Recovery Instance with the Failback Client...
Pairing completed
Establishing a connection between the Failback Client and the AWS Replication Agent running on the Recovery Instance over port 1500...
```

> **Important**
>
> TCP Port 1500 inbound must be open on the recovery instance for the pairing to succeed.

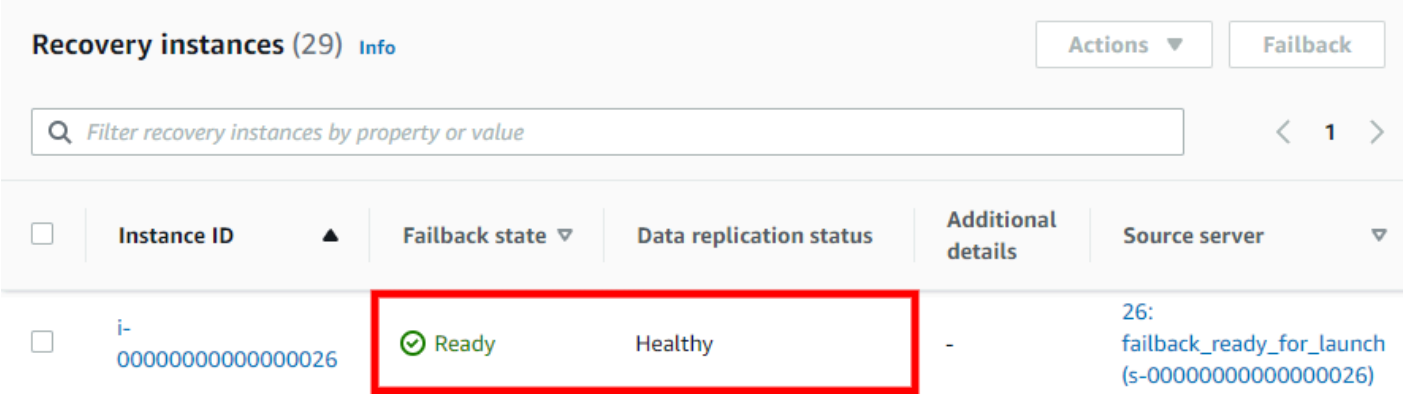
16. Data replication will begin.

```
Connection established. Replication in progress...
```

You can monitor data replication progress on the **Recovery instances** page in the AWS Elastic Disaster Recovery Console.

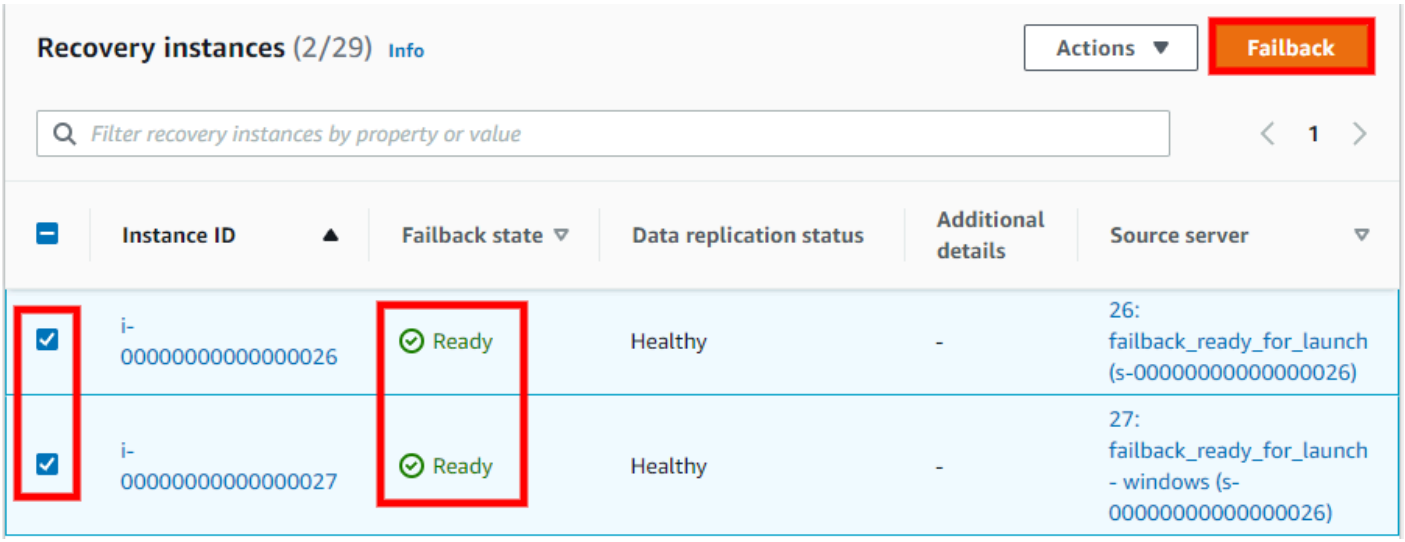


17. Once data replication has been completed, the Recovery instance on the **Recovery instances** page will show the **Ready** status under the **Failback state** column and the **Healthy** status under the **Data replication status** column.



18. Once all of the recovery instances you are planning to fail back show the statuses above, select the checkbox to the left of each Instance ID and choose **Failback**. This will stop data replication and will start the conversion process. This will finalize the failback process and create a replica of each recovery instance on the corresponding source server.

    Select the checkbox to the left of one or more recovery instances that are in the **Ready** state and click **Failback** to continue the failback process after performing a failback with the Elastic Disaster Recovery Failback Client. This action will stop data replication and will start the conversion process. This will finalize the failback process and will create a replica of each recovery instance on the corresponding source server.



When the **Continue with failback for X instances** dialog appears, click **Failback**.

This action will create a Job, which you can follow on the **Recovery job history** page. Learn more about the recovery job history page.

19. Once the failback is complete, the Failback Client will show that the failback has been completed successfully.

```
Failback completed successfully.
```

20. You can opt to either terminate, delete, or disconnect the Recovery instance. Learn more about each action.

Failback Client Program Arguments

The arguments supported by Failback Client LiveCD process are:

- --aws-access-key-id AWS_ACCESS_KEY_ID
- --aws-secret-access-key AWS_SECRET_ACCESS_KEY
- --aws-session-token AWS_SESSION_TOKEN
- --region REGION
- --endpoint ENDPOINT
- --default-endpoint
- --recovery-instance-id RECOVERY_INSTANCE_ID
- --dm-value-format {dev-name,by-path,by-id,by-uuid,all-strict}
- --device-mapping DEVICE_MAPPING] [--no-prompt

- --log-console
- --log-file LOG_FILE

All arguments are optional.

[--device-mapping DEVICE_MAPPING]

`--device-mapping` argument will skip mapping auto-detection and manual mapping and use the mapping provided in this parameter.

There are three formats supported:

1. Classic CE format of key-value CSV string as one line.

   You may use either ":" or "=" as CSV fields separator which is more sutable for Windows drive letters. Examples are:

   ```
   recovery_device1=local_device1,recovery_device2=local_device2,recovery_device3=EXCLUDE, . .
   ```

   ```
   recovery_device1:local_device1,recovery_device2:local_device2, . . .
   ```

2. JSON format:

   ```
   '{"/dev/xvdb":"/dev/sdb","/dev/xvdc":"/dev/sdc","recovery_device3":"local_device3"}'
   ```

3. JSON list DRS API format:

   ```
   '[{"recoveryInstanceDeviceName": "recovery_device1","failbackClientDeviceName":
    "local_device1"},{"recoveryInstanceDeviceName" . . .: }]'
   ```

No matter which format you choose, you need to provide either valid Failback Client device name or EXCLUDE for each Recovery Instance device.

[dm-value-format DM_VALUE_FORMAT]

`--dm-value-format` allows to use Failback Client persistent block devices identifiers in --device-mapping argument.

Such persistent identifiers will always refer to the same block devices after Failback Client reboot.

Possible --dm-value-format choices are:

1. "dev-name" - default format for using /dev/sda, /dev/xvda, /dev/nvme3n1 etc
2. "by-path" - from ls -l /dev/disk/by-id/ e.g. pci-0000:00:10.0-scsi-0:0:3:0, pci-0000:00:1e.0-nvme-1, pci-0000:02:01.0-ata-1, xen-vbd-768 etc
3. "by-id" - from ls -l /dev/disk/by-id/ e.g. device serial numbers
4. "by-uuid" - UUIDs from ls -l /dev/disk/by-uuid/
5. "all-strict" - all of the above mixed

We will use the example of SCSI identifiers from the command output below:

```
# root@ubuntu:~# ls -l /dev/disk/by-path/
total 0
lrwxrwxrwx 1 root root  9 Jun 27 12:25 pci-0000:00:10.0-scsi-0:0:0:0 -> ./../sda
lrwxrwxrwx 1 root root 10 Jun 27 12:25 pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../
sda1
lrwxrwxrwx 1 root root  9 Jun 27 12:25 pci-0000:00:10.0-scsi-0:0:1:0 -> ../../sdb
lrwxrwxrwx 1 root root  9 Jun 27 12:25 pci-0000:00:10.0-scsi-0:0:2:0 -> ../../sdc
lrwxrwxrwx 1 root root  9 Jun 27 12:25 pci-0000:00:10.0-scsi-0:0:3:0 -> ../../sdd
```

To use block device SCSI identifies like 'pci-0000:00:10.0-scsi-0:0:0:0' you need to add to command line:`--dm-value-format by-path`

The examples of valid --device-mapping for `--dm-value-format by-path` are:

```
/dev/nvme2n1=pci-0000:00:10.0-scsi-0:0:0:0,/dev/nvme0n1=pci-0000:00:10.0-scsi-0:0:1:0,/
dev/nvme3n1=pci-0000:00:10.0-scsi-0:0:2:0...
```

```
'{"/dev/nvme2n1":"pci-0000:00:10.0-scsi-0:0:0:0","/dev/nvme0n1":"pci-0000:00:10.0-
scsi-0:0:1:0","/dev/nvme3n1":"pci-0000:00:10.0-scsi-0:0:2:0", . . .}'
```

No matter which format you choose, you need to provide either valid Failback Client device name or EXCLUDE for each Recovery Instance device.

Performing a failback with the DRS Mass Failback Automation client

DRS allows you to perform a scalable failback for vCenter with the DRS Mass Failback Automation Client (DRSFA Client). This allows you to perform a one-click or custom failback for multiple vCenter machines at once.

> **Note**
>
> The DRSFA client only works with vCenters source servers.

> **Note**
>
> The DRSFA client was only tested on vCenter versions 6.7 and 7.0.

DRSFA prerequisites

The following are the prerequisites for performing failback automation with the DRSFA client:

1. Ensure that you meet all of the network requirements.
2. Ensure that you have initialized DRS.
3. Each server that is being failed back must have at least 3 GB of ram.
4. Each server that is being failed back must have the hardware clock set to UTC rather than Local Time.
5. The recovery instance used as a source for failback must have permissions to access AWS Elastic Disaster Recovery via API calls. This is done using instance profile for the underlying EC2 instance. The instance profile must include the AWSElasticDisasterRecoveryRecoveryInstancePolicy in addition to any other policy you require the EC2 instance to have. By default, the launch settings that DRS creates for source servers already have an instance profile defined that includes that policy and that instance profile will be used when launching a Recovery Instance.
6. Inbound port TCP 1500 must be open on the Recovery instance in AWS.
7. The server on which the DRSFA client is ran needs to be able to communicate with your vCenter environment.
8. The server on which the DRSFA client is ran must have at least 4 GB of ram.

9. The server on which the DRSFA client is ran must run Python 3.9.4 with pip installed (other versions of Python will not work).

> **Note**
>
> The installation procedure shown below uses Ubuntu 20.04 which has the required Python version preinstalled.

10. The server on which the DRSFA client is ran requires the following tools for DRSFA Client installation. The installer will attempt to install them if they are not already present::

    build-essential curl genisoimage git libbz2-dev libffi-dev liblzma-dev libncurses5-dev libncursesw5-dev libreadline-dev libsqlite3-dev libssl-dev llvm make tk-dev unzip wget xz-utils zlib1g-dev

    a. To see the list of python libraries required for the DRSFA Client to run, see the requirements.txt file (https://drsfa-us-west-2.s3.us-west-2.amazonaws.com/requirements.txt). These libraries will be installed automatically by DRSFA Client.

11. The vCenter source servers must have two CD ROM devices with IDE controllers attached to run the DRSFA client - one for the DRS Failback Client and one for the drs_failback_automation_seed.iso

> **Note**
>
> If no attached CD ROM devices are found, the DRSFA client will attempt to add the CD ROM devices.

12. The DRS Failback Client must be uploaded to your vCenter Datastore.

13. We recommend using the latest version of the DRS Failback Client. Download the latest version of the DRS Failback Clientand upload it to your vCenter datastore.

14. We recommend running SHA512 checksum verification on the DRS Failback Client prior to using it with the DRSFA client. You can verify the checksum at the following address: `https://aws-elastic-disaster-recovery-hashes-{REGION}.s3.amazonaws.com/latest/failback_livecd/aws-failback-livecd-64bit.iso.sha512`

15. We recommend running SHA512 checksum verification on the drs_failback_automation_seed.iso file prior to using it with the DRSFA client.

16. The DRSFA client does not require root privileges. We recommend low privileges for running the client.
17. You need to have the following vCenter API credentials and permissions: 'Virtual machine' : ['Change Settings', 'Guest operation queries', 'Guest operation program execution', 'Connect devices', 'Power off', 'Power on'. 'Add or remove device', 'Configure CD media] 'Datastore': ['Browse datastore']
18. vCenter credentials should only be constrained to the VMs you plan to failback.
19. You should be able to fail back all of the Recovery instances in a single AWS Region simultaneously with the aid of the DRSFA Client as long as your vCenter hardware supports the failback load.

Security best practices

The following are security best practices for using the DRSFA Client:

1. Follow the least privilege principle and set the appropriate permissions on the folder where the JSON generated by the client will be stored.
2. Ensure that you are always using the latest version of the DRSFA Client. The client will automatically check and verify that you are using the latest version upon startup.
3. You should not provide any additional permissions to the DRSFA Client other than the ones listed in the prerequisites.
4. Ensure that you follow the AWS recommended password policy when setting the password for the VM that hosts the DRS Failback Client when generating the drs_failback_automation_seed.iso file.
5. Ensure that you manually verify the DRSFA client hashes when automatic hash verification is not performed. The hash verification hint is shown when the DRSFA client is installed.
6. Ensure that only trusted administrators have access to the vCenter environment. The DRSFA Client will consider the customer executing scripts and every person with access to the datastore as a single trust entity
7. We suggest performing a hash verification on the DRS Failback Client and the drs_failback_automation_seed.iso file before proceeding. The hash is exported to the `drs_failback_automation_seed.iso.sha512` file once the seed iso is created.
8. We suggest using low level privilege when running the DRSFA client.

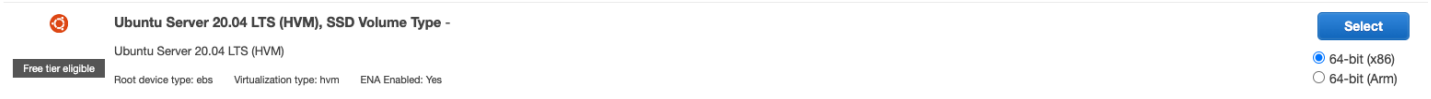
9. We suggest following the least privilege principle and setting the appropriate permissions on the folder where the Failback Client and seed.iso files will be stored.
10. The vCenter credentials used should only have permissions to the VMs involved in the failback attempt.

Installing the DRSFA Client

Prior to running the DRSFA Client, you must first install it. Installing the client is a one-time operation.

The DRSFA client was fully tested on Ubuntu 20.04 and an installation script for this version is provided. Use the following vanilla AMI or public ISO to run the client locally in your vCenter environment.

Follow the Create your EC2 resources and launch your EC2 instance guidelines as per the EC2 documentation. When asked to select an AMI, select the option below instead of the Amazon Linux 2 AMI and then proceed according to the documentation. Use the following AMI from EC2: Ubuntu Server 20.04 LTS (HVM), SSD Volume Type:



Download the Ubuntu Server 20.04 LTS server install image ISO from the Ubuntu download site.

Once your VM instance is set up and ready, connect to the Ubuntu instance and run command prompt and download the DRSFA client using the following command:

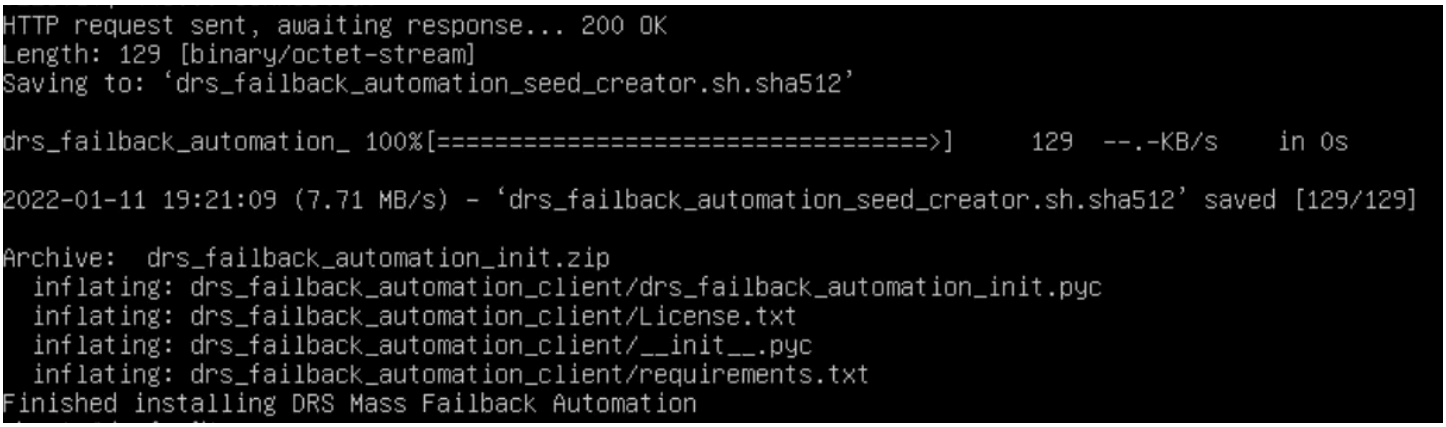
```
wget https://drsfa-us-west-2.s3.us-west-2.amazonaws.com/
drs_failback_automation_installer.sh
```

```
HTTP request sent, awaiting response... 200 OK
Length: 3422 (3.3K) [binary/octet-stream]
Saving to: 'drs_failback_automation_installer.sh'

drs_failback_automation_ 100%[=================================>]   3.34K  --.-KB/s    in 0s

2022-01-11 17:37:57 (33.8 MB/s) - 'drs_failback_automation_installer.sh' saved [3422/3422]
```

> **Note**
>
> You should verify the hash of the installer after running the installation command: `https://drsfa-hashes-us-west-2.s3.us-west-2.amazonaws.com/drs_failback_automation_installer.sh.sha512`

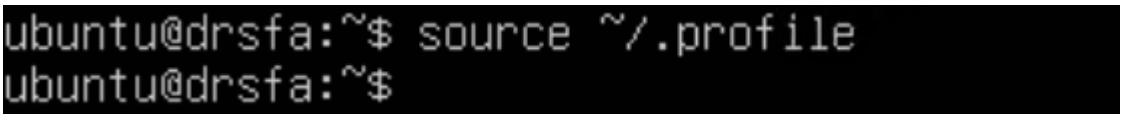
Use the following command to execute the installation script:

```
bash drs_failback_automation_installer.sh
```



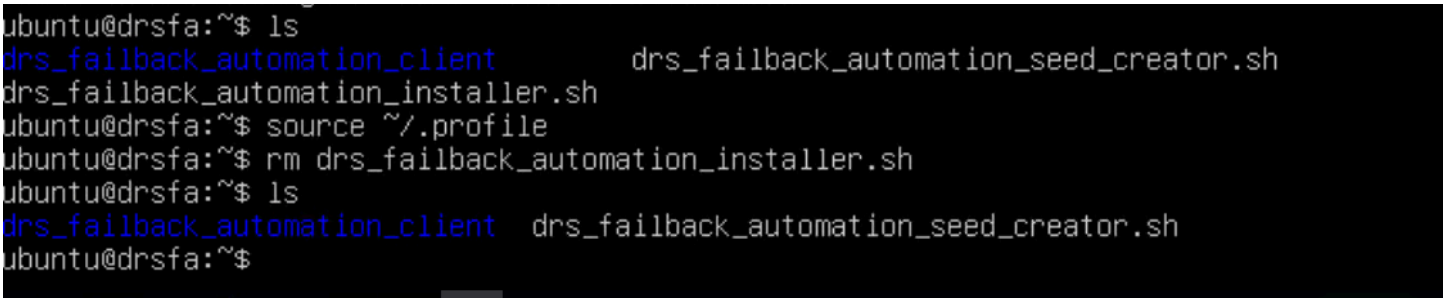
> **Note**
>
> This command may ask for a sudo password if you use the Ubuntu ISO. Enter the password but **do not** run this command as sudo.

```
source ~/.profile
```



The DRSFA client has a one-time installation. The DRSFA client will be installed in the `drs_failback_automation_client` directory. Once you've successfully ran the command above and installed the client, you can delete the DRSFA client installer from your server by running the following command:

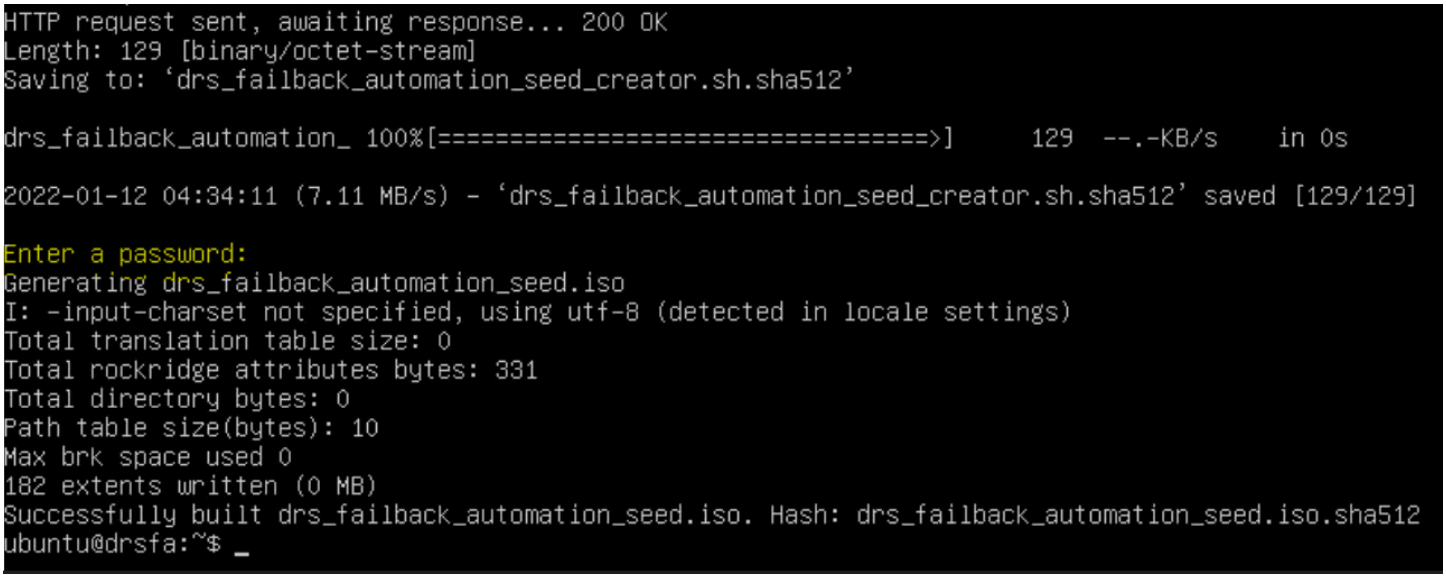
```
rm drs_failback_automation_installer.sh
```



Once installation is complete, you will need to set up a password for the VM on which the DRSFA client is ran. This is done by generating a seed.iso file that you must upload to your Datastore. Run the following commands to generate the seed.iso file:

```
bash drs_failback_automation_seed_creator.sh
```

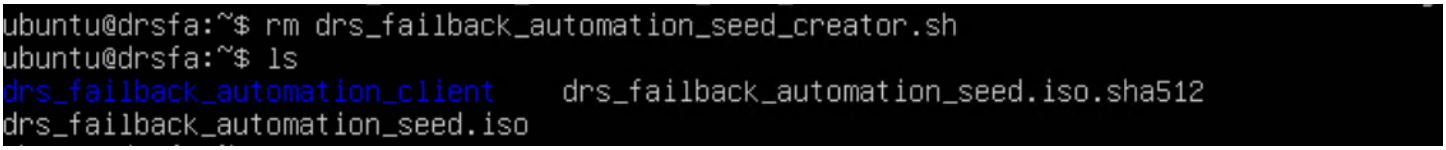
You will be prompted to enter a password. Ensure that you enter a unique password that following the AWS recommended password policy.



Two files will be generated, the `drs_failback_automation_seed.iso` file and the `drs_failback_automation_seed.iso.sha512` hash. Upload the seed.iso file to the same Datastore where the DRS Failback Client ISO file is stored.

Once the `drs_failback_automation_seed.iso` file is generated, you can run the following command to delete the seed creator:

`rm drs_failback_automation_seed_creator.sh`



Once you have completed the initial installation, you can generate the required credentials and run the DRSFA client.

Generating IAM credentials and configuring Cloudwatch logging

In order to run the DRSFA Client, you must first generate the required AWS credentials.

> **Important**
>
> Temporary credentials have many advantages. You don't need to rotate them or revoke them when they're no longer needed, and they cannot be reused after they expire. You can specify for how long the credentials are valid, up to a maximum limit. Because they provide enhanced security, using temporary credentials is considered best practice and the recommended option.

Temporary credentials

To create temporary credentials, take the following steps:

1. Create a new IAM Role with the **AWSElasticDisasterRecoveryAgentInstallationPolicy** policy.
2. Request temporary security credentials via AWS STS using the AssumeRole API.

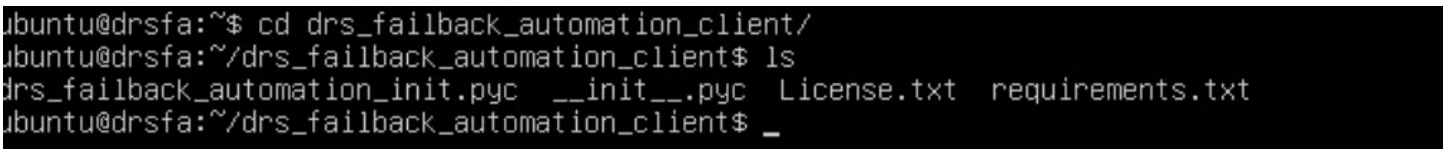
Once your credentials are generated, you should create a logGroup for CloudWatch logging named **DRS_Mass_Failback_Automation**. If this log group is not created or if it's created with the wrong name, the DRSFA client will still work, but logs will not be sent to CloudWatch. Learn more about working with log groups in the Amazon CloudWatch Logs documentation.

Running the DRSFA client

Once you have installed the DRSFA client, you can run it by following these instructions:

`cd` into the `drs_failback_automation_client` directory and enter the following parameters in a single line or settings the environment variables one by one, replace the defaults with your

specific parameters and paths followed by the `python drs_failback_automation_init.pyc` command and press enter.



- AWS_REGION=XXXXX – The AWS Region in which your Recovery instances are located.
- AWS_ACCESS_KEY=XXXXX – The AWS Access Key you generated for the DRSFA client.
- AWS_SECRET_ACCESS_KEY=XXXXXX – The AWS Secret Access Key you generated for the DRSFA client.
- DRS_FAILBACK_CLIENT_PASSWORD = XXXXXX – The custom password you set for the Failback Client in the drs_failback_automation_seed.iso file.
- VCENTER_HOST=XX.XX.XXX.XXX – The IP address of the vCenter Host.
- VCENTER_PORT=XXX – The vCenter Port (usually 443)
- VCENTER_USER=sample@vsphere.local – The vCenter username
- VCENTER_PASSWORD=samplepassword – The vCenter password
- VCENTER_DATASTORE=DatastoreX – The Datastore within vCenter where the Failback Client ISO file (aws-failback-livecd-64bit.iso) and seed.iso file (drs_failback_automation_seed.iso) are stored.
- VCENTER_FAILBACK_CLIENT_PATH='samplepath/aws-failback-livecd-64bit.iso' – Failback Client ISO path in the Datastore.
- VCENTER_SEED_ISO_PATH='samplepath/drs_failback_automation_seed.iso' – The seed.iso file path in the Datastore.

You should enter all of the parameters in a single line or enter the environmental variables individually one by one. Once you have entered your parameters, enter the `python drs_failback_automation_init.pyc` command and press enter. The full parameters and command should look like the following example:

```
AWS_REGION=XXXX AWS_ACCESS_KEY=XXXX AWS_SECRET_ACCESS_KEY=XXXX
DRS_FAILBACK_CLIENT_PASSWORD=XXXX VCENTER_HOST=XXXX VCENTER_PORT=XXXX
VCENTER_USER=XXXX VCENTER_PASSWORD=XXXX VCENTER_DATASTORE=XXXX
VCENTER_FAILBACK_CLIENT_PATH=XXXX VCENTER_SEED_ISO_PATH=XXXX python
drs_failback_automation_init.pyc
```

```
ubuntu@drsfa:~/drs_failback_automation_client$ AWS_REGION=          AWS_ACCESS_KEY=
     AWS_SECRET_ACCESS_KEY=                                          DRS_FAILBACK_CLIENT_PASSWORD='
          VCENTER_HOST=                VCENTER_PORT=    VCENTER_USER=
 VCENTER_PASSWORD=            VCENTER_DATASTORE=           VCENTER_FAILBACK_CLIENT_PATH=          i/a
ws-failback-livecd-64bit.iso' VCENTER_SEED_ISO_PATH='          drs_failback_automation_seed.iso' DIS
ABLE_SSL_VERIFICATION=TRUE python3 drs_failback_automation_init.pyc
Looking for DRS Mass Failback Automation latest version
Found software and configuration files locally. Verifying latest version...
DRS Mass Failback Automation Client was updated successfully
```

Note

SSL verification is active by default. If you want to deactivate SSL verification, then add the following parameter: DISABLE_SSL_VERIFICATION=true

Note

By default, the DRSFA client will initiate a failback for 10 servers at once (if failing back more than 10 servers). However, if you would like to change the default value, use the THREAD_POOL_SIZE parameter.

One-click failback

Once the client has connected successfully and finished verification, select the **One-Click Failback** option under **What would you like to do?**

```
ubuntu@drsfa:~$ cd drs_failback_automation_client/
ubuntu@drsfa:~/drs_failback_automation_client$ python3 drs_failback_automation_init.pyc
Looking for DRS Mass Failback Automation latest version
Required software and configuration files downloaded successfully
? Would you like to continue?  Yes
The aws-failback-livecd-64bit.iso file was found in datastore datastore1. TIP: Ensure that the DRS F
ailback Client ISO file is up to date. You can use this hash to ensure you have the latest version h
ttps://aws-elastic-disaster-recovery-hashes-eu-west-1.s3.amazonaws.com/latest/failback_livecd/aws-fa
ilback-livecd-64bit.iso.sha512
The drs_failback_automation_seed.iso file was found in datastore datastore1
Welcome to the DRS Mass Failback Automation CLI
? What would you like to do?  (Use arrow keys)
 ♦ One-Click Failback
   Perform a Custom Failback
   Generate a default failback configuration file
   Find servers in vCenter
   Help
   Exit
```

Enter a custom prefix for the results output for this failback operation. This file will be saved in the `/drs_failback_automation_client/results/Failback` directory.

```
? What would you like to do?  One-Click Failback
? Enter a custom prefix for the results output:  drsfa_1_
```

If failback replication has already been started for some of the Recovery instances, the console will prompt you whether you want to skip the instances that are already in failback or restart replication for those instances.

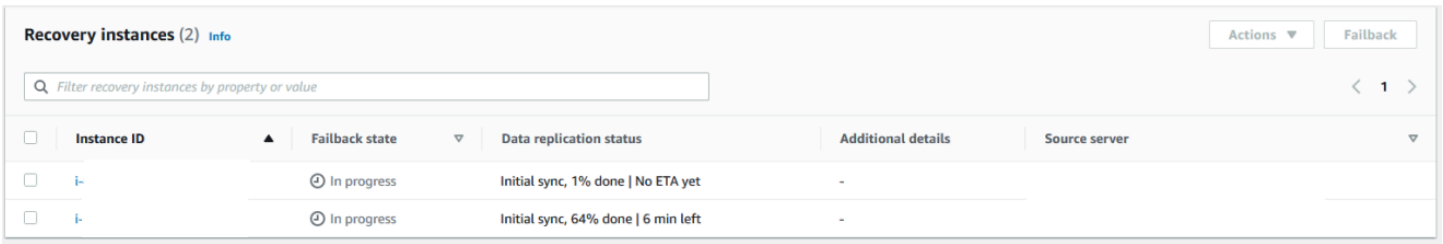
```
Note that you already started failback replication for some of your servers, we will skip those so t
hat they don't restart the process.
? Would you like us to restart the machines?  (Use arrow keys)
 ♦ No. Skip those instances.
   Yes. Restart all.
```

The DRSFA client will list the Recovery instances that are currently present in your AWS Account. The client will then prompt you **Would you like to continue?** . Enter **Y** to continue.

```
The following Recovery instances will be failed back to their original VMs:
i-
i-
? Would you like to continue?  (Y/n)
```

The client will initiate failback. You can see the failback progress on the **Recovery instances** page in the DRS Console.

```
Initiating failback for account               region eu-west-1


21:48:39: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:49:42: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:50:45: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:51:48: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:52:50: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:53:53: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:54:55: 2 total servers. 1 currently replicating, 1 initiating replication, 0 skipped, 0 failed
21:55:57: 2 total servers. 2 currently replicating, 0 initiating replication, 0 skipped, 0 failed
```

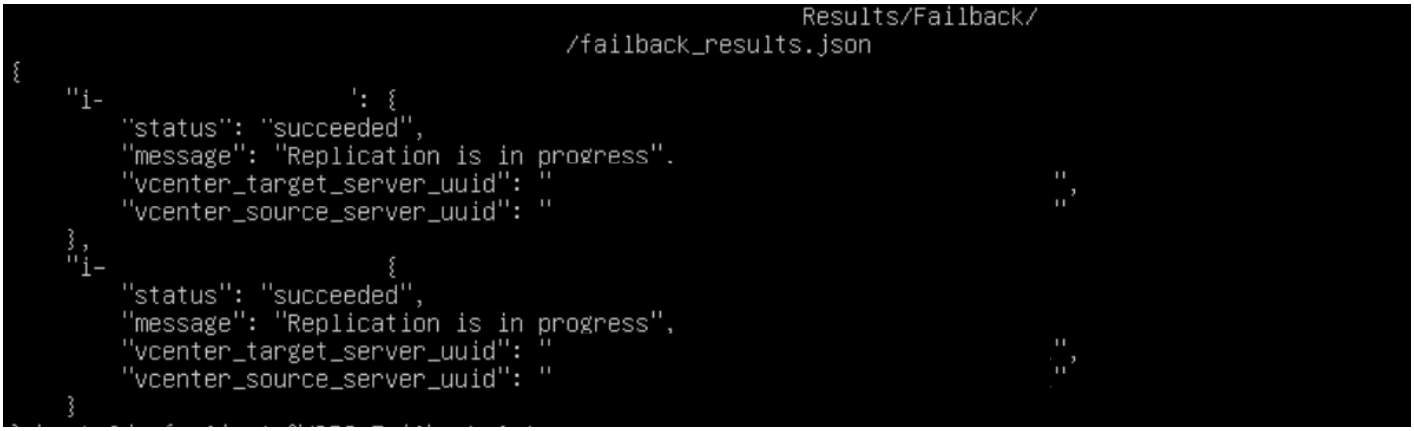


Once the failback has been complete, the DRSFA client will display the results of the failback, including the number of servers for which replication has successfully been initiated and the number of servers for which the failback operation failed.

The full results of the failback will be exported as a JSON file to the failback client folder path under the `/drs_failback_automation_client/results/Failback` folder with the custom prefix you set, the AWS account ID, the AWS Region, and a timestamp.

The JSON file will display the following:

- The AWS ID of the Recovery instance
- The status of the failback (succeeded, skipped, or failed)
- A message (which provides the cause for failure in the case of failure)
- The vCenter VM UUID



If failback failed for any of your machines, you can troubleshoot the failure by looking at the machine configuration `failback_hosts_settings.json` file in the same folder.

Here, you can see the exact configurations of the failed machines. You can then fix any problems and use the custom failback flow explained below to fail back these specific machines.

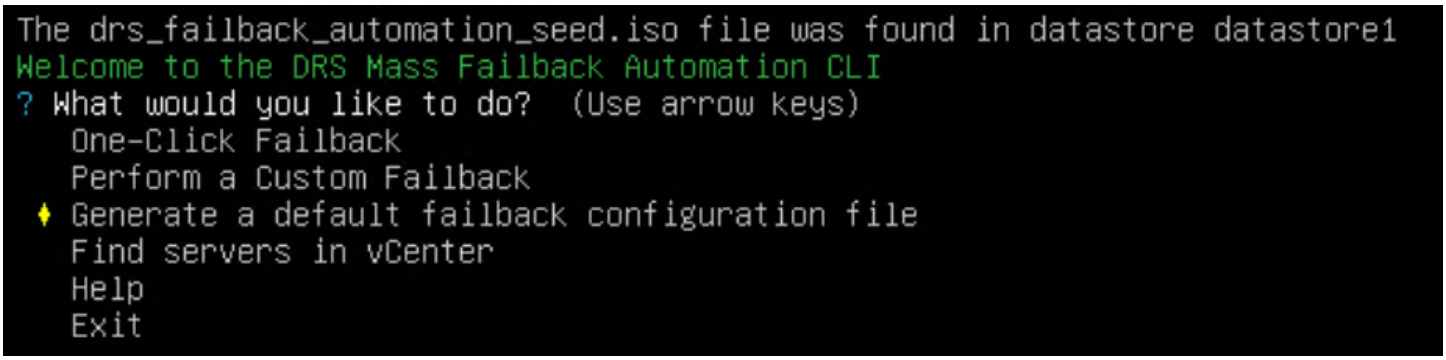
Custom failback

The custom failback option gives you more control and flexibility over the failback process. When utilizing the custom failback option, you will first create a failback configuration file, in which you can edit specific settings for each individual machine, and you will then use this file to perform a failback in a flow that is similar to that of the one-step failback.

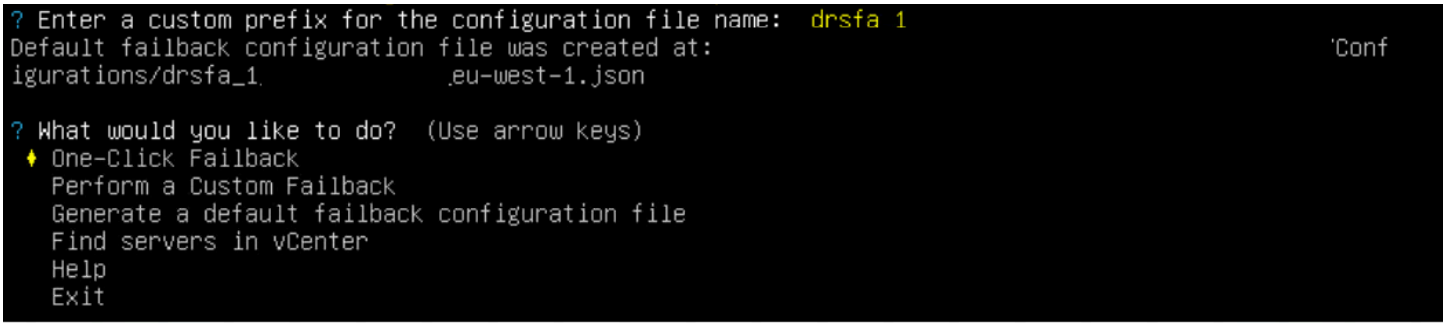
Generating the configuration file

To use the custom failback option, you can either create a custom configuration JSON file or generate a default failback configuration file through the client.

To generate a default failback configuration file, once the client has connected successfully and finished verification, select the **Generate a default failback configuration file** option under **What would you like to do?**



Enter a custom prefix for the configuration file name. The configuration file will be created as a JSON file in the `/drs_failback_automation_client/ Configurations` /folder with the following name: "{prefix}_{account_id}_{region}.json"



You can edit any of the fields in the file in order to correctly configure it. The file will display the following fields for each machine. You can edit every field to have absolute control over your failback configuration for each machine. Ensure to save your changes.

- NETMASK
- VCENTER_MACHINE_UUID
- PROXY
- DNS
- CONFIG_NETWORK
- IPADDR
- GATEWAY
- SOURCE_SERVER_ID
- DEVICE_MAPPING

> **Note**
>
> - The `CONFIG_NETWORK` value should be set to "DHCP" if you are using DHCP. The value should be set to "STATIC" if you want to manually configure the network settings. If CONFIG_NETWORK is set to "DHCP", then the `DNS, IPADDR, GATEWAY, NETMASK`, and `PROXY` parameters are ignored but should not be deleted.
> - If you are using a proxy server, leave the `PROXY` field as an empty string, do not remove it.
> - If a source server does not have an attached recovery instance, the file will still be generated, but the **SOURCE SERVER ID** field will be empty.

You can edit any of the fields in the file in order to correctly configure it. The file will display the following fields for each machine. You can edit every field to have absolute control over your failback configuration for each machine. Ensure to save your changes.

Custom device mapping parameter

Custom "DEVICE_MAPPING" field is passed to the LiveCD failback process as --device-mapping argument. Learn more about using --device-mapping program argument

There are three formats supported:

1. Classic CE format of key-value CSV string as one line.

You may use either ":" or "=" as CSV fields separator which is more sutable for Windows drive letters. Examples are:

```
"DEVICE_MAPPING":
 "recovery_device1=local_device1,recovery_device2=local_device2,recovery_device3=EXCLUDE"
```

```
"DEVICE_MAPPING": "recovery_device1:local_device1,recovery_device2:local_device2"
```

2. JSON format:

```
"DEVICE_MAPPING": {
    "/dev/xvdb":"/dev/sdb",
    "/dev/xvdc":"/dev/sdc",
    "recovery_device3":"local_device3"
}
```

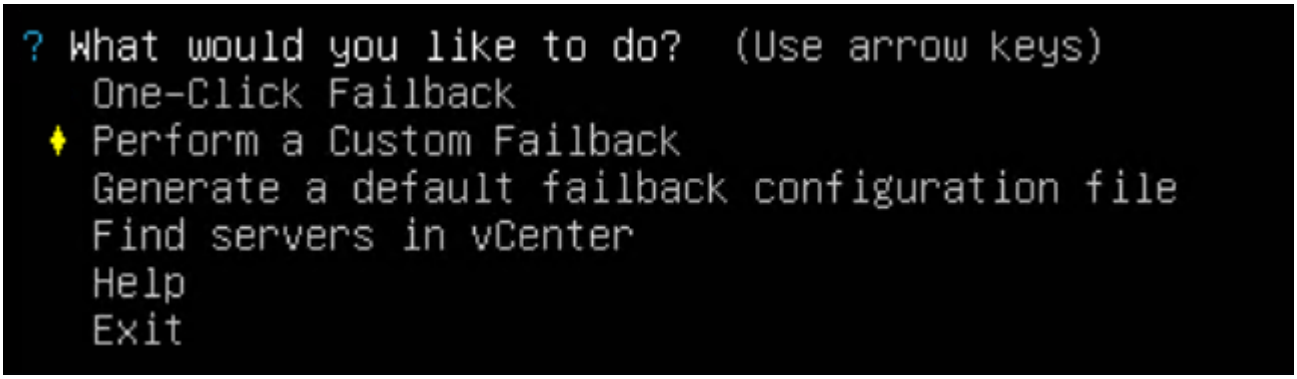
3. JSON list DRS API format:

```
[
    {
    "recoveryInstanceDeviceName": "recovery_device1",
    "failbackClientDeviceName": "local_device1"
    },
    {
    "recoveryInstanceDeviceName": "recovery_device2",
    "failbackClientDeviceName": "local_device2"
    }
]
```

No matter which format you choose, you need to provide either valid Failback Client device name or EXCLUDE for each Recovery Instance device.

Performing the custom failback

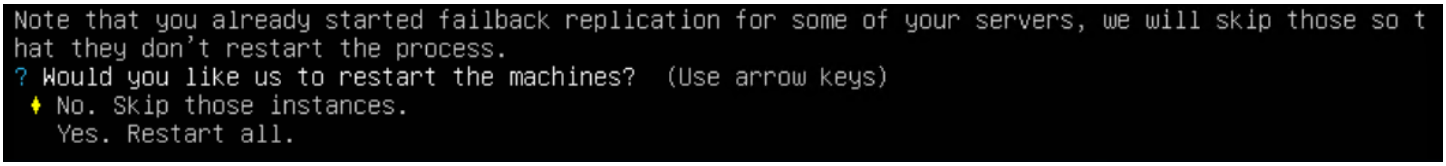
Once you are done editing your configuration file, rerun the DRSFA client and select the **Perform a Custom Failback** option.



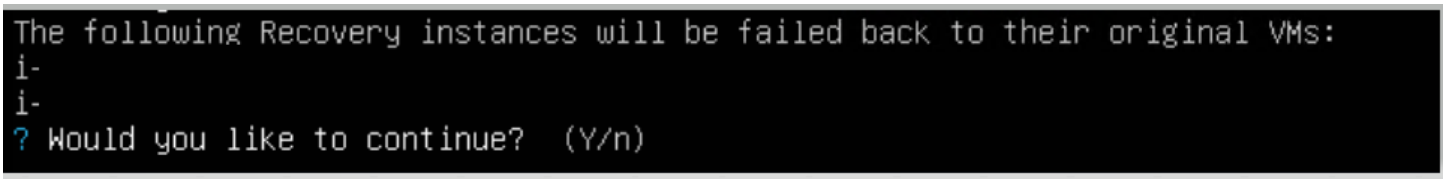
Select your configuration file. You can either define a custom path or select the default path that's automatically displayed by the client.

Enter a custom prefix for the results output for this failback operation. This file will be saved in the `/drs_failback_automation_client/Results/Failback` directory.

If failback replication has already been started for some of the recovery instances, the console will prompt you whether you want to skip the instances that are already in failback or restart replication for those instances.

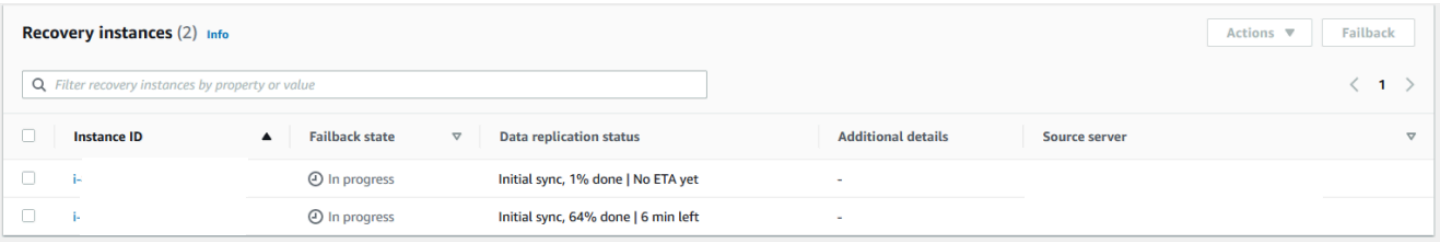


The Client will identify the recovery instances that will be failed back to their original VMs and list them. The client will then prompt you whether you would like to continue. Choose **Y** to continue.



The Client will initiate failback. You can see the failback progress on the **Recovery instances** page in the AWS DRS Console.

```
Initiating failback for account              region eu-west-1


21:48:39: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:49:42: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:50:45: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:51:48: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:52:50: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:53:53: 2 total servers. 0 currently replicating, 2 initiating replication, 0 skipped, 0 failed
21:54:55: 2 total servers. 1 currently replicating, 1 initiating replication, 0 skipped, 0 failed
21:55:57: 2 total servers. 2 currently replicating, 0 initiating replication, 0 skipped, 0 failed
```

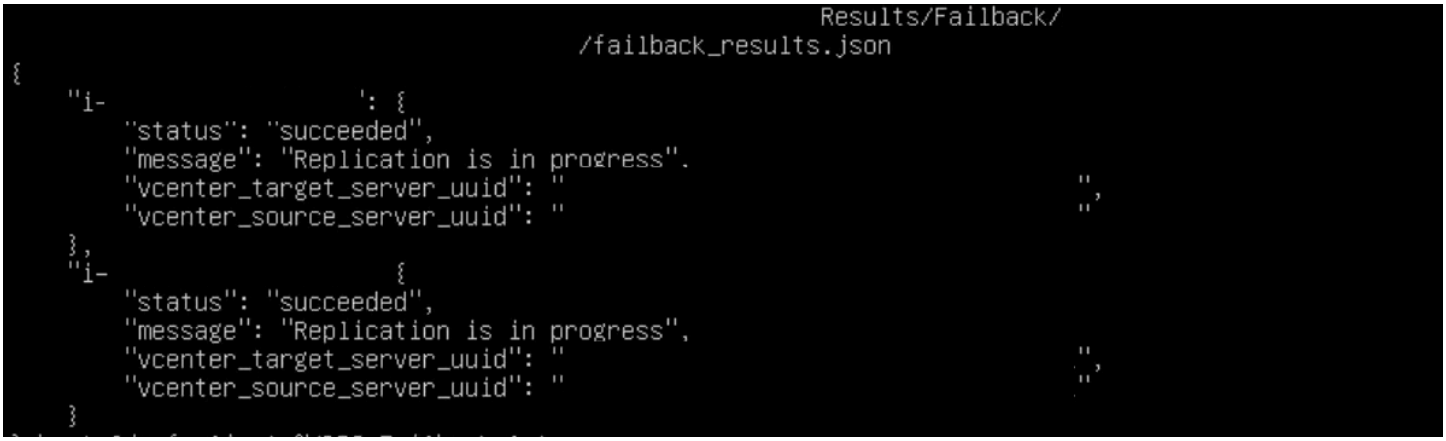


Once the failback has been complete, the DRSFA client will display the results of the failback, including the number of servers for which replication has successfully been initiated and the number of servers for which the failback operation failed.

The full results of the failback will be exported as a JSON file to the failback client folder path under the `/drs_failback_automation_client/Results/Failback` folder with the custom prefix you set, the AWS account ID, the AWS Region, and a timestamp.

The JSON file will display the following:

- The AWS ID of the Recovery instance
- The status of the failback (succeeded, skipped, or failed)
- A message (which provides the cause for failure in the case of failure)
- The vCenter VM UUID
- The vCenter UUID of the original source server

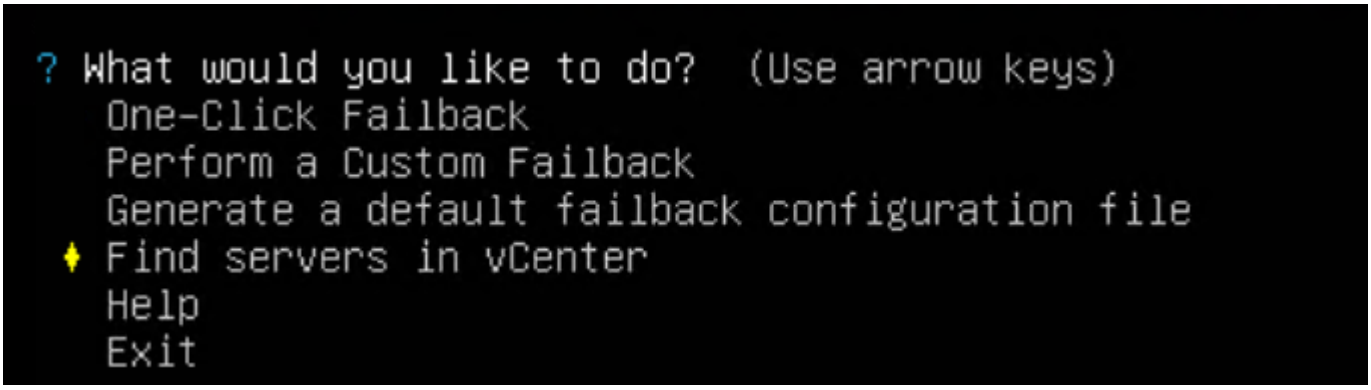


If failback failed for any of your machines, you can troubleshoot the failure by looking at the machine configuration `failback_hosts_settings.json` file in the same folder.

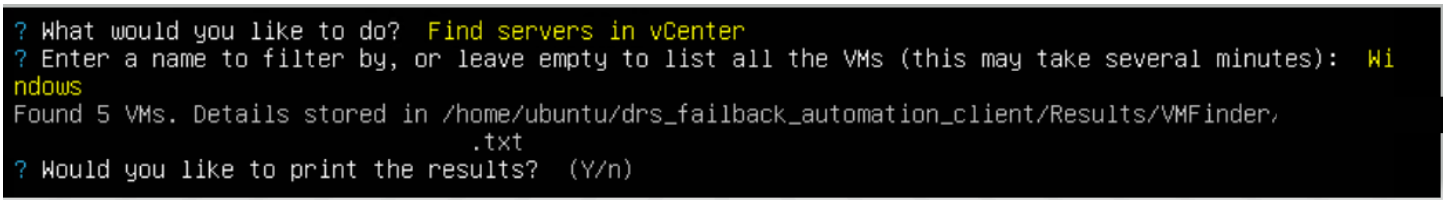
Here, you can see the exact configurations of the failed machines. You can then fix any problems and use the custom failback flow explained below to fail back these specific machines.

Find servers in vCenter

Select the **Find servers in vCenter** option to find machines in vCenter. This makes it easier to discover the disks/volumes of your machines for custom failback.



Enter a name to filter or press Enter to see all results. Choose **Yes** to print your results.



The results will be exported to the `Results/VMFinder` folder in the DRSFA client folder. The results will be named after the vCenter IP and the time stamp. `{vcenter_host}_{ts}.txt`

The following will be displayed for each server:

- Name
- UUID
- Disk and volume info

Upgrading the DRSFA Client

Most of DRSFA components are upgraded automatically upon execution. However, in certain scenarios, you will see a message informing you that you need to upgrade the DRSFA Client manually.

To complete the upgrade, take the following steps:

1. Change directory (cd) into the directory where the installation originally took place.
2. Download the DRSFA installer:

```
wget https://drsfa-us-west-2.s3.us-west-2.amazonaws.com/
drs_failback_automation_installer.sh
```

> **Note**
>
> You should verify the hash of the installer after running the installation command: `https://drsfa-hashes-us-west-2.s3.us-west-2.amazonaws.com/drs_failback_automation_installer.sh.sha512`

3. Run the installer.

```
bash drs_failback_automation_installer.sh
```

4. Remove the installer.

```
rm drs_failback_automation_installer.sh
```

Troubleshooting

- To troubleshoot the DRSFA Client, review the `drs_failback_automation.log` file that is generated in the `/drs_failback_automation_client/` folder on the server from which the client is ran.
- To find the log for a specific server, open the VM, and find the `drs_failback_automation.log` and `failback.log` file, which can be used for troubleshooting.

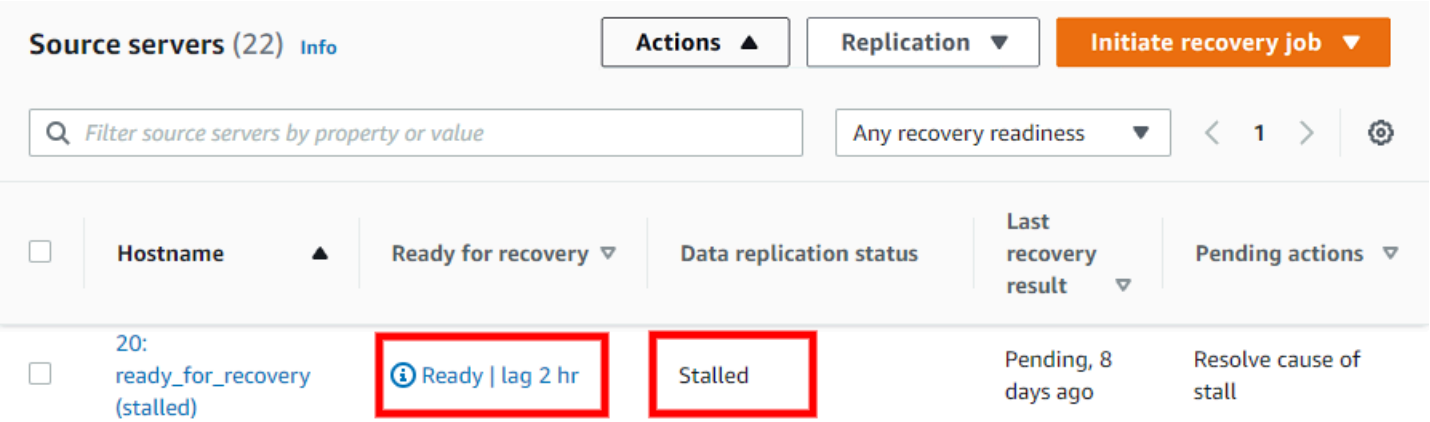
Failback notes

Using the faillback client to perform a failback to the original source server

When using the failback client, you can fail back to the original source server or a different source server using AWS Elastic Disaster Recovery.

Te ensure that the original source server has not been deleted and still exists, check its status in the AWS DRS console. Source servers that have been deleted or no longer exist will show as having **Lag** and being **Stalled**.

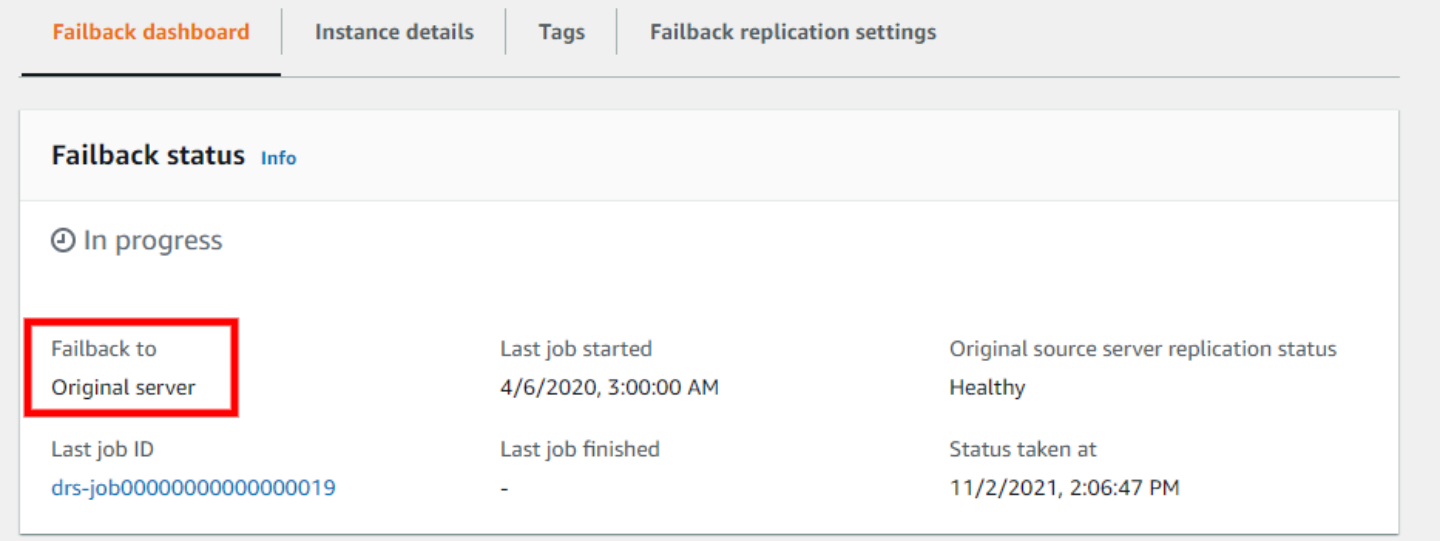
> **Note**
>
> After failing back to the original source server, you don't need to reinstall the DRS agent to start replication back to AWS.



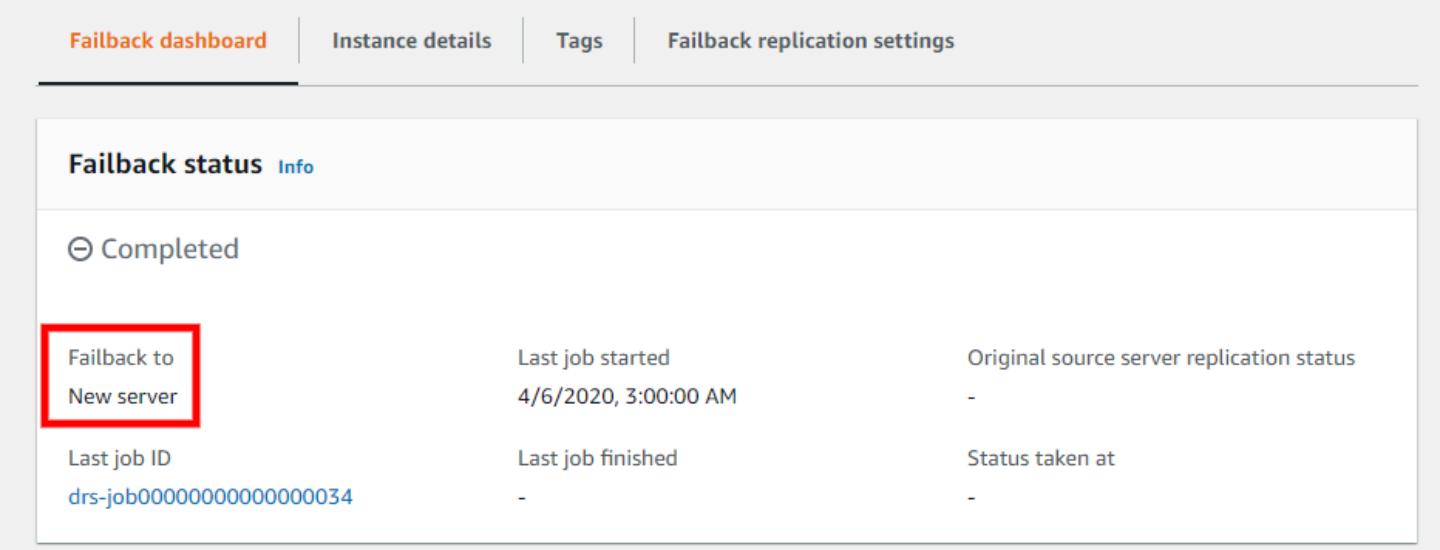
If the original source server is healthy and you decide to fail back to it, it will undergo a rescan until it reaches the **Ready** status.

You can tell whether you are failing back to the original or a new source servers in the recovery instance details view under **Failback status**.

Original server:



New server:



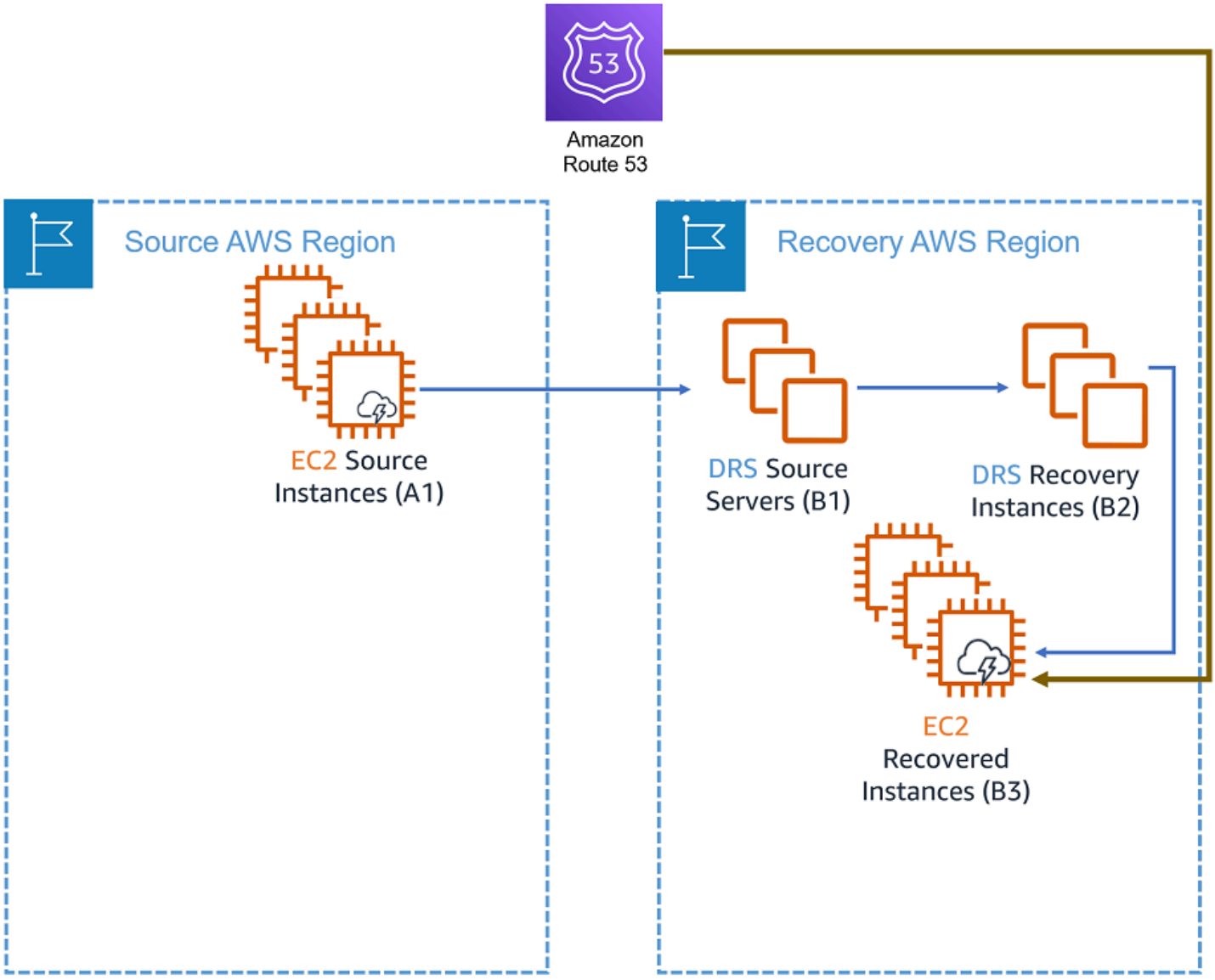
Performing a cross-Region failback

AWS Elastic Disaster Recovery (AWS DRS) allows you to perform failover and failback your EC2-based applications from one AWS Region to another AWS Region. The failover process is the same as failing over into an AWS Region from a source outside of AWS, but the failback process is different. The instructions below describe the complete cross-Region failover and failback process. In the examples, we use us-east-1 as the source AWS Region and us-east-2 as the recovery AWS Region, but any combination of AWS Regions that are supported by DRS will work.

> **Note**
>
> Cross-Partition failback features between commercial, and AWS GovCloud partitions are not supported. Cross-Region failback features within the AWS GovCloud partition are available between AWS GovCloud Regions (us-gov-west-1 and us-gov-east-1)

Overview and prerequisites

The failback process starts after the failover process ends. During failover, AWS DRS allows you to replace the EC2 source instance (A1) with the EC2 recovered instance (B3). The current AWS resource state is illustrated in the following diagram:



After performing a recovery, your applications are running on EC2 instances in the recovery region. However, these recovered instances (marked B3 in the diagram above) are not protected against other potential outages. In order to avoid data loss, you should start a reversed replication immediately. Starting reversed replication involves copying the data from the EC2 recovered instances (B3) to the original region, an operation that takes time and incurs cross-Region data transfer costs.

Once replication has reached a healthy state, failing back to the source region is possible using the DRS console on that region, assuming DRS has been initialized in the source region.

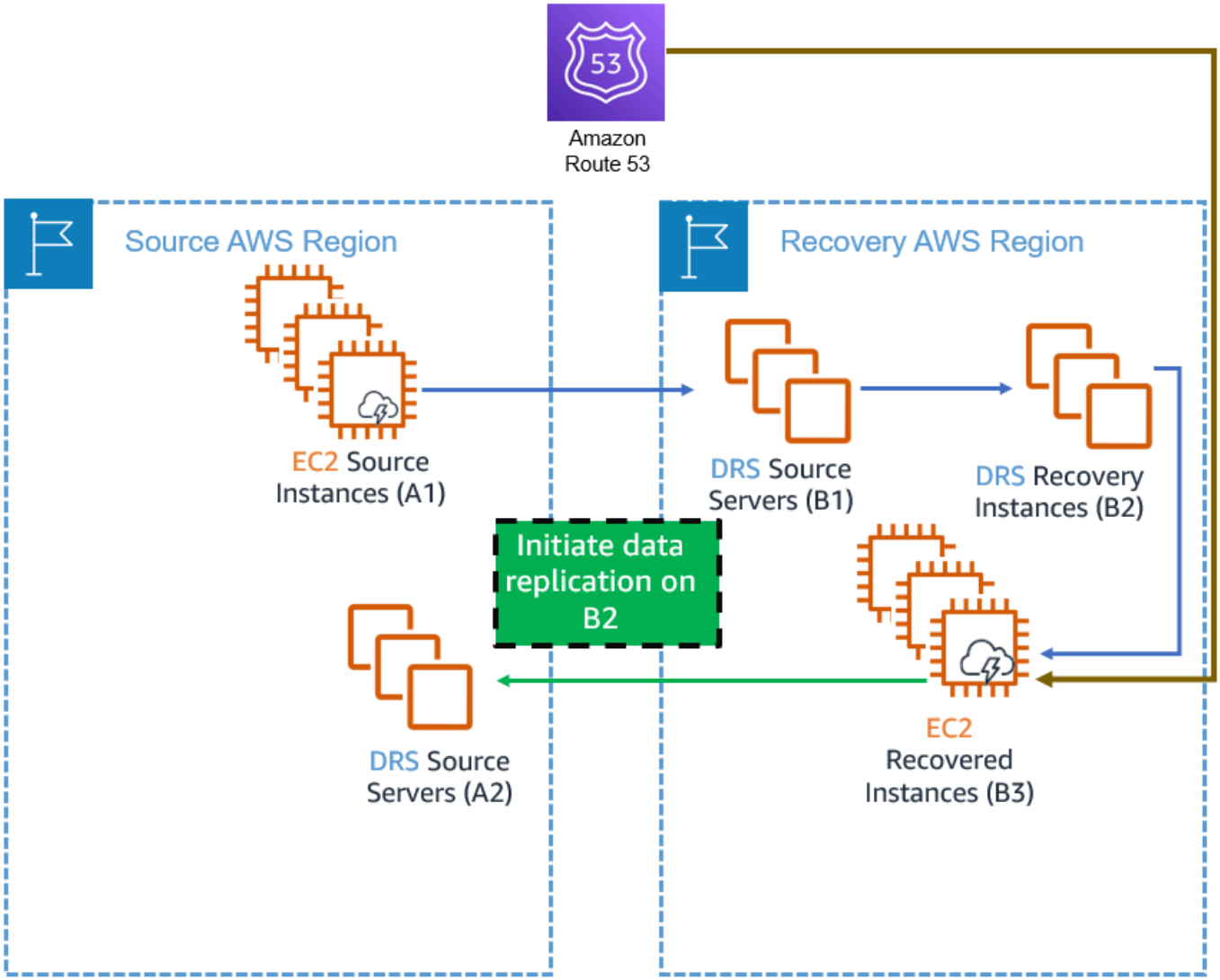
Important

- To ensure operational continuity, initialize the AWS DRS in advance in both the source and target AWS Regions, and conduct regular failover and failback drills.
- Before starting a failback, make sure the EC2 recovered instances (B3) have a network interface while meeting the specified network requirements.
- Access to EC2 instance metadata is required. If you have a custom network setup that modifies the operating system route, ensure that access to metadata is intact. Learn how to verify metadata access for Linux and for Windows.
- EC2 Instances that have failed over must resolve via DNS the regional DRS endpoint of the failback region. The resolved endpoint must be accessible from the EC2 Instance via TCP 443.

Performing cross-region failback

1. **Start reversed replication.**
   a. Go to the recovery AWS Region (in this example, us-east-2).
   b. Choose the **AWS Elastic Disaster Recovery** service.
   c. Navigate to the **Recovery instances** page.
   d. Select the servers that you want to protect and click **Start reversed replication**.

   e. A Source server (A2) will be created in the source region, as shown in the following diagram.



Note

All server data is transferred over the wire during this step. This process could take some time and will result in cross-Region data transfer costs. Moreover, starting reversed replication creates additional replication resources (A2). To avoid double billing, you can stop replicating the source instances (A1) by navigating to the AWS DRS source server in the recovery region (B1) and clicking **Stop replication** in the replication drop-down menu. Make sure that you only stop the replication after validating the failover instances because once replication is stopped, all previous points in time are deleted.

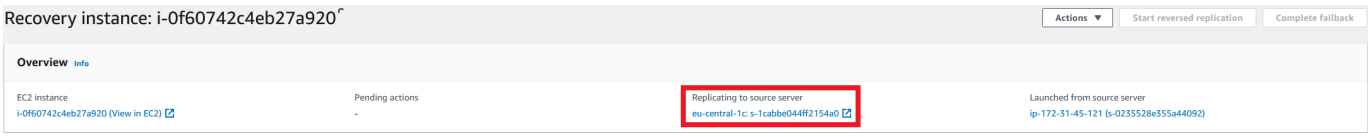
> **Important**
>
> Once replication is stopped, all previous points in time are deleted. This is done to minimize costs.

2. **Launch, validate, and redirect traffic.**

   After the **Reversed direction launch state** is marked as **Ready**, take the following steps to complete the failback:

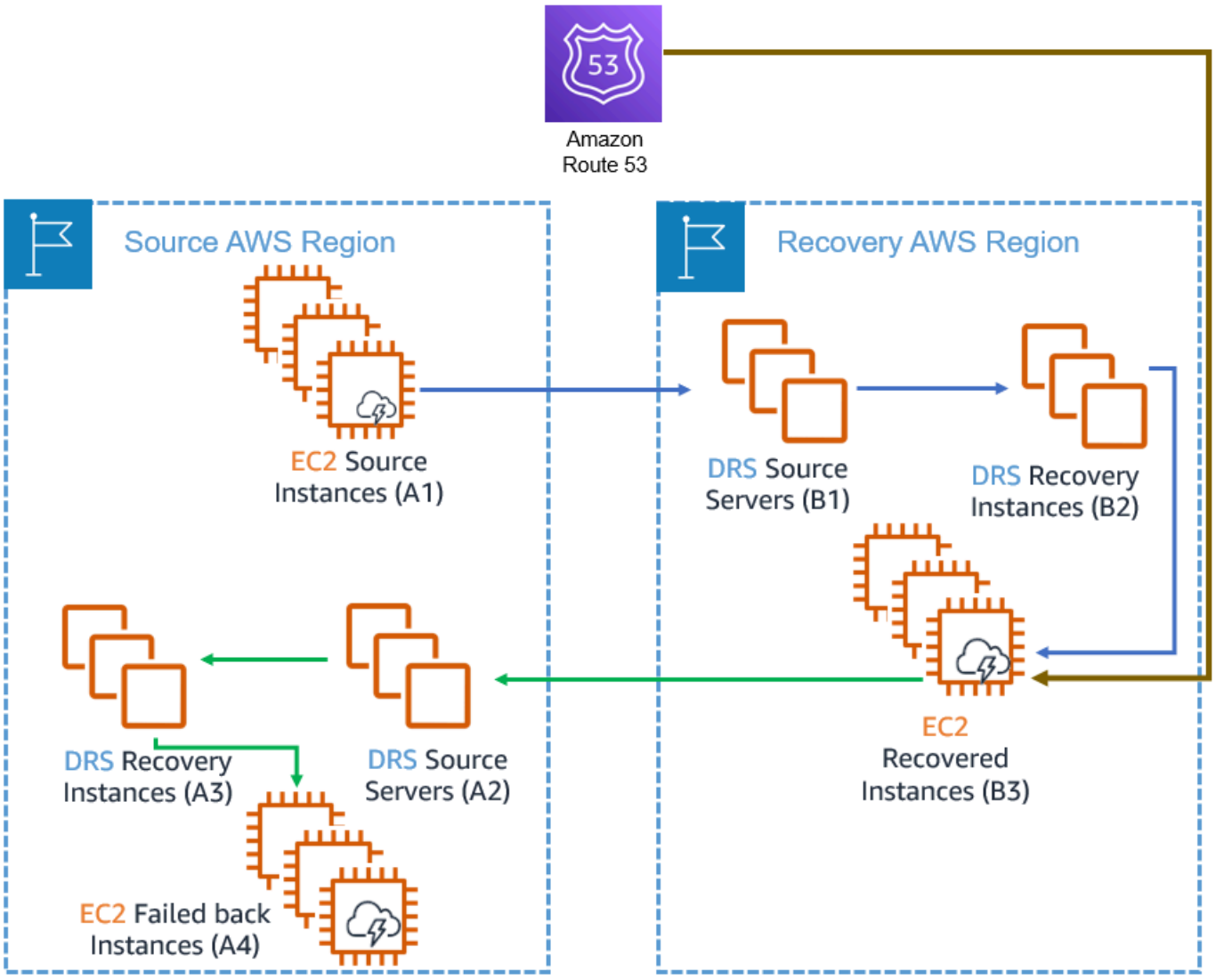
   a. Find the relevant source servers (A2) in the source region by clicking the **Replicating to source server** link in the recovery instance (B2).

   > **Note**
   >
   > You can also find it directly on the **Source servers** page in AWS DRS console at the source region.



   b. If the state is **Ready** (or **Ready with lag**), click **Launch for failback** under **Initiate recovery job**.

   > **Important**
   >
   > Make sure that your applications (A4) are working as expected. If you run into any issues, you can relaunch the instances and try again. Until you opt to failback, your recovery instances (B3) will continue to run in your recovery AWS Region to ensure business continuity.



c. Redirect traffic to failed back instances (A4), which will now become your new primary instances. Traffic redirection is not conducted using DRS. Choose a service according to your preferences (consider using Amazon Route 53).

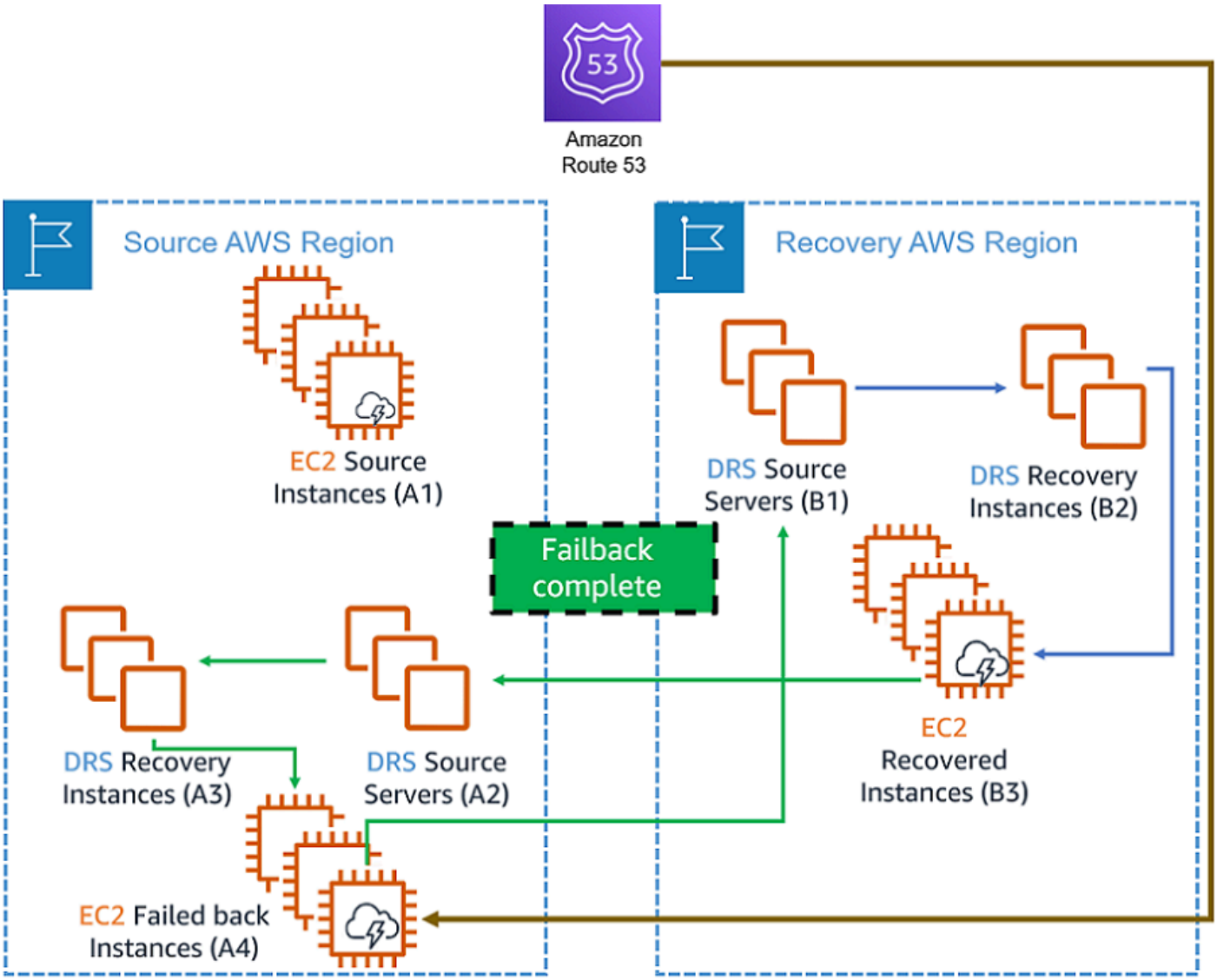
3. **Protect your new failed back instances.**

> **Important**
>
> Do not perform this step when performing a drill. This step replaces the instances that AWS DRS replicates (from the Source instances, A1, to the failed back instances, A4). In a drill, the source instances (A1) are still your production environment.

The newly launched failed-back instances (A4) are not protected. In order to protect them, follow these steps:

a. Navigate to the recovery instance (A3) in the source region.

b. Click **Start reversed replication**. This step will replace the Instances that the Source Server (B1) protects (A4 instead of A1).



4. **Clean your environment.**

After the failover to failback cycle is complete, you may be left with multiple AWS resources that you no longer need and that are costly to maintain. These include the source and failover EC2 instances (A1,B3), the recovery instances (B2, A3), and the Source servers (A2). Consider removing them.

Cleanup steps:

a. **Stop replication on the source servers (A2) of the source region.**

Navigate to the source server in the source region (A2), and click on **Stop replication** under the **Replication menu**. This step is required before terminating the recovery instance (B2).

b. **Terminate the recovery instances (B2).**

These instances, launched in your recovery AWS Region, are no longer needed now that you have launched new primary instances in your original source AWS Region. To terminate these instances, navigate to the AWS DRS Console in your recovery AWS Region (B2). After termination, those instances will no longer appear in the **Recovery Instances** page of the DRS Console. This process also terminates the recovered EC2 instances (B3).

c. **Terminate the source region EC2 instances (A1).**

These have now been replaced by the new instances launched in step 2 above (EC2 failed back instances, A3). You might have stopped these instances after the failover, and you can now terminate them using the AWS EC2 Console.

d. **Remove the recovery instance (A3) in the source region.**

Navigate to the **Recovery instances** in the AWS DRS console. Select the relevant recovery instance and click **Delete server** under the **Action** drop-down menu.

> **Note**
>
> If you have started reversed replication for the recovery instance (A3), you will not be able to disconnect it. To remove the recovery instances (A3) in the source region, simply delete the server. This will ensure that the newly launched failed-back instances (A4) remains protected.

e. **Remove the source servers (A2) in the source region.**

Navigate to the **Source servers** in the AWS DRS console. Select the relevant source server and select **Disconnect from AWS** under the **Actions** drop-down menu. Then, select **Delete server** under the same **Actions** menu.

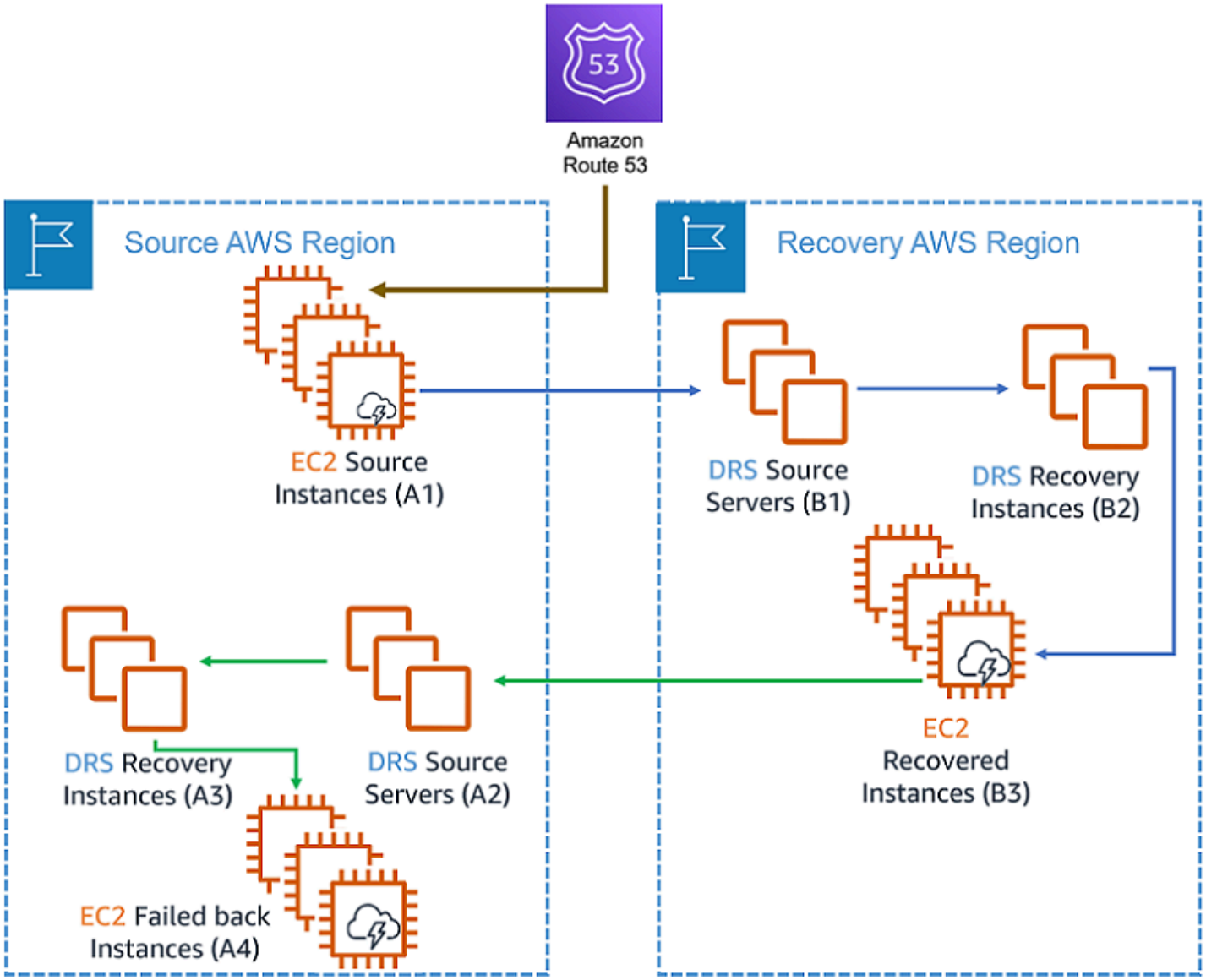
Performing a drill

To conduct a drill, follow the steps 1 and 2 as described above, and then perform a different cleanup process as described below.

> **Note**
>
> 1. Do not to stop the source server (B1) in the recovery AWS region as recommended in the note of step 1-e.
> 2. Do not perform step 3, Protecting the failed back instances would affect your production data.

Cleaning up after a drill

After a successful drill your AWS environment should look like this:



The only two AWS resources that need to remain are your actual production environment (A1) and its replication backup (B1). Since DRS protects replication servers, you must stop the replication first.

1. Stop the replication of the Source servers (A2) in the Source region.

> **Important**
>
> Make sure you don't stop replicating the Source servers (B1) in the recovery region.

2. Terminate the recovery instances (A3) in the source region and the recovery instances (B2) in the recovery region. As a result of this action, both the recovered instances (B3) and the failback instances (A4) are terminated as well.

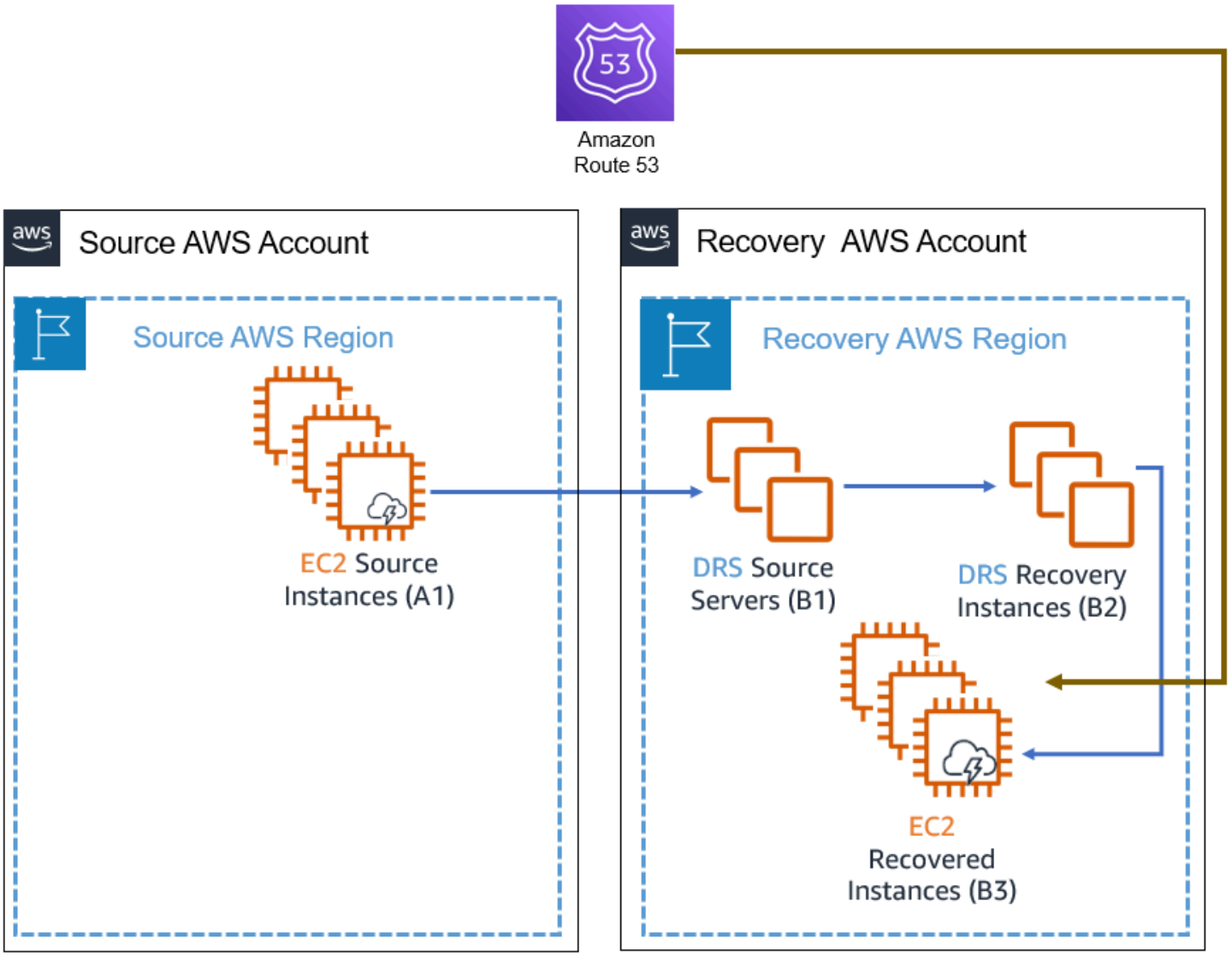
> **Note**
>
> Performing cross-region replication, failover and failback accrues additional costs, not detailed in the AWS DRS pricing examples. These additional costs consist of cross-Region data transfer costs during initial data replication, ongoing data replication, and failback replication; as well as the cost of replication resources (such as Amazon EBS volumes, snapshots, and more), used for failback replication; and also the DRS hourly billing for failback source servers.

Performing a cross-account failback

AWS Elastic Disaster Recovery (AWS DRS) allows you to perform failover and failback your EC2-based applications from one AWS account to another AWS account. The failover process is the same as failing over into an AWS account from a source outside of AWS, but the failback process is different. The instructions below describe the complete cross-account failover and failback process.

Overview and prerequisites

The failback process starts after the failover process ends. During failover, AWS DRS allows you to replace the EC2 source instance (A1) with the EC2 recovered instance (B3). The current AWS resource state is illustrated in the following diagram:



After performing a recovery, your applications are running on EC2 instances in the recovery account and region. However, these recovered instances (marked B3 in the diagram above) are not protected against other potential outages. In order to avoid data loss, you should start a reversed replication immediately. Starting reversed replication is only possible if the service is initialized in the recovery account and region. See initialize the AWS DRS.

Starting reversed replication involves copying the data from the EC2 recovered instances (B3) to the original account and region, an operation that takes time and possibly incurs cross-Region data transfer costs if the source region differs from the recovery region.

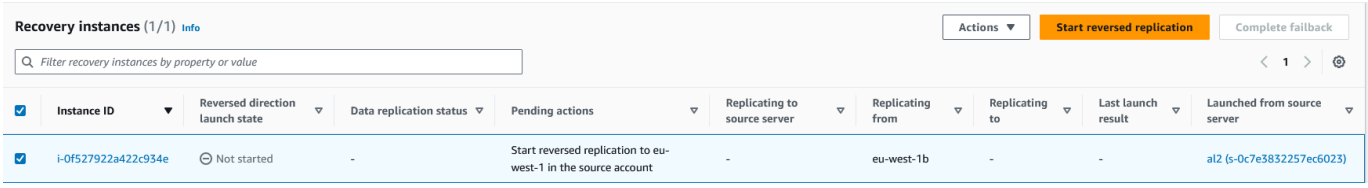
Once replication has reached a healthy state, failing back to the source account (after starting reversed replication) is possible using the DRS console on the source account and region, assuming DRS has been initialized in the source account and region.

Important

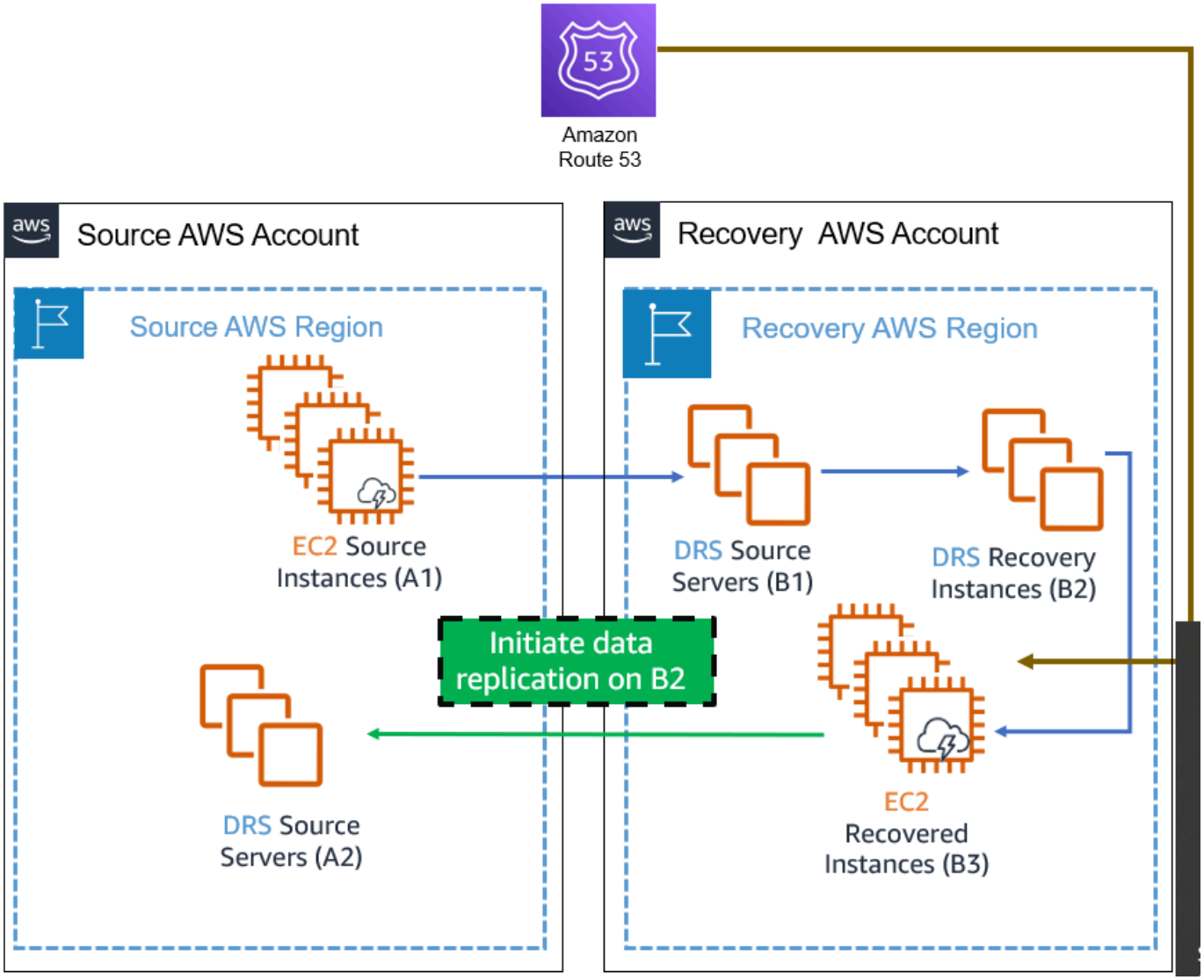
- To ensure operational continuity, initialize the AWS DRS in advance in both the source and target AWS accounts and regions, and conduct regular failover and failback drills.
- Create the roles, identified as **Failback and in-AWS right-sizing roles** via Trusted Account page in advance, for both directions: from source account to recovery account and from recovery account to source account.
- Before starting a failback, make sure the EC2 recovered instances (B3) have a network interface while meeting the specified network requirements.
- Access to EC2 instance metadata is required. If you have a custom network setup that modifies the operating system route, ensure that access to metadata is intact. Learn how to verify metadata access for Linux and for Windows.

Performing cross-account failback

1. **Start reversed replication.**
   a. Log in to the recovery account and select the recovery region (the account and region where the recovery instances were launched in).
   b. Open the **AWS Elastic Disaster Recovery** service console.
   c. Navigate to the **Recovery instances** page.
   d. Select the servers that you want to protect and click **Start reversed replication**.



   e. A Source server (A2) will be created in the source account and region, as shown in the following diagram.



Note

All server data is transferred over the wire during this step. This process could take some time and possibly result in cross-Region data transfer costs if the source region differs from the recovery region. Moreover, starting reversed replication creates additional replication resources (A2). To avoid double billing, you can stop replicating the source instances (A1) by navigating to the AWS DRS source server in the recovery account and region (B1) and clicking **Stop replication** in the replication drop-down menu. Make sure that you only stop the replication after validating the recovery instances because once replication is stopped, all previous points in time are deleted.

> **Important**
>
> Once replication is stopped, all previous points in time are deleted. This is done to minimize costs.

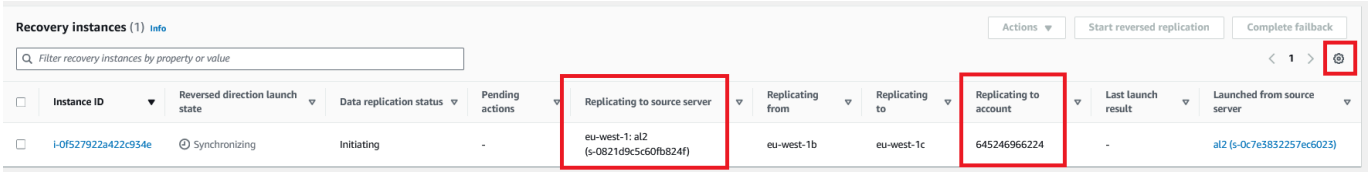
2. **Launch, validate, and redirect traffic.**

   After the **Reversed direction launch state** is marked as **Ready**, take the following steps to complete the failback:

   a. Find the relevant source servers (A2) in the source account and region by information in the **Replicating to source server** and **Replicating to account** columns of the recovery instance (B2)

> **Note**
>
> You can also find it directly on the **Source servers** page in AWS DRS console at the source account and region.

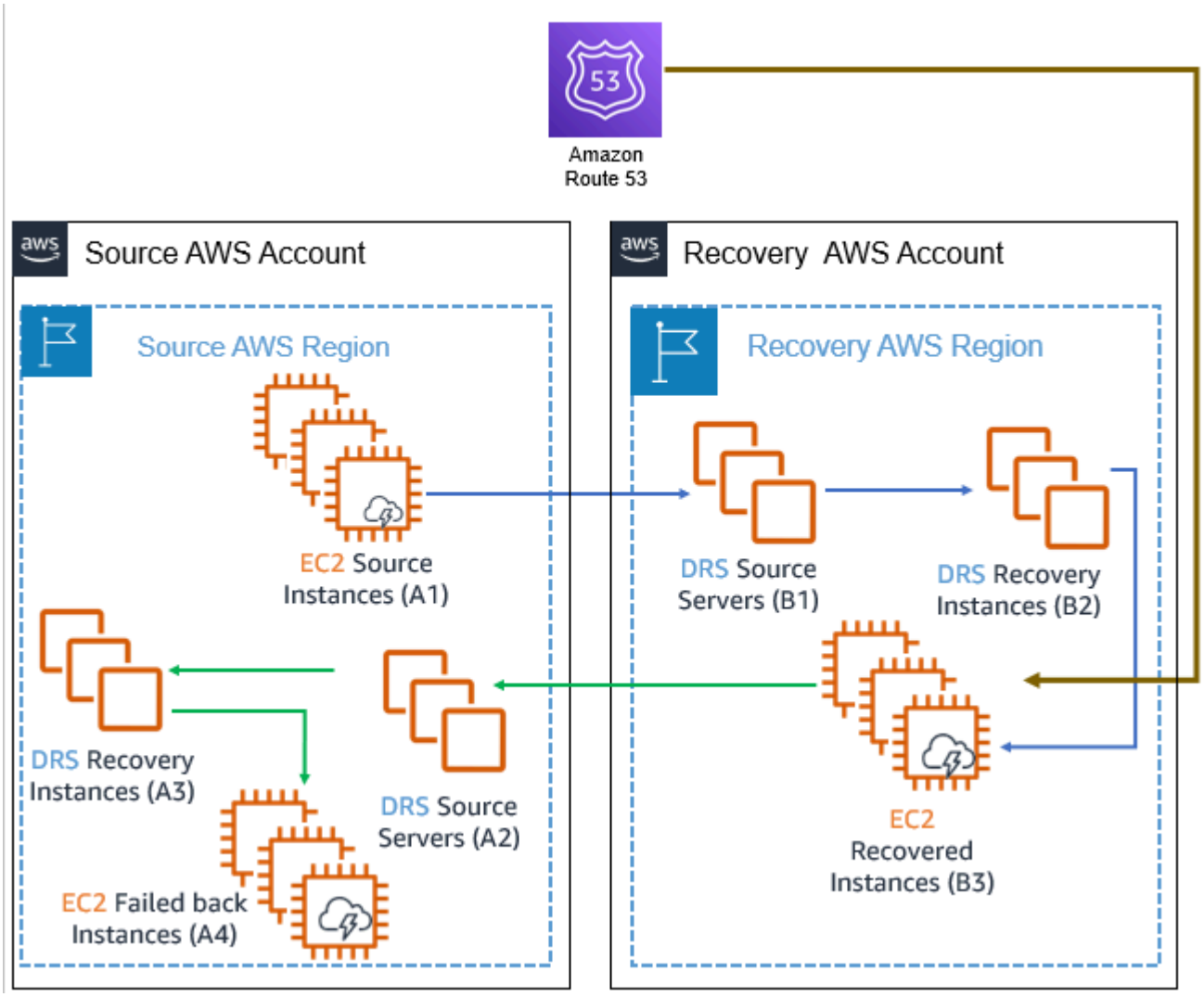
> **Note**
>
> Column **Replicating to account** is not visible by default and can be visible by toggling of the column in preferences of Recovery instances page



   b. If the state is **Ready** (or **Ready with lag**), click **Launch for failback** under **Initiate recovery job**.

> **Important**
>
> Make sure that your applications (A4) are working as expected. If you run into any issues, you can relaunch the instances and try again. Until you opt to failback, your

recovery instances (B3) will continue to run in your recovery account and region to ensure business continuity.



c. Redirect traffic to failed back instances (A4), which will now become your new primary instances. Traffic redirection is not conducted using DRS -> You need to perform traffic redirection either using your systems, or by utilizing a custom post-launch action. Choose a service according to your preferences (consider using Amazon Route 53).

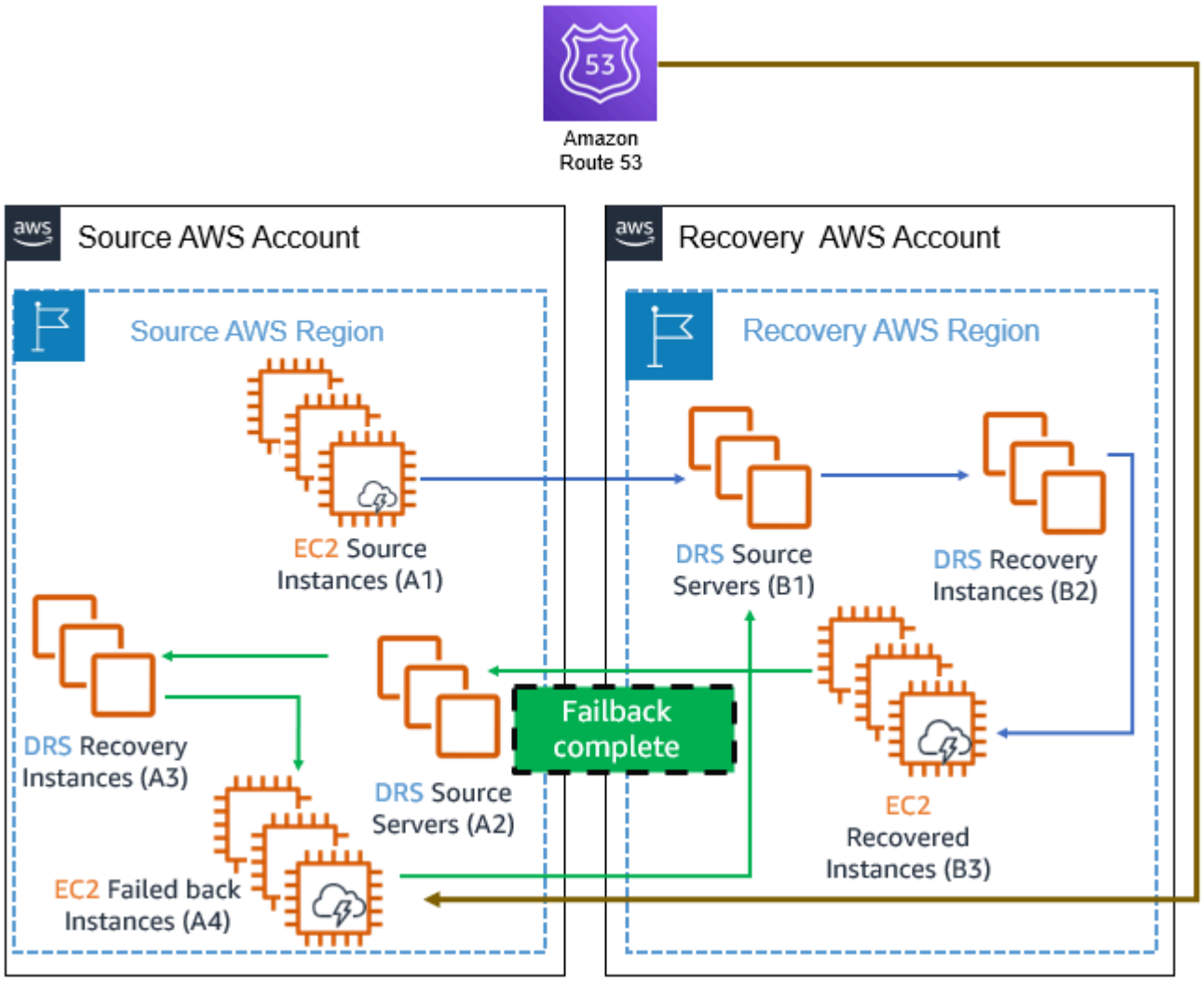
3. **Protect your new failed back instances.**

> **Important**
>
> Do not perform this step when performing a drill. This step replaces the instances that AWS DRS replicates (from the Source instances, A1, to the failed back instances, A4). In a drill, the source instances (A1) are still your production environment.

The newly launched failed-back instances (A4) are not protected. In order to protect them, follow these steps:

a. Navigate to the recovery instance (A3) in the source account and region.

b. Click **Start reversed replication**. This step will replace the Instances that the Source Server (B1) protects (A4 instead of A1).



4. **Clean your environment.**

After the failover to failback cycle is complete, you may be left with multiple AWS resources that you no longer need and that are costly to maintain. These include the source and failover EC2 instances (A1,B3), the recovery instances (B2, A3), and the Source servers (A2). Consider removing them.

Cleanup steps:

a. **Stop replication on the source servers (A2) of the source account and region.**

Navigate to the source server in the source account and region (A2), and click on **Stop replication** under the **Replication menu**. This step is required before terminating the recovery instance (B2).

b. **Terminate the recovery instances (B2).**

These instances, launched in your recovery account and region, are no longer needed now that you have launched new primary instances in your original source account and region. To terminate these instances, navigate to the AWS DRS Console in your recovery account and region (B2). After termination, those instances will no longer appear in the **Recovery Instances** page of the DRS Console. This process also terminates the recovered EC2 instances (B3).

c. **Terminate the EC2 instances (A1) on the source account and region.**

These have now been replaced by the new instances launched in step 2 above (EC2 failed back instances, A3). You might have stopped these instances after the failover, and you can now terminate them using the AWS EC2 Console.

d. **Remove the recovery instance (A3) in the source account and region.**

Navigate to the **Recovery instances** in the AWS DRS console. Select the relevant recovery instance and click **Delete server** under the **Action** drop-down menu.

> **Note**
>
> If you have started reversed replication for the recovery instance (A3), you will not be able to disconnect it. To remove the recovery instances (A3) in the source account and region, simply delete the server. This will ensure that the newly launched failed-back instances (A4) remains protected.

e. **Remove the source servers (A2) in the source account and region**

Navigate to the **Source servers** in the AWS DRS console. Select the relevant source server and select **Disconnect from AWS** under the **Actions** drop-down menu. Then, select **Delete server** under the same **Actions** menu.

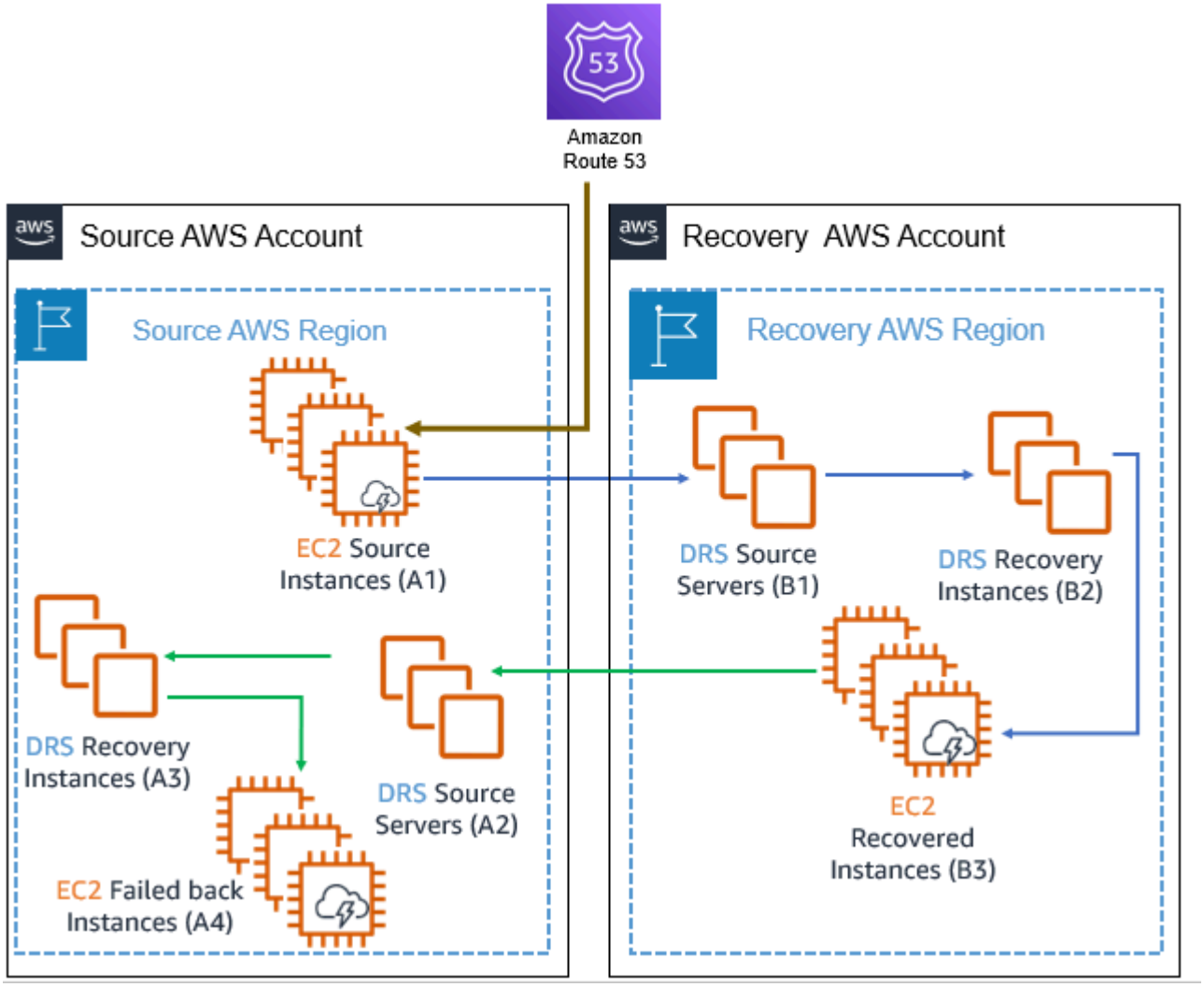
Performing a drill

To conduct a drill, follow the steps 1 and 2 as described above, and then perform a different cleanup process as described below.

> **Note**
>
> 1. Do not to stop the source server (B1) in the recovery account and region as recommended in the note of step 1-e.
> 2. Do not perform step 3, Protecting the failed back instances would affect your production data.

Cleaning up after a drill

After a successful drill your AWS environment should look like this:



The only two AWS resources that need to remain are your actual production environment (A1) and its replication backup (B1). Since DRS protects replication servers, you must stop the replication first.

1. Stop the replication of the Source servers (A2) in the source account and region.

> **Important**
>
> Make sure you don't stop replicating the Source servers (B1) in the recovery account and region.

2. Terminate the recovery instances (A3) in the source account and region and the recovery instances (B2) in the recovery account and region. As a result of this action, both the recovered instances (B3) and the failback instances (A4) are terminated as well.

Note

Performing cross-account replication, failover and failback accrues additional costs, not detailed in the AWS DRS pricing examples. These additional costs consist of cross-Region data transfer costs during initial data replication, ongoing data replication, and failback replication if the source region differs from the recovery region; as well as the cost of replication resources (such as Amazon EBS volumes, snapshots, and more), used for failback replication; and also the DRS hourly billing for failback source servers.

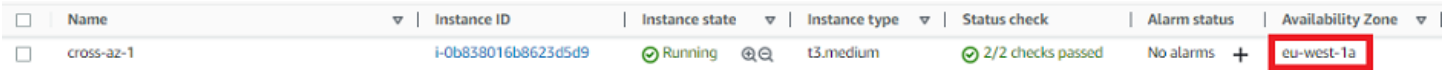
Cross-Availability-Zone recovery

You can use DRS to replicate and recover EC2 instances across Availability Zones.

Cross Availability Zone (AZ) setup

Initial settings

In order to replicate an EC2 instance across availability zones, the replication settings and launch settings should be set to replicate into an availability zone different from the one hosting your protected EC2 instance. To find out which availability zone hosts an instance, visit the AWS EC2 console.



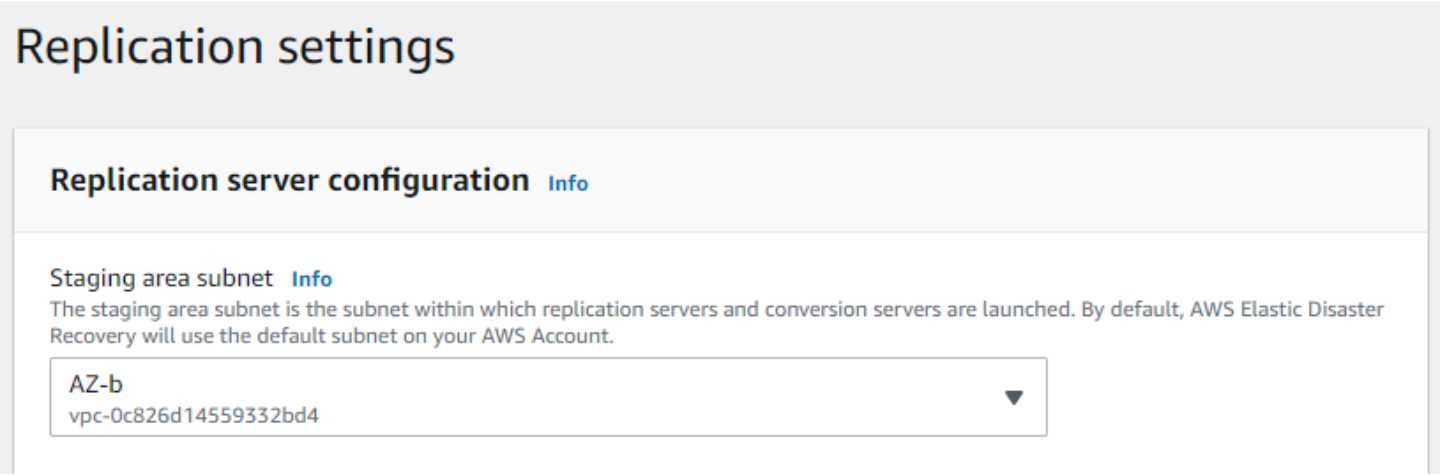
Configure the replication settings and launch template to use a subnet hosted on an availability zone different from the one hosting the EC2 instance being protected.

Example

If the protected EC2 is hosted on availability zone eu-west-1a, as shown in the screen above, the replication settings subnet (and launch template subnet) will be hosted on another availability zone in the same region, for example, eu-west-1b.

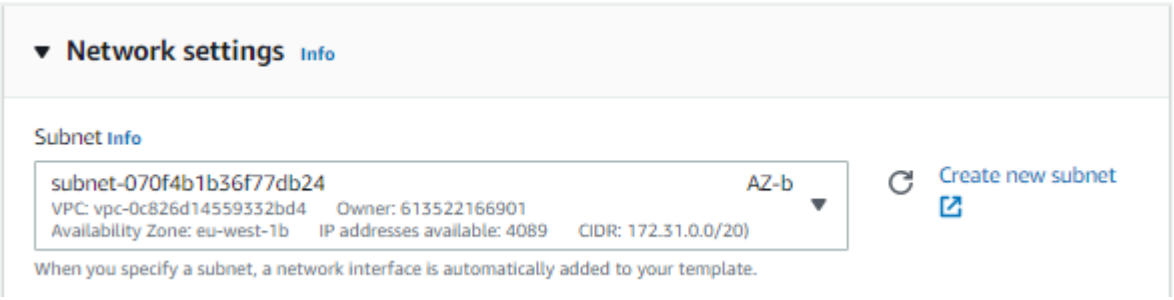
Selecting a subnet for replication is done from the replication settings page for the source server. Information about each subnet, including which availability zone hosts it, can be found on the Amazon VPC console.

Replication settings



Launch settings

Learn how to modify the launch template.



Launching a Recovery Instance

To recover the protected EC2 instance, follow these instructions.

Protecting your Recovered Instance

Once a recovery instance has been successfully launched inside a target availability zone and failed over, this recovery instance should be protected by DRS.

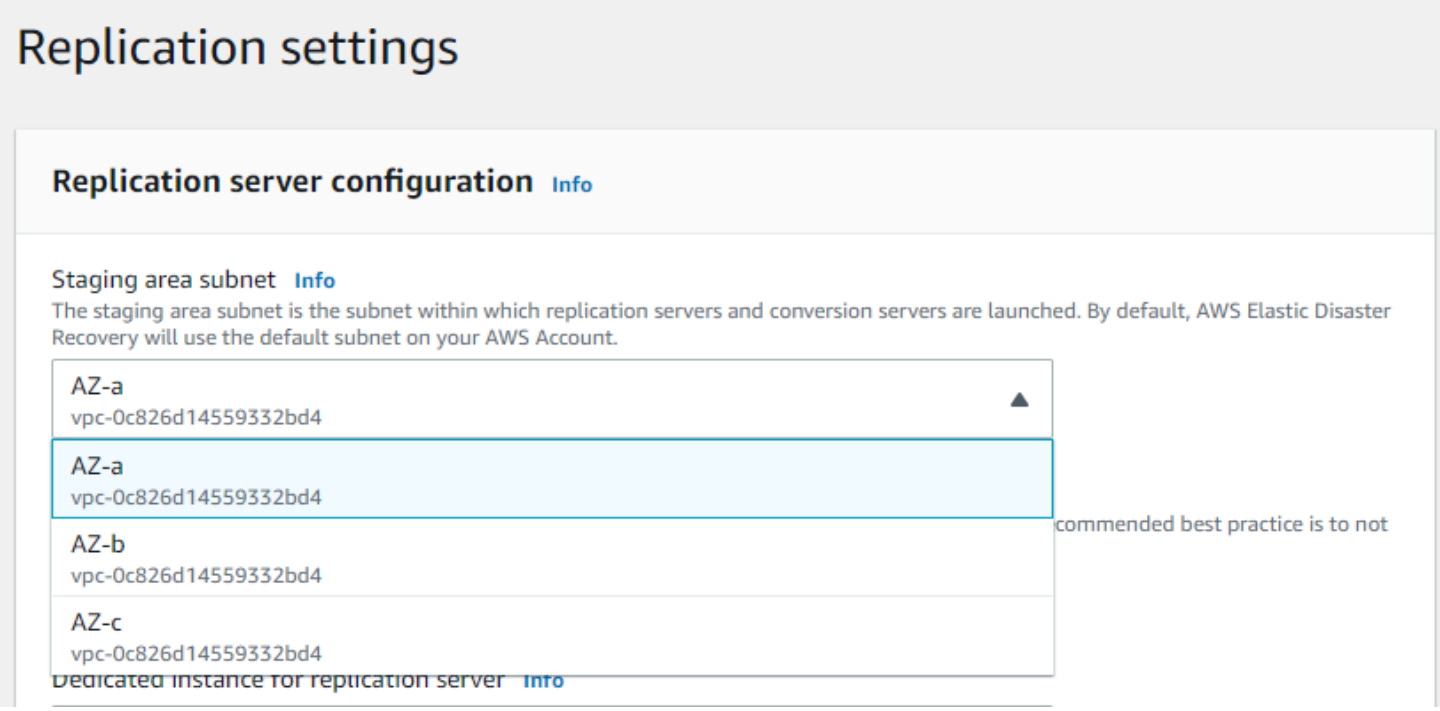
To protect this recovery instance:

- Replication settings and launch template subnets should be changed to a subnet hosted on an availability zone different from the one hosting the EC2 instance that is associated with the recovery instance.
- You must start the replication from the new Recovery EC2 Instance instead of the original EC2 instance.

Example

If a recovery instance was created and the underlying EC2 instance is hosted on availability zone "eu-west-1**b**", the replication settings and launch template can be modified to use a subnet hosted on availability zone "eu-west-1**a**".

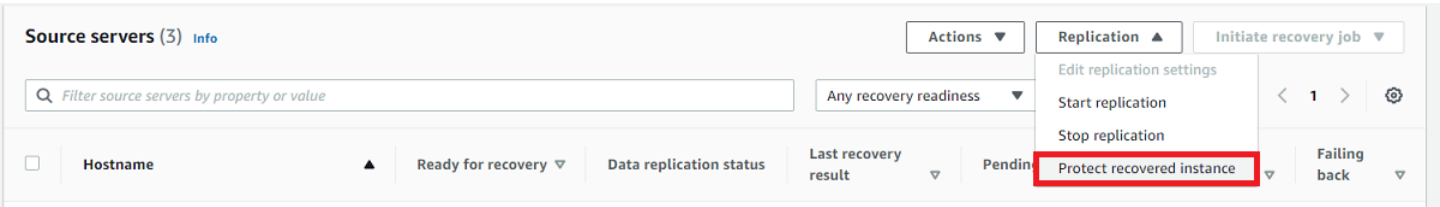
Modify the replication settings to replicate to the original availability zone.



Modify the launch settings to the original availability zone.

In order to modify the launch template follow these instructions.

Protect your recovered instance.



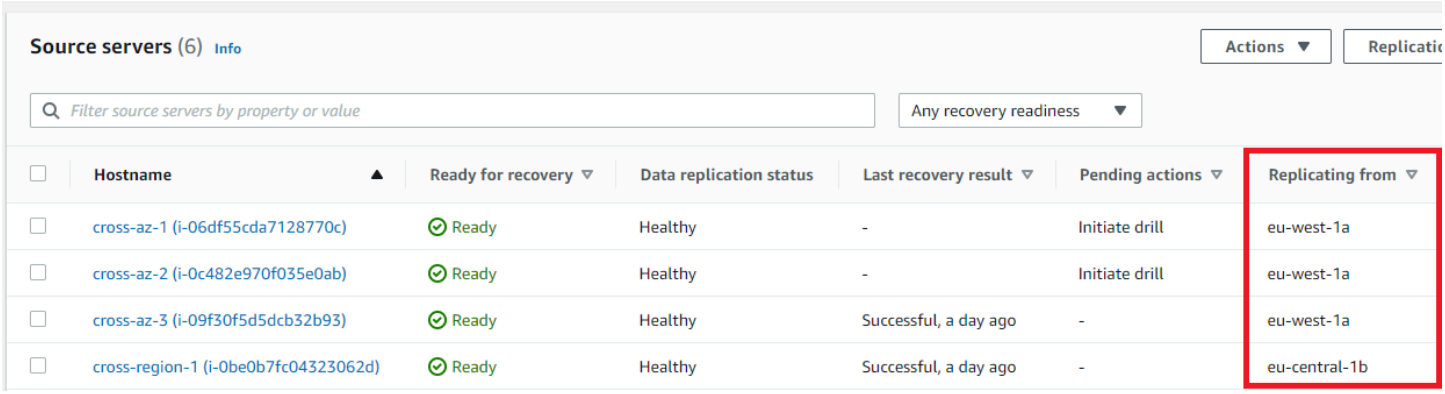
Protecting your recovered instance also stops the replication of the original EC2 instance. For example, if the original EC2 instance is hosted in availability zone "eu-west-1a" and is recovered to a subnet hosted in availability zone eu-west-1b, starting the replication on the recovered instance back to eu-west-1a also stops the replication of the original instance hosted in eu-west-1a.

Starting the replication for a recovered instance only initiates a rescan (to apply the new instance's changes on the last snapshot) instead of a full synchronization. The reason is that all the replication resources associated with the original instance, such as point in time snapshots, configuration, and job logs are retained. After the replication has started, there is no need to keep the original instance for replication purposes.

The availability zone hosting the EC2 instance that is being protected can be viewed on the **Source servers** list (**Replicating from** column).

> **Note**
>
> One of the major benefits of cross AZ replication is that the replication agent only needs to rescan the differences between the latest point in time snapshot and the current source server data. This saves both time and resources. All points-in-time snapshots, configuration, and job logs will be retained. You can now terminate the original EC2 instance in eu-west-**1a**. Your recovered instances are now protected.

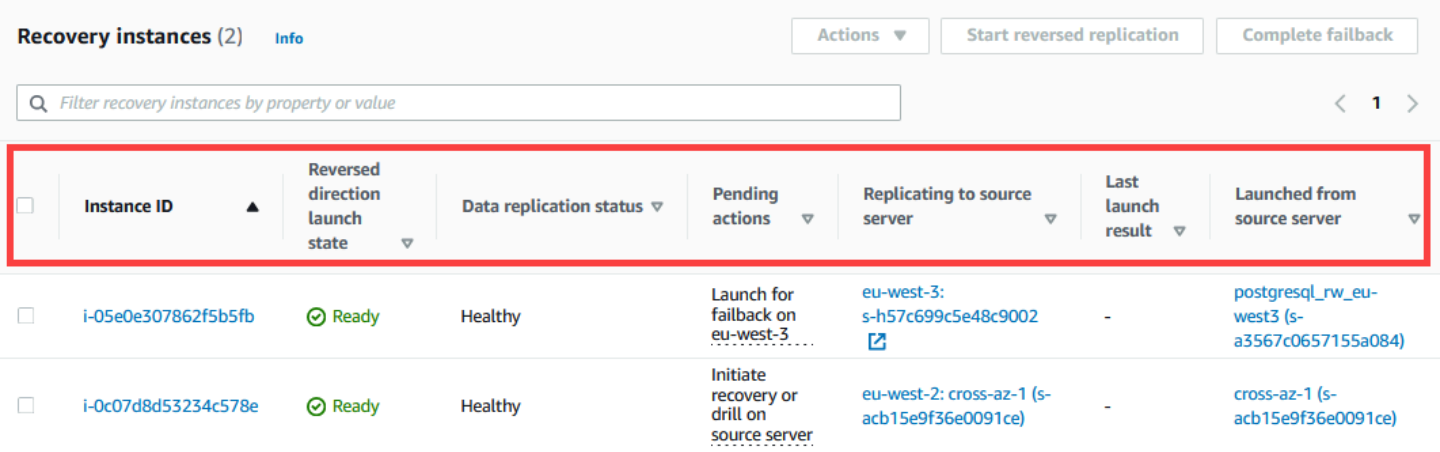
You can view the source environment availability zone from the **Source servers** list.



Recovery Instances page

Recovery instances overview

You can manage your recovery instances on the **Recovery instances** page.



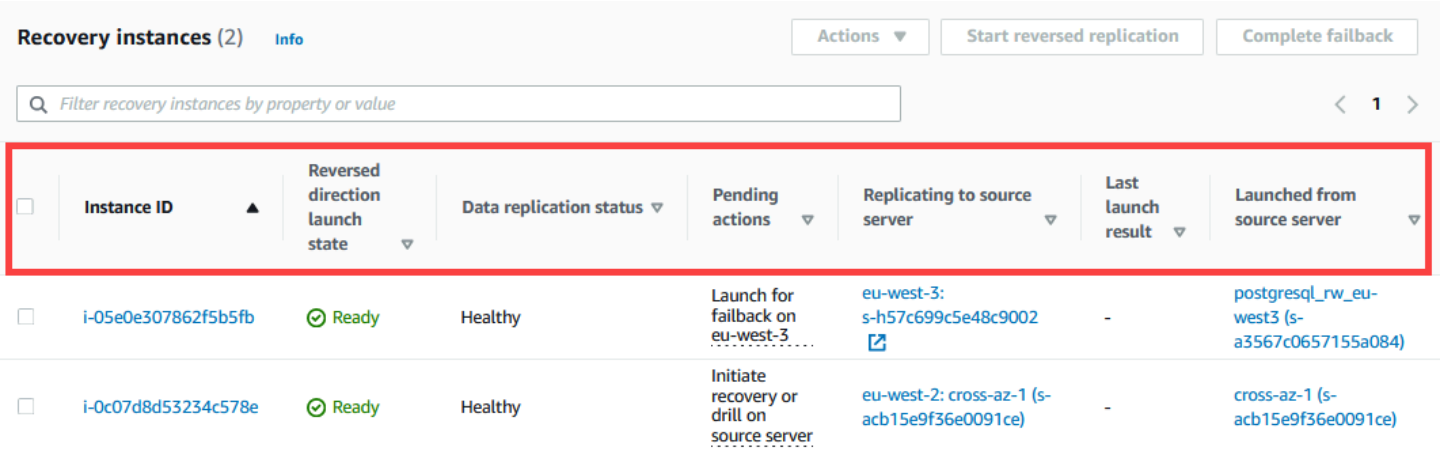
This page displays all of the recovery instances that you have launched in AWS for your source servers, as well as recovery instances that you have added to directly to Elastic Disaster Recovery.

It allows you to monitor the data replication status of your recovery instances, view recovery instance details, start reversed replication, edit recovery instance failback settings for on-premises failback, view post-launch actions run results and terminate recovery instances.

Monitoring recovery instances

You can fully monitor your recovery instances on the **Recovery instances** page.

The page shows all of your recovery instances and sorts them by **Instance ID**, **Reversed direction launch state**, **Data replication status**, **Pending actions**, **Replicating to source server**, **Last launch result**, and **Launched from source server.**



You can sort your recovery instances alphabetically in descending or ascending order by choosing the arrow next to the various category headers (with the exception of data replication status).

You can filter the recovery instances page by a variety of properties within the **Filter by property of value** box.

Recovery instance categories

The following is a breakdown of each category header:

Instance ID

The **Instance ID** category shows the ID of the recovery instance. Choose the specific Instance ID to open the recovery instance details view. Learn more about the recovery instance details view.

Reversed direction launch state

The **Reversed direction launch state** shows the current state of the Reversed direction launch for the recovery instance. Possible states include:

- **Not started** – Reversed replication has not been started for the recovery instance.
- **Synchronizing** – Reversed replication has been started for the recovery instance and is currently in process.
- **Ready** – Reversed replication has completed initiation and is not ready to be launched.
- **Completed** – Failback process to the on-premises server has been successfully completed. This value does not appear for in-AWS launch flows.

- **Error** – There was an error during the reversed replication process. You can learn more about the cause of the error in the **Data replication status** and **Pending actions** columns.

Data replication status

The **Data replication status** category shows the current data replication status of the recovery instance. Possible states include:

- **Not started** – Data replication has not been started for the recovery instance. This indicates that failback has not been started for the instance.
- **Initiating** – Data replication is initiating. This indicates that reversed replication has been initiated for the instance.
- **Initial sync** – The recovery instance is undergoing the initial sync process after reversed replication has been initiated. The Elastic Disaster Recovery Console will show the percentage completed and the time left.
- **Rescanning** – The recovery instance is undergoing a rescan. The AWS Elastic Disaster Recovery Console will show the percentage completed and the time left.
- **Healthy** – The data replication process has been completed and the recovery instance is ready for launch.
- **Lag** – The recovery instance is currently experiencing lag. Open the recovery instance details view to learn more.
- **Stalled** – The recovery instance is experiencing a stall. Open the **Recovery instance details** view to learn more.
- **Completed** – The failback process has been completed and as a result data replication has been successfully completed and stopped. This value is only relevant to on-premises failback and does not appear in in-AWS flows.
- **Disconnected** – The recovery instance has been disconnected from AWS Elastic Disaster Recovery. As a result, data replication has been stopped.

Pending actions

The **Pending actions** column provides additional details, when relevant, about the next actions that should be performed in order to progress the current flow or to initiate the reversed replication. Possible values include:

- **Launch for failback on {region}** – This status indicates that reversed replication has reached a healthy state. To launch for failback, click on the link under **replicating to source server**.
- **Use failback client** – To start the replication back to the on-premises server, use the Failback Client. This value is only relevant to on-premises failback.
- **Start reversed replication to {region name}** – Click **start reversed replication** to initiate reversed replication to the specified region. This value only applies to In-AWS and cross-region replications.

Replicating to source server

The **Replicating to source server** category identifies the source server to which the recovery instance is replicating. when you start reversed replication, it is managed through this source server. Launch operations are performed by navigating to this source server and initiating the operation from that screen.

The following are displayed in order: (region & ID)

- The source server region
- The source server's ID

Click on the source server links to be redirected to the source server details view of the source server that is associated with the specific recovery instance. Learn more about the server details view. Note that if the source server is located in another region (marked by an external icon), clicking the link will open the source server's details page in a different tab.

Last launch results

This category indicated the results of the last launch. Possible values include:

- Launch successful
- Failback successful
- Launch failed
- Failback failed

Launched from source server

The **Launched from source server** column identifies the source server from which the recovery instance was launched.

The following are displayed in order:

- The source server hostname
- The source server's ID

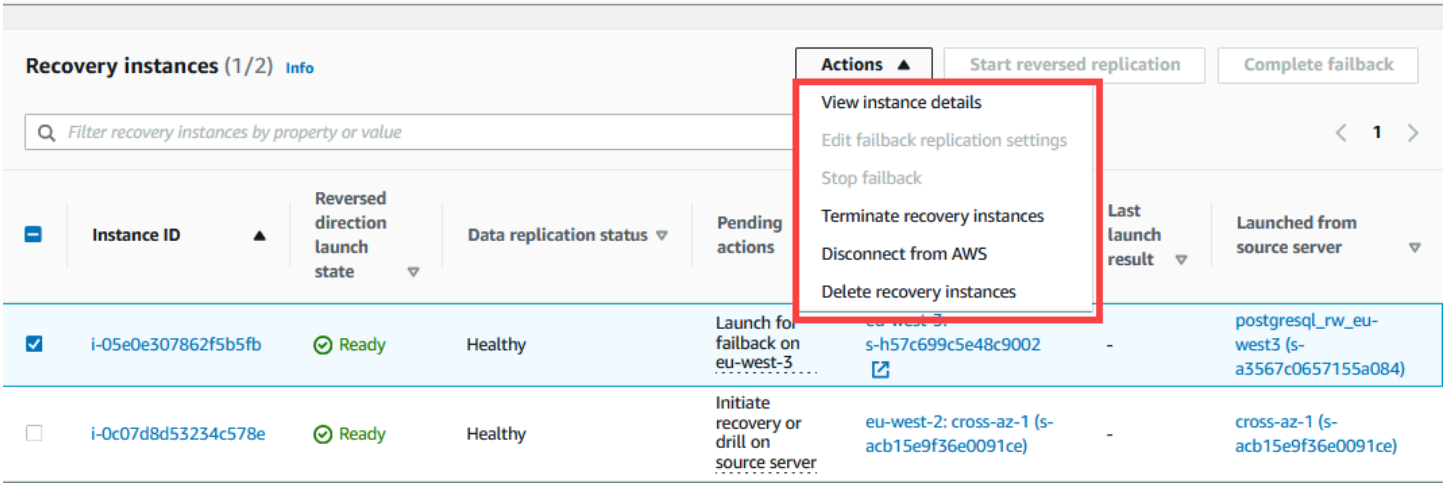
Click on the source server links to be redirected to the source server details view of the source server that is associated with the specific recovery instance. Learn more about the server details view.

Recovery instances actions

The recovery instances page allows you to perform a variety of actions, including viewing recovery instance details, adding recovery instances, editing the failback replication settings, terminating recovery instances, and continuing the failback process.

Actions menu

You can perform various actions from the **Actions** menu.



View instance details

Select the checkbox to the left of any recovery instance and choose the **View instance details** option under the **Actions** menu to open the **Recovery instance details** view. Learn more about the recovery instance details view.

Edit failback replication settings

Select the checkbox to the left of one or more recovery instances and choose the **Edit failback replication settings** option under the **Actions** menu to edit the failback replication settings the selected recovery instances. The failback replication settings configure the replication to the on-premises servers during an on-premises failback process. This does not apply to in-AWS replication, which is managed on the **replicating to source server** source servers. Learn more about Failback replication settings.

Stop failback

Select the checkbox to the left of one or more recovery instances which are in the **Synchronizing** state, and choose the **Stop**option under the **Actions** menu to stop the failback process for the selected recovery instance or instances. This will return the instances' **Reversed replication launch state** to **Not started** and will stop any ongoing failback process. The Failback client will indicate that the failback has been stopped. To restart failback, reboot the machine in the Failback Client. Note that the **Stop failback** state is only relevant to on-premises flows.

When the **Stop failback for recovery instances** dialog appears, click **Stop failback**.

Terminate recovery instances

Select the checkbox to the left of one or more recovery instances and choose the **Terminate recovery instances** option under the **Actions** menu to terminate the recovery instance or instances. This will remove all of resources associated with the selected recovery instance or instances from Elastic Disaster Recovery and will terminate all related EC2 resources. You should perform this action if you no longer need the recovery instance (after having successfully completed a launch, or if you decide that you no longer want to protect the paired source server).

When the **Terminate recovery instances** dialog appears, click **Terminate**.

Disconnect from AWS

Select the checkbox to the left of one or more recovery instances and choose the **Disconnect from AWS** option under the **Actions** menu to disconnect the recovery instance or instances from AWS. This will delete the AWS Replication Agent from the recovery instance or instances, but will keep the recovery instance Elastic Disaster Recovery resources and the EC2 resources intact. You may want to disconnect from AWS if you do not want to perform a launch for the specific recovery instance or instances and do not want to accrue additional costs for data replication, but do still want the recovery instance to appear in the Elastic Disaster Recovery Console.

When the **Disconnect X recovery instances from service** dialog appears, click **Disconnect**.

Delete recovery instances

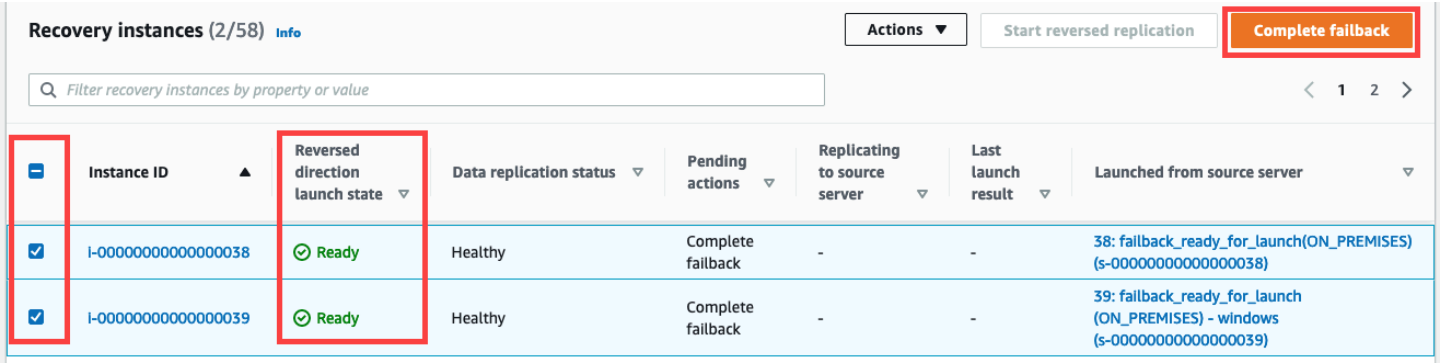
Select the checkbox to the left of one or more recovery instances and choose the **Delete recovery instances** option under the **Actions** menu to delete the recovery instance or instances. This will remove all of resources associated with the selected recovery instance or instances from Elastic Disaster Recovery but will not terminate all related EC2 resources and the instance will keep on running on Amazon EC2.

You may want to delete the recovery instance or instances if you already failed over into AWS, but have then decided to stay in AWS permanently instead of failing back to your original source servers and do not want to incur any more costs associated with Elastic Disaster Recovery resources. You may also want to delete the recovery instance or instances if you performed an in-AWS launch but do not want to start reversed replication back to the original region. Note that you can only delete recovery instances that have already been disconnected from AWS.

When the **Delete recovery instance** dialog appears, click **Delete**.

Failback

Select the checkbox to the left of one or more recovery instances that are in the **Ready** state and choose the **Complete failback** option to continue the failback process after performing a failback with the Elastic Disaster Recovery Failback Client. This action will stop data replication and will start the conversion process. This will finalize the failback process and will create a replica of each recovery instance on the corresponding source server.



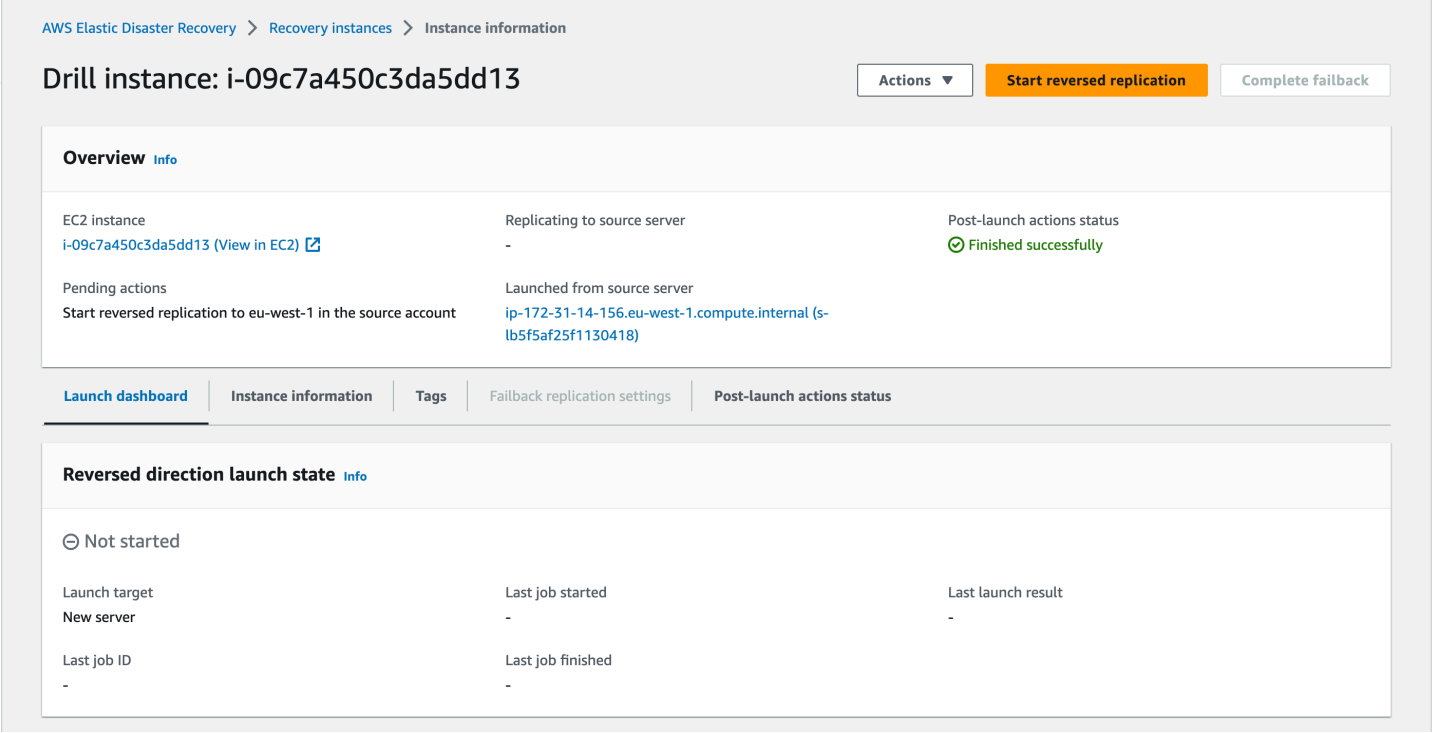
Important

Ensure that you complete the entire failback process with the Elastic Disaster Recovery Failback Client prior to choosing the Failback option.

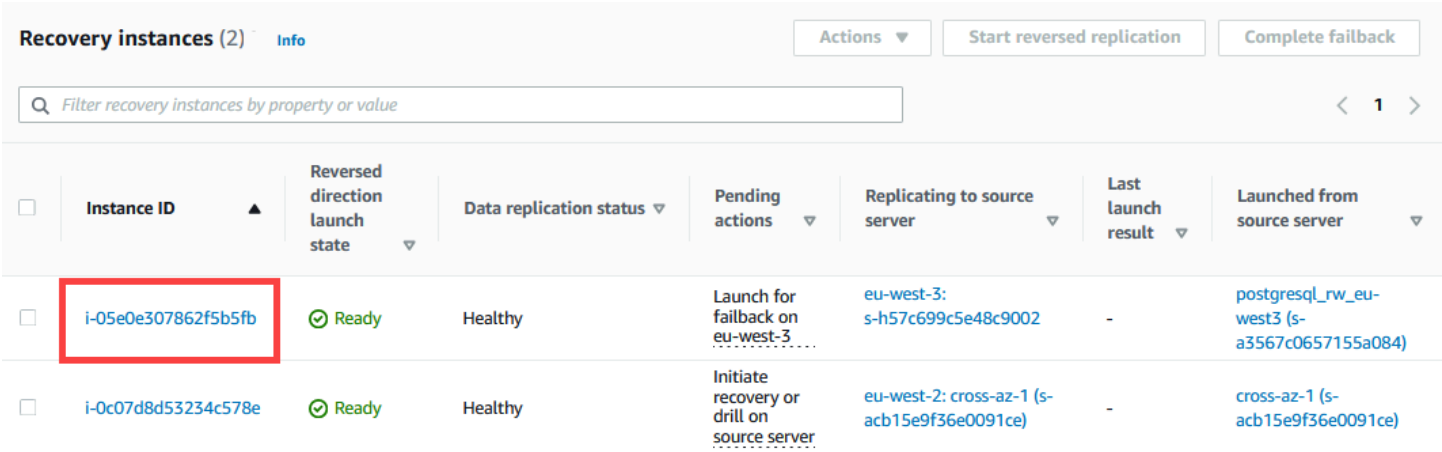
When the **Continue with failback for X instances** dialog appears, click **Failback.**

Recovery instance details view

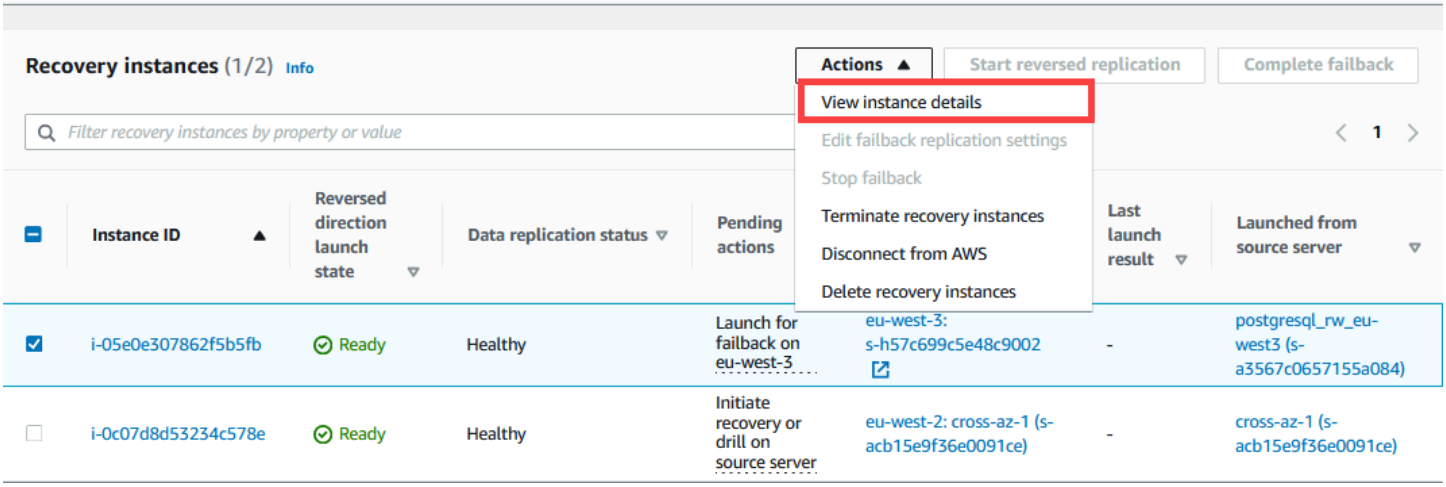
The recovery instance details view provides an in-depth overview of the recovery instance, including the instance's reversed direction launch process, post-launch action runs and data replication status while allowing you to control instance tags and the instance's failback settings.



You can access the recovery instance details view by choosing the instance ID of the recovery instance you wish to access under the **Instance ID** column.

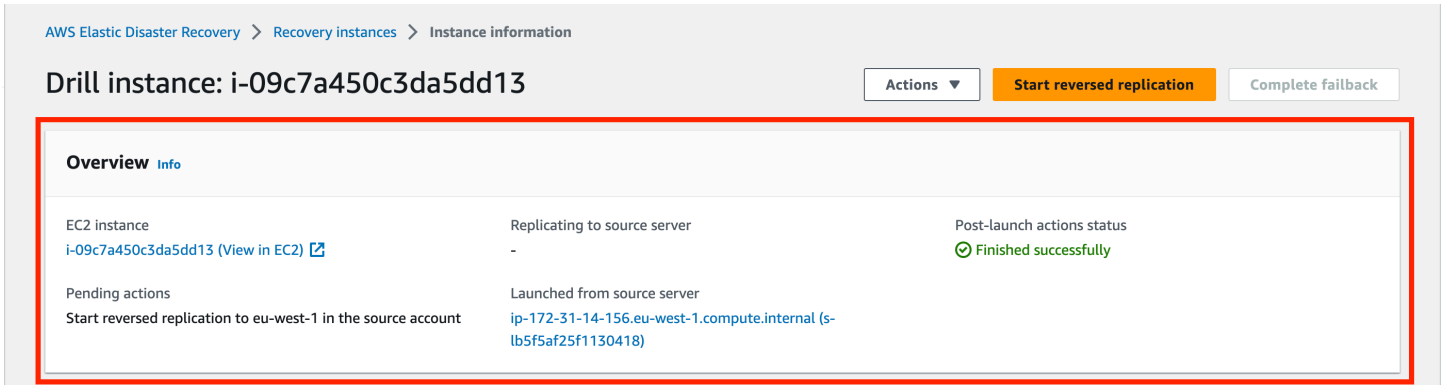


You can also access the recovery instance details view by selecting the checkbox to the left of the recovery instance and choosing the **View instance details** option under the **Actions** menu.



The recovery instance information page shows the **Instance ID** at the top.

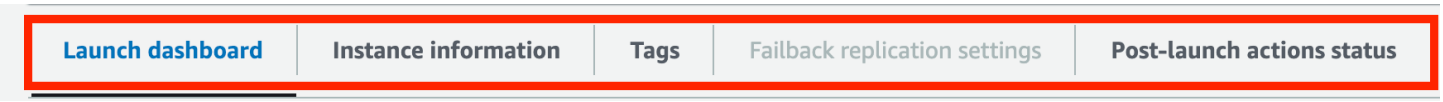
The **Overview** panel provides an overview of the failback process, including:



- **EC2 instance** – the ID of the recovery instance in EC2. Choose the **View in EC2** option to open the AWS EC2 Console.
- **Pending actions** – information derived from the **Pending actions** column (for example, **Launch for failback on {region}**).
- **Replicating to source server** – the source server to which the recovery instance is replicating. Choose the source server ID to open the **Source server details view** page for the specific source server.

- **Launched from source server** – the source server source server from which the recovery instance was launched. Choose the source server ID to open the **Source server details view** page for the specific source server.
- **Post-launch actions status** – shows the status of the last post-launch actions run on this instance.

The recovery instance details page is divided into the following sections:



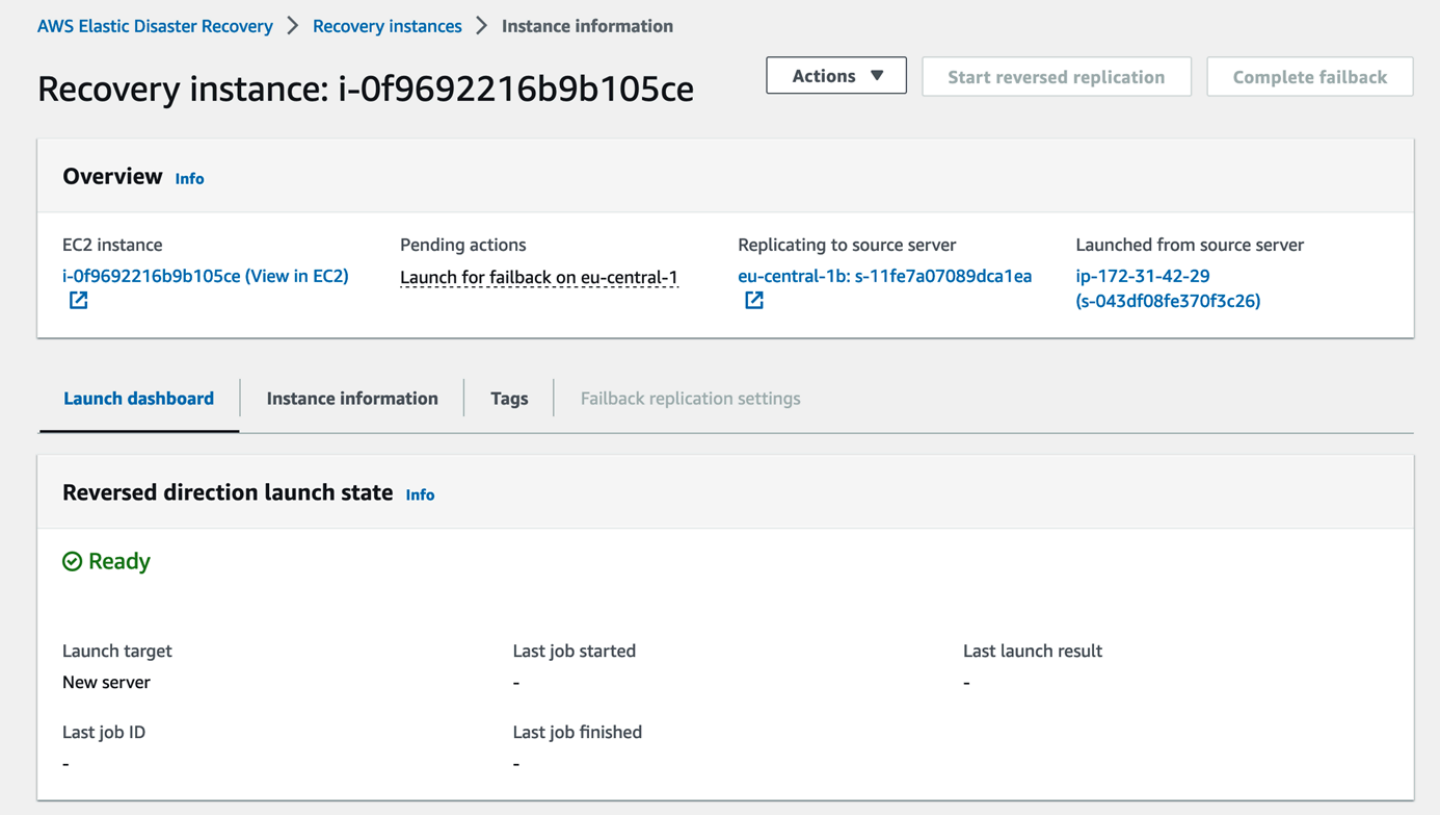
- **Launch dashboard** - see the current status of failback or reversed direction replication and launch.
- **Instance information** - view information about the underlying EC2 instance.
- **Tags** - manage the tags of the recovery instance.
- **Failback replication settings** - configure settings for failback to on-premises servers. Not relevant for in-AWS launches.
- **Post-launch actions status** - the progress or result of the last post-launch actions run.

Launch dashboard

The **Launch** dashboard provides a detailed overview of the reversed direction launch process.

Reversed direction launch state

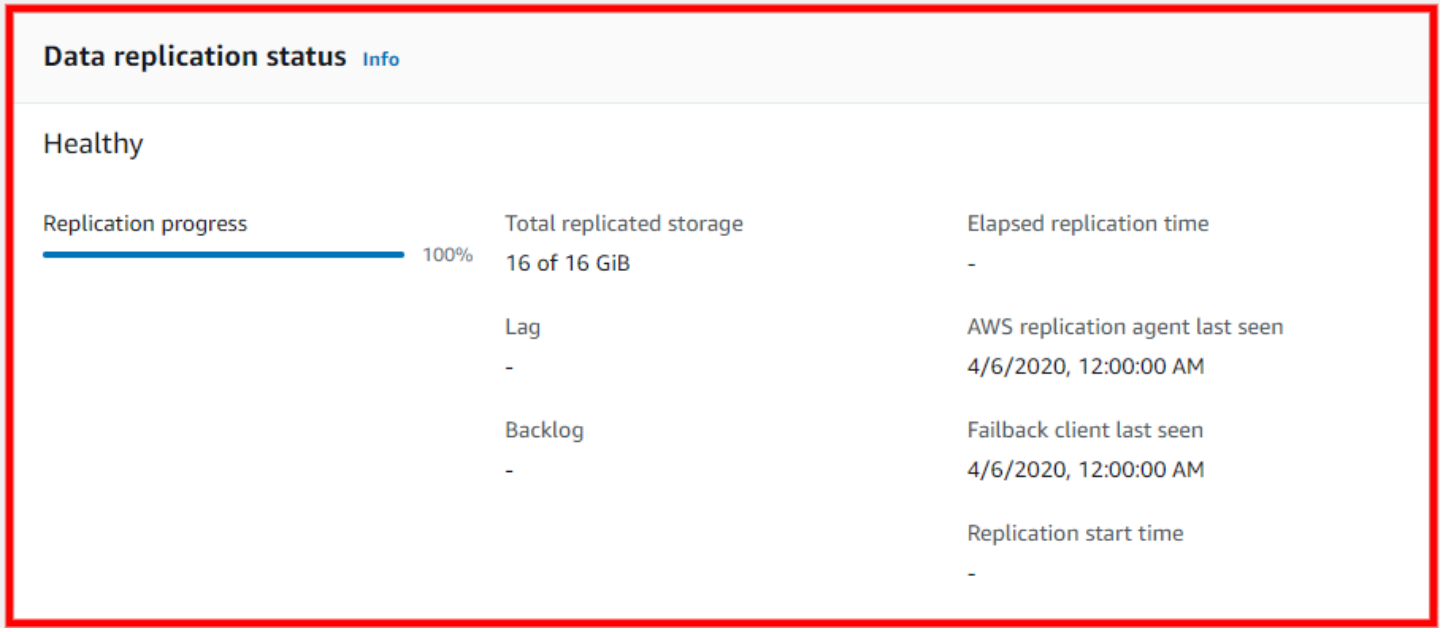
The **Reversed direction launch state** panel provides an overview of the reversed direction launch process, including:



- The current state of the failback.
- **Launch target** – the server onto which the recovery instance is launching into. This will indicate whether the recovery instance is launching into the original server or to a new server. Note that for in-AWS flows, this value is always a new server.
- **Last job ID** – the date and time of the last failback Job was started for the recovery instance.
- **Last job started** – the date and time the last failback job was started for the recovery instance.
- **Last job finished** – the date and time the last failback job was finished for the recovery instance.
- **Last launch results** – the results of the most recent launch.

Data replication status

The Data replication status panel shows the current data replication status state for the recovery instance, including:



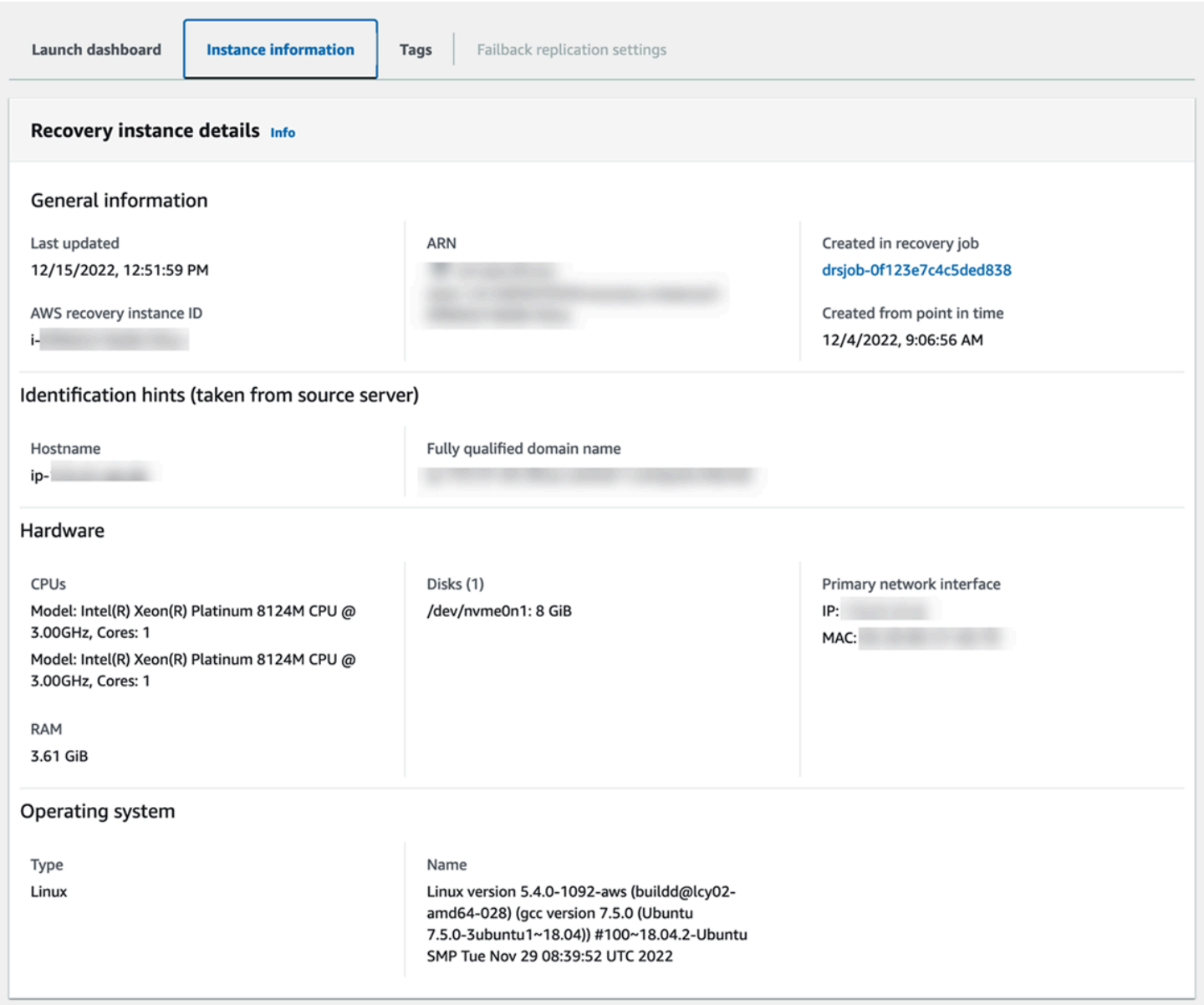
- **Replication progress** – the progress of the replication of the recovery instance in percent completed.
- **Total replicated storage** – the total amount of storage replicated in GiB.
- **Lag** – the total Lag time, if any.
- **Backlog** – the total backlog amount and time to clear, if any.
- **Elapsed replication time** – time elapsed since replication began.
- **AWS replication agent last seen** – the date and time connectivity was last established between the recovery instance and the AWS Replication Agent.
- **Failback client last seen** – the date and time connectivity was last established between the recovery instance and the Failback client.
- **Replication start time** – and date and time replication was started for the recovery instance.

Events and metrics

The Events and metrics section contains external links to monitor your recovery instance in AWS CloudTrail. Learn more about monitoring DRS with CloudTrail.

Instance information

The **Instance information** tab shows a variety of general server information, hardware, and network information.

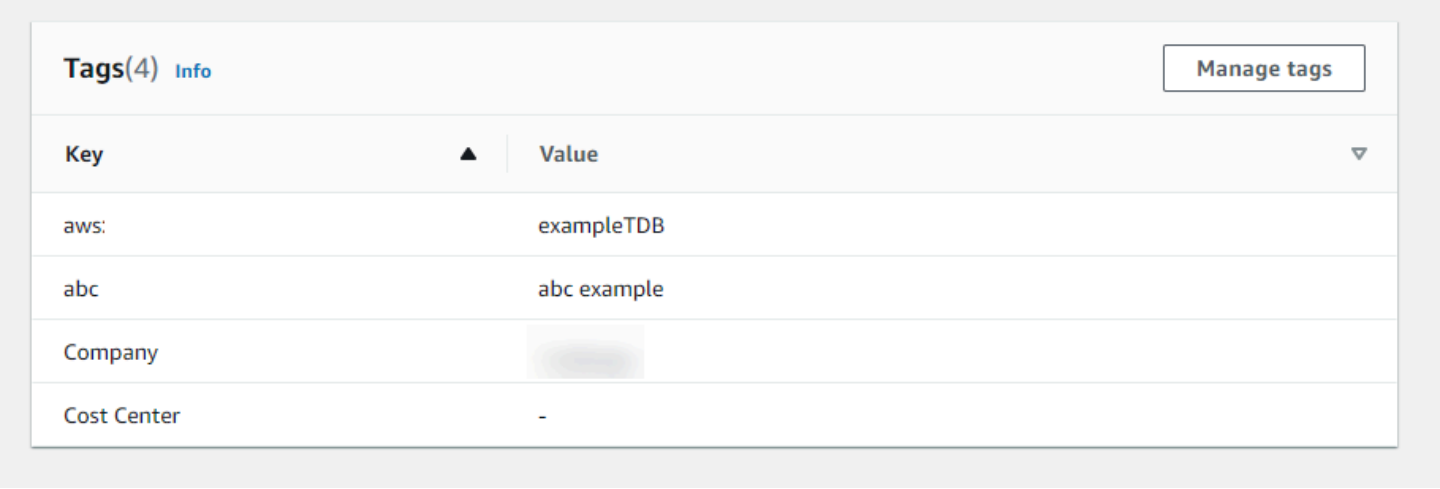


Information shown includes:

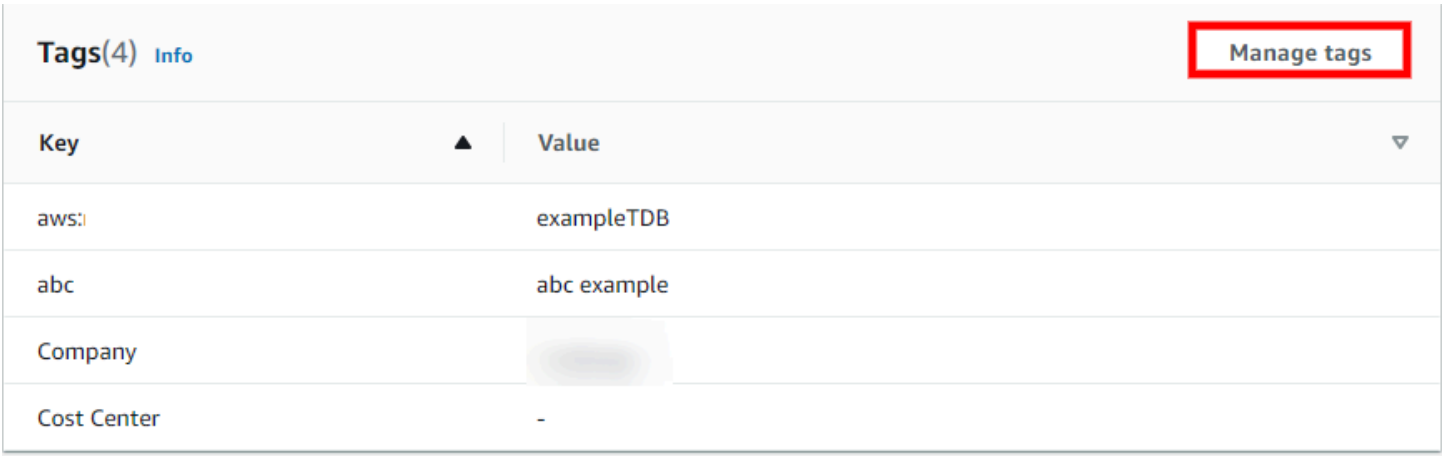
- **Last updated**
- **AWS recovery instance ID**
- **Created in recovery job**
- **Hostname**
- **Fully qualified domain name**
- **CPUs**
- **Disks**
- **Primary network interface**
- **Operating system** information

Tags

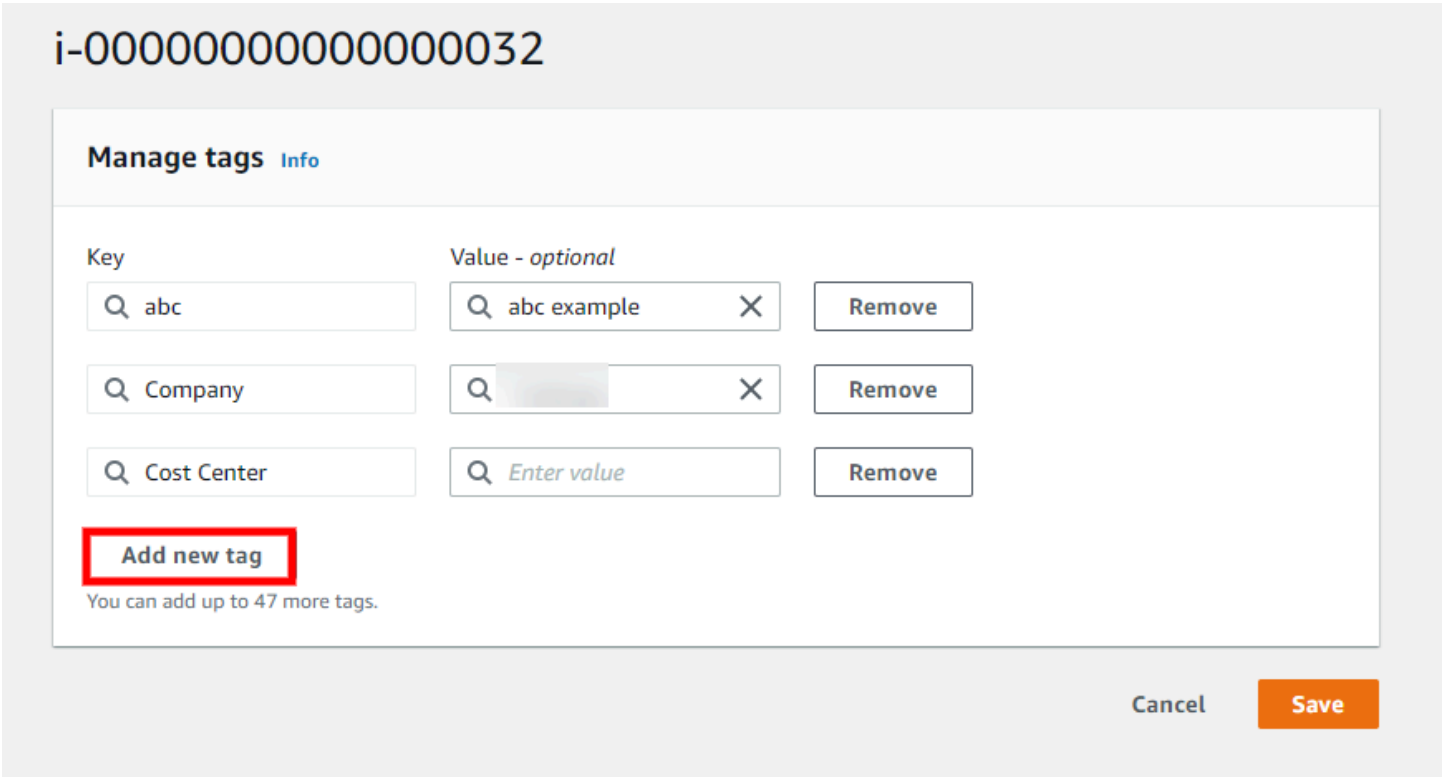
The Tags section shows any tags that have been assigned to the server. A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Learn more about AWS tags in this Amazon EC2 article.



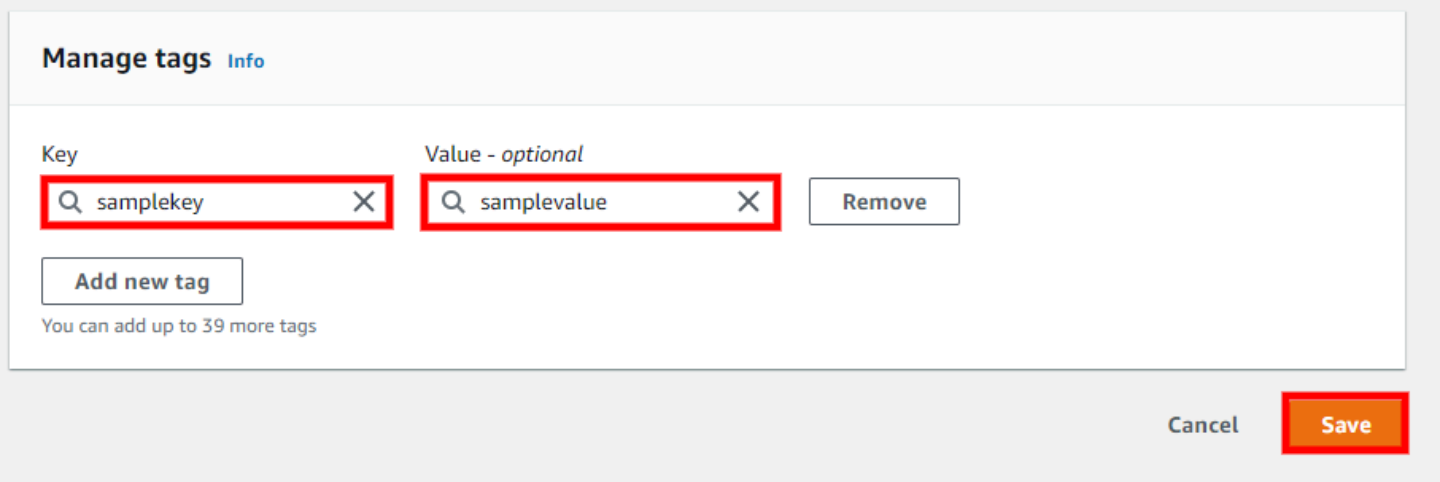
Choose **Manage tags** to add or remove tags.



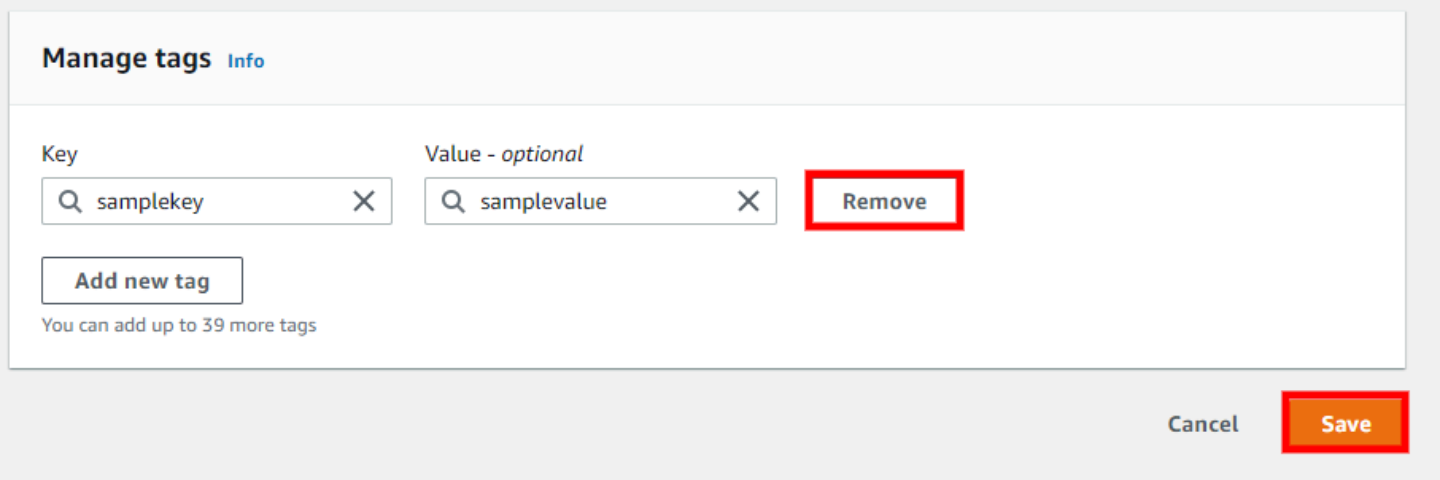
The **Manage tags** page will open. Choose **Add new tag** to add a new tag.



Add a tag **Key** and an optional tag **Value**. Choose **Save** to save your added tags.



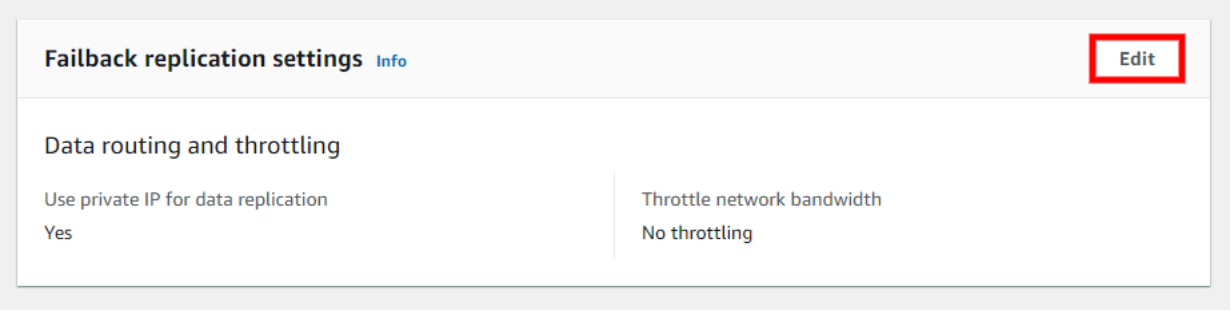
To remove a tag, choose **Remove** to the right of the tag you want to remove, and then choose **Save**.



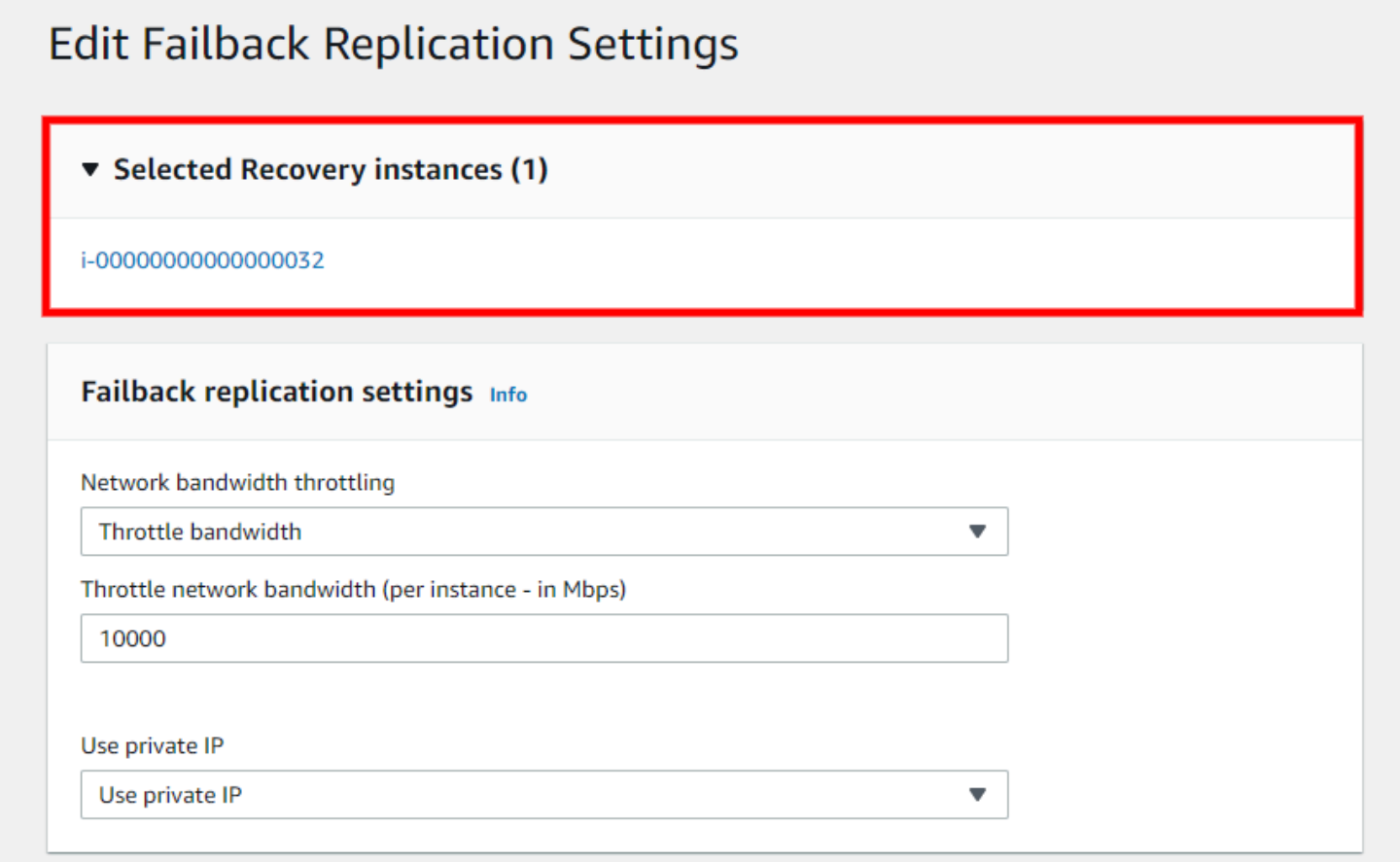
Failback replication settings

The Failback replication settings tab allows you to edit various failback replication settings for the recovery instance prior to performing a failback.

Choose **Edit** to edit the settings.

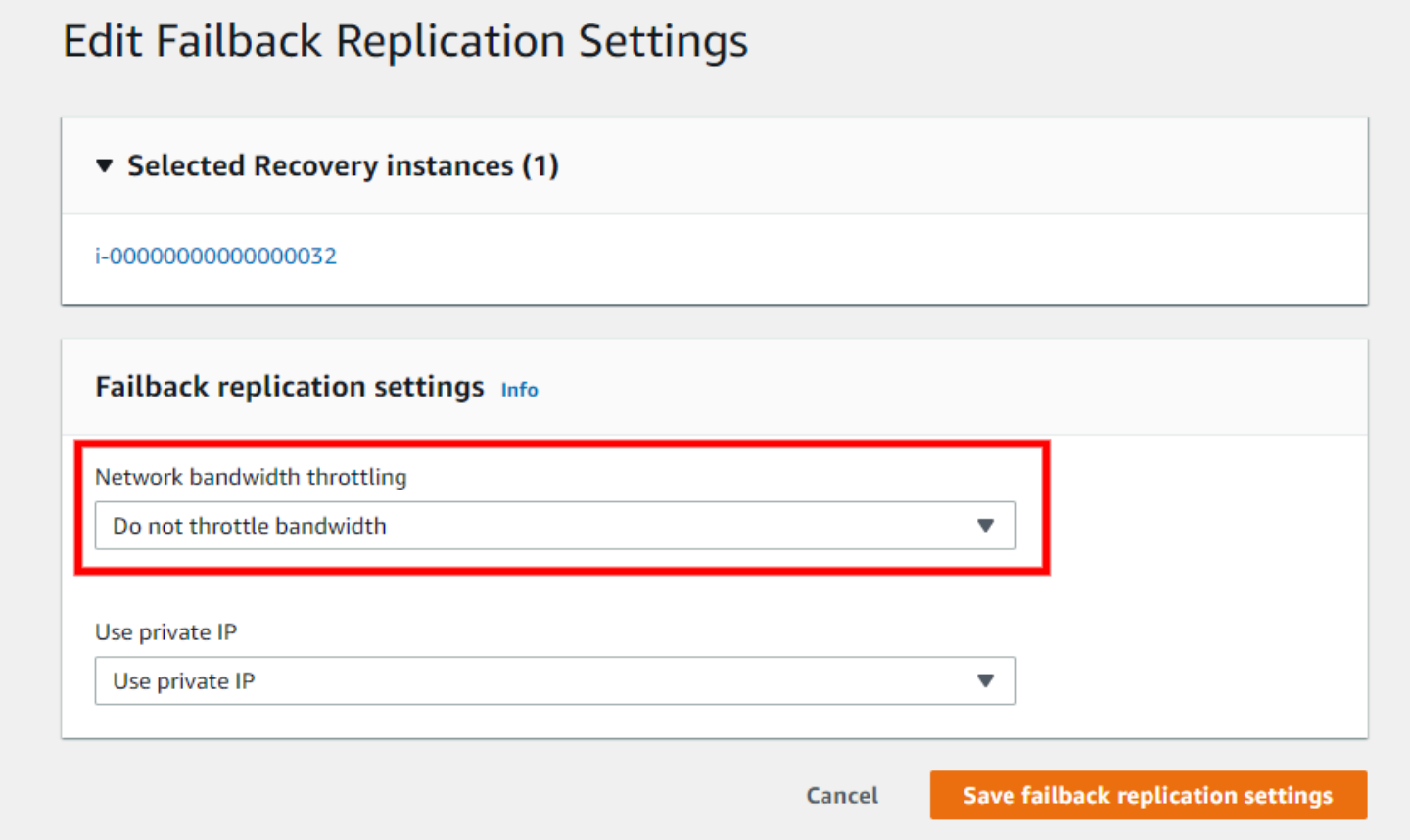


You can configure the failback replication settings for multiple recovery instances at once. The **Selected recovery instances** box shows for which recovery instances you are updating the settings.



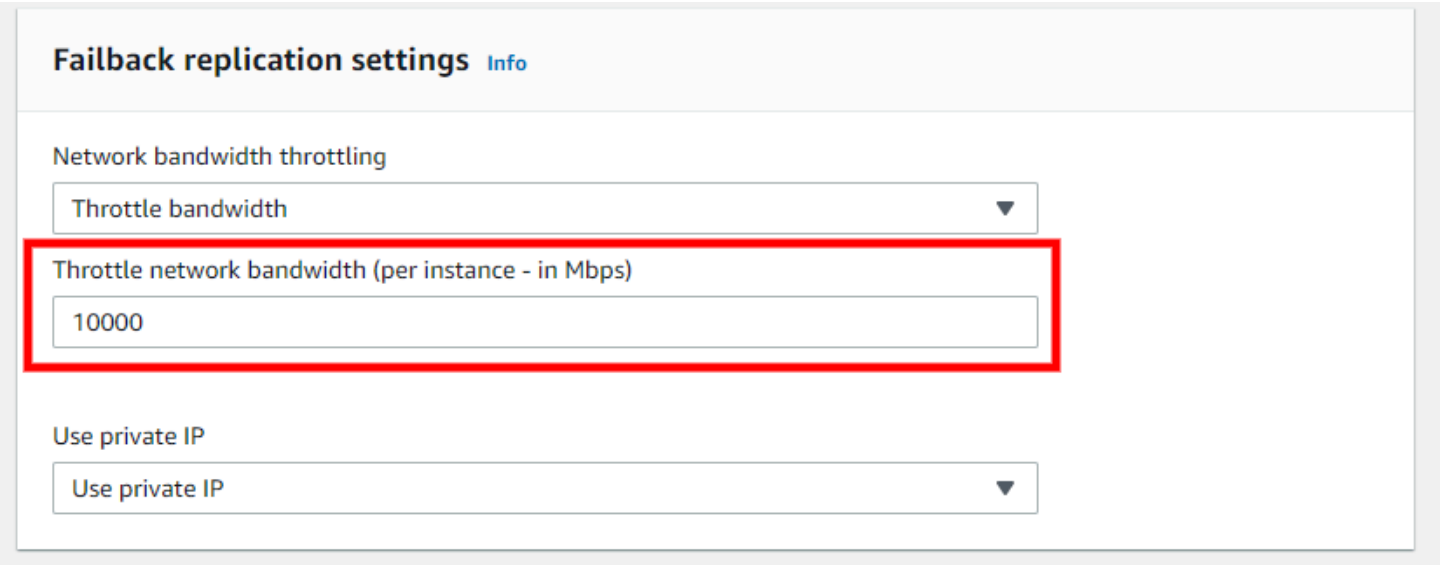
Network bandwidth throttling

You can control the amount of network bandwidth used for data replication per server. By default, Elastic Disaster Recovery will use all available network bandwidth utilizing five concurrent connections.



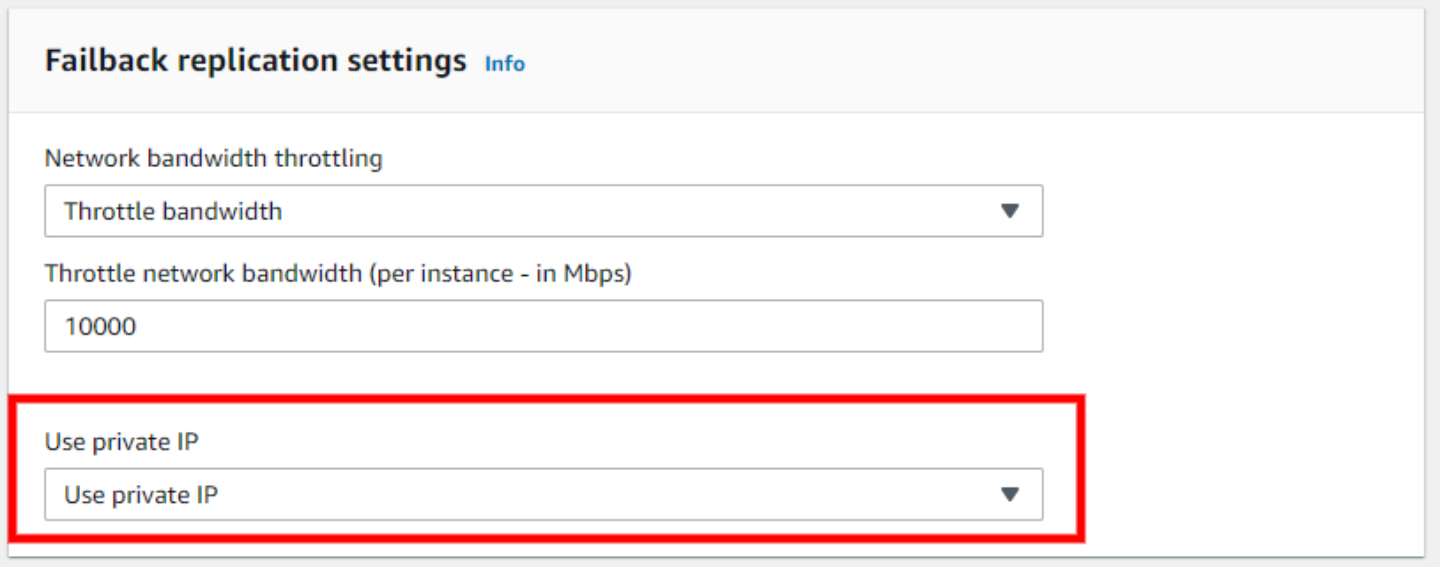
Choose **Throttle bandwidth** if you want to control the transfer rate of data sent from your recovery instances to your source servers during failback over TCP Port 1500. Otherwise, choose **Do not throttle bandwidth**.

If you chose to throttle bandwidth, then the **Throttle network bandwidth (per instance, in Mbps) box will appear**. Enter your desired bandwidth in Mbps.



Use private IP

By default, data is sent from the recovery instance to the source servers over the public internet, using the public IP that was automatically assigned to the Replication Servers. Transferred data is always encrypted in transit.



Choose the **Use private IP** option if you want to route the replicated data from your recovery instance to your source servers through a private network with a VPN, AWS Direct Connect, VPC peering, or another type of existing private connection. You should use this option if you want to:

- Allocate a dedicated bandwidth for replication;
- Use another level of encryption;
- Add another layer of security by transferring the replicated data from one private IP address (source) to another private IP address (on AWS).

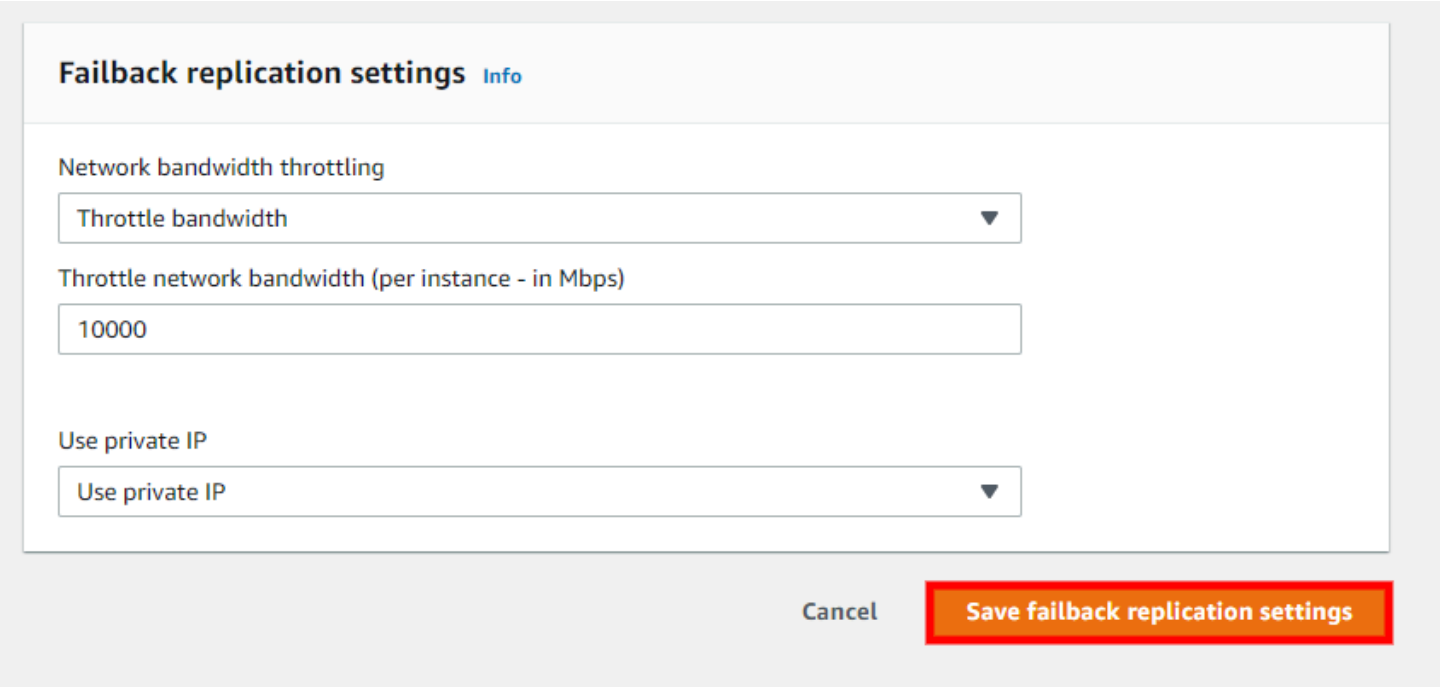
> **Important**
>
> Data replication will not work unless you have already set up the VPN, AWS Direct Connect, or VPC peering in the AWS Console.

Note

- If you selected the default subnet, it is highly unlikely that the private IP is activated for that subnet. Ensure that Private IP (VPN, AWS Direct Connect, or VPC peering) is activated for your chosen subnet if you wish to use this option.
- Choosing the **Use Private IP** option will not create a new private connection.

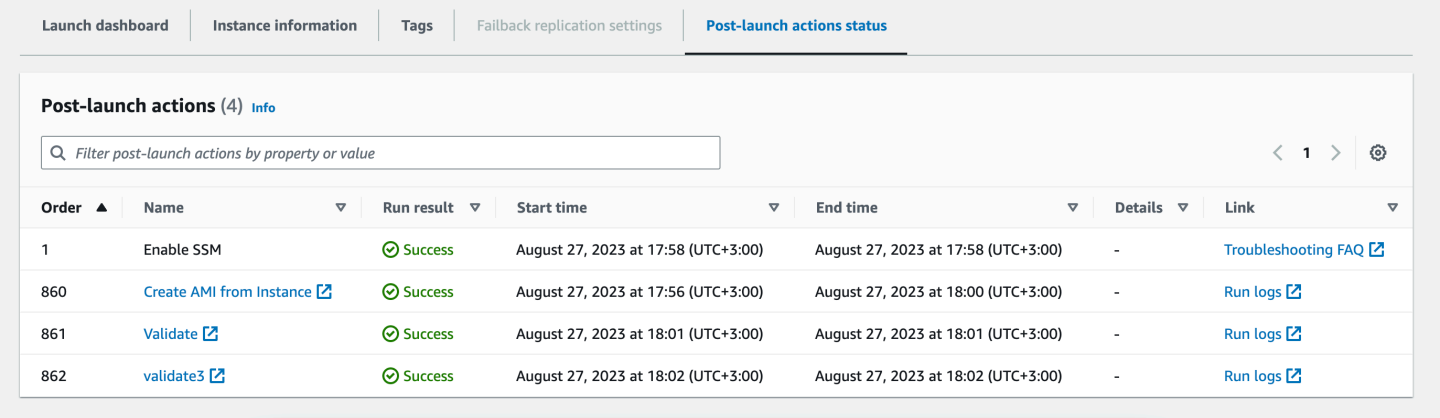
Saving failback replication settings

Once you have configured your failback replication settings, choose **Save failback replication settings.**



Post-launch actions status

The **Post-launch actions** view shows the current run status of post-launch actions.



The status includes:

- **Order** - the running order of the action.
- **Name** - the name of the action is a link to the detailed run status in the AWS Systems Manager console.
- **Run result** – provides the current action run status.
- **Start time** – the time when the action script started to run. This column will be empty for actions that have not yet started running.
- **End time** – the time when the action script run ended. This column will be empty for actions that have not yet completed running.
- **Details** – error messages will be shown in this column.
- **Link** – provides a link to resources created by this action if there are any, or to the action run logs in the AWS Systems Manager console.

Recovery job history

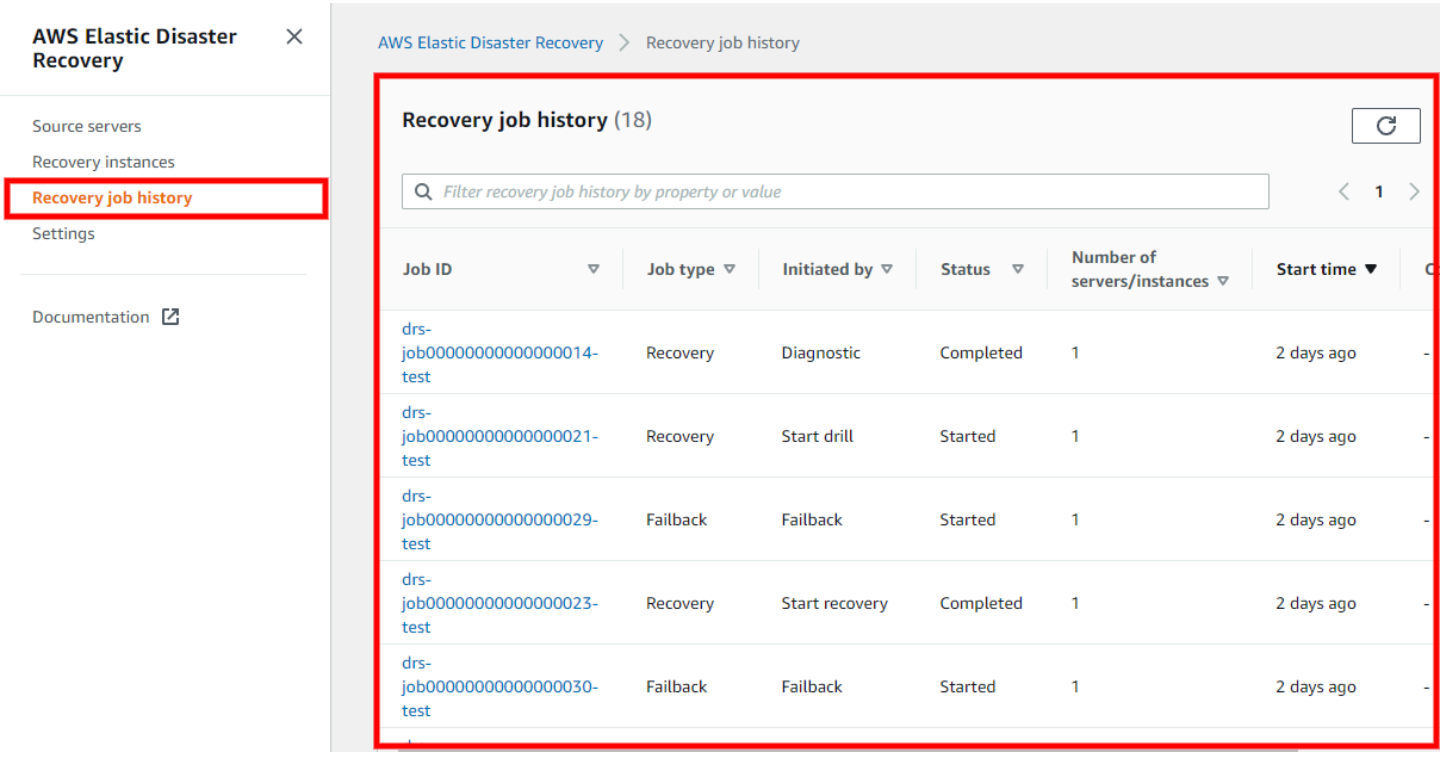
The Recovery launch history page provides an in-depth overview for operations (Jobs) performed in Elastic Disaster Recovery.

Topics

Recovery job history

The **Recovery job history** page allows you to track and manage all operations performed in Elastic Disaster Recovery.

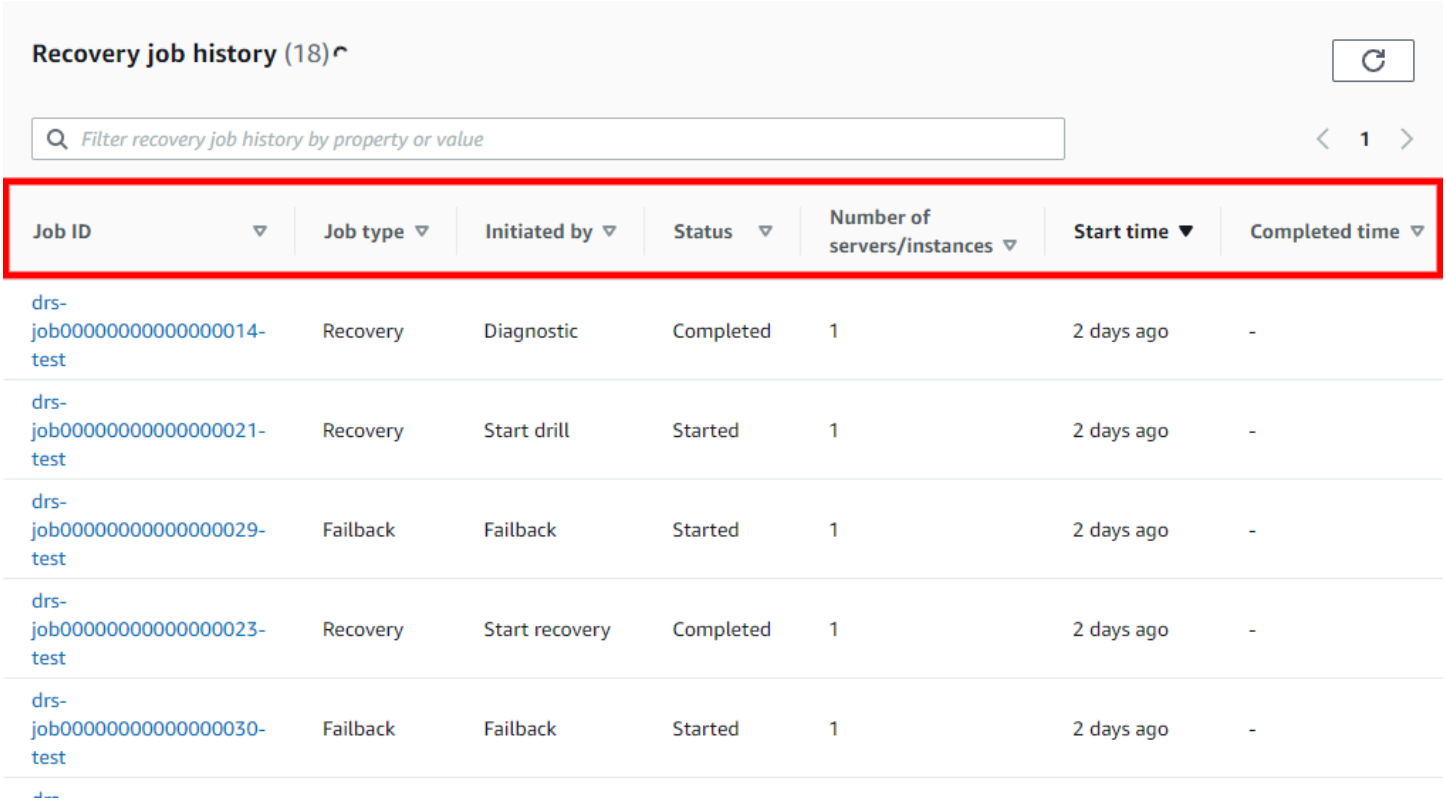
You can access the Recovery job history page by choose **Recovery job history** on the left-hand navigation menu.



Topics

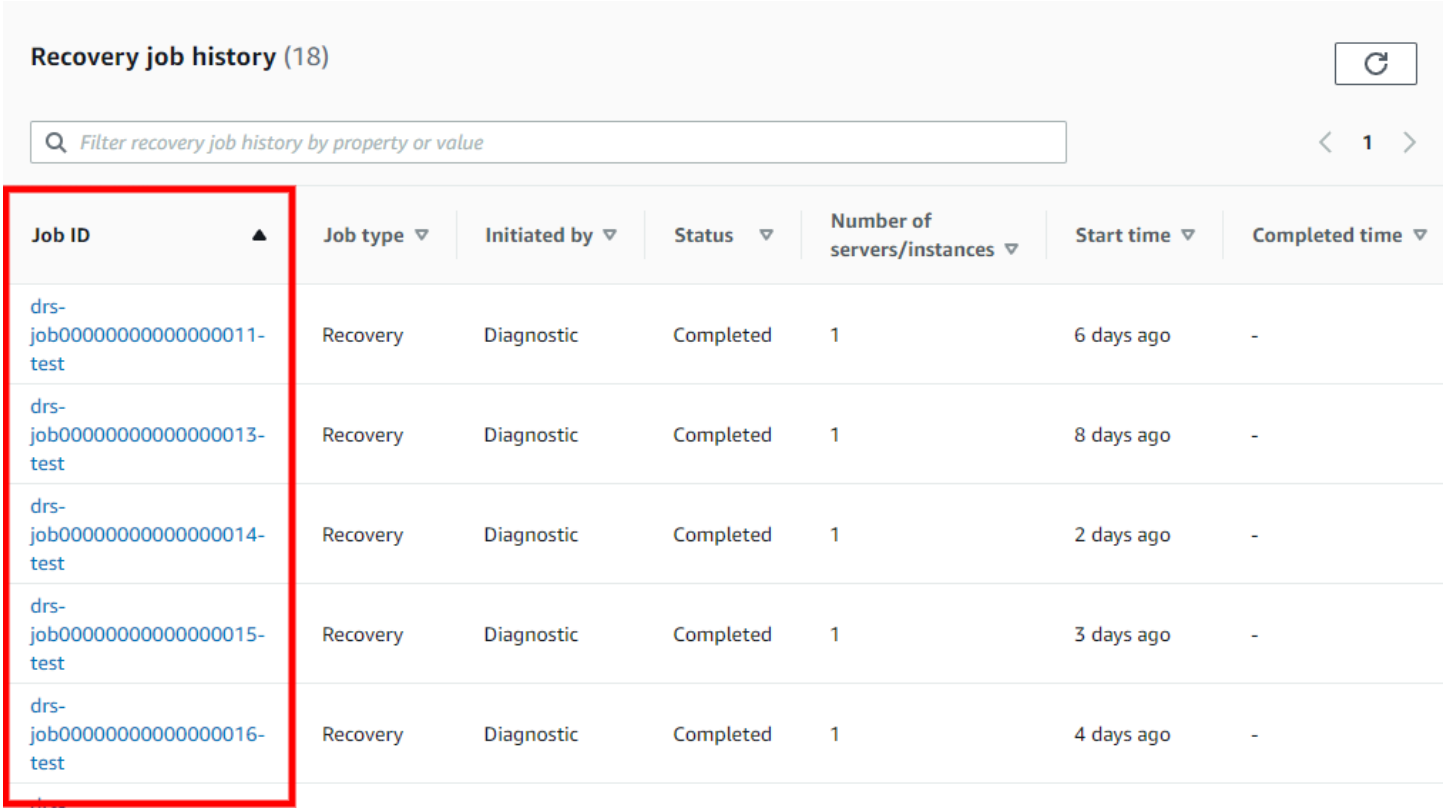
Overview

The Recovery job history tab shows all of the operations (referred to as "Jobs") performed on your account. Each Job corresponds to a single operation (ex. Launch Recovery instance, Launch Drill instance, etc.) Each Job is composed of one or more servers. The main Recovery job history view allows you to easily identify all key Job parameters, including:

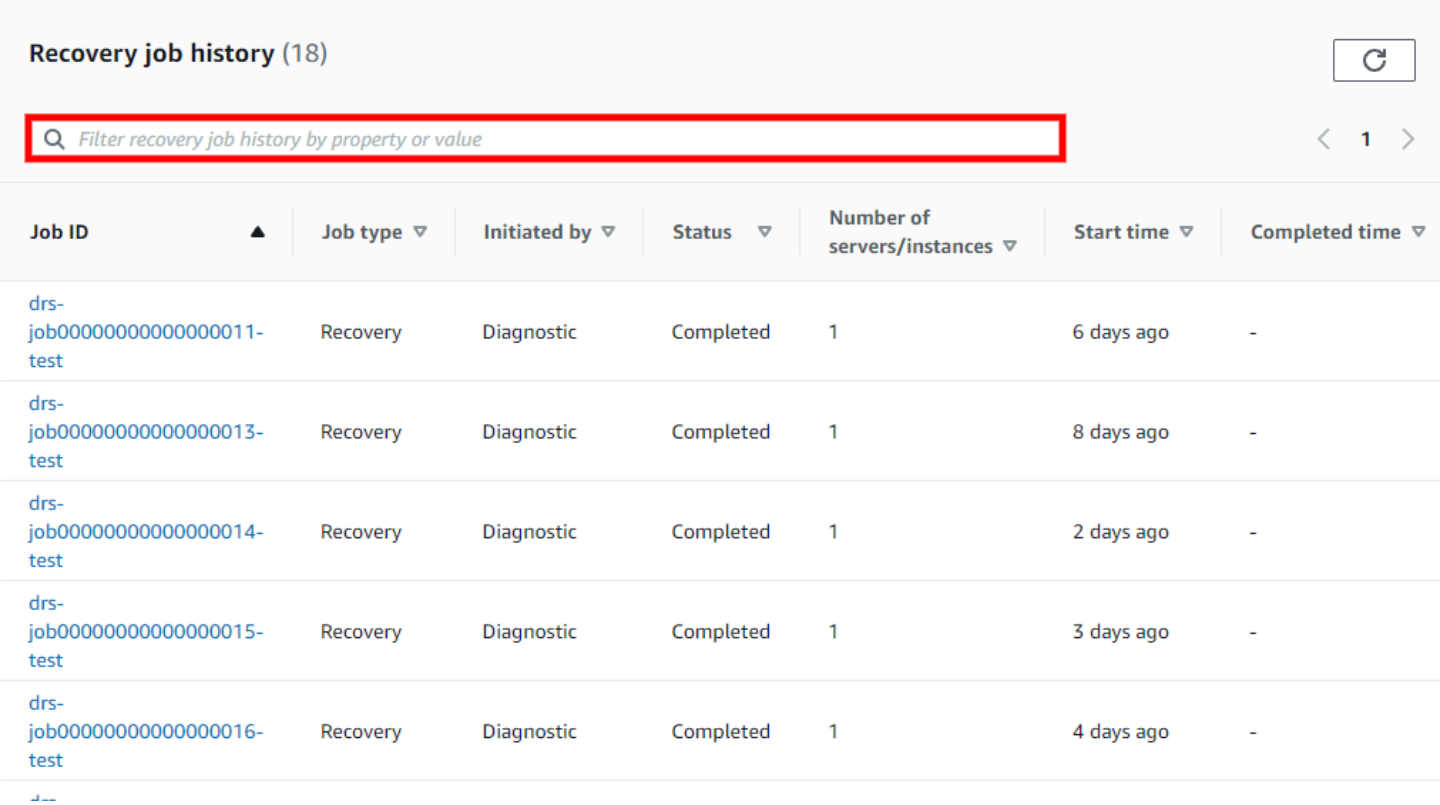


- **Job ID** - The unique ID of the Job.
- **Job Type** - The type of Job (Recovery, Failback, or Terminate)
- **Initiated By** - The command or action that initiated the Job (ex. Drill, Recovery, Failback)
- **Status** - The status of the Job (Pending, Completed, or Started)
- **Servers** - The number of servers that are included in the Job.
- **Start Time** - The time the job was started.
- **Completed Time** - The time the Job was completed (blank if the job was not completed)

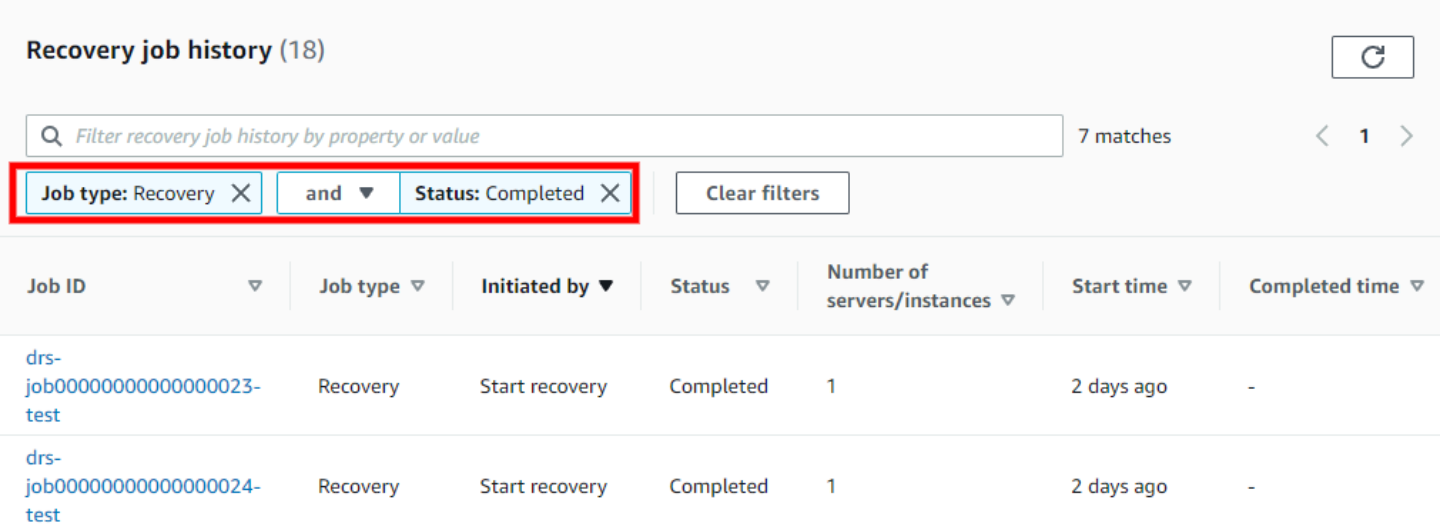
To sort the Recovery job history by any column (for example, **Job ID**), click the column header.



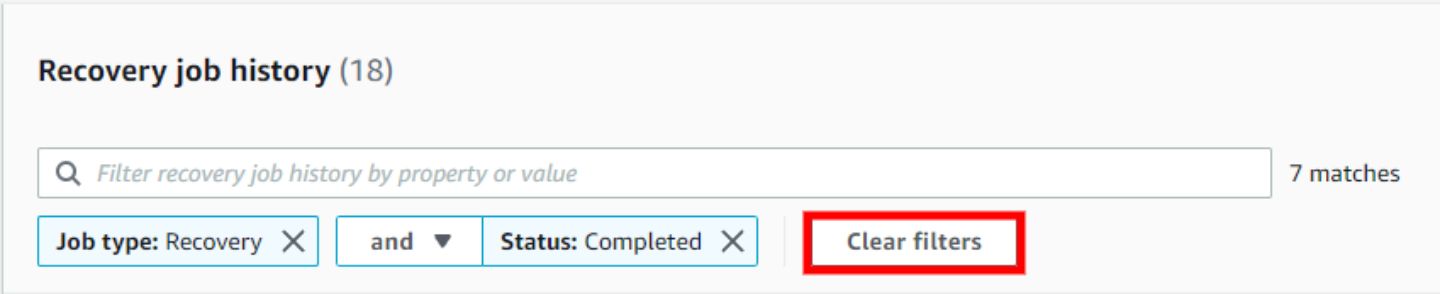
You can search for specific Jobs by any of the available fields within the **Find launch history by property or value** search bar.



Example: Filtered search for the values **Job type: Recovery** and **Status: Completed**, only showing completed Recovery Jobs.



Choose **Clear filters** to clear the search results and return to the default Job History view.

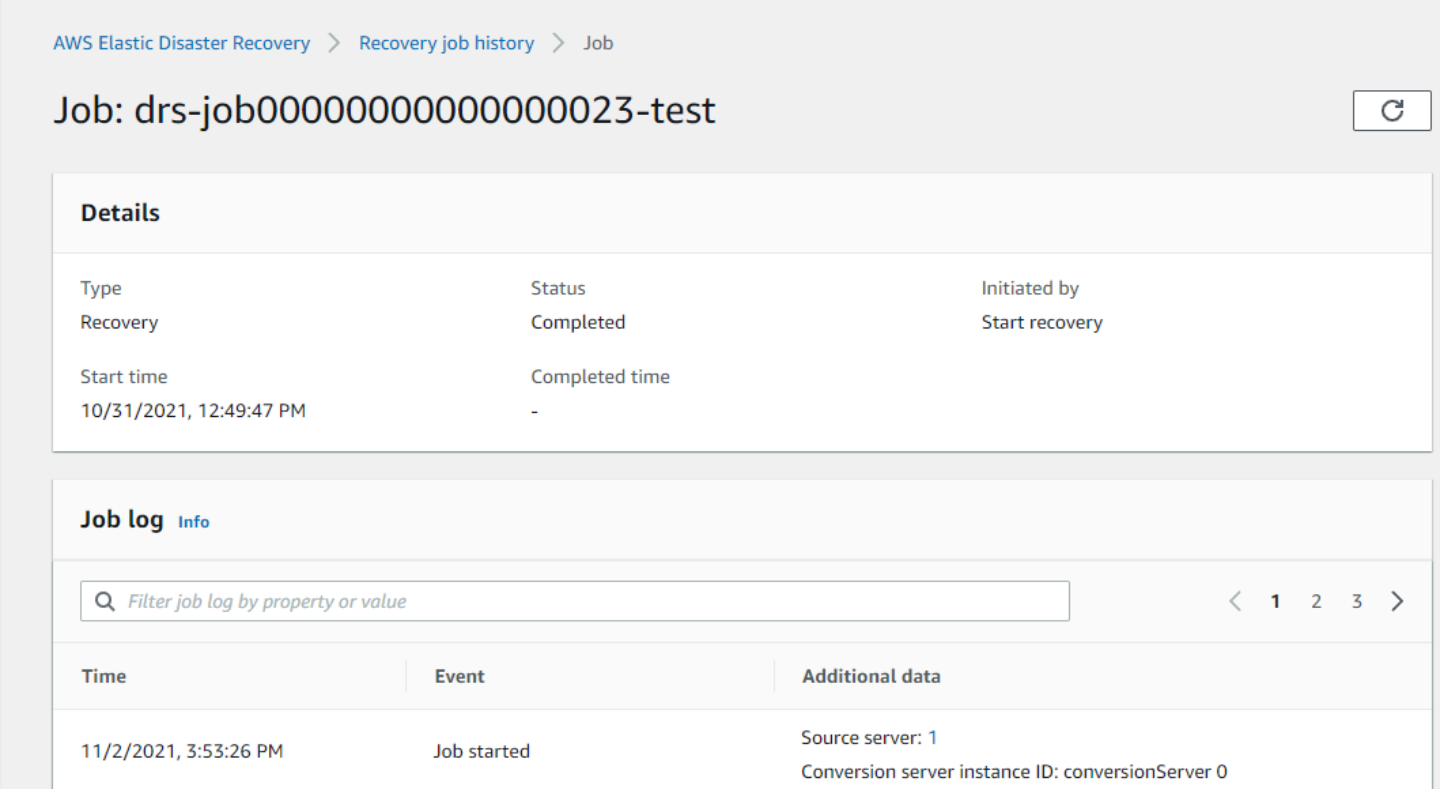


Job Details

You can view a detailed breakdown of each individual job by choosing the Job ID. Choose the **Job ID** of any Job to open the Job details view.

Job ID	Job type	Initiated by	Status	Number of servers/instances	Start time
drs-job00000000000000014-test	Recovery	Diagnostic	Completed	1	3 hours ago
drs-job00000000000000028-test	Failback	Continue failback	Started	1	3 hours ago

The Job details view is composed of three sections:

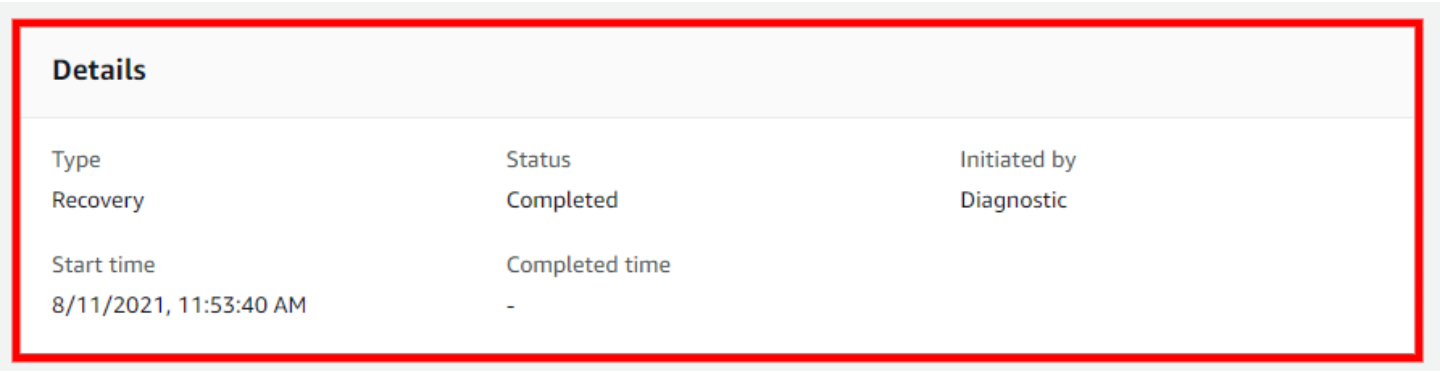


Topics

- Details
- Job log
- Jobs - Source servers

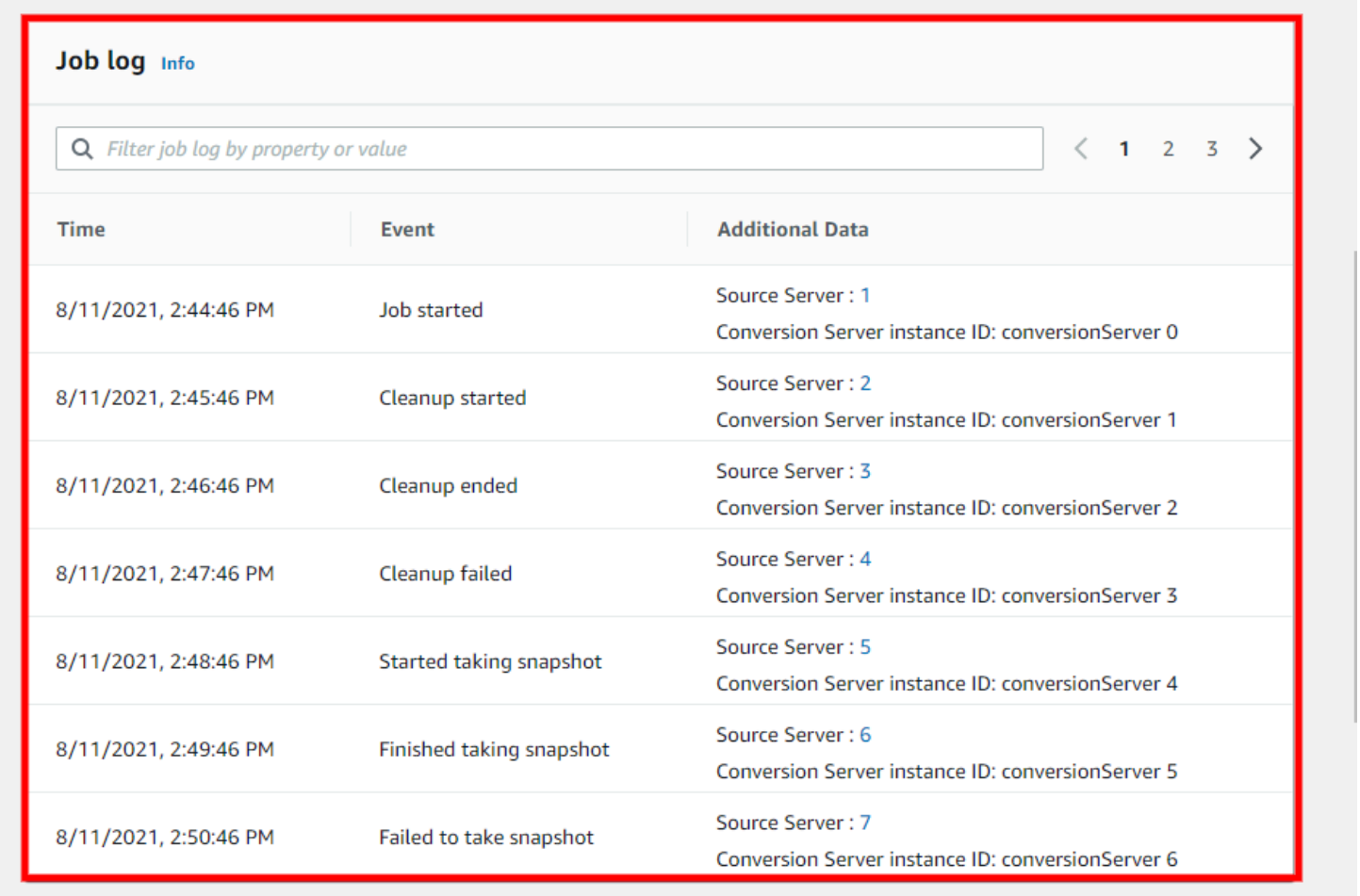
Details

The **Details** section shows the same information as the main Job log page, including the **Type, Status, Initiated By, Start time,** and **Completed time**.



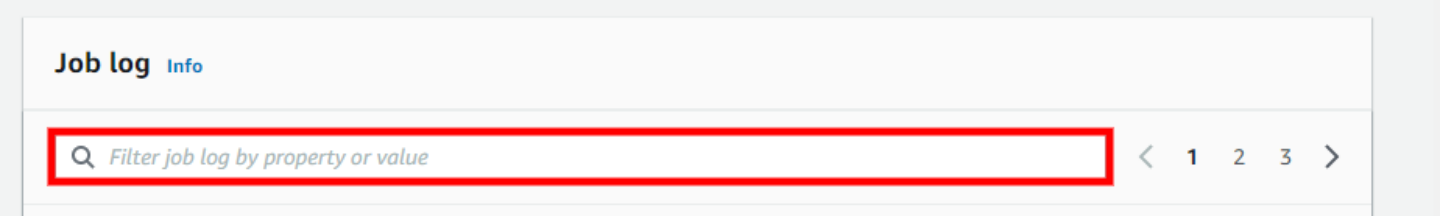
Job log

The Job log section shows a detailed log of all of the operations performed during the Job.

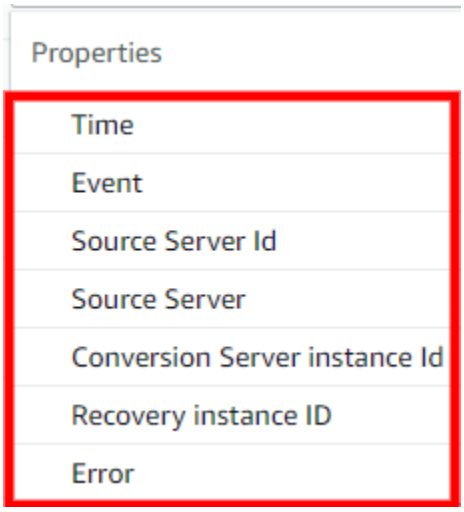


You can use this section to troubleshoot any potential issues and determine in which step of the launch process they occurred.

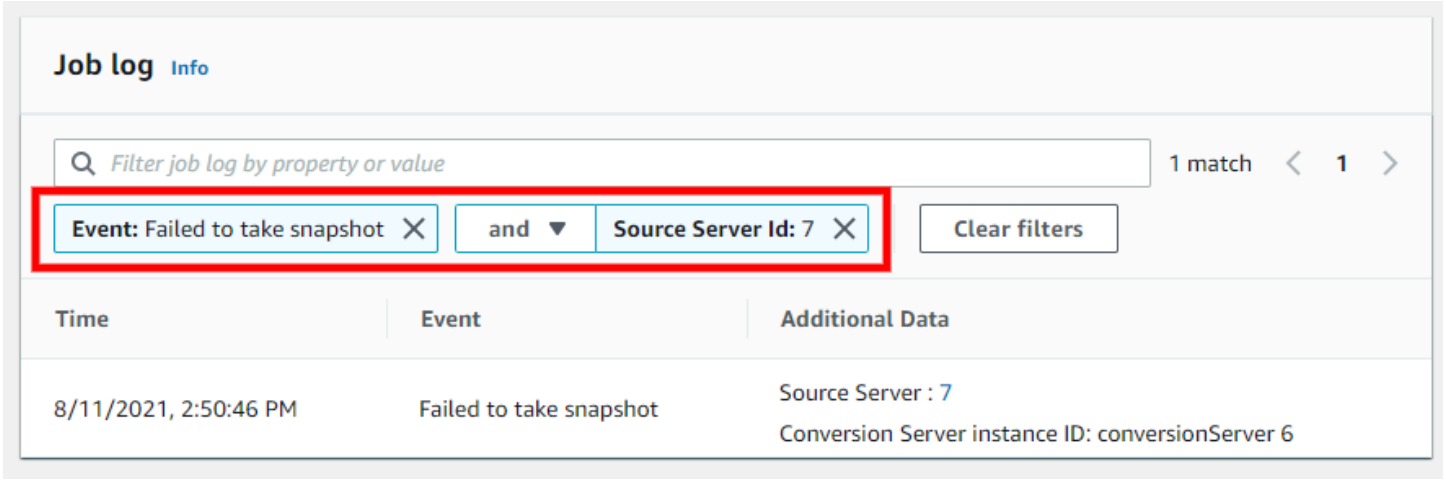
You can use the **Filter job log by property or value** search bar to filter the Job log.



You can filter by a variety of properties, including **Time, Event, Source Server Id, Source server hostname, Conversion Server instance Id, Drill/Recovery instance ID,** and **Error.**



You can filter by multiple values at once (for example, Job log filtered by **Event: Failed to take snaphot** and a specific **Source Server Id: 7**).

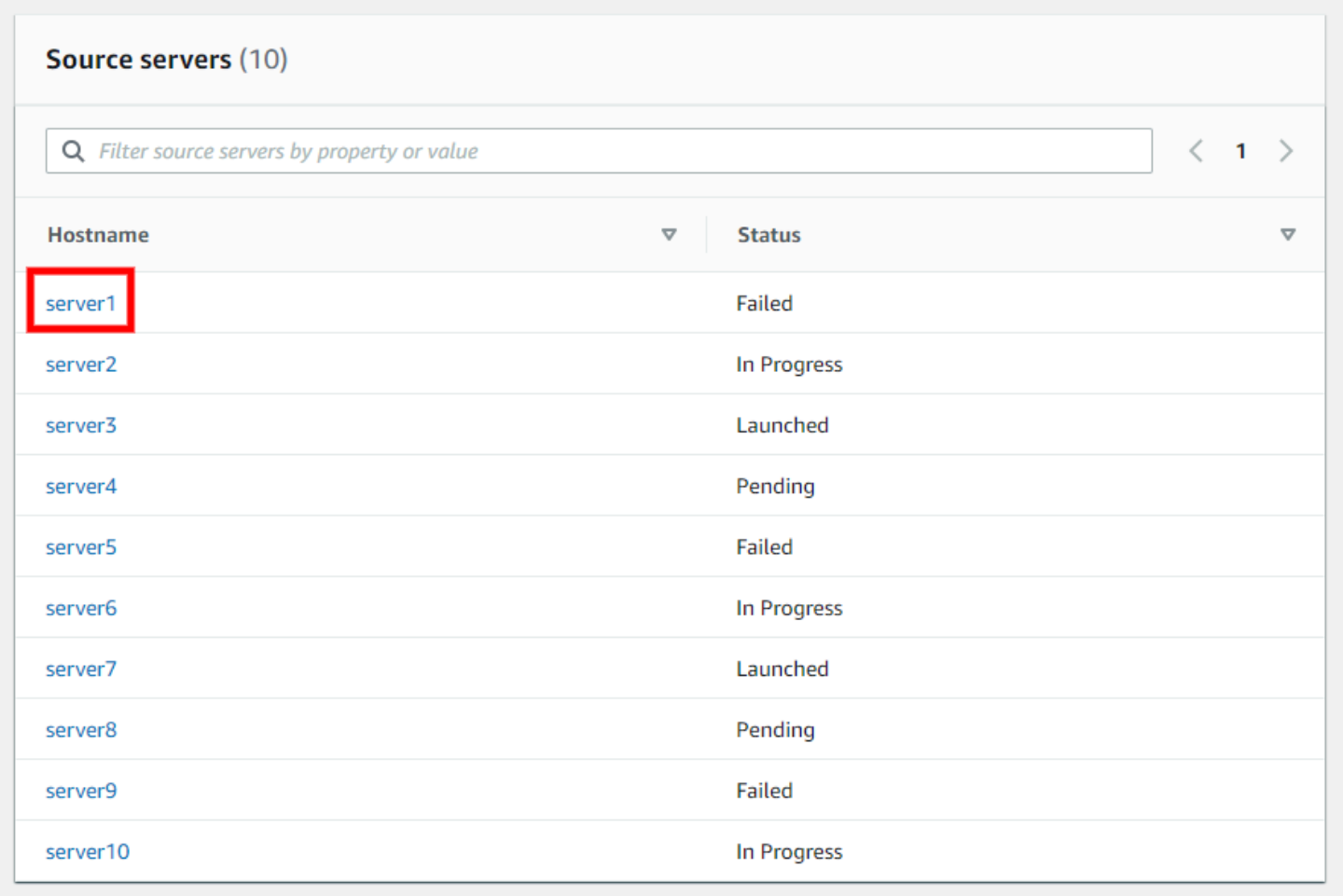


Jobs - Source servers

The Source servers section shows a list of all source servers involved in the Job and their status.

You can use the **Filter source servers by property or value** search bar to filter by **Hostname** or **Status**.

Choose the Hostname of any of Source server from the list to open the Server Details view for that server. Learn more about the Source Server details view.



Using multiple staging accounts with AWS DRS

AWS Elastic Disaster Recovery (AWS DRS) currently limits customers to 300 replicating source servers per account per AWS Region, due to various storage and API limitations. Customers who want to replicate and protect more than 300 source servers can use multiple staging accounts for replication, and recover their source servers into these accounts or into a single target AWS account. Customers who want to recover into a target account can manage the recovery for all the source servers in the staging accounts from that target account.

Use cases for this feature:

- You have more than 300 replicating servers and want to manage them from a single account.
- You have multiple AWS accounts with any number of servers and want to manage these servers from a single account.
- You want to manage your source servers in different AWS accounts for various business or security reasons and want to manage them from a single account.
- You have replicating servers that you would like to be able to recover to multiple different AWS accounts.

Overview

The multiple staging account feature is configured similarly to the standard AWS DRS configuration, but includes several extra steps required to configure the target AWS account.

For each staging account, you must first:

1. Initialize AWS DRS.
2. Define your replication configuration template.
3. Install the AWS Replication Agent on each source server.
4. Configure the individual source server replication settings.
5. Share the EBS encryption key with the target account.
6. Create a role to allow access into the staging account from the target account

For each target account, you must first:

1. Initialize the target account.

Once all of your source servers have been added to your staging accounts and are replicating successfully (are in the **Healthy** data replication state), you can use the target AWS account to launch Drill and Recovery instances for each server.

> **Note**
>
> You can only update the default replication template for the source servers from the staging account and not from the target account. Also, disconnection and deletion of the staging account's source servers are done from the staging account (to stop replication and save on resource usage). Source servers can be extended into many target accounts, or deleted from them.

> **Note**
>
> Source servers that reside in the staging account but are managed in the target account are called "extended source servers". An extended source server for which the staging source server has been deleted, or the role revoked, will remain in the target account, but will be marked with an extension error. An extended source server can be deleted at any time from the target account.

> **Note**
>
> Source servers that are EC2 instances, and have one or more marketplace licenses associated, cannot be extended into the target AWS account, unless the source AWS account (the AWS account that owns the EC2 instance) creates a failback and in-AWS account role for the staging account. This is required to provide permissions to get the marketplace license data from the source account when the server is extended. Create a Failback and in-AWS right-sizing role for trusted account for any staging account on the source account (the AWS account that owns the EC2 instance).

On a target account, the source servers list view shows all the source servers that were extended into the account, or those that are replicating in it.

Extending source servers from a staging account into a target AWS account

You can extend source servers from both new and existing AWS DRS accounts into a target AWS account.

Onboarding a new staging account

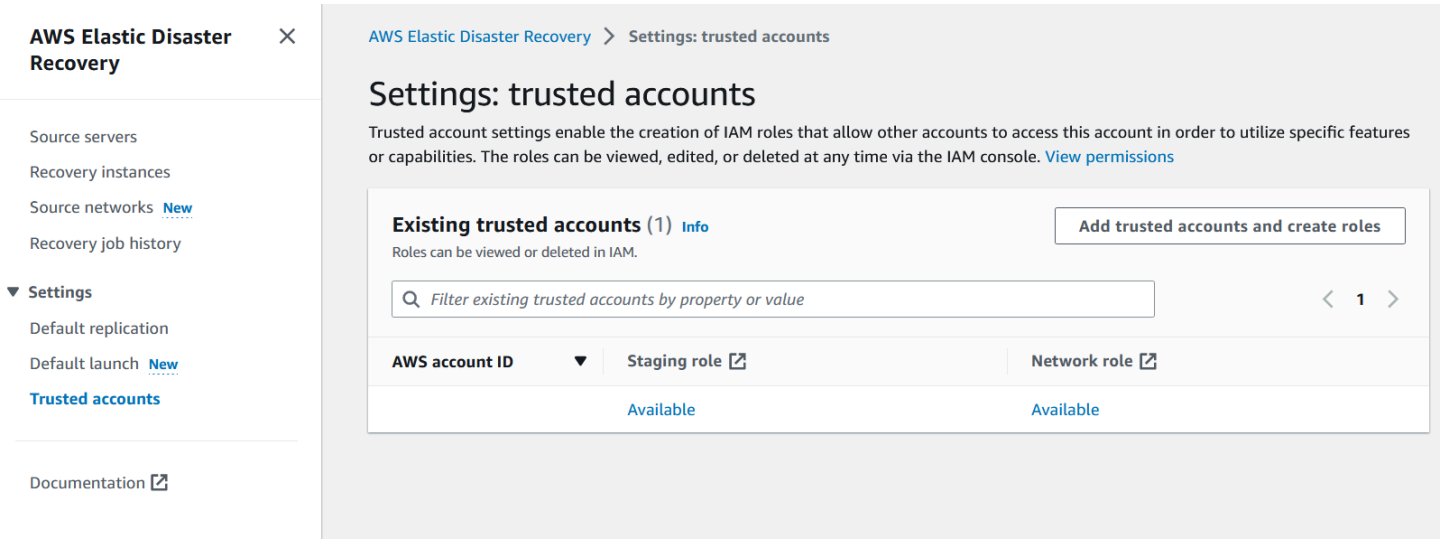
To use an account as a staging account in any AWS Region, you must first initialize AWS DRS in the AWS Region of the staging account, and add roles for the target account or accounts you plan to use.

During initialization, you will need to define the default replication settings, as described in the quick start guide.

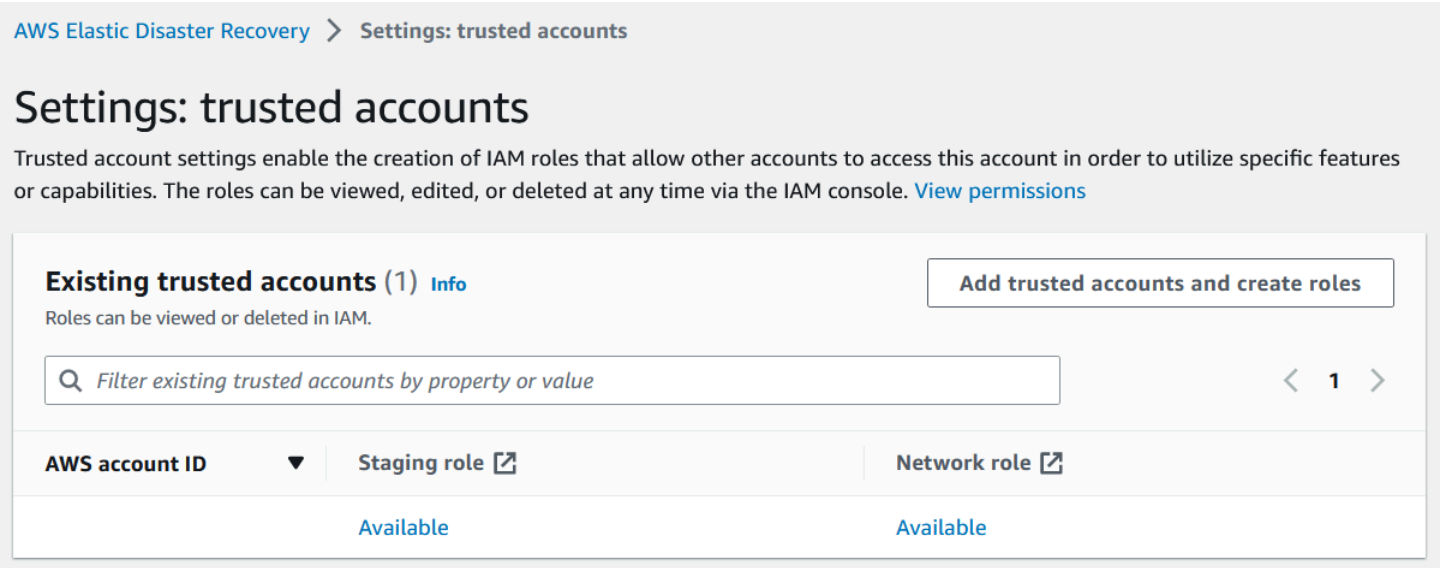
> **Note**
>
> If your volumes are encrypted, you must use a custom encryption using a customer managed key when defining the EBS encryption. This key must be shared with the target account (see instructions below), to facilitate recovery in the target account.

After the initialization of the staging account, add IAM roles for the target accounts on the **Settings: trusted accounts** page of AWS DRS in the staging account. The roles are used to allow the target account to extend source servers from the staging account and to recover them in the target account.

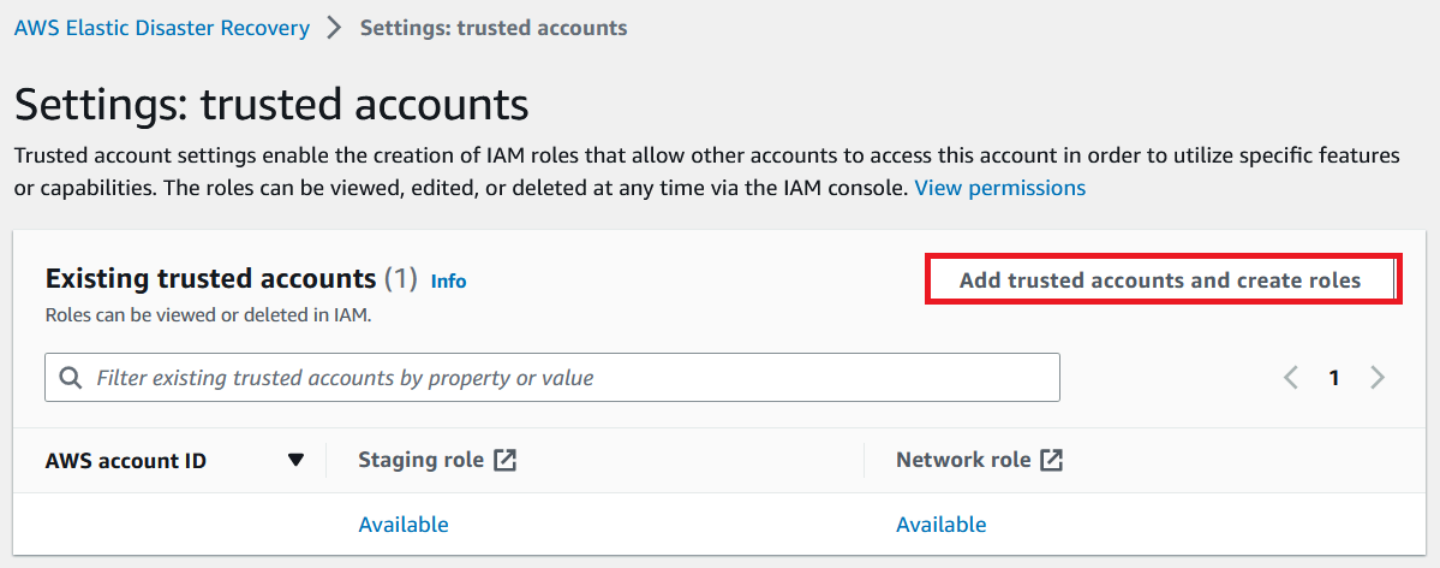
> **Note**
>
> Commercial AWS accounts can only be extended to other Commercial AWS accounts and Gov Cloud AWS accounts can only be extended to other Gov Cloud AWS accounts.



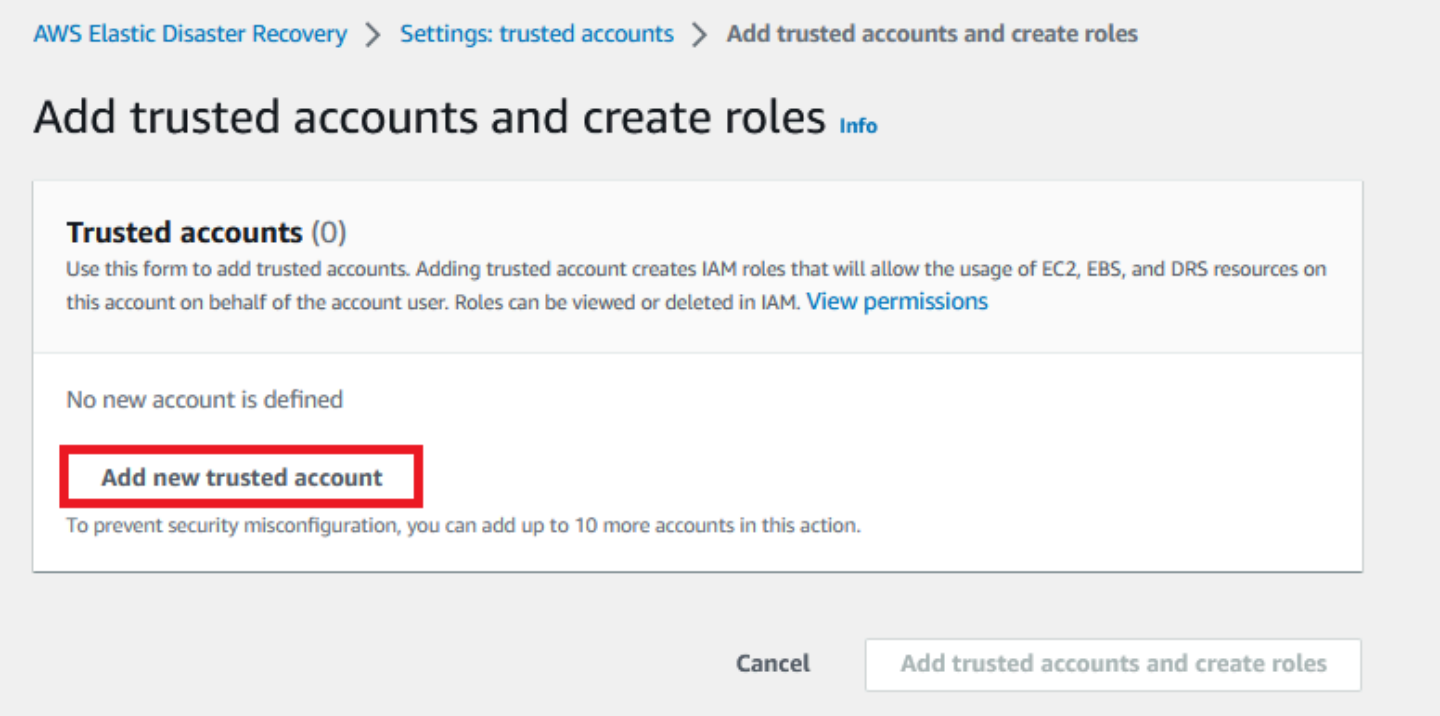
Under trusted accounts settings, you will find the **Existing trusted accounts** pane. Here, you can manage existing staging account IAM roles. These IAM roles are used to associate the staging account with the target account.



Use **Add trusted accounts and create roles** to add roles for any trusted account you plan to use.



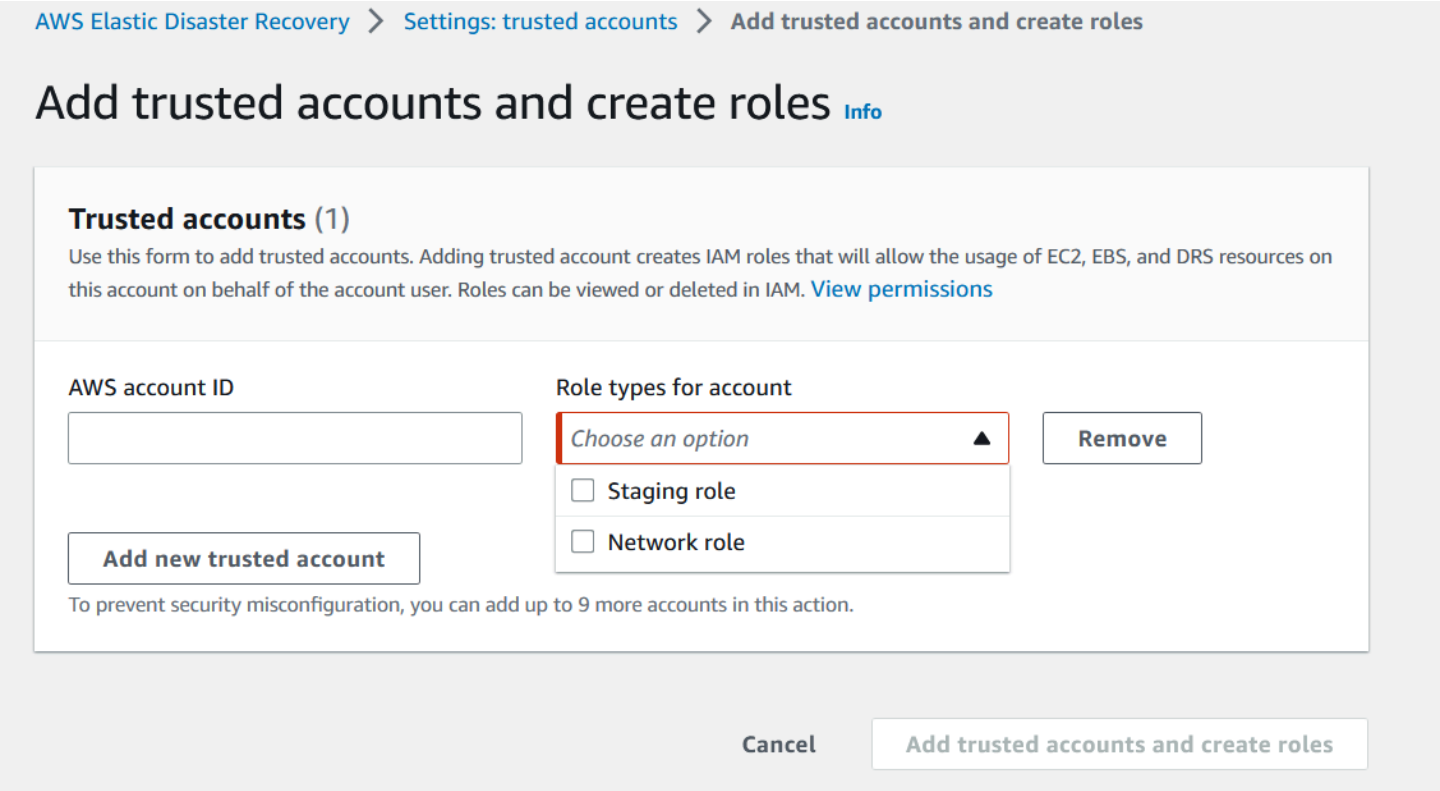
On the **Add trusted accounts and create roles** page, choose **Add new trusted account**.



Enter the AWS account IDs of the trusted account and select staging role. This will automatically generate a service IAM role that will allow the use of Amazon EC2, Amazon EBS and AWS DRS resources in the staging account on behalf of a trusted account's user.

Choose **Add new trusted account** to add more than one trusted account at once. You can add up to 10 trusted accounts at once.

Once you have added your accounts, choose **Add trusted accounts and create roles**.



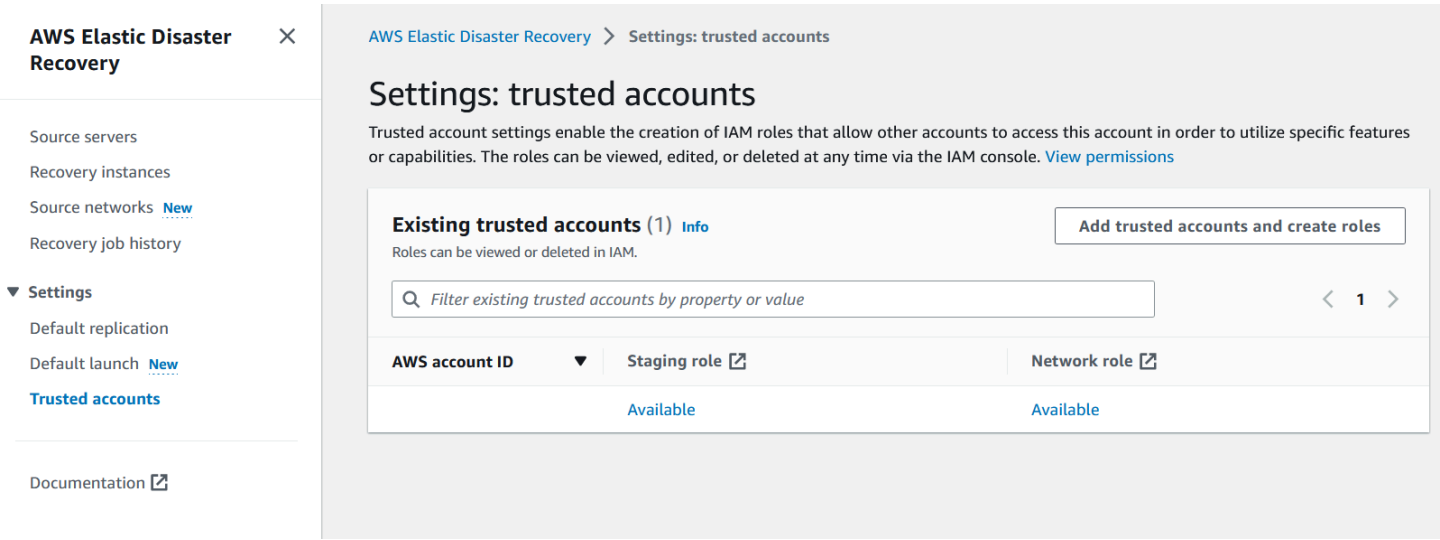
Using an existing account as a staging account

To use an account as a staging account, the default replication settings and replication settings of each source server that is to be extended into a target account should be reviewed, and EBS encryption must be set to use custom encryption using a customer managed key.

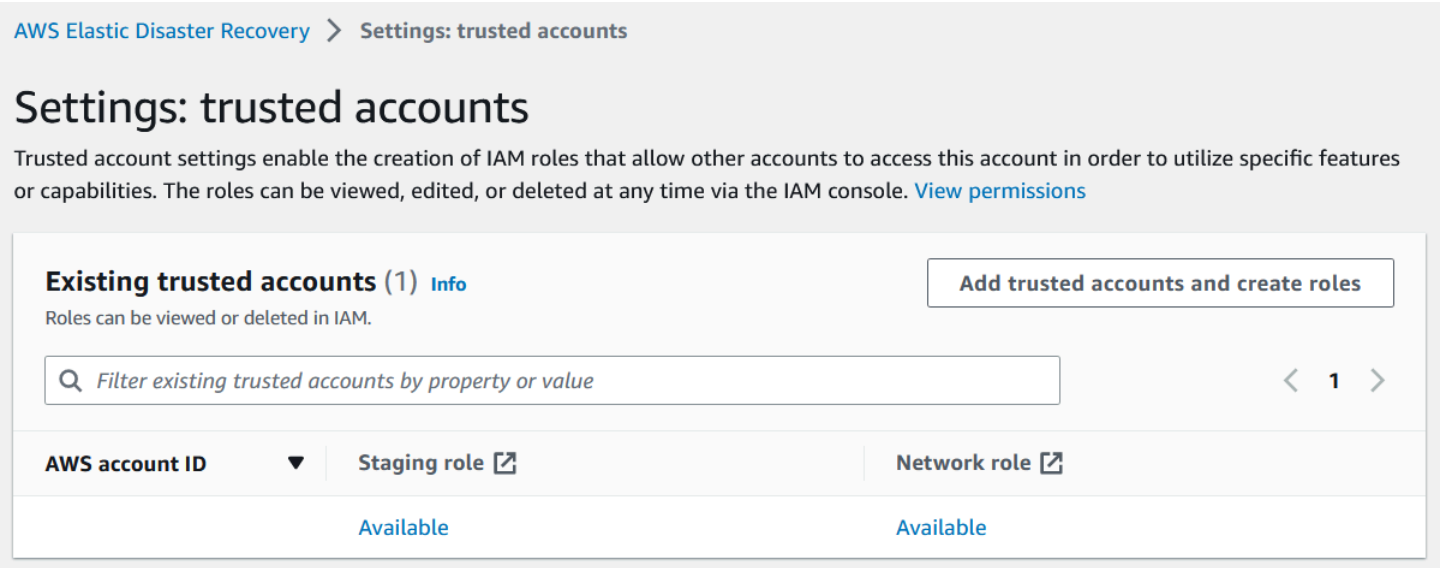
> **Note**
>
> This may trigger a full resync of the replicated data for a source server that had the default key, if that source server's encryption key was modified.

Share the customer managed key (or keys) with the target account (as described below).

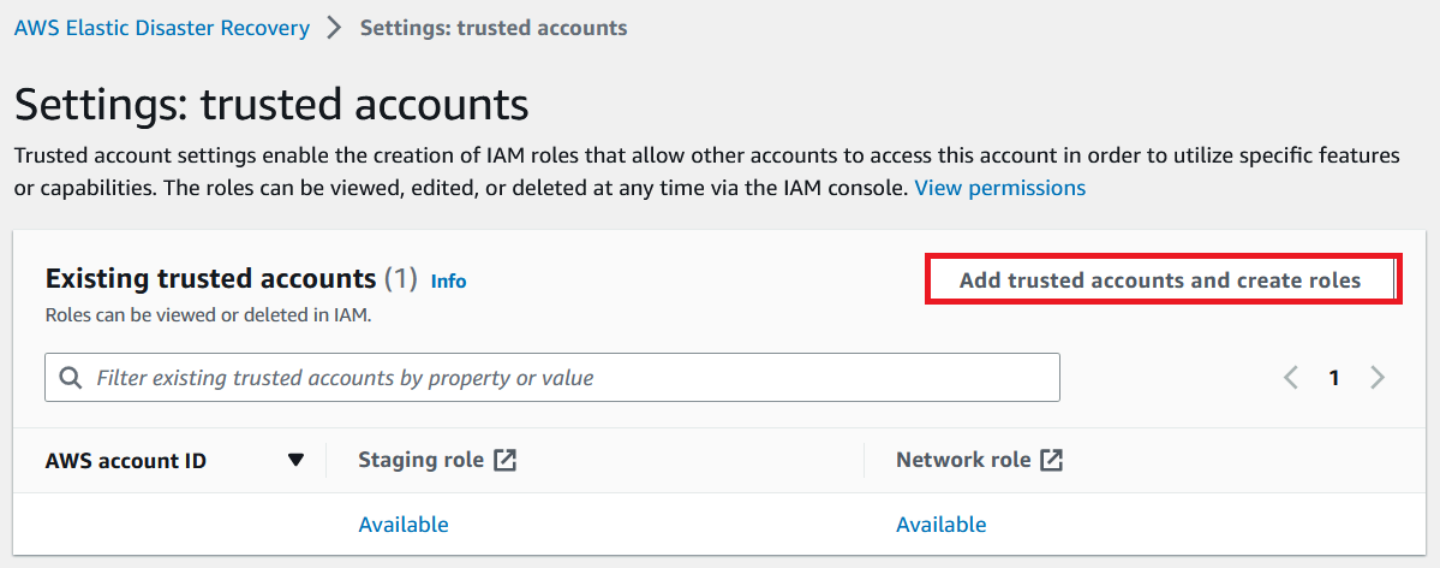
IAM roles are automatically created for the target accounts on the **Settings: trusted account** page of AWS DRS in the staging account. These roles are used to allow the target account to extend source servers from the staging account and to recover them in the target account.



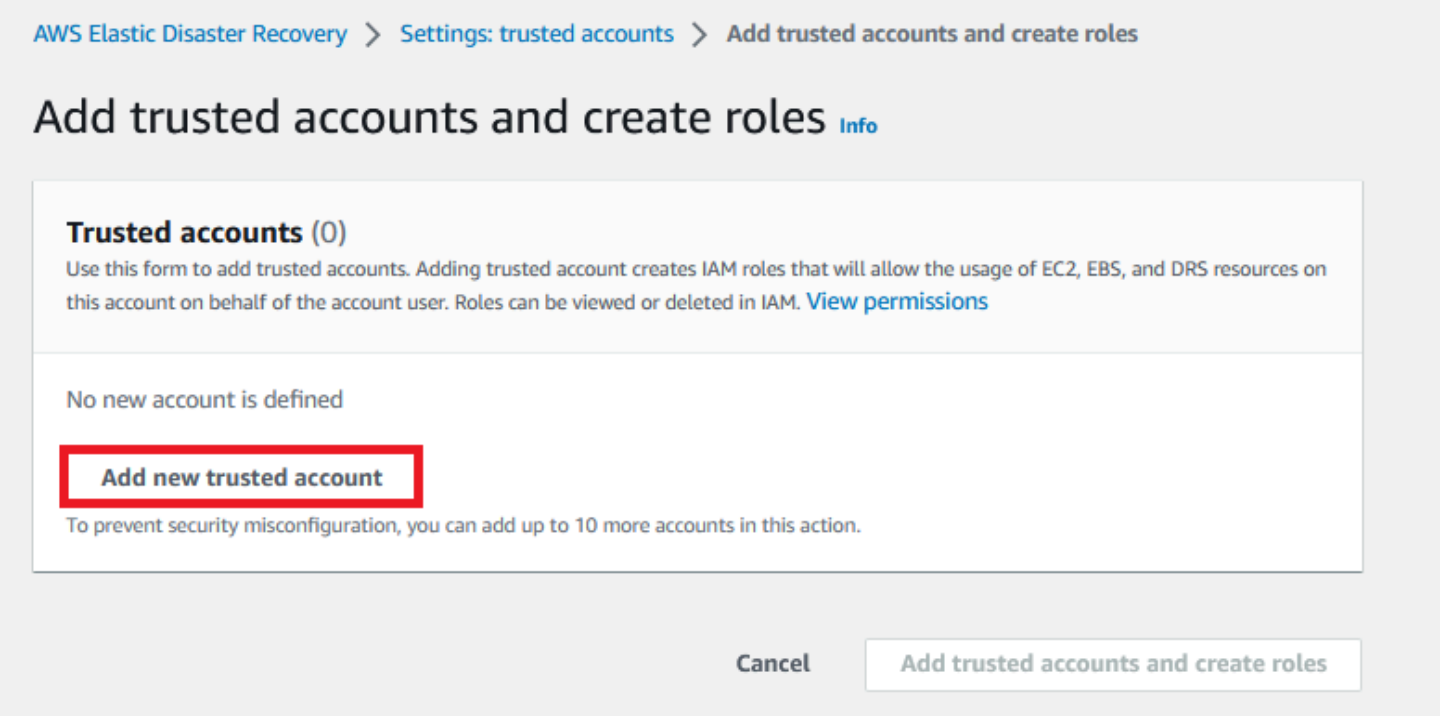
Under trusted accounts settings, you will find the **Existing trusted accounts** category. Here, you can manage existing staging accounts with links to IAM roles associated for each account. These IAM roles are used to associate the staging account with the trusted account.



Use **Add trusted accounts and create roles** to add roles for any trusted account you plan to use.



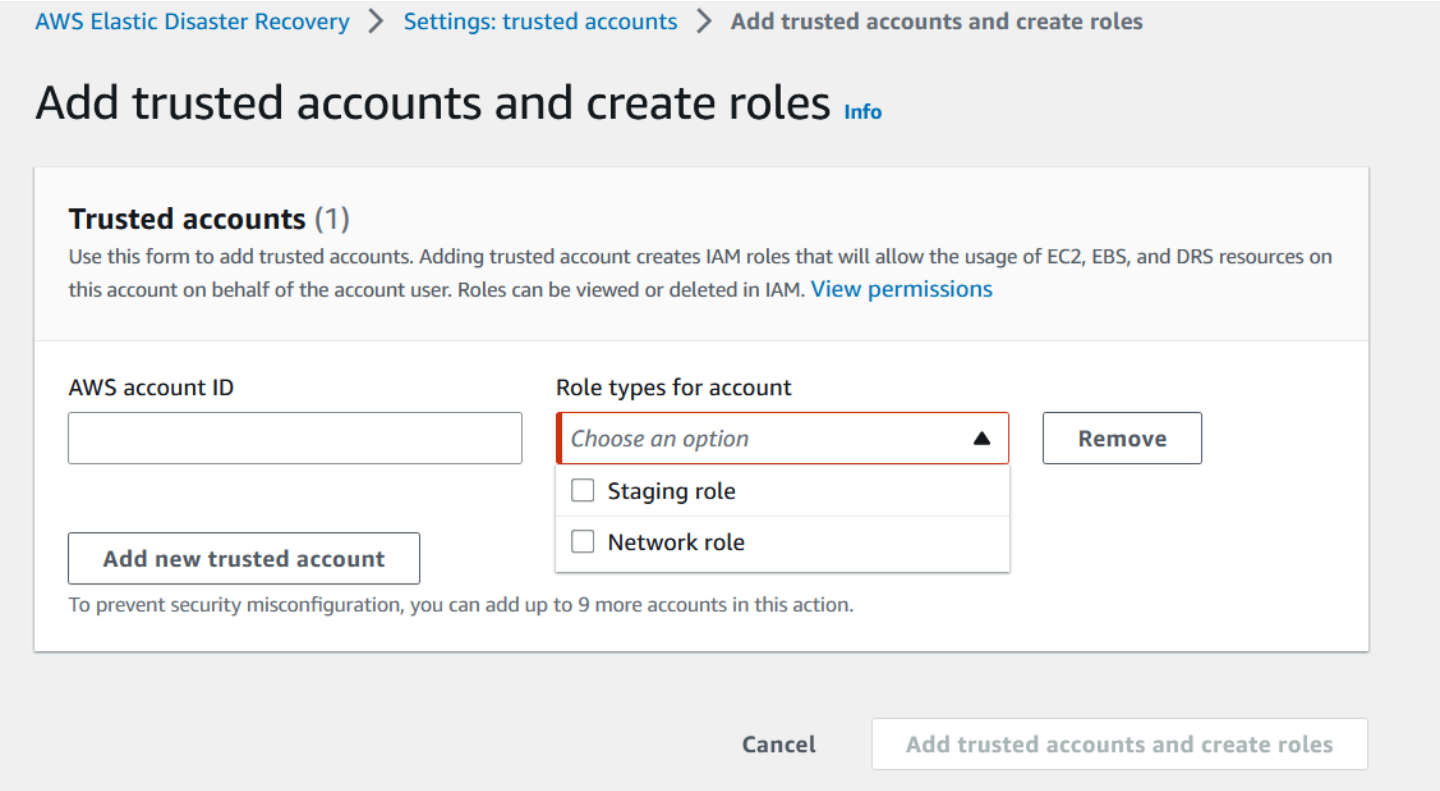
On the **Add trusted accounts and create roles** page, choose **Add new trusted account**.



Enter the AWS account IDs of the target account. This will automatically generate a service IAM role that will allow the use of Amazon EC2, Amazon EBS and AWS DRS resources in the staging account on behalf of a target account's user.

Choose **Add new trusted account** to add more than one trusted account at once. You can add up to 10 trusted accounts at once.

Once you have added your accounts, choose **Add trusted accounts and create roles**.



Share the EBS encryption key with the target account

Sharing the EBS encryption key is mandatory only if your volumes are encrypted.

In order for the target account to be able to successfully read the EBS snapshots of the replication servers in the staging account, you must share the EBS encryption key configured in the staging account with the target account. This can be done by following the instructions in the Allowing users in other accounts to use a KMS key documentation.

You must set the following statement policies on your staging account's KMS key in order to be able to recover extended source servers on a specific target account. Ensure that you properly assign the `$STAGING_ACCOUNT_ID` and `$TARGET_ACCOUNT_ID` and `$REGION` variables.

Note that if this is a key you already have been using, you will need to attach this policy in addition to the existing one.

```
[
 {
  "Sid": "Allow access to share snapshots with a target account",
  "Effect": "Allow",
  "Principal": {
   "AWS": [
    "arn:aws:iam::$STAGING_ACCOUNT_ID:role/service-role/DRSStagingAccountRole_
$TARGET_ACCOUNT_ID"
   ]
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
   "StringEquals": {
    "kms:CallerAccount": "$STAGING_ACCOUNT_ID",
    "kms:ViaService": "ec2.$REGION.amazonaws.com"
   }
  }
 },
 {
  "Sid": "Allow a target account to use this KMS key via EC2",
  "Effect": "Allow",
  "Principal": {
   "AWS": "arn:aws:iam::$TARGET_ACCOUNT_ID:root"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
   "StringEquals": {
    "kms:CallerAccount": "$TARGET_ACCOUNT_ID",
    "kms:ViaService": "ec2.$REGION.amazonaws.com"
   }
  }
 }
]
```

Managing extended source servers within the target AWS account

In order to manage extended source servers within the target accounts, you should extend source servers you wish to recover in the target account into that account from any staging account.

Initializing the target account

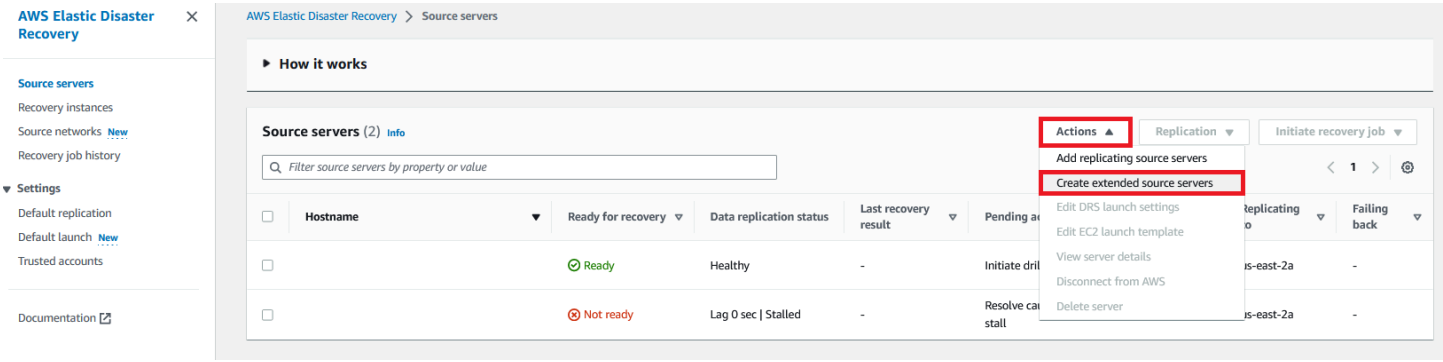
If you plan on using an AWS account and AWS Region in which AWS DRS has not been initialized, the service can be initialized either from the AWS DRS console or from the API. If you choose to initialize the service from the API, using the InitializeService API, you can skip creating the default replication settings if you plan to use the service only from the API and do not plan to have source servers replicating on this account. If you initialize the service through the AWS DRS console, the initialization wizard still creates the default replication settings, and the wizard will also run if you use the console after initializing a service without creating the default replication settings.

Create extended source servers

In order to add your source servers from your staging accounts into your target account, you must extend the source servers from the staging account to the target account.

> **Important**
>
> You must repeat the steps below for every staging account you want to associate with the target account.

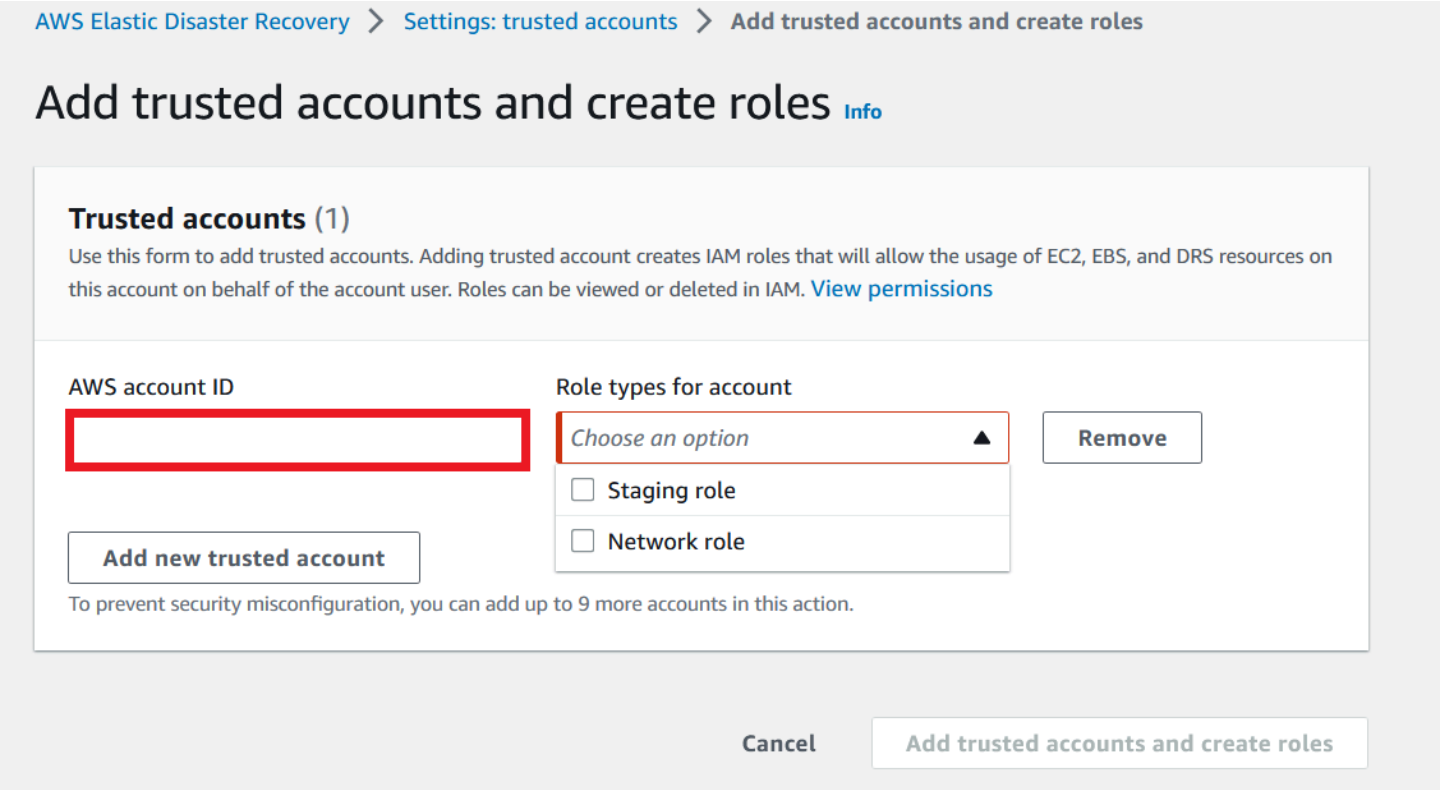
Navigate to the **Source servers** view within the target account, open the **Actions** menu, and choose **Create extended source servers**. This will extend the source servers from the staging accounts into the target account, allowing you to manage all of the source servers in your staging accounts through a single target account.



The **Create extended source servers** wizard will appear. The wizard is composed of three steps:

- Configure access
- Extend source servers
- Review and create

First you must configure access. Under **Configure access > Staging account configuration**, enter the ID of the staging account in which you created the IAM roles in the previous step. Choose **Next**.



Select the source servers you want to extend from the staging account into the target account by checking the box to the left of the source server **Hostname**. This will create a new source server resource that will inherit the replication configuration and points in time from the base source server in the staging account. Only source servers that have not already been extended will be shown. Once you have selected your source servers, choose **Next**.

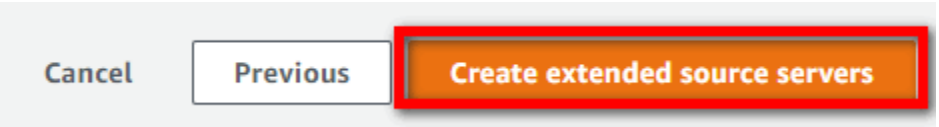
> **Note**
>
> The Extend source servers page will only show 30 source servers per page. If you have many source servers in your staging account and want to extend them all to your target account, then choose the **Select all X source servers** option.

Note

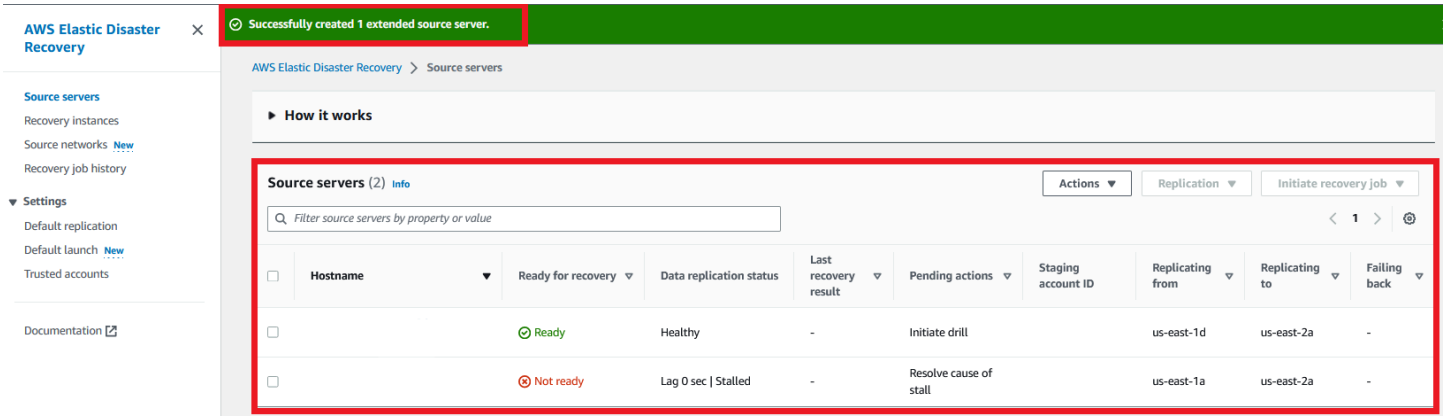
You can filter the source servers shown by Hostname or Source Server ID through the **Filter. ...** box.

You can add **Tags** if you wish to or you can skip this step. Add tags if you wish to, and then choose **Next**. Learn more about adding tags in AWS DRS.

Finally, **Review and create** the extended source servers. Review the information on the page and then choose **Create extended source servers**.

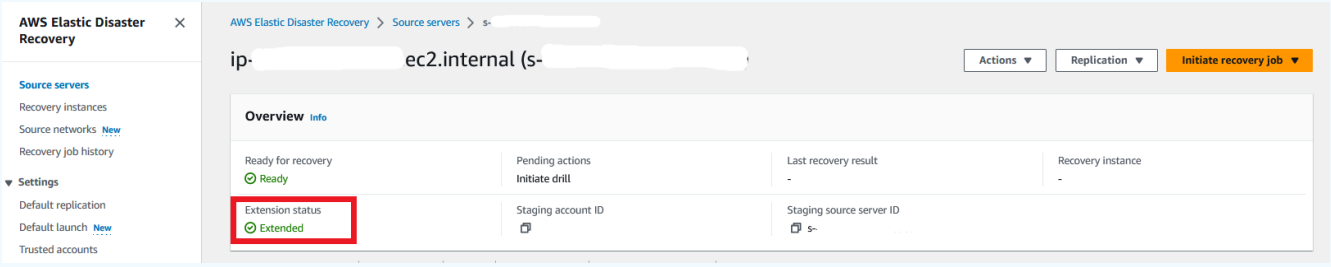


The AWS DRS console show the **Successfully created X extended source servers** message and you will see your extended source servers in your target account.



Note

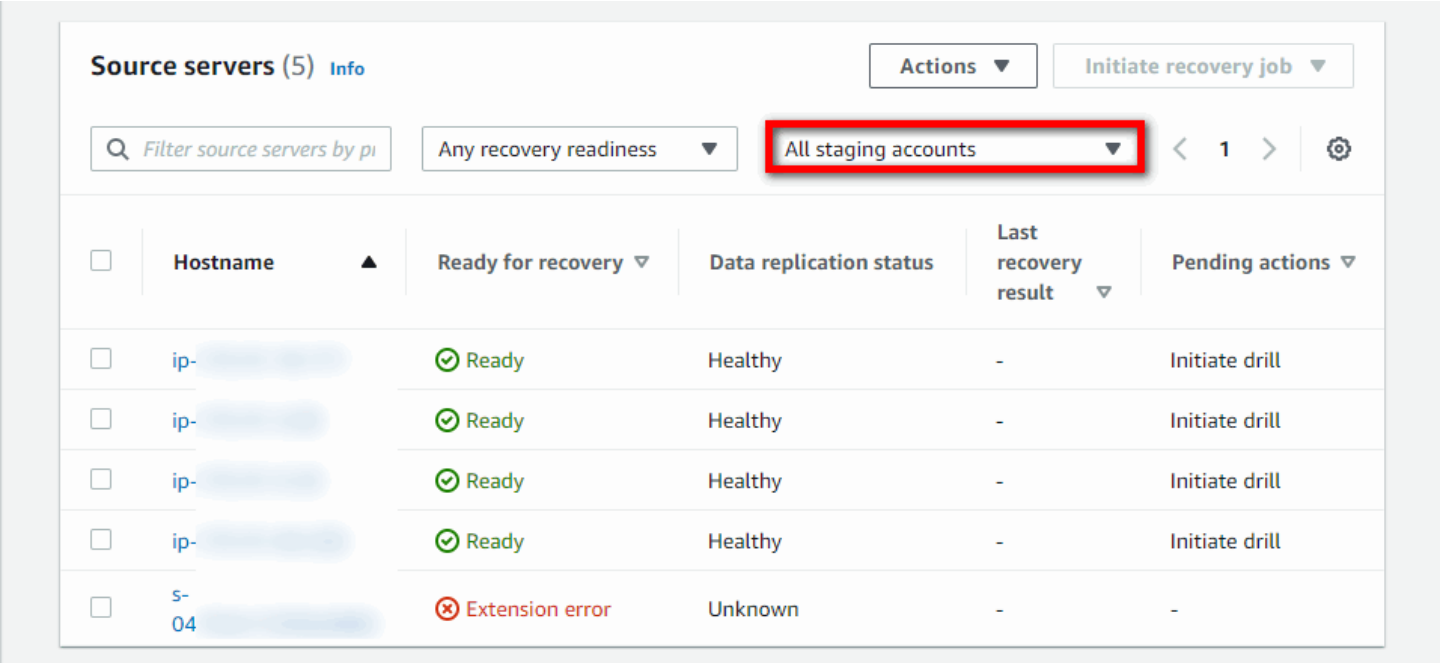
Extended source servers will show **Extended** under **Extension status** in the **Source server details > Overview section.**



Manage source servers

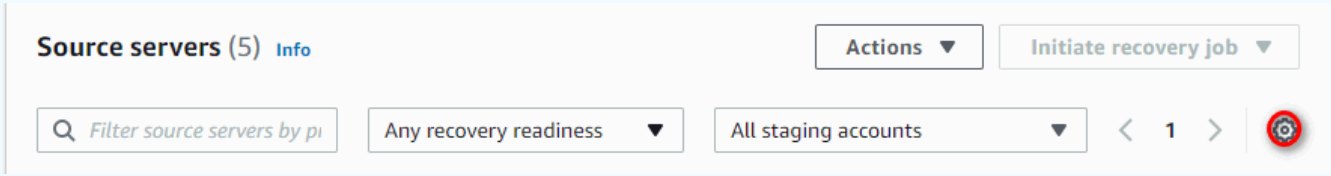
Once you have extended your source servers from every staging account into the target account, you can manage the source servers from the target account.

Source servers are grouped by staging account. You can choose the staging account under the **Source servers** header.

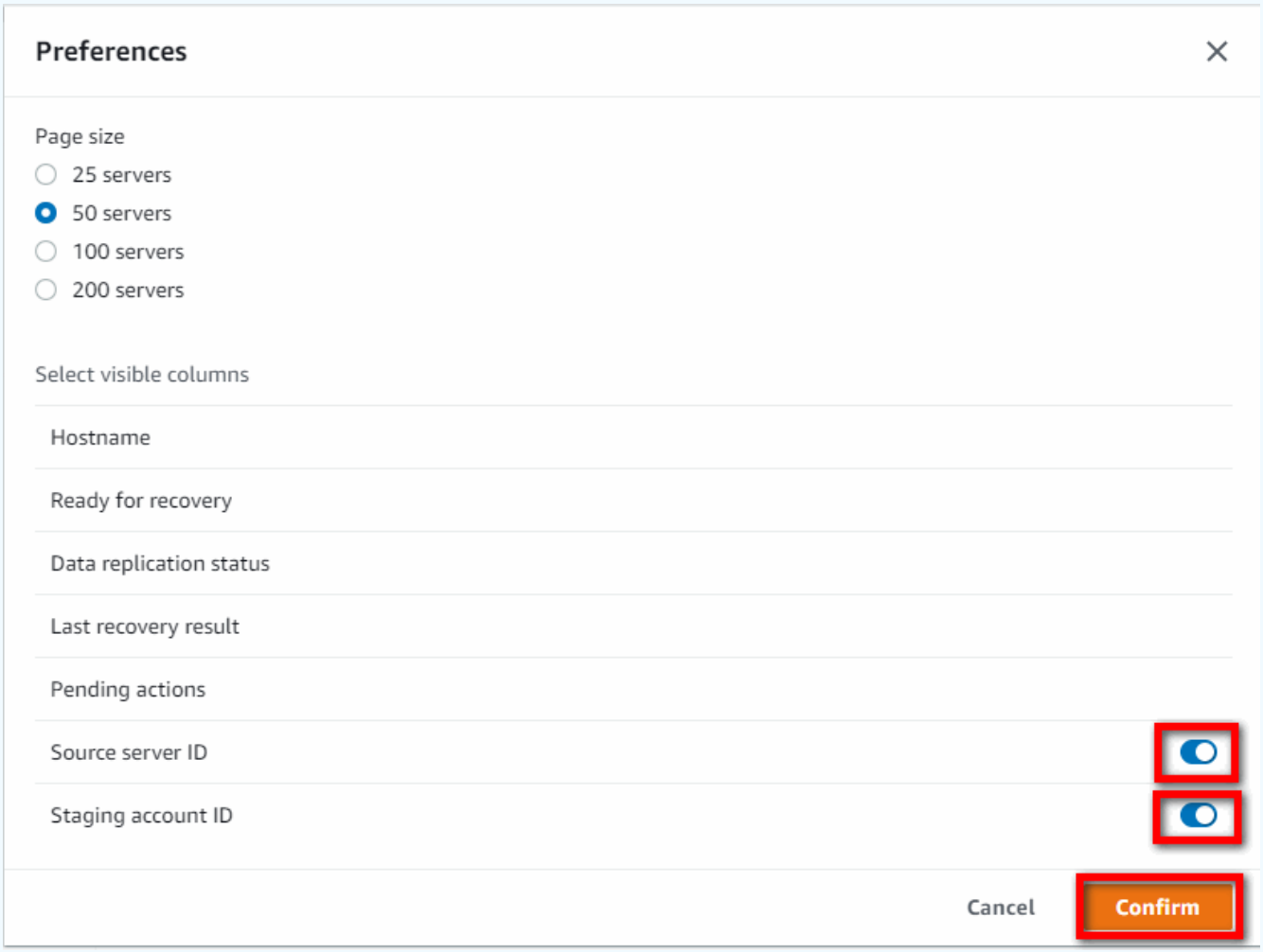


Note

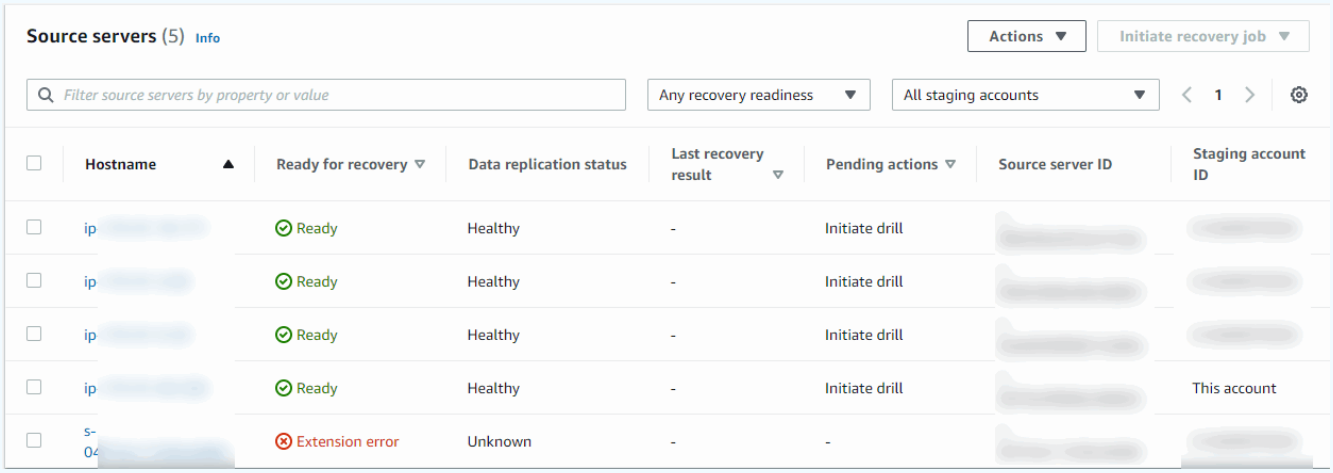
If you want to see the Source Server ID and Staging Account ID of each source server in the **source servers** view, you can add those columns by choosing the **Preferences** wheel.



From **Preferences**, toggle the **Source server ID** and **Staging account ID** options and choose **Confirm**.



You will now see these columns added to the main source servers view.



You can now perform normal AWS DRS operations for the source servers, including:

- Configuring individual source server settings
- Configure launch settings
- Launching Drill and Recovery instances
- Performing a failover and failback

> **Note**
>
> You will not be able to edit the replication settings and disks for individual extended source servers from the target account. You must edit these from the staging account.

> **Note**
>
> The AWS Replication Agent will stop replicating automatically after failing back from a recovery instance of an extended source server to the original server.

Removing an extended source server

If you need to delete an extended source server, do this from the account it was extended to (and where it is no longer needed). Deleting an extended source server has no effect on the replication of the source server into the staging account.

You can always recreate an extended source server after it was deleted, using 'Create extended source server' on the same staging source server.

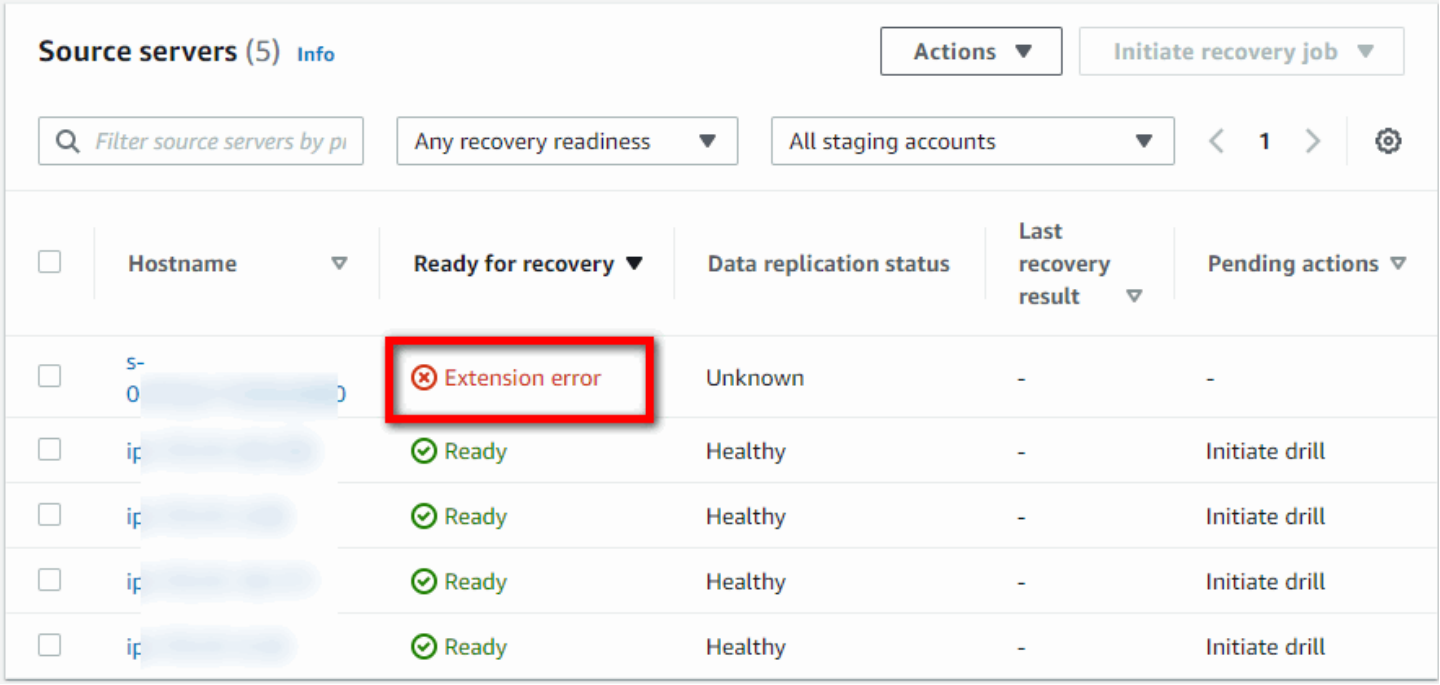
> **Note**
>
> To delete the staging account source server (the source server that is used to replicate data into the staging account), it must first be disconnected, and then it can be deleted from the staging account.

> **Note**
>
> You cannot change the staging account (the account where replication takes place) for a server that has been extended. For example, if a source server is replicating into account A,

and has been extended into account B, you cannot reinstall the agent on the source server to replicate into an account that is different than A while it is still extended into account B.

Troubleshooting

If your source server shows **Extension error** under the **Ready for recovery** category in the target account, then the source server was most likely deleted from the staging account.



Navigate to the source server details page by choosing the server's hostname in order to see the extension error details.

Working with AWS DRS and AWS Outposts

AWS Elastic Disaster Recovery now supports AWS Outpost racks. AWS Outposts require specific configurations in the default replication settings, individual source server replication settings, default launch templates and source server launch templates to work. The following sections explain how to use AWS Outpost racks with DRS, how to troubleshoot key AWS Outpost issues, and how to monitor AWS Outposts with DRS. Learn more about AWS Outposts.

Considerations when using AWS Outposts:

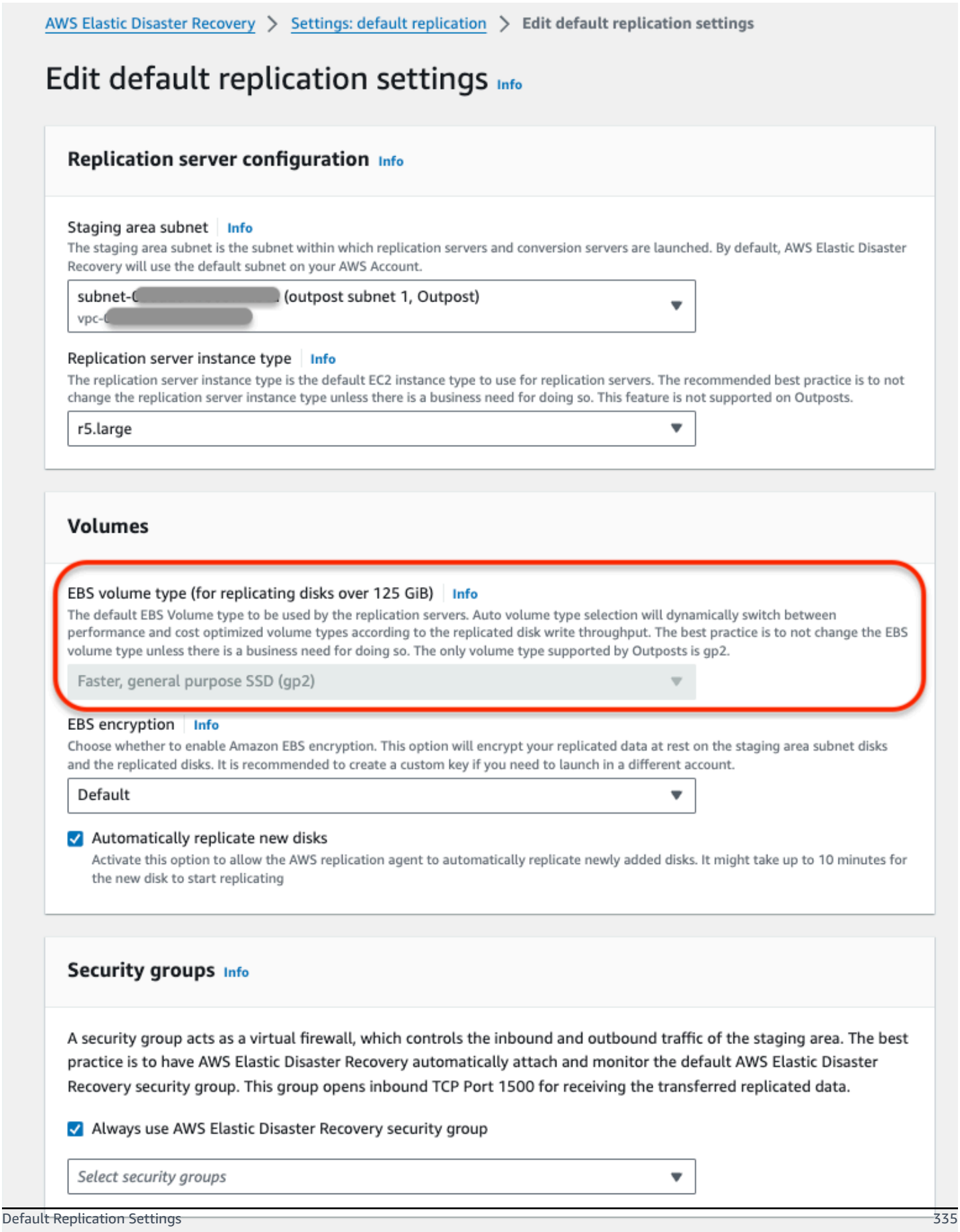
- To use an AWS Outpost, you need to have a subnet within the Outpost selected to be used for replication or recovery. If you select an Outpost subnet for replication, a subnet in the same Outpost must be selected for recovery.
- Once a subnet within an AWS Outpost is selected for replication, all replication resources (including replication servers, conversion servers, EBS volumes, and snapshots) will be created and saved within the Outpost.
- If a subnet within an AWS Outpost is selected for launch, then the recovery instances, EBS volumes and snapshots used for recovery are created and saved within the Outpost.

Default Replication Settings

Selecting an Outpost subnet on the *Settings: default replication* page means that newly added source servers will automatically start replicating to the Outpost.

Subnets that are within Outposts will have the word "Outpost" appended after the subnet name in the subnet selector drop down.

Once the Subnet is chosen, you will have to select the replication server instance type. Only instance types that are supported by the chosen Outpost will be shown.


AWS Elastic Disaster Recovery > Settings: default replication > Edit default replication settings
Edit default replication settings Info
Replication server configuration Info
Staging area subnet Info
The staging area subnet is the subnet within which replication servers and conversion servers are launched. By default, AWS Elastic Disaster Recovery will use the default subnet on your AWS Account.
subnet- (outpost subnet 1, Outpost)
vpc-
Replication server instance type Info
The replication server instance type is the default EC2 instance type to use for replication servers. The recommended best practice is to not change the replication server instance type unless there is a business need for doing so. This feature is not supported on Outposts.
r5.large
Volumes
EBS volume type (for replicating disks over 125 GiB) Info
The default EBS Volume type to be used by the replication servers. Auto volume type selection will dynamically switch between performance and cost optimized volume types according to the replicated disk write throughput. The best practice is to not change the EBS volume type unless there is a business need for doing so. The only volume type supported by Outposts is gp2.
Faster, general purpose SSD (gp2)
EBS encryption Info
Choose whether to enable Amazon EBS encryption. This option will encrypt your replicated data at rest on the staging area subnet disks and the replicated disks. It is recommended to create a custom key if you need to launch in a different account.
Default
Automatically replicate new disks
Activate this option to allow the AWS replication agent to automatically replicate newly added disks. It might take up to 10 minutes for the new disk to start replicating
Security groups Info
A security group acts as a virtual firewall, which controls the inbound and outbound traffic of the staging area. The best practice is to have AWS Elastic Disaster Recovery automatically attach and monitor the default AWS Elastic Disaster Recovery security group. This group opens inbound TCP Port 1500 for receiving the transferred replicated data.
Always use AWS Elastic Disaster Recovery security group
Select security groups

Outposts only support GP2 disks. As such, you will not be able to change the default disk type in the replication settings.

All Outpost volumes must be encrypted, and so you must select an encryption key when a subnet within an Outpost is selected. You will not be able to select not to encrypt. Ensure that you choose the correct volume encryption key you want to use. You can use the default KMS key for your account, or select a customer managed key (CMK).

Other replication settings can be set normally. Learn more about default replication settings.

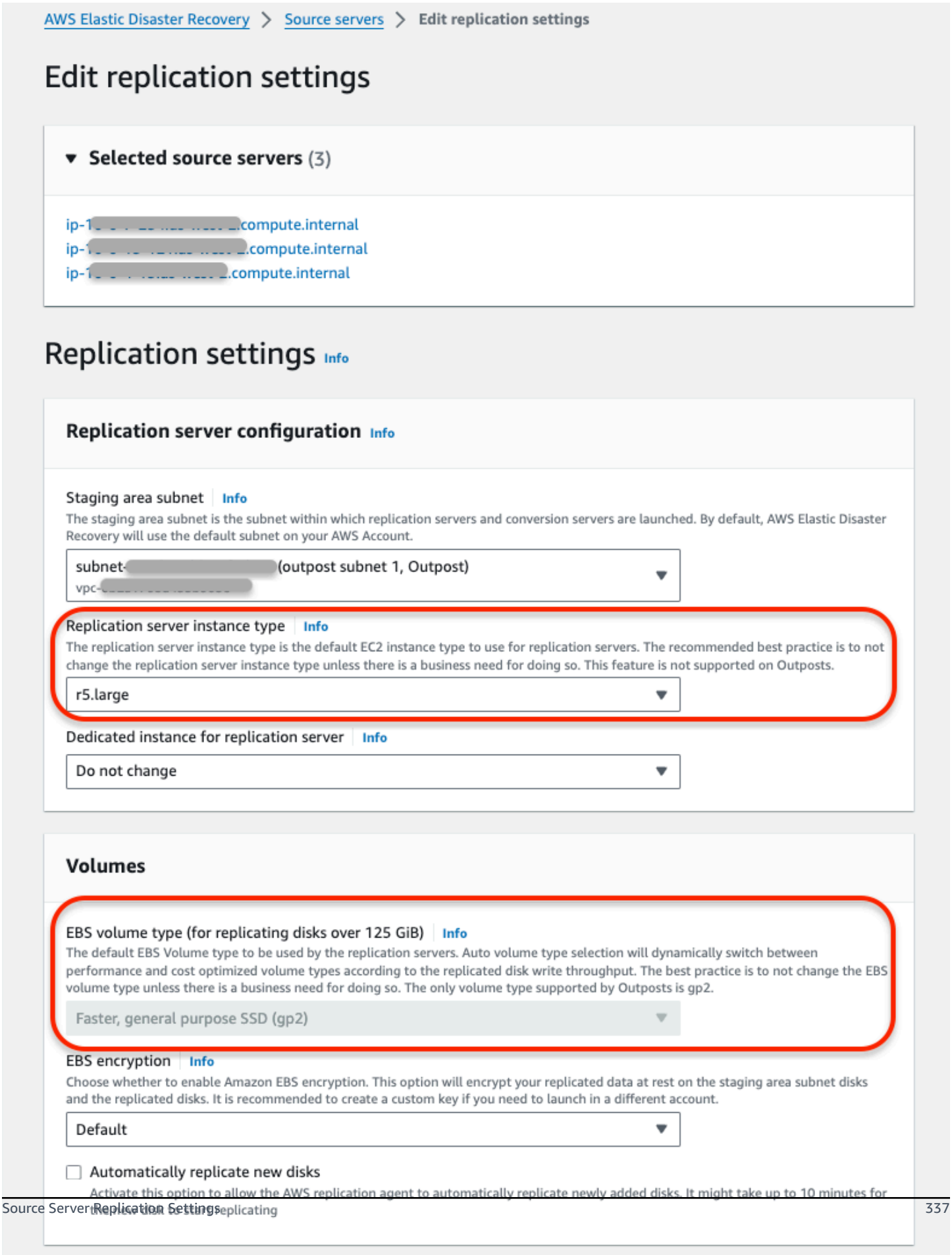
> **Note**
>
> When selecting an Outpost subnet in your replication settings, ensure that your launch template specifies a subnet on the same Outpost. Not doing so may result in an error during recovery.

Source Server Replication Settings

Use these settings to have a specific source server or multiple source servers replicate into an AWS Outpost by selecting a subnet within an AWS Outpost.

Subnets that are within Outposts will have the word "Outpost" appended after the subnet name in the subnet selector drop down.

Once the Subnet is chosen, you will have to select the replication server instance type. Only instance types that are supported by the chosen Outpost will be shown.


AWS Elastic Disaster Recovery > Source servers > Edit replication settings
Edit replication settings
Selected source servers (3)
ip-1[redacted].compute.internal
ip-1[redacted].compute.internal
ip-1[redacted].compute.internal
Replication settings Info
Replication server configuration Info
Staging area subnet Info
The staging area subnet is the subnet within which replication servers and conversion servers are launched. By default, AWS Elastic Disaster Recovery will use the default subnet on your AWS Account.
subnet-[redacted] (outpost subnet 1, Outpost)
vpc-[redacted]
Replication server instance type Info
The replication server instance type is the default EC2 instance type to use for replication servers. The recommended best practice is to not change the replication server instance type unless there is a business need for doing so. This feature is not supported on Outposts.
r5.large
Dedicated instance for replication server Info
Do not change
Volumes
EBS volume type (for replicating disks over 125 GiB) Info
The default EBS Volume type to be used by the replication servers. Auto volume type selection will dynamically switch between performance and cost optimized volume types according to the replicated disk write throughput. The best practice is to not change the EBS volume type unless there is a business need for doing so. The only volume type supported by Outposts is gp2.
Faster, general purpose SSD (gp2)
EBS encryption Info
Choose whether to enable Amazon EBS encryption. This option will encrypt your replicated data at rest on the staging area subnet disks and the replicated disks. It is recommended to create a custom key if you need to launch in a different account.
Default
Automatically replicate new disks
Activate this option to allow the AWS replication agent to automatically replicate newly added disks. It might take up to 10 minutes for
[redacted] replicating

Outposts only support GP2 disks. As such, you will not be able to change the default disk type in the replication settings.

All Outpost volumes must be encrypted and you must select an encryption key when a subnet within an Outpost is selected. You will not be able to select *not to encrypt*. Ensure that you choose the correct volume encryption key you want to use. You can use the default KMS key for your account or select a customer managed key (CMK).

Other replication settings can be set normally. Learn more about replication settings.

Note

You cannot edit the replication settings of multiple source servers if some of the source servers are replicating to Outpost subnets and others are replicating to non-Outpost subnets.

Note

When selecting an Outpost subnet in your replication settings, ensure that your launch template specifies a subnet on the same Outpost. Not doing so may result in an error during recovery.

Default Launch Template

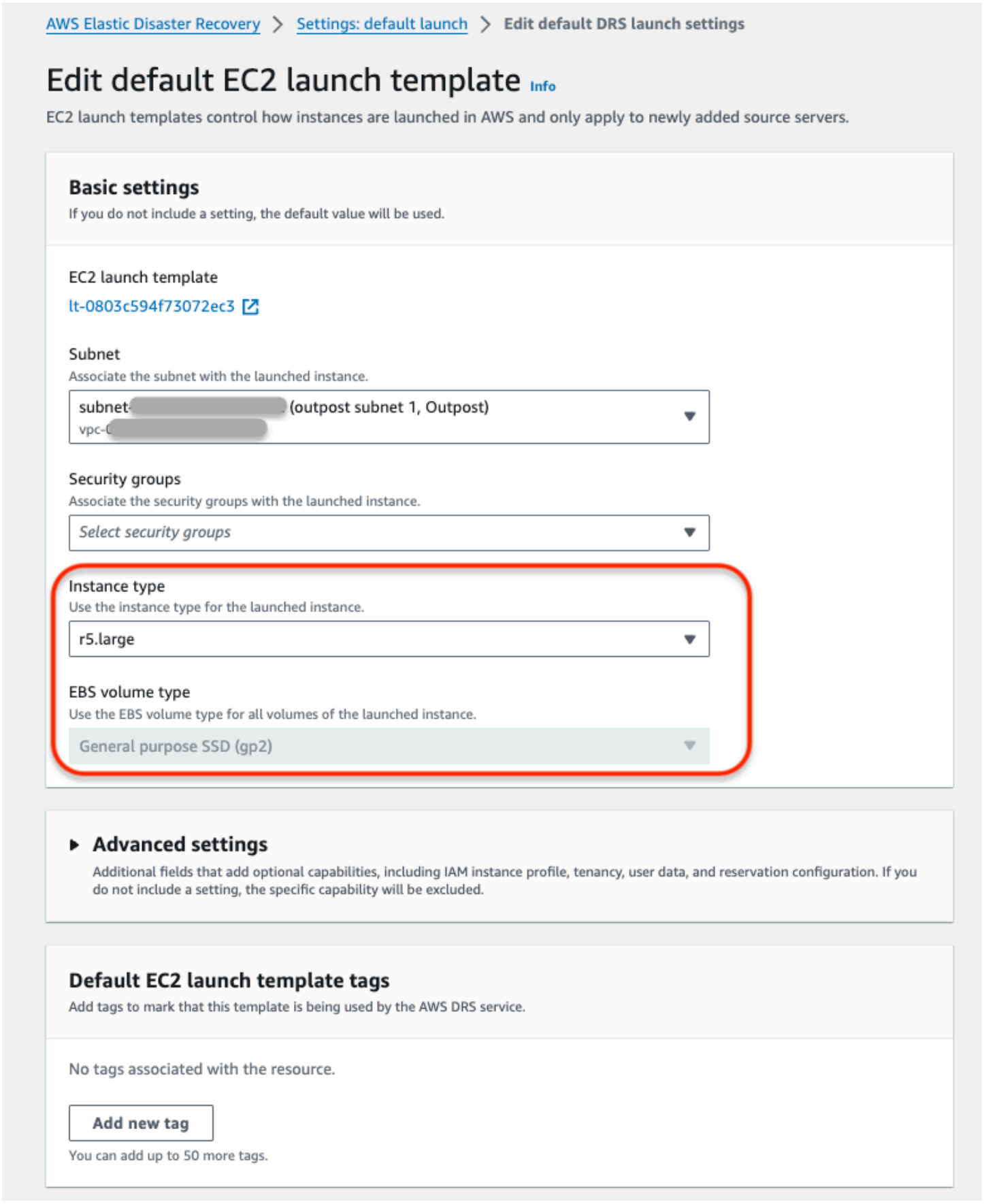
Selecting an Outpost subnet in the default launch template means that newly added source servers will launch into the Outpost. You must select the instance type from the list of available instance types on your Outpost.

Note

When working with Outposts, if you selected a subnet within an AWS Outpost in the default replication settings, you must also choose a subnet within the same Outpost in the default launch template. Otherwise, newly added source servers will replicate into an Outpost but launch outside of an Outpost which may result in an error.

> **Note**
>
> Network replication and recovery does not create a subnet within the Outpost.

Outposts only support GP2 disks. As such, you will not be able to change the default disk type per volume in the default launch template.

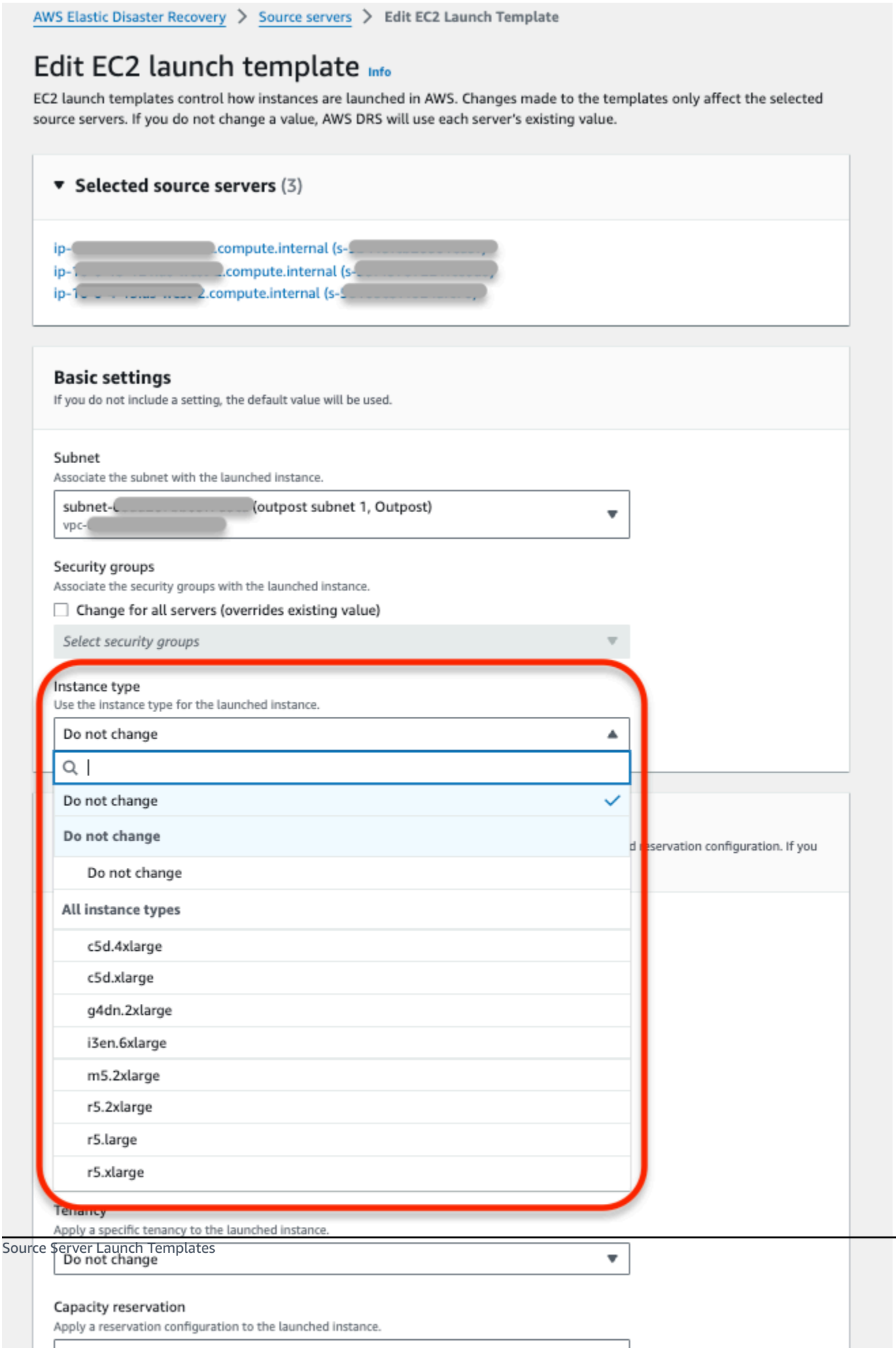

AWS Elastic Disaster Recovery > Settings: default launch > Edit default DRS launch settings
Edit default EC2 launch template Info
EC2 launch templates control how instances are launched in AWS and only apply to newly added source servers.
Basic settings
If you do not include a setting, the default value will be used.
EC2 launch template
lt-0803c594f73072ec3
Subnet
Associate the subnet with the launched instance.
subnet- (outpost subnet 1, Outpost)
vpc-
Security groups
Associate the security groups with the launched instance.
Select security groups
Instance type
Use the instance type for the launched instance.
r5.large
EBS volume type
Use the EBS volume type for all volumes of the launched instance.
General purpose SSD (gp2)
Advanced settings
Additional fields that add optional capabilities, including IAM instance profile, tenancy, user data, and reservation configuration. If you do not include a setting, the specific capability will be excluded.
Default EC2 launch template tags
Add tags to mark that this template is being used by the AWS DRS service.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.

Source Server Launch Templates

When working with Outposts, if you selected a subnet within an AWS Outpost in the replication settings for any source server, you must also choose a subnet within the same Outpost in the launch template of that source server. You must select the instance type to launch into from the list of available instance types on your Outpost. Not doing so may result in an error during recovery.

> **Note**
>
> Network replication and recovery does not create a subnet within the Outpost.

Outposts only support GP2 disks. As such, you will not be able to change the default disk type per volume in the default launch template.

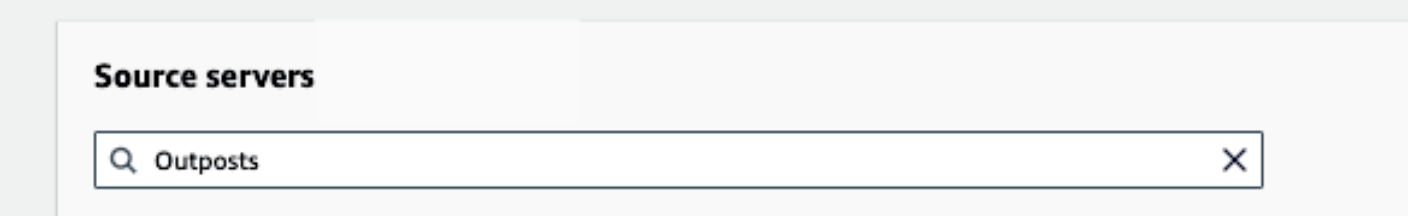

AWS Elastic Disaster Recovery > Source servers > Edit EC2 Launch Template
Edit EC2 launch template Info
EC2 launch templates control how instances are launched in AWS. Changes made to the templates only affect the selected source servers. If you do not change a value, AWS DRS will use each server's existing value.
Selected source servers (3)
Basic settings
If you do not include a setting, the default value will be used.
Subnet
Associate the subnet with the launched instance.
(outpost subnet 1, Outpost)
Security groups
Associate the security groups with the launched instance.
Change for all servers (overrides existing value)
Select security groups
Instance type
Use the instance type for the launched instance.
Do not change
Do not change
Do not change
Do not change
All instance types
c5d.4xlarge
c5d.xlarge
g4dn.2xlarge
i3en.6xlarge
m5.2xlarge
r5.2xlarge
r5.large
r5.xlarge
Apply a specific tenancy to the launched instance.
Do not change
Capacity reservation
Apply a reservation configuration to the launched instance.

> **Note**
>
> You cannot edit the launch templates of multiple source servers if some of the source servers are configured to launch to Outpost subnets and the others are configured to launch to non-Outpost subnets.

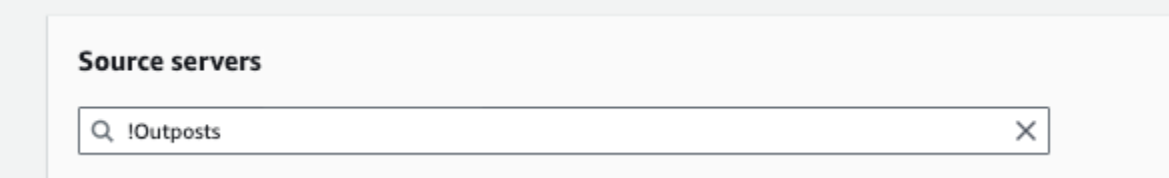
Source Server Page

You can find which of your servers are replicating into an AWS Outpost rack, as these are marked with the value (Outpost) in the **Replicating to** field. The following are search options for source servers replicating to an Outpost:

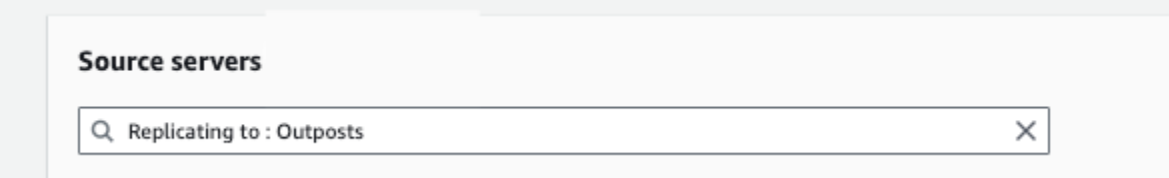
- You can enter the text "*Outpost*" in the search field to only display source servers that are replicating to an AWS Outpost rack.



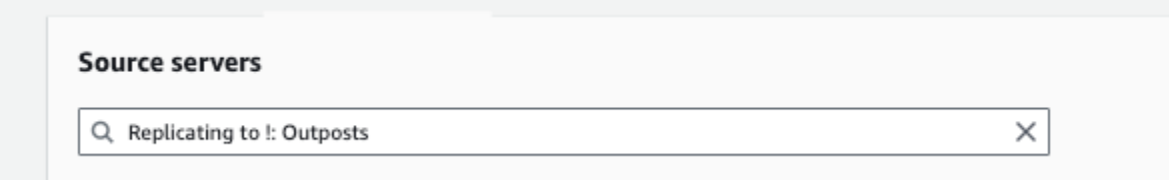
- You can enter the text "*!Outpost*" to only display source servers that are not replicating to an AWS Outpost rack.



- You can enter "*Replicating to: Outpost*" in the search field to only display source servers that are replicating to an AWS Outpost rack.



- You can enter "*Replicating to !: Outpost*" in the search field to only display source servers that are not replicating to an AWS Outpost rack.



Important Outpost Notes

Outpost Storage

AWS Outposts have a certain storage capacity. You should actively monitor available storage capacity in your Outpost. **If you run out of storage capacity, you may not be able to use DRS with that Outpost environment.** For example, if S3 runs out of capacity on the Outpost then DRS won't be able to create any new snapshots, so no new points in time will be created.

Be aware of EBS storage capacity on the Outpost. **DRS requires storage capacity roughly on a 2:1 ratio (excluding any launched instances on the Outpost).** For example, if you have source volumes amounting to a total of 1 TiB of storage, then DRS will require at least 1 TiB for replication resources, and another 1 TiB for conversion or recovery instance creation.

Replication and Launch Subnets

When using DRS with Outposts it is imperative that the subnet in replication setting and launch settings are on the same Outpost. If the subnets are not on the same Outpost, it may result in a failure during recovery.

Instance Types and Operating Systems (OSs)

> **Note**
>
> AWS Outposts only support Nitro instance types.

- **Linux** - RHEL 7.0+ and CentOS 7.0+
- **Windows** - Windows Server 2008 R2, Windows Server 2012 R2, Windows Server 2016, and higher.

Monitoring

You can monitor Outpost metrics for S3, EC2, and EBS capacity through CloudWatch. You can create a dashboard with these metrics:

- EBS Metrics: EBSVolumeTypeCapacityAvailability, EBSVolumeTypeCapacityUtilization
- EC2 Metrics: InstanceTypeCapacityUtilization, AvailableInstanceType_Count
- S3 Metrics: OutpostTotalBytes, OutpostFreeBytes

Learn more about CloudWatch S3 monitoring.

Learn more about CloudWatch Metrics for AWS Outposts.

Security in AWS Elastic Disaster Recovery

Topics

Overview

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs . To learn about the compliance programs that apply to AWS Elastic Disaster Recovery (AWS DRS), see AWS Services in Scope by Compliance Program .
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using AWS DRS. It shows you how to configure AWS DRS to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Elastic Disaster Recovery resources.

The customer is responsible for making sure that no misconfigurations are present during and after the recovery process, including:

1. The replication server should be accessed only from the CIDR range of the source servers. Proper security groups rules should be assigned to the replication server after it is created.
2. After the recovery, the customer should make sure that only allowed ports are exposed to the public internet.
3. Hardening of OS packages and other software deployed in the servers is completely under the customer's responsibility and we recommend the following:
   a. Packages should be up to date and free of known vulnerabilities.
   b. Only necessary OS/application services should be up and running.
4. Activating the Anti-DDOS protection (AWS Shield) in the customer's AWS Account to eliminate the risk of denial of service attacks on the replication servers as well as the migrated servers.

Identity and access management for AWS Elastic Disaster Recovery

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS resources. IAM allows you to create users and groups under your AWS account. You control the permissions that users have to perform tasks using AWS resources. You can use IAM for no additional charge.

By default, IAM users don't have permissions for AWS Elastic Disaster Recovery (AWS DRS) resources and operations. To allow IAM users to manage AWS DRS resources, you must create an IAM policy that explicitly grants them permissions, and attach the policy to the users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more information, see Policies and Permissions in the *IAM User Guide* guide.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using

credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

Authenticating with identities in AWS Elastic Disaster Recovery

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Grant permission to tag resources during creation

Some resource-creating Amazon DRS API actions allow you to specify tags when you create the resource. You can use resource tags to implement attribute-based control (ABAC).

To allow users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `drs:CreateSourceServerForDrs` for source server or `drs:CreateRecoveryInstanceForDrs` for Recovery instances. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `drs:TagResource` action to verify that users have permissions to create tags. Therefore, users must also have explicit permissions to use the `drs:TagResource` action.

In the IAM policy definition for the `drs:TagResource` action, use the Condition element with the `drs:CreateAction` condition key to give tagging permissions to the action that creates the resource.

The following example demonstrates a policy that allows an agent installer to create a source server or recover instance and apply any tags to the resource on creation. The installer is not permitted to tag any existing resources (it cannot call the `drs:TagResource` action directly).

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
```

```
    "drs:GetAgentInstallationAssetsForDrs",
    "drs:SendClientLogsForDrs",
    "drs:CreateSourceServerForDrs",
    "drs:CreateRecoveryInstanceForDrs",
    "drs:DescribeRecoveryInstances"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": "drs:TagResource",
   "Resource": "arn:aws:drs:*:*:source-server/*",
   "Condition": {
    "StringEquals": {
     "drs:CreateAction": "CreateSourceServerForDrs"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": "drs:TagResource",
   "Resource": "arn:aws:drs:*:*:recovery-instance/*",
   "Condition": {
    "StringEquals": {
     "drs:CreateAction": "CreateRecoveryInstanceForDrs"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": "drs:IssueAgentCertificateForDrs",
   "Resource": "arn:aws:drs:*:*:source-server/*"
  }
 ]
}
```

The `drs:TagResource` action is only evaluated if tags are applied during the resource-creating action. Therefore, an installer that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `drs:TagResource` action if no tags are specified in the request. However, if the installer attempts to create a resource with tags, the request fails if the installer does not have permissions to use the `drs:TagResource` action.

AWS managed policies for AWS Elastic Disaster Recovery

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the*IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the*IAM User Guide*. AWS Elastic Disaster Recovery read-only permissions are included in the general IAM ReadOnlyAccess policy.

Topics

- AWS managed policy: AWSElasticDisasterRecoveryServiceRolePolicy
- AWS managed policy: AWSElasticDisasterRecoveryStagingAccountPolicy
- AWS managed policy: AWSElasticDisasterRecoveryStagingAccountPolicy_v2
- AWS managed policy: AWSElasticDisasterRecoveryEc2InstancePolicy
- AWS managed policy: AWSElasticDisasterRecoveryCrossAccountReplicationPolicy
- AWS managed policy: AWSElasticDisasterRecoveryNetworkReplicationPolicy
- AWS managed policy: AWSElasticDisasterRecoveryLaunchActionsPolicy
- AWS managed policy: AWSElasticDisasterRecoveryConsoleFullAccess_v2
- Elastic Disaster Recovery updates for AWS managed policies

AWS managed policy: AWSElasticDisasterRecoveryAgentPolicy

This policy gives the AWS Replication Agent, which is used with AWS Elastic Disaster Recovery (AWS DRS) to replicate source servers to AWS, permissions to communicate with AWS DRS to receive instructions and to send logs and metrics. We do not recommend that you attach this policy to your users or roles.

Permissions details

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DRSAgentPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:SendAgentMetricsForDrs",
    "drs:SendAgentLogsForDrs",
    "drs:UpdateAgentSourcePropertiesForDrs",
    "drs:UpdateAgentReplicationInfoForDrs",
    "drs:UpdateAgentConversionInfoForDrs",
    "drs:GetAgentCommandForDrs",
    "drs:GetAgentConfirmedResumeInfoForDrs",
    "drs:GetAgentRuntimeConfigurationForDrs",
    "drs:UpdateAgentBacklogForDrs",
    "drs:GetAgentReplicationInfoForDrs",
```

```
    "drs:IssueAgentCertificateForDrs"
   ],
   "Resource": "arn:aws:drs:*:*:source-server/${aws:SourceIdentity}"
  },
  {
   "Sid": "DRSAgentPolicy2",
   "Effect": "Allow",
   "Action": [
    "drs:GetAgentInstallationAssetsForDrs"
   ],
   "Resource": "*"
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryAgentInstallationPolicy

This policy allows installing the AWS Replication Agent, which is used with AWS Elastic Disaster Recovery (AWS DRS) to recover external servers to AWS. Attach this policy to your users or roles whose credentials you provide during the installation step of the AWS Replication Agent.

Permissions details

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DRSAgentInstallationPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:GetAgentInstallationAssetsForDrs",
    "drs:SendClientLogsForDrs",
    "drs:SendClientMetricsForDrs",
    "drs:CreateSourceServerForDrs",
    "drs:CreateRecoveryInstanceForDrs",
    "drs:DescribeRecoveryInstances",
    "drs:CreateSourceNetwork"
   ],
   "Resource": "*"
  },
  {
```

```
    "Sid": "DRSAgentInstallationPolicy2",
    "Effect": "Allow",
    "Action": "drs:TagResource",
    "Resource": "arn:aws:drs:*:*:source-server/*",
    "Condition": {
     "StringEquals": {
      "drs:CreateAction": "CreateSourceServerForDrs"
     }
    }
   },
   {
    "Sid": "DRSAgentInstallationPolicy3",
    "Effect": "Allow",
    "Action": "drs:TagResource",
    "Resource": "arn:aws:drs:*:*:source-server/*",
    "Condition": {
     "StringEquals": {
      "drs:CreateAction": "CreateRecoveryInstanceForDrs"
     }
    }
   },
   {
    "Sid": "DRSAgentInstallationPolicy4",
    "Effect": "Allow",
    "Action": "drs:TagResource",
    "Resource": "arn:aws:drs:*:*:source-network/*",
    "Condition": {
     "StringEquals": {
      "drs:CreateAction": "CreateSourceNetwork"
     }
    }
   },
   {
    "Sid": "DRSAgentInstallationPolicy5",
    "Effect": "Allow",
    "Action": "drs:IssueAgentCertificateForDrs",
    "Resource": "arn:aws:drs:*:*:source-server/*"
   }
  ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryConversionServerPolicy

This policy is attached to the AWS Elastic Disaster Recovery conversion server's instance role.

This policy allows AWS Elastic Disaster Recovery (AWS DRS) Conversion Servers, which are EC2 instances launched by AWS DRS, to communicate with the DRS service. An IAM role with this policy is attached (as an EC2 instance Profile) by DRS to the DRS Conversion Servers, which are automatically launched and terminated by DRS, when needed. We do not recommend that you attach this policy to your users or roles. DRS Conversion Servers are used by Elastic Disaster Recovery when users choose to recover source servers using the AWS DRS console, CLI, or API.

This policy is attached to the AWS Elastic Disaster Recovery Conversion server's instance role. This policy allows the AWS DRS Conversion Server, which are EC2 instances launched by AWS DRS, to communicate with AWS DRS. An IAM role with this policy is attached (as an EC2 Instance Profile) by AWS DRS to the AWS DRS conversion servers, which are automatically launched and terminated by AWS DRS, when needed. We do not recommend that you attach this policy to your users or roles. AWS DRS conversion servers are used by Elastic Disaster Recovery when users choose to recover source servers using the AWS DRS console, CLI, or API.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
          "Sid": "DRSConversionServerPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:SendClientMetricsForDrs",
    "drs:SendClientLogsForDrs"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSConversionServerPolicy2",
   "Effect": "Allow",
   "Action": [
    "drs:GetChannelCommandsForDrs",
    "drs:SendChannelCommandResultForDrs"
   ],
   "Resource": "*"
```

```
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryFailbackPolicy

This policy allows using the AWS Elastic Disaster Recovery Failback Client, which is used to failback recovery instances back to your original source infrastructure. We do not recommend that you attach this policy to your users or roles.

This policy is used by AWS Elastic Disaster Recovery to refresh credentials for the AWS Elastic Disaster Recovery Failback Client. We do not recommend that you attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
          "Sid": "DRSFailbackPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:SendClientMetricsForDrs",
    "drs:SendClientLogsForDrs"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSFailbackPolicy2",
   "Effect": "Allow",
   "Action": [
    "drs:GetChannelCommandsForDrs",
    "drs:SendChannelCommandResultForDrs"
   ],
   "Resource": "*"
```

```
  },
  {
          "Sid": "DRSFailbackPolicy3",
   "Effect": "Allow",
   "Action": [
    "drs:DescribeReplicationServerAssociationsForDrs",
    "drs:DescribeRecoveryInstances"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSFailbackPolicy4",
   "Effect": "Allow",
   "Action": [
    "drs:GetFailbackCommandForDrs",
    "drs:UpdateFailbackClientLastSeenForDrs",
    "drs:NotifyAgentAuthenticationForDrs",
    "drs:UpdateAgentReplicationProcessStateForDrs",
    "drs:NotifyAgentReplicationProgressForDrs",
    "drs:NotifyAgentConnectedForDrs",
    "drs:NotifyAgentDisconnectedForDrs",
    "drs:NotifyConsistencyAttainedForDrs",
    "drs:GetFailbackLaunchRequestedForDrs",
    "drs:IssueAgentCertificateForDrs"
   ],
   "Resource": "arn:aws:drs:*:*:recovery-instance/${aws:SourceIdentity}"
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryFailbackInstallationPolicy

You can attach the AWSElasticDisasterRecoveryFailbackInstallationPolicy policy to your IAM identities.

This policy allows installing the AWS Elastic Disaster Recovery Failback Client, which is used to failback Recovery Instances back to your original source infrastructure. Attach this policy to your users or roles whose credentials you provide when running the AWS Elastic Disaster Recovery Failback Client.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
          "Sid": "DRSFailbackInstallationPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:SendClientLogsForDrs",
    "drs:SendClientMetricsForDrs",
    "drs:DescribeRecoveryInstances",
    "drs:DescribeSourceServers"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSFailbackInstallationPolicy2",
   "Effect": "Allow",
   "Action": [
    "drs:TagResource",
    "drs:IssueAgentCertificateForDrs",
    "drs:AssociateFailbackClientToRecoveryInstanceForDrs",
    "drs:GetSuggestedFailbackClientDeviceMappingForDrs",
    "drs:UpdateAgentReplicationInfoForDrs",
    "drs:UpdateFailbackClientDeviceMappingForDrs"
   ],
   "Resource": "arn:aws:drs:*:*:recovery-instance/*"
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryConsoleFullAccess

This policy provides full access to all public APIs of AWS Elastic Disaster Recovery (AWS DRS), as well as permissions to read KMS key, License Manager, Resource Groups, Elastic Load Balancing, IAM, and EC2 information. It also includes EC2 actions that allow to launch, delete, or modify replication servers and recovery instances. These EC2 actions are limited only to resources which the service creates with a specific AWS-only tag. policy to your users or roles.

AWSElasticDisasterRecoveryConsoleFullAccess includes access to your AWS managed keys. However, it does not include access to your customer managed keys, so if you use CMK you will need to add a policy statement to allow the usage of your KMS keys.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ConsoleFullAccess1",
   "Effect": "Allow",
   "Action": [
    "drs:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess2",
   "Effect": "Allow",
   "Action": [
    "kms:ListAliases",
    "kms:DescribeKey"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess3",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceTypes",
    "ec2:DescribeInstanceAttribute",
    "ec2:DescribeInstanceStatus",
    "ec2:DescribeInstanceTypeOfferings",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeSecurityGroups",
```

```
    "ec2:DescribeSnapshots",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:GetEbsEncryptionByDefault",
    "ec2:GetEbsDefaultKmsKeyId",
    "ec2:DescribeKeyPairs",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeHosts"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess4",
   "Effect": "Allow",
   "Action": "license-manager:ListLicenseConfigurations",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess5",
   "Effect": "Allow",
   "Action": "resource-groups:ListGroups",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess6",
   "Effect": "Allow",
   "Action": "elasticloadbalancing:DescribeLoadBalancers",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess7",
   "Effect": "Allow",
   "Action": [
    "iam:ListInstanceProfiles",
    "iam:ListRoles"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess8",
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": [
    "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryConversionServerRole",
```

```
    "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryRecoveryInstanceRole"
   ],
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "ec2.amazonaws.com"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess9",
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteSnapshot"
   ],
   "Resource": "arn:aws:ec2:*:*:snapshot/*",
   "Condition": {
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess10",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateLaunchTemplateVersion",
    "ec2:ModifyLaunchTemplate",
    "ec2:DeleteLaunchTemplateVersions",
    "ec2:CreateTags",
    "ec2:DeleteTags"
   ],
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess11",
   "Effect": "Allow",
```

```
    "Action": [
     "ec2:CreateLaunchTemplate"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess12",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess13",
    "Effect": "Allow",
    "Action": [
     "ec2:StartInstances",
     "ec2:StopInstances",
     "ec2:TerminateInstances",
     "ec2:ModifyInstanceAttribute",
     "ec2:GetConsoleOutput",
     "ec2:GetConsoleScreenshot"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
```

```
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess14",
   "Effect": "Allow",
   "Action": [
    "ec2:RevokeSecurityGroupEgress",
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:AuthorizeSecurityGroupEgress"
   ],
   "Resource": "arn:aws:ec2:*:*:security-group/*",
   "Condition": {
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess15",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess16",
   "Effect": "Allow",
   "Action": "ec2:CreateSecurityGroup",
   "Resource": "arn:aws:ec2:*:*:vpc/*"
  },
  {
```

```
    "Sid": "ConsoleFullAccess17",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSecurityGroup"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess18",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess19",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
```

```
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess20",
   "Effect": "Allow",
   "Action": [
    "ec2:DetachVolume",
    "ec2:AttachVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:instance/*",
   "Condition": {
    "Null": {
     "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess21",
   "Effect": "Allow",
   "Action": [
    "ec2:DetachVolume",
    "ec2:AttachVolume",
    "ec2:StartInstances",
    "ec2:GetConsoleOutput",
    "ec2:GetConsoleScreenshot"
   ],
   "Resource": "arn:aws:ec2:*:*:instance/*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
    },
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "drs.amazonaws.com"
     ]
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess22",
```

```
    "Effect": "Allow",
    "Action": [
     "ec2:AttachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess23",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess24",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
```

```
    "Sid": "ConsoleFullAccess25",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:subnet/*",
     "arn:aws:ec2:*:*:image/*",
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess26",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:snapshot/*",
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
       "CreateSecurityGroup",
       "CreateVolume",
       "CreateSnapshot",
       "RunInstances"
      ]
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
```

```
    "Sid": "ConsoleFullAccess27",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
       "CreateLaunchTemplate"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess28",
    "Effect": "Allow",
    "Action": [
     "cloudformation:DescribeStacks",
     "cloudformation:ListStacks"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess29",
    "Effect": "Allow",
    "Action": [
     "s3:GetBucketLocation",
     "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
   }
  ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryReadOnlyAccess

You can attach the AWSElasticDisasterRecoveryReadOnlyAccess policy to your IAM identities.

This policy provides permissions to all read-only public APIs of AWS Elastic Disaster Recovery (AWS DRS), as well as some read-only APIs of IAM, EC2 and SSM in order to list and view installed roles Recovery Instances, Source Servers and post-launch actions. Attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DRSReadOnlyAccess1",
   "Effect": "Allow",
   "Action": [
    "drs:DescribeJobLogItems",
    "drs:DescribeJobs",
    "drs:DescribeRecoveryInstances",
    "drs:DescribeRecoverySnapshots",
    "drs:DescribeReplicationConfigurationTemplates",
    "drs:DescribeSourceServers",
    "drs:GetFailbackReplicationConfiguration",
    "drs:GetLaunchConfiguration",
    "drs:GetReplicationConfiguration",
    "drs:ListExtensibleSourceServers",
    "drs:ListStagingAccounts",
    "drs:ListTagsForResource",
    "drs:ListLaunchActions"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess2",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess4",
   "Effect": "Allow",
   "Action": "iam:ListRoles",
   "Resource": "*"
  },
```

```
  {
   "Sid": "DRSReadOnlyAccess5",
   "Effect": "Allow",
   "Action": "ssm:ListCommandInvocations",
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess6",
   "Effect": "Allow",
   "Action": "ssm:GetParameter",
   "Resource": "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*"
  },
  {
   "Sid": "DRSReadOnlyAccess7",
   "Effect": "Allow",
   "Action": [
    "ssm:DescribeDocument",
    "ssm:GetDocument"
   ],
   "Resource": [
    "arn:aws:ssm:*:*:document/AWS-CreateImage",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateNetworkConnectivity",
    "arn:aws:ssm:*:*:document/AWSMigration-VerifyMountedVolumes",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateHttpResponse",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateDiskSpace",
    "arn:aws:ssm:*:*:document/AWSMigration-VerifyProcessIsRunning",
    "arn:aws:ssm:*:*:document/AWSMigration-LinuxTimeSyncSetting",
    "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure"
   ]
  },
  {
   "Sid": "DRSReadOnlyAccess8",
   "Effect": "Allow",
   "Action": [
    "ssm:GetAutomationExecution"
   ],
   "Resource": "arn:aws:ssm:*:*:automation-execution/*",
   "Condition": {
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    }
   }
  }
```

```
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryReplicationServerPolicy

This policy is attached to the AWS Elastic Disaster Recovery replication server's instance role.

This policy allows the AWS Elastic Disaster Recovery (AWS DRS) replication servers, which are Amazon EC2 instances launched by Elastic Disaster Recovery, to communicate with the DRS service, and to create EBS snapshots in your AWS account. An IAM role with this policy is attached (as an EC2 instance profile) by AWS DRS to the AWS DRS replication servers which are automatically launched and terminated by AWS DRS, as needed. AWS DRS replication servers are used to facilitate data replication from your external servers to AWS, as part of the recovery process managed by AWS DRS. We do not recommend that you attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
          "Sid": "DRSReplicationServerPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:SendClientMetricsForDrs",
    "drs:SendClientLogsForDrs"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSReplicationServerPolicy2",
   "Effect": "Allow",
   "Action": [
    "drs:GetChannelCommandsForDrs",
    "drs:SendChannelCommandResultForDrs"
   ],
   "Resource": "*"
  },
```

```
{
        "Sid": "DRSReplicationServerPolicy3",
 "Effect": "Allow",
 "Action": [
  "drs:GetAgentSnapshotCreditsForDrs",
  "drs:DescribeReplicationServerAssociationsForDrs",
  "drs:DescribeSnapshotRequestsForDrs",
  "drs:BatchDeleteSnapshotRequestForDrs",
  "drs:NotifyAgentAuthenticationForDrs",
  "drs:BatchCreateVolumeSnapshotGroupForDrs",
  "drs:UpdateAgentReplicationProcessStateForDrs",
  "drs:NotifyAgentReplicationProgressForDrs",
  "drs:NotifyAgentConnectedForDrs",
  "drs:NotifyAgentDisconnectedForDrs",
  "drs:NotifyVolumeEventForDrs",
  "drs:SendVolumeStatsForDrs"
 ],
 "Resource": "*"
},
{
        "Sid": "DRSReplicationServerPolicy4",
 "Effect": "Allow",
 "Action": [
  "ec2:DescribeInstances",
  "ec2:DescribeSnapshots"
 ],
 "Resource": "*"
},
{
        "Sid": "DRSReplicationServerPolicy5",
 "Effect": "Allow",
 "Action": [
  "ec2:CreateSnapshot"
 ],
 "Resource": "arn:aws:ec2:*:*:volume/*",
 "Condition": {
  "Null": {
   "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
  }
 }
},
{
        "Sid": "DRSReplicationServerPolicy6",
 "Effect": "Allow",
```

```
    "Action": [
     "ec2:CreateSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
           "Sid": "DRSReplicationServerPolicy7",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateSnapshot"
     }
    }
   }
  ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryRecoveryInstancePolicy

This policy is attached to the instance role of AWS Elastic Disaster Recovery's recovery instance.

This policy allows the AWS Elastic Disaster Recovery (AWS DRS) recovery instance, which are EC2 instances launched by AWS DRS - to communicate with the AWS DRS service, and to be able to failback to their original source infrastructure. An IAM role with this policy is attached (as an Amazon EC2 Instance Profile) by AWS DRS to the AWS DRS recovery instances. We do not recommend that you attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
```

```
{
        "Sid": "DRSRecoveryInstancePolicy1",
 "Effect": "Allow",
 "Action": [
  "drs:SendAgentMetricsForDrs",
  "drs:SendAgentLogsForDrs",
  "drs:UpdateAgentSourcePropertiesForDrs",
  "drs:UpdateAgentReplicationInfoForDrs",
  "drs:UpdateAgentConversionInfoForDrs",
  "drs:GetAgentCommandForDrs",
  "drs:GetAgentConfirmedResumeInfoForDrs",
  "drs:GetAgentRuntimeConfigurationForDrs",
  "drs:UpdateAgentBacklogForDrs",
  "drs:GetAgentReplicationInfoForDrs",
  "drs:UpdateReplicationCertificateForDrs",
  "drs:NotifyReplicationServerAuthenticationForDrs"
 ],
 "Resource": "arn:aws:drs:*:*:recovery-instance/*",
 "Condition": {
  "StringEquals": {
   "drs:EC2InstanceARN": "${ec2:SourceInstanceARN}"
  }
 }
},
{
        "Sid": "DRSRecoveryInstancePolicy2",
 "Effect": "Allow",
 "Action": [
  "drs:DescribeRecoveryInstances"
 ],
 "Resource": "*"
},
{
        "Sid": "DRSRecoveryInstancePolicy3",
 "Effect": "Allow",
 "Action": [
  "ec2:DescribeInstanceTypes"
 ],
 "Resource": "*"
},
{
        "Sid": "DRSRecoveryInstancePolicy4",
 "Effect": "Allow",
 "Action": [
```

```
    "drs:GetAgentInstallationAssetsForDrs",
    "drs:SendClientLogsForDrs",
    "drs:CreateSourceServerForDrs"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSRecoveryInstancePolicy5",
   "Effect": "Allow",
   "Action": [
    "drs:TagResource"
   ],
   "Resource": "arn:aws:drs:*:*:source-server/*",
   "Condition": {
    "StringEquals": {
     "drs:CreateAction": "CreateSourceServerForDrs"
    }
   }
  },
  {
          "Sid": "DRSRecoveryInstancePolicy6",
   "Effect": "Allow",
   "Action": [
    "drs:SendAgentMetricsForDrs",
    "drs:SendAgentLogsForDrs",
    "drs:UpdateAgentSourcePropertiesForDrs",
    "drs:UpdateAgentReplicationInfoForDrs",
    "drs:UpdateAgentConversionInfoForDrs",
    "drs:GetAgentCommandForDrs",
    "drs:GetAgentConfirmedResumeInfoForDrs",
    "drs:GetAgentRuntimeConfigurationForDrs",
    "drs:UpdateAgentBacklogForDrs",
    "drs:GetAgentReplicationInfoForDrs"
   ],
   "Resource": "arn:aws:drs:*:*:source-server/*"
  },
  {
          "Sid": "DRSRecoveryInstancePolicy7",
   "Effect": "Allow",
   "Action": [
    "sts:AssumeRole",
    "sts:TagSession"
   ],
   "Resource": [
```

```
    "arn:aws:iam::*:role/service-role/DRSCrossAccountAgentAuthorizedRole_*"
   ],
   "Condition": {
    "StringLike": {
     "aws:RequestTag/SourceInstanceARN": "${ec2:SourceInstanceARN}"
    },
    "ForAnyValue:StringEquals": {
     "sts:TransitiveTagKeys": "SourceInstanceARN"
    }
   }
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryServiceRolePolicy

This policy allows AWS Elastic Disaster Recovery to manage AWS resources on your behalf.

This policy is attached to the AWSServiceRoleForElasticDisasterRecovery role.

Permissions details

This policy includes permissions to do the following:

- ec2 – Retrieve and modify resources needed to support failover and failback of source servers and source networks.
- cloudwtach – Retrieve disk usage to allow cost optimization
- iam – Acquire the permissions required for recovery
- kms – Allow using encrypted volumes
- drs – Retrieve tags and set tags for DRS resources, create DRS resources on failover

```
{
 "Version": "2012-10-17",
 "Statement": [
        {
            "Sid": "DRSServiceRolePolicy1",
            "Effect": "Allow",
            "Action": [
                "drs:ListTagsForResource"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy2",
            "Effect": "Allow",
            "Action": [
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:recovery-instance/*"
        },
        {
            "Sid": "DRSServiceRolePolicy3",
            "Effect": "Allow",
            "Action": [
                "drs:CreateRecoveryInstanceForDrs",
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:source-server/*"
        },
        {
            "Sid": "DRSServiceRolePolicy4",
            "Effect": "Allow",
            "Action": "iam:GetInstanceProfile",
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy5",
            "Effect": "Allow",
            "Action": "kms:ListRetirableGrants",
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy6",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeLaunchTemplateVersions",
```

```
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshots",
                "ec2:DescribeSubnets",
                "ec2:DescribeVolumes",
                "ec2:DescribeVolumeAttribute",
                "ec2:GetEbsDefaultKmsKeyId",
                "ec2:GetEbsEncryptionByDefault",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeVpcs",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeRouteTables",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeManagedPrefixLists",
    "ec2:GetManagedPrefixListEntries",
    "ec2:GetManagedPrefixListAssociations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy7",
            "Effect": "Allow",
            "Action": [
                "ec2:RegisterImage"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy8",
            "Effect": "Allow",
            "Action": [
                "ec2:DeregisterImage"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy9",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:snapshot/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy10",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy11",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteVolume",
                "ec2:ModifyVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy12",
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "ec2:ModifyInstanceAttribute",
                "ec2:GetConsoleOutput",
                "ec2:GetConsoleScreenshot"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy13",
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy14",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy15",
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy16",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
        {
            "Sid": "DRSServiceRolePolicy17",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplate"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy18",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
```

```
        },
        {
            "Sid": "DRSServiceRolePolicy19",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:snapshot/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy20",
            "Effect": "Allow",
            "Action": [
                "ec2:DetachVolume",
                "ec2:AttachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy21",
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy22",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
        },
        {
            "Sid": "DRSServiceRolePolicy23",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy24",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:image/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:launch-template/*"
            ]
        },
        {
            "Sid": "DRSServiceRolePolicy25",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryReplicationServerRole",
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryConversionServerRole",
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceRole"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy26",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:snapshot/*",
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "CreateLaunchTemplate",
                        "CreateSecurityGroup",
                        "CreateVolume",
                        "CreateSnapshot",
                        "RunInstances"
                    ]
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy27",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:image/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
```

```
            "Sid": "DRSServiceRolePolicy28",
            "Effect": "Allow",
            "Action": "cloudwatch:GetMetricData",
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryStagingAccountPolicy

This policy allows read-only access to AWS Elastic Disaster Recovery (AWS DRS) resources such as source servers and jobs. It also allows creating a converted snapshot and sharing that EBS snapshot with a specified account.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
          "Sid": "DRSStagingAccountPolicy1",
   "Effect": "Allow",
   "Action": [
    "drs:DescribeSourceServers",
    "drs:DescribeRecoverySnapshots",
    "drs:CreateConvertedSnapshotForDrs",
    "drs:GetReplicationConfiguration",
    "drs:DescribeJobs",
    "drs:DescribeJobLogItems"
   ],
   "Resource": "*"
  },
  {
          "Sid": "DRSStagingAccountPolicy2",
   "Effect": "Allow",
   "Action": [
    "ec2:ModifySnapshotAttribute"
   ],
   "Resource": "arn:aws:ec2:*:*:snapshot/*",
```

```
   "Condition": {
    "StringEquals": {
     "ec2:Add/userId": "${aws:SourceIdentity}"
    },
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    }
   }
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryStagingAccountPolicy_v2

This policy is used by AWS Elastic Disaster Recovery (AWS DRS) to recover source servers into a separate target account and to allow failing back. We do not recommend that you attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
 {
     "Sid": "DRSStagingAccountPolicyv21",
            "Effect": "Allow",
            "Action": [
                "drs:DescribeSourceServers",
                "drs:DescribeRecoverySnapshots",
                "drs:CreateConvertedSnapshotForDrs",
                "drs:GetReplicationConfiguration",
                "drs:DescribeJobs",
                "drs:DescribeJobLogItems"
            ],
            "Resource": "*"
        },
 {
     "Sid": "DRSStagingAccountPolicyv22",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:Add/userId": "${aws:SourceIdentity}"
                },
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
 {
     "Sid": "DRSStagingAccountPolicyv23",
            "Effect": "Allow",
            "Action":  "drs:IssueAgentCertificateForDrs",
            "Resource": [
                "arn:aws:drs:*:*:source-server/*"
            ]
        }
    ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryEc2InstancePolicy

This policy allows installing and using the AWS Replication Agent, which is used by AWS Elastic Disaster Recovery (AWS DRS) to recover source servers that run on EC2 (cross-Region, cross-AZ or cross-Account). An IAM role with this policy should be attached (as an EC2 Instance Profile) to the EC2 Instances.

Permissions details

This policy includes the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DRSEc2InstancePolicy1",
            "Effect": "Allow",
```

```
            "Action": [
                "drs:GetAgentInstallationAssetsForDrs",
                "drs:SendClientLogsForDrs",
                "drs:SendClientMetricsForDrs",
                "drs:CreateSourceServerForDrs",
                "drs:CreateSourceNetwork"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSEc2InstancePolicy2",
            "Effect": "Allow",
            "Action": [
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:source-server/*",
            "Condition": {
                "StringEquals": {
                    "drs:CreateAction": "CreateSourceServerForDrs"
                }
            }
        },
        {
            "Sid": "DRSEc2InstancePolicy3",
            "Effect": "Allow",
            "Action": [
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:source-network/*",
            "Condition": {
                "StringEquals": {
                    "drs:CreateAction": "CreateSourceNetwork"
                }
            }
        },
        {
            "Sid": "DRSEc2InstancePolicy4",
            "Effect": "Allow",
            "Action": [
                "drs:SendAgentMetricsForDrs",
                "drs:SendAgentLogsForDrs",
                "drs:UpdateAgentSourcePropertiesForDrs",
                "drs:UpdateAgentReplicationInfoForDrs",
                "drs:UpdateAgentConversionInfoForDrs",
```

```
                "drs:GetAgentCommandForDrs",
                "drs:GetAgentConfirmedResumeInfoForDrs",
                "drs:GetAgentRuntimeConfigurationForDrs",
                "drs:UpdateAgentBacklogForDrs",
                "drs:GetAgentReplicationInfoForDrs"
            ],
            "Resource": "arn:aws:drs:*:*:source-server/*"
        },
        {
            "Sid": "DRSEc2InstancePolicy5",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/DRSCrossAccountAgentAuthorizedRole_*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/SourceInstanceARN": "${ec2:SourceInstanceARN}"
                },
                "ForAnyValue:StringEquals": {
                    "sts:TransitiveTagKeys": "SourceInstanceARN"
                }
            }
        }
    ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryCrossAccountReplicationPolicy

This policy allows AWS Elastic Disaster Recovery (DRS) to support cross-account replication and cross-account failback.

Permissions details

This policy includes the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "CrossAccountPolicy1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:DescribeVolumeAttribute",
                "ec2:DescribeInstances",
                "drs:DescribeSourceServers",
                "drs:DescribeReplicationConfigurationTemplates",
                "drs:CreateSourceServerForDrs"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CrossAccountPolicy2",
            "Effect": "Allow",
            "Action": [
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:source-server/*",
            "Condition": {
                "StringEquals": {
                    "drs:CreateAction": "CreateSourceServerForDrs"
                }
            }
        }
    ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryNetworkReplicationPolicy

This policy allows AWS Elastic Disaster Recovery (DRS) to support network replication.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
```

```
  {
   "Sid": "DRSNetworkReplicationPolicy1",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeVpcAttribute",
    "ec2:DescribeInternetGateways",
    "ec2:DescribeVpcs",
    "ec2:DescribeSubnets",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeInstances",
    "ec2:DescribeManagedPrefixLists",
    "ec2:GetManagedPrefixListEntries",
    "ec2:GetManagedPrefixListAssociations"
   ],
   "Resource": "*"
  }
 ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryLaunchActionsPolicy

You can attach the AWSElasticDisasterRecoveryLaunchActionsPolicy policy to your IAM identities.

This policy allows you to use Amazon SSM and additional services required permissions to run post-launch actions in AWS Elastic Disaster Recovery (AWS DRS). Attach this policy to your IAM roles or users.

Permissions details

This policy includes the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LaunchActionsPolicy1",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeParameters"

            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy2",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/*",
                "arn:aws:ssm:*:*:automation-definition/*:*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                },
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy3",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
```

```
            "Resource": [
                "arn:aws:ssm:*::document/AWS-*",
                "arn:aws:ssm:*::document/AWSCodeDeployAgent-*",
                "arn:aws:ssm:*::document/AWSConfigRemediation-*",
                "arn:aws:ssm:*::document/AWSConformancePacks-*",
                "arn:aws:ssm:*::document/AWSDisasterRecovery-*",
                "arn:aws:ssm:*::document/AWSDistroOTel-*",
                "arn:aws:ssm:*::document/AWSDocs-*",
                "arn:aws:ssm:*::document/AWSEC2-*",
                "arn:aws:ssm:*::document/AWSEC2Launch-*",
                "arn:aws:ssm:*::document/AWSFIS-*",
                "arn:aws:ssm:*::document/AWSFleetManager-*",
                "arn:aws:ssm:*::document/AWSIncidents-*",
                "arn:aws:ssm:*::document/AWSKinesisTap-*",
                "arn:aws:ssm:*::document/AWSMigration-*",
                "arn:aws:ssm:*::document/AWSNVMe-*",
                "arn:aws:ssm:*::document/AWSNitroEnclavesWindows-*",
                "arn:aws:ssm:*::document/AWSObservabilityExporter-*",
                "arn:aws:ssm:*::document/AWSPVDriver-*",
                "arn:aws:ssm:*::document/AWSQuickSetupType-*",
                "arn:aws:ssm:*::document/AWSQuickStarts-*",
                "arn:aws:ssm:*::document/AWSRefactorSpaces-*",
                "arn:aws:ssm:*::document/AWSResilienceHub-*",
                "arn:aws:ssm:*::document/AWSSAP-*",
                "arn:aws:ssm:*::document/AWSSAPTools-*",
                "arn:aws:ssm:*::document/AWSSQLServer-*",
                "arn:aws:ssm:*::document/AWSSSO-*",
                "arn:aws:ssm:*::document/AWSSupport-*",
                "arn:aws:ssm:*::document/AWSSystemsManagerSAP-*",
                "arn:aws:ssm:*::document/AmazonCloudWatch-*",
                "arn:aws:ssm:*::document/AmazonCloudWatchAgent-*",
                "arn:aws:ssm:*::document/AmazonECS-*",
                "arn:aws:ssm:*::document/AmazonEFSUtils-*",
                "arn:aws:ssm:*::document/AmazonEKS-*",
                "arn:aws:ssm:*::document/AmazonInspector-*",
                "arn:aws:ssm:*::document/AmazonInspector2-*",
                "arn:aws:ssm:*::document/AmazonInternal-*",
                "arn:aws:ssm:*::document/AwsEnaNetworkDriver-*",
                "arn:aws:ssm:*::document/AwsVssComponents-*",
                "arn:aws:ssm:*::automation-definition/AWS-*:*",
                "arn:aws:ssm:*::automation-definition/AWSCodeDeployAgent-*:*",
                "arn:aws:ssm:*::automation-definition/AWSConfigRemediation-*:*",
                "arn:aws:ssm:*::automation-definition/AWSConformancePacks-*:*",
                "arn:aws:ssm:*::automation-definition/AWSDisasterRecovery-*:*",
```

```
                "arn:aws:ssm:*::automation-definition/AWSDistroOTel-*:*",
                "arn:aws:ssm:*::automation-definition/AWSDocs-*:*",
                "arn:aws:ssm:*::automation-definition/AWSEC2-*:*",
                "arn:aws:ssm:*::automation-definition/AWSEC2Launch-*:*",
                "arn:aws:ssm:*::automation-definition/AWSFIS-*:*",
                "arn:aws:ssm:*::automation-definition/AWSFleetManager-*:*",
                "arn:aws:ssm:*::automation-definition/AWSIncidents-*:*",
                "arn:aws:ssm:*::automation-definition/AWSKinesisTap-*:*",
                "arn:aws:ssm:*::automation-definition/AWSMigration-*:*",
                "arn:aws:ssm:*::automation-definition/AWSNVMe-*:*",
                "arn:aws:ssm:*::automation-definition/AWSNitroEnclavesWindows-*:*",
                "arn:aws:ssm:*::automation-definition/AWSObservabilityExporter-*:*",
                "arn:aws:ssm:*::automation-definition/AWSPVDriver-*:*",
                "arn:aws:ssm:*::automation-definition/AWSQuickSetupType-*:*",
                "arn:aws:ssm:*::automation-definition/AWSQuickStarts-*:*",
                "arn:aws:ssm:*::automation-definition/AWSRefactorSpaces-*:*",
                "arn:aws:ssm:*::automation-definition/AWSResilienceHub-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSAP-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSAPTools-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSQLServer-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSSO-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSupport-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSystemsManagerSAP-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonCloudWatch-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonCloudWatchAgent-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonECS-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonEFSUtils-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonEKS-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInspector-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInspector2-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInternal-*:*",
                "arn:aws:ssm:*::automation-definition/AwsEnaNetworkDriver-*:*",
                "arn:aws:ssm:*::automation-definition/AwsVssComponents-*:*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy4",
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                },
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy5",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSDRS": "false"
                },
                "StringEquals": {
                    "aws:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy6",
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListCommandInvocations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "LaunchActionsPolicy7",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocumentVersions",
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": "arn:aws:ssm:*:*:document/*"
        },
        {
            "Sid": "LaunchActionsPolicy8",
            "Effect": "Allow",
            "Action": [
                "ssm:GetAutomationExecution"
            ],
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy9",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/
ManagedByAWSElasticDisasterRecoveryService-*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm.amazonaws.com"
                }
            }
        },
        {
```

```
            "Sid": "LaunchActionsPolicy10",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/
ManagedByAWSElasticDisasterRecoveryService-*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy11",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "drs.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS managed policy: AWSElasticDisasterRecoveryConsoleFullAccess_v2

You can attach the **AWSElasticDisasterRecoveryConsoleFullAccess_v2** policy to your IAM identities.

Allows full administrative access to AWS Elastic Disaster Recovery (AWS DRS) Console. Attach this policy to your users or roles.

Permissions details

This policy includes permissions to do the following:

- `drs` – All apis.
- `kms` – List aliases and describe keys.
- `ec2` – Describe account attributes, availability zones, images, instance (including types, statuses, type offerings), subnets, volumes, ebs encryption by default, ebs default kms key id, key/pairs, capacity reservations and hosts. Describe, create and delete snapshots. Describe and create launch templates. Start, run, stop and terminate instances. Describe and modify instance attributes. Create, attach and detach volumes. Describe, create, modify and delete launch template version. Create and delete tags. Get console output and screenshots. Describe and create security groups. Authorize and revoke security group egress. Authorize security group ingress.
- `licence manager` – List license configurations.
- `resource groups` – List groups.
- `elastic load balancing` – Describe load balancers..
- `iam` – List instance profiles and roles, passRole.
- `cloudformation` – Describe and list stacks.
- `s3` – Get bucket location and list all my buckets.
- `ssm` – Describe instance information, send command, start automation execution. List documents and command invocations. Get and put parameters. Describe and get document. Get automation executions.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ConsoleFullAccess1",
   "Effect": "Allow",
   "Action": [
    "drs:*"
   ],
```

```
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess2",
    "Effect": "Allow",
    "Action": [
     "kms:ListAliases",
     "kms:DescribeKey"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess3",
    "Effect": "Allow",
    "Action": [
     "ec2:DescribeAccountAttributes",
     "ec2:DescribeAvailabilityZones",
     "ec2:DescribeImages",
     "ec2:DescribeInstances",
     "ec2:DescribeInstanceTypes",
     "ec2:DescribeInstanceAttribute",
     "ec2:DescribeInstanceStatus",
     "ec2:DescribeInstanceTypeOfferings",
     "ec2:DescribeLaunchTemplateVersions",
     "ec2:DescribeLaunchTemplates",
     "ec2:DescribeSecurityGroups",
     "ec2:DescribeSnapshots",
     "ec2:DescribeSubnets",
     "ec2:DescribeVolumes",
     "ec2:GetEbsEncryptionByDefault",
     "ec2:GetEbsDefaultKmsKeyId",
     "ec2:DescribeKeyPairs",
     "ec2:DescribeCapacityReservations",
     "ec2:DescribeHosts"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess4",
    "Effect": "Allow",
    "Action": "license-manager:ListLicenseConfigurations",
    "Resource": "*"
   },
   {
```

```
    "Sid": "ConsoleFullAccess5",
    "Effect": "Allow",
    "Action": "resource-groups:ListGroups",
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess6",
    "Effect": "Allow",
    "Action": "elasticloadbalancing:DescribeLoadBalancers",
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess7",
    "Effect": "Allow",
    "Action": [
     "iam:ListInstanceProfiles",
     "iam:ListRoles"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess8",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryConversionServerRole",
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryRecoveryInstanceRole",
     "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole"
    ],
    "Condition": {
     "StringEquals": {
      "iam:PassedToService": "ec2.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess9",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition":
```

```
    {
     "Null":
     {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool":
     {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess10",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplateVersion",
     "ec2:ModifyLaunchTemplate",
     "ec2:DeleteLaunchTemplateVersions",
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess11",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplate"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess12",
    "Effect": "Allow",
```

```
    "Action": [
     "ec2:DeleteVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess13",
    "Effect": "Allow",
    "Action": [
     "ec2:StartInstances",
     "ec2:StopInstances",
     "ec2:TerminateInstances",
     "ec2:ModifyInstanceAttribute",
     "ec2:GetConsoleOutput",
     "ec2:GetConsoleScreenshot"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess14",
    "Effect": "Allow",
    "Action": [
     "ec2:RevokeSecurityGroupEgress",
     "ec2:AuthorizeSecurityGroupIngress",
     "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
```

```
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess15",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess16",
   "Effect": "Allow",
   "Action": "ec2:CreateSecurityGroup",
   "Resource": "arn:aws:ec2:*:*:vpc/*"
  },
  {
   "Sid": "ConsoleFullAccess17",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": "arn:aws:ec2:*:*:security-group/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
```

```
    }
   },
   {
    "Sid": "ConsoleFullAccess18",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess19",
    "Effect": "Allow",
    "Action":
    [
     "ec2:CreateSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess20",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume",
     "ec2:AttachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
```

```
    "Null": {
     "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess21",
   "Effect": "Allow",
   "Action": [
    "ec2:DetachVolume",
    "ec2:AttachVolume",
    "ec2:StartInstances",
    "ec2:GetConsoleOutput",
    "ec2:GetConsoleScreenshot"
   ],
   "Resource": "arn:aws:ec2:*:*:instance/*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
    },
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "drs.amazonaws.com"
     ]
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess22",
   "Effect": "Allow",
   "Action": [
    "ec2:AttachVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
```

```
    }
   },
   {
    "Sid": "ConsoleFullAccess23",
    "Effect": "Allow",
    "Action":
    [
     "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess24",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess25",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:subnet/*",
     "arn:aws:ec2:*:*:image/*",
     "arn:aws:ec2:*:*:network-interface/*",
```

```
    "arn:aws:ec2:*:*:launch-template/*"
   ],
   "Condition": {
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess26",
   "Effect": "Allow",
   "Action": "ec2:CreateTags",
   "Resource": [
    "arn:aws:ec2:*:*:security-group/*",
    "arn:aws:ec2:*:*:volume/*",
    "arn:aws:ec2:*:*:snapshot/*",
    "arn:aws:ec2:*:*:instance/*"
   ],
   "Condition": {
    "StringEquals": {
     "ec2:CreateAction": [
      "CreateSecurityGroup",
      "CreateVolume",
      "CreateSnapshot",
      "RunInstances"
     ]
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess27",
   "Effect": "Allow",
   "Action": "ec2:CreateTags",
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "StringEquals": {
     "ec2:CreateAction": [
      "CreateLaunchTemplate"
     ]
    }
   }
```

```
  },
  {
   "Sid": "ConsoleFullAccess28",
   "Effect": "Allow",
   "Action": [
    "cloudformation:DescribeStacks",
    "cloudformation:ListStacks"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess29",
   "Effect": "Allow",
   "Action": [
    "s3:GetBucketLocation",
    "s3:ListAllMyBuckets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess30",
   "Effect": "Allow",
   "Action": [
    "ssm:DescribeInstanceInformation",
    "ssm:DescribeParameters"

   ],
   "Resource": [
    "*"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "drs.amazonaws.com"
     ]
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess31",
   "Effect": "Allow",
   "Action": [
    "ssm:SendCommand",
    "ssm:StartAutomationExecution"
```

```
    ],
    "Resource": [
     "arn:aws:ssm:*:*:automation-definition/AWS-CreateImage:$DEFAULT",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateNetworkConnectivity",
     "arn:aws:ssm:*:*:document/AWSMigration-VerifyMountedVolumes",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateHttpResponse",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateDiskSpace",
     "arn:aws:ssm:*:*:document/AWSMigration-VerifyProcessIsRunning",
     "arn:aws:ssm:*:*:document/AWSMigration-LinuxTimeSyncSetting",
     "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess32",
    "Effect": "Allow",
    "Action": [
     "ssm:SendCommand"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     },
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess33",
    "Effect": "Allow",
    "Action": [
```

```
    "ssm:ListDocuments",
    "ssm:ListCommandInvocations"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess34",
   "Effect": "Allow",
   "Action": [
    "ssm:GetParameter",
    "ssm:PutParameter"
   ],
   "Resource": "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*",
   "Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess35",
   "Effect": "Allow",
   "Action": [
    "ssm:DescribeDocument",
    "ssm:GetDocument"
   ],
   "Resource": "arn:aws:ssm:*:*:document/*"
  },
  {
   "Sid": "ConsoleFullAccess36",
   "Effect": "Allow",
   "Action": [
    "ssm:GetParameters"
   ],
   "Resource": [
    "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": "ssm.amazonaws.com"
    }
   }
  },
  {
```

```
    "Sid": "ConsoleFullAccess37",
    "Effect": "Allow",
    "Action":
    [
     "ssm:GetAutomationExecution"
    ],
    "Resource": "arn:aws:ssm:*:*:automation-execution/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   }
  ]
}
```

Elastic Disaster Recovery updates for AWS managed policies

View details about updates to AWS managed policies for AWS Elastic Disaster Recovery since March 1, 2021.

AWS Elastic Disaster Recovery policy updates

Change	Description	Date
AWSElasticDisasterRecoveryConsoleFullAccess_v2– Updated policy AWSElasticDisasterRecoveryLaunchActionsPolicy– Updated policy	Created new revisions of AWSElasticDisasterRecoveryConsoleFullAccess_v2 and AWSElasticDisasterRecoveryLaunchActionsPolicy managed policies, to support additional parameter types in SSM Parameters Store for post-launch actions.	May 19, 2024
AWSElasticDisasterRecoveryServiceRolePolicy – Updated policy	Created revision of the AWSElasticDisasterRecoveryServiceRolePolicy policy, to	January 28, 2024

Change	Description	Date
	support replicating marketplace licenses to launched instances.	
AWSElasticDisasterRecoveryCrossAccountReplicationPolicy – Updated policy	Created revision of the AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy, to support replicating marketplace licenses to launched instances.	January 28, 2024
AWSElasticDisasterRecoveryNetworkReplicationPolicy AWSElasticDisasterRecoveryServiceRolePolicy	Created new revisions of managed policies to support managed prefix lists for DRS network replication and recovery.	January 3rd, 2024

Change	Description	Date
	Created new revisions of managed policies to support DRS to GovCloud and added Sid to statements in managed policies	November 27, 2023

Change	Description	Date
AWSElasticDisasterRecoveryReplicationServerPolicy		
AWSElasticDisasterRecoveryCrossAccountReplicationPolicy – Updated policy	Created revision of AWSElasticDisasterRecoveryCrossAccountReplicationPolicy to support DRS in GovCloud	November 27, 2023
AWSElasticDisasterRecoveryReadOnlyAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy with additional read-only permissions for post-launch actions.	November 27, 2023
AWSElasticDisasterRecoveryConsoleFullAccess_v2 New policy	AWS Elastic Disaster Recovery added a new policy. This policy provides access to use DRS console. Attach this policy to your IAM roles or users.	November 27, 2023
AWSElasticDisasterRecoveryConsoleFullAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow launching into an existing instance.	October 15, 2023
AWSElasticDisasterRecoveryConsoleFullAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow launching into an existing instance.	October 15, 2023

Change	Description	Date
AWSElasticDisasterRecoveryLaunchActionsPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow launching into an existing instance tagged with a specific AWS-only key-value pair.	October 15, 2023
AWSElasticDisasterRecoveryEc2InstancePolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow sending installation result metrics to AWS Elastic Disaster Recovery.	October 10, 2023
AWSElasticDisasterRecoveryAgentInstallationPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow sending installation result metrics to AWS Elastic Disaster Recovery.	October 10, 2023
AWSElasticDisasterRecoveryLaunchActionsPolicy New policy	AWS Elastic Disaster Recovery added a new policy. This policy provides access to use post-launch actions. Attach this policy to your IAM roles or users.	September 13, 2023
AWSElasticDisasterRecoveryReadOnlyAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy with new read-only APIs for post-laun ch actions.	September 13, 2023
AWSElasticDisasterRecoveryAgentInstallationPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow network replication and recovery.	June 13, 2023

Change	Description	Date
AWSElasticDisasterRecoveryEc2InstancePolicy – Updated policy	This policy was updated to allow network replication and recovery.	June 13, 2023
AWSElasticDisasterRecoveryConsoleFullAccess– Updated policy	This policy was updated to support network replication and recovery.	June 13, 2023
AWSElasticDisasterRecoveryNetworkReplicationPolicy – New policy	This policy is used by AWS Elastic Disaster Recovery (DRS) to support network replication.	June 13, 2023
AWSElasticDisasterRecoveryServiceRolePolicy – Updated policy	This policy was updated to support network replication and recovery.	June 13, 2023
AWSElasticDisasterRecoveryCrossAccountReplicationPolicy – New policy	This policy is used by AWS Elastic Disaster Recovery (DRS) to support replication and failback.	May 14, 2023
AWSElasticDisasterRecoveryRecoveryInstancePolicy – Updated policy	This policy was updated to support failback by the agent after reverse replication.	May 14, 2023
AWSElasticDisasterRecoveryEc2InstancePolicy – Updated policy	This policy was updated to support replication by the agent.	May 14, 2023
AWSElasticDisasterRecoveryFullAccess– Updated policy	This policy was updated to support default EC2 launch templates and bulk editing of source server EC2 launch templates.	April 19, 2023

Change	Description	Date
AWSElasticDisasterRecoveryCrossAccountReplicationPolicy – New policy	This policy is used by AWS Elastic Disaster Recovery (DRS) to support cross-account replication and cross-account failback.	May 7, 2023
AWSElasticDisasterRecoveryRecoveryInstancePolicy – Updated policy	This policy was updated to support cross-account failback by the agent after reverse replication.	May 7, 2023
AWSElasticDisasterRecoveryEc2InstancePolicy – Updated policy	This policy was updated to support cross-account replication by the agent.	May 7, 2023
AWSElasticDisasterRecoveryConsoleFullAccess– Updated policy	This policy was updated to support default EC2 launch templates and bulk editing of source server EC2 launch templates.	April 16, 2023
AWSElasticDisasterRecoveryAgentPolicy – Updated policy	This policy was updated to support the kernel upgrade feature.	April 1, 2023
AWSElasticDisasterRecoveryStagingAccountPolicy_v2 – New policy	This policy was updated to support the kernel upgrade feature.	December 11, 2022

Change	Description	Date
AWSElasticDisasterRecoveryAgentInstallationPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to properly support agent installation on Recovery Instances. This policy allows installing the AWS Replicati on Agent, which is used with AWS Elastic Disaster Recovery (AWS DRS) to recover external servers to AWS. Attach this policy to your users or roles whose credentials you provide during the installation step of the AWS Replication Agent.	November 14, 2022
AWSElasticDisasterRecoveryRecoveryInstancePolicy – Updated policy	AWS Elastic Disaster Recovery updated this policy to include permissions which allow DRS Recovery Instances that originated from EC2 instances to replicate back to their origin locations in a failback scenario. As an additional security mechanism, Elastic Disaster Recovery will block requests that are not targeted at the source server the EC2 instance is associated with.	October 24, 2022

Change	Description	Date
AWSElasticDisasterRecoveryAgentInstallationPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to include resource tagging. This policy allows installing the AWS Replication Agent, which is used with AWS Elastic Disaster Recovery (AWS DRS) to recover external servers to AWS. Attach this policy to your users or roles whose credentials you provide during the installation step of the AWS Replication Agent.	June 28, 2022
AWSElasticDisasterRecoveryFailbackInstallationPolicy – Updated policy	AWS Elastic Disaster Recovery updated this policy to include a new permission (drs:UpdateAgentReplicationInfoForDrs). This permission is needed to complete the failback process in some cases.	June 22, 2022
AWSElasticDisasterRecoveryServiceRolePolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow DRS to call cloudwatch:GetMetricData and also ec2:ModifyVolume on EBS volumes of the replication server in order to support the automatic volume type selection feature.	June 21st, 2022

Change	Description	Date
AWSElasticDisasterRecoveryReplicationServerPolicy – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow replication servers to call drs:NotifyVolumeEventForDrs and drs:SendVolumeStatsForDrs.	June 21st, 2022
AWSElasticDisasterRecoveryConsoleFullAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy to allow listing IAM roles.	May 26th, 2022
AWSElasticDisasterRecoveryReadOnlyAccess – Updated policy	AWS Elastic Disaster Recovery updated the policy with new read-only APIs of DRS and also added a permission that allows to list IAM roles.	May 26th, 2022
AWSElasticDisasterRecoveryEc2InstancePolicy – Updated policy	AWS Elastic Disaster Recovery added a new policy. This policy allows installing and using the AWS Replication Agent, which is used by AWS Elastic Disaster Recovery (DRS) to recover source servers that run on EC2 (cross-region or cross-AZ). An IAM role with this policy should be attached (as an EC2 Instance Profile) to the EC2 Instances.	April 6, 2022
AWSElasticDisasterRecoveryReadOnlyAccess – Updated policy	AWS Elastic Disaster Recovery updated this policy.	April 3, 2022

Change	Description	Date
AWSElasticDisasterRecoveryStagingAccountPolicy – New policy	AWS Elastic Disaster Recovery added a new policy. This policy allows read-only access to AWS Elastic Disaster Recovery (DRS) resources such as source servers and jobs. It also allows creating a converted snapshot and sharing that EBS snapshot with a specified account.	February 24, 2022
AWSElasticDisasterRecoveryAgentPolicy – New policy	AWS Elastic Disaster Recovery added a new policy. This policy allows using the AWS Replication Agent, which is used with AWS Elastic Disaster Recovery to recover source servers to AWS. We do not recommend that you attach this policy to your users or roles.	November 17, 2021

Change	Description	Date
AWSElasticDisasterRecoveryConversionServerPolicy New policy	AWS Elastic Disaster Recovery added a new policy. This policy is attached to the AWS Elastic Disaster Recovery Conversion server's instance role. This policy allows Elastic Disaster Recovery (DRS) Conversion Servers, which are EC2 instances launched by Elastic Disaster Recovery, to communicate with the DRS service. An IAM role with this policy is attached (as an EC2 Instance Profile) by DRS to the DRS Conversion Servers, which are automatically launched and terminated by DRS, when needed. We do not recommend that you attach this policy to your users or roles. AWS DRS conversio n servers are used by AWS Elastic Disaster Recovery when users choose to recover source servers using the Elastic Disaster Recovery console, CLI, or API.	November 17, 2021

Change	Description	Date
AWSElasticDisasterRecoveryFailbackPolicy - New policy	AWS Elastic Disaster Recovery added a new policy. This policy allows using the AWS Elastic Disaster Recovery Failback Client, which is used to failback Recovery Instances back to your original source infrastructure. We do not recommend that you attach this policy to your users or roles.	November 17, 2021
AWSElasticDisasterRecoveryFailbackInstallationPolicy – New policy	AWS Elastic Disaster Recovery added a new policy. You can attach the AWSElasticDisasterRecoveryFailbackInstallationPolicy policy to your IAM identities. This policy allows installing the AWS Elastic Disaster Recovery Failback Client, which is used to failback recovery instances back to your original source infrastructure. Attach this policy to your users or roles whose credentials you provide when running the EAWS Elastic Disaster Recovery Failback Client.	November 17, 2021

Change	Description	Date
AWSElasticDisasterRecoveryConsoleFullAccess – New policy	AWS Elastic Disaster Recovery added a new policy. This policy provides full access to all public APIs of AWS Elastic Disaster Recovery (AWS DRS), as well as permissions to read KMS key, License Manager, Resource Groups, Elastic Load Balancing, IAM, and Amazon EC2 information. Attach this policy to your users or roles.	November 17, 2021

Change	Description	Date
AWSElasticDisasterRecoveryReplicationServerPolicy – New policy	AWS Elastic Disaster Recovery added a new policy. This policy is attached to the Elastic Disaster Recovery Replication server's instance role. This policy allows the Elastic Disaster Recovery (DRS) Replication Servers, which are EC2 instances launched by Elastic Disaster Recovery, to communicate with the DRS service, and to create EBS snapshots in your AWS account. An IAM role with this policy is attached (as an EC2 Instance Profile) by Elastic Disaster Recovery to the DRS Replication Servers which are automatically launched and terminated by DRS, as needed. DRS Replication Servers are used to facilitat e data replication from your external servers to AWS, as part of the recovery process managed by DRS. We do not recommend that you attach this policy to your users or roles.	November 17, 2021

Change	Description	Date
AWSElasticDisasterRecoveryRecoveryInstancePolicy – New policy	AWS Elastic Disaster Recovery added a new policy. This policy is attached to the instance role of Elastic Disaster Recovery's Recovery Instance. This policy allows the Elastic Disaster Recovery (DRS) Recovery Instance, which are EC2 instances launched by Elastic Disaster Recovery - to communicate with the DRS service, and to be able to failback to their original source infrastru cture. An IAM role with this policy is attached (as an EC2 Instance Profile) by Elastic Disaster Recovery to the DRS recovery instances. We do not recommend that you attach this policy to your users or roles.	November 17, 2021
AWSElasticDisasterRecoveryServiceRolePolicy – New policy	AWS Elastic Disaster Recovery added a new policy. This policy allows Elastic Disaster Recovery to manage AWS resources on your behalf.	November 17, 2021
AWS Elastic Disaster Recovery started tracking changes	AWS Elastic Disaster Recovery started tracking changes for AWS managed policies.	November 17, 2021

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as a user, role, or group. These policies control what actions that identity can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM Policies in the IAM User Guide.

Identity-based policies can be further categorized as inline policies or managed policies. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing Between Managed Policies and Inline Policies in the IAM User Guide.

Using identity-based policies

By default, IAM users and roles don't have permission to create or modify AWS Elastic Disaster Recovery resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to

perform specific API operations on the specified resources they need. The administrator must then attach those policies to the users or groups that require those permissions. To learn how to attach policies to a user or group, see Adding and removing IAM identity permissions in the IAM User Guide. To learn how to create an IAM identity-based policy using example JSON policy documents, see Creating policies on the JSON tab in the IAM User Guide.

Topics

Customer-managed policies in AWS Elastic Disaster Recovery

You can create your own custom IAM policies to allow permissions for AWS Elastic Disaster Recovery actions and resources. You can attach these custom policies to the users, roles, or groups that require those permissions. You can also create your own custom IAM policies for integration betweenAWS Elastic Disaster Recovery and other AWS services. The following example IAM policies grant permissions for various AWS Elastic Disaster Recovery actions. Use them to limit AWS Elastic Disaster Recovery access for your users and roles.

Console Full Access Policy - AWSElasticDisasterRecoveryConsoleFullAccess

This policy provides full access to all public APIs of AWS Elastic Disaster Recovery (AWS DRS), as well as permissions to read KMS key, License Manager, Resource Groups, Elastic Load Balancing, IAM, and Amazon EC2 information. Attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ConsoleFullAccess1",
   "Effect": "Allow",
```

```
    "Action": [
     "drs:*"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess2",
    "Effect": "Allow",
    "Action": [
     "kms:ListAliases",
     "kms:DescribeKey"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess3",
    "Effect": "Allow",
    "Action": [
     "ec2:DescribeAccountAttributes",
     "ec2:DescribeAvailabilityZones",
     "ec2:DescribeImages",
     "ec2:DescribeInstances",
     "ec2:DescribeInstanceTypes",
     "ec2:DescribeInstanceAttribute",
     "ec2:DescribeInstanceStatus",
     "ec2:DescribeInstanceTypeOfferings",
     "ec2:DescribeLaunchTemplateVersions",
     "ec2:DescribeLaunchTemplates",
     "ec2:DescribeSecurityGroups",
     "ec2:DescribeSnapshots",
     "ec2:DescribeSubnets",
     "ec2:DescribeVolumes",
     "ec2:GetEbsEncryptionByDefault",
     "ec2:GetEbsDefaultKmsKeyId",
     "ec2:DescribeKeyPairs",
     "ec2:DescribeCapacityReservations",
     "ec2:DescribeHosts"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess4",
    "Effect": "Allow",
    "Action": "license-manager:ListLicenseConfigurations",
```

```
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess5",
    "Effect": "Allow",
    "Action": "resource-groups:ListGroups",
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess6",
    "Effect": "Allow",
    "Action": "elasticloadbalancing:DescribeLoadBalancers",
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess7",
    "Effect": "Allow",
    "Action": [
     "iam:ListInstanceProfiles",
     "iam:ListRoles"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess8",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryConversionServerRole",
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryRecoveryInstanceRole"
    ],
    "Condition": {
     "StringEquals": {
      "iam:PassedToService": "ec2.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess9",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
```

```
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess10",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplateVersion",
     "ec2:ModifyLaunchTemplate",
     "ec2:DeleteLaunchTemplateVersions",
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess11",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplate"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess12",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteVolume"
```

```
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess13",
    "Effect": "Allow",
    "Action": [
     "ec2:StartInstances",
     "ec2:StopInstances",
     "ec2:TerminateInstances",
     "ec2:ModifyInstanceAttribute",
     "ec2:GetConsoleOutput",
     "ec2:GetConsoleScreenshot"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess14",
    "Effect": "Allow",
    "Action": [
     "ec2:RevokeSecurityGroupEgress",
     "ec2:AuthorizeSecurityGroupIngress",
     "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
```

```
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess15",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess16",
   "Effect": "Allow",
   "Action": "ec2:CreateSecurityGroup",
   "Resource": "arn:aws:ec2:*:*:vpc/*"
  },
  {
   "Sid": "ConsoleFullAccess17",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": "arn:aws:ec2:*:*:security-group/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
```

```
{
 "Sid": "ConsoleFullAccess18",
 "Effect": "Allow",
 "Action": [
  "ec2:CreateSnapshot"
 ],
 "Resource": "arn:aws:ec2:*:*:volume/*",
 "Condition": {
  "Null": {
   "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
  },
  "Bool": {
   "aws:ViaAWSService": "true"
  }
 }
},
{
 "Sid": "ConsoleFullAccess19",
 "Effect": "Allow",
 "Action": [
  "ec2:CreateSnapshot"
 ],
 "Resource": "arn:aws:ec2:*:*:snapshot/*",
 "Condition": {
  "Null": {
   "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
  },
  "Bool": {
   "aws:ViaAWSService": "true"
  }
 }
},
{
 "Sid": "ConsoleFullAccess20",
 "Effect": "Allow",
 "Action": [
  "ec2:DetachVolume",
  "ec2:AttachVolume"
 ],
 "Resource": "arn:aws:ec2:*:*:instance/*",
 "Condition": {
  "Null": {
   "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
  },
```

```
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess21",
   "Effect": "Allow",
   "Action": [
    "ec2:DetachVolume",
    "ec2:AttachVolume",
    "ec2:StartInstances",
    "ec2:GetConsoleOutput",
    "ec2:GetConsoleScreenshot"
   ],
   "Resource": "arn:aws:ec2:*:*:instance/*",
   "Condition": {
    "StringEquals": {
     "ec2:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
    },
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "drs.amazonaws.com"
     ]
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess22",
   "Effect": "Allow",
   "Action": [
    "ec2:AttachVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
```

```
    "Sid": "ConsoleFullAccess23",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess24",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess25",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:subnet/*",
     "arn:aws:ec2:*:*:image/*",
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
     "Bool": {
```

```
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess26",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:snapshot/*",
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
       "CreateSecurityGroup",
       "CreateVolume",
       "CreateSnapshot",
       "RunInstances"
      ]
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess27",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
       "CreateLaunchTemplate"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess28",
    "Effect": "Allow",
```

```
    "Action": [
     "cloudformation:DescribeStacks",
     "cloudformation:ListStacks"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess29",
    "Effect": "Allow",
    "Action": [
     "s3:GetBucketLocation",
     "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
   }
  ]
}
```

Console Full Access Policy - AWSElasticDisasterRecoveryConsoleFullAccess_v2

Allows full administrative access to AWS Elastic Disaster Recovery (AWS DRS) Console. Attach this policy to your users or roles.

Permissions details

This policy includes permissions to do the following:

- `drs` – All apis.
- `kms` – List aliases and describe keys.
- `ec2` – Describe account attributes, availability zones, images, instance (including types, statuses, type offerings), subnets, volumes, ebs encryption by default, ebs default kms key id, key/pairs, capacity reservations and hosts. Describe, create and delete snapshots. Describe and create launch templates. Start, run, stop and terminate instances. Describe and modify instance attributes. Create, attach and detach volumes. Describe, create, modify and delete launch template version. Create and delete tags. Get console output and screenshots. Describe and create security groups. Authorize and revoke security group egress. Authorize security group ingress.
- `licence manager` – List license configurations.

- `resource groups` – List groups.
- `elastic load balancing` – Describe load balancers..
- `iam` – List instance profiles and roles, passRole.
- `cloudformation` – Describe and list stacks.
- `s3` – Get bucket location and list all my buckets.
- `ssm` – Describe instance information, send command, start automation execution. List documents and command invocations. Get and put parameters. Describe and get document. Get automation executions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ConsoleFullAccess1",
   "Effect": "Allow",
   "Action": [
    "drs:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess2",
   "Effect": "Allow",
   "Action": [
    "kms:ListAliases",
    "kms:DescribeKey"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess3",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceTypes",
    "ec2:DescribeInstanceAttribute",
    "ec2:DescribeInstanceStatus",
```

```
    "ec2:DescribeInstanceTypeOfferings",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSnapshots",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:GetEbsEncryptionByDefault",
    "ec2:GetEbsDefaultKmsKeyId",
    "ec2:DescribeKeyPairs",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeHosts"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess4",
   "Effect": "Allow",
   "Action": "license-manager:ListLicenseConfigurations",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess5",
   "Effect": "Allow",
   "Action": "resource-groups:ListGroups",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess6",
   "Effect": "Allow",
   "Action": "elasticloadbalancing:DescribeLoadBalancers",
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess7",
   "Effect": "Allow",
   "Action": [
    "iam:ListInstanceProfiles",
    "iam:ListRoles"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess8",
```

```
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryConversionServerRole",
     "arn:aws:iam::*:role/service-role/AWSElasticDisasterRecoveryRecoveryInstanceRole",
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole"
    ],
    "Condition": {
     "StringEquals": {
      "iam:PassedToService": "ec2.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess9",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteSnapshot"
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess10",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplateVersion",
     "ec2:ModifyLaunchTemplate",
     "ec2:DeleteLaunchTemplateVersions",
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
```

```
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess11",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateLaunchTemplate"
   ],
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "Null": {
     "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess12",
   "Effect": "Allow",
   "Action":
   [
    "ec2:DeleteVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "Null": {
     "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess13",
   "Effect": "Allow",
   "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances",
    "ec2:TerminateInstances",
    "ec2:ModifyInstanceAttribute",
    "ec2:GetConsoleOutput",
    "ec2:GetConsoleScreenshot"
   ],
```

```
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess14",
    "Effect": "Allow",
    "Action": [
     "ec2:RevokeSecurityGroupEgress",
     "ec2:AuthorizeSecurityGroupIngress",
     "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess15",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
```

```
{
 "Sid": "ConsoleFullAccess16",
 "Effect": "Allow",
 "Action": "ec2:CreateSecurityGroup",
 "Resource": "arn:aws:ec2:*:*:vpc/*"
},
{
 "Sid": "ConsoleFullAccess17",
 "Effect": "Allow",
 "Action":
 [
  "ec2:CreateSecurityGroup"
 ],
 "Resource": "arn:aws:ec2:*:*:security-group/*",
 "Condition": {
  "Null": {
   "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
  },
  "Bool": {
   "aws:ViaAWSService": "true"
  }
 }
},
{
 "Sid": "ConsoleFullAccess18",
 "Effect": "Allow",
 "Action": [
  "ec2:CreateSnapshot"
 ],
 "Resource": "arn:aws:ec2:*:*:volume/*",
 "Condition": {
  "Null": {
   "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
  },
  "Bool": {
   "aws:ViaAWSService": "true"
  }
 }
},
{
 "Sid": "ConsoleFullAccess19",
 "Effect": "Allow",
 "Action": [
  "ec2:CreateSnapshot"
```

```
    ],
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess20",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume",
     "ec2:AttachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess21",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume",
     "ec2:AttachVolume",
     "ec2:StartInstances",
     "ec2:GetConsoleOutput",
     "ec2:GetConsoleScreenshot"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "StringEquals": {
      "ec2:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
     },
     "ForAnyValue:StringEquals": {
```

```
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess22",
    "Effect": "Allow",
    "Action":
    [
     "ec2:AttachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool":
     {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess23",
    "Effect": "Allow",
    "Action": [
     "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:*:*:volume/*",
    "Condition":
    {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess24",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
```

```
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "Null": {
      "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
     },
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess25",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:subnet/*",
     "arn:aws:ec2:*:*:image/*",
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition":
    {
     "Bool": {
      "aws:ViaAWSService": "true"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess26",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:snapshot/*",
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
```

```
      "CreateSecurityGroup",
      "CreateVolume",
      "CreateSnapshot",
      "RunInstances"
     ]
    },
    "Bool": {
     "aws:ViaAWSService": "true"
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess27",
   "Effect": "Allow",
   "Action": "ec2:CreateTags",
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "StringEquals": {
     "ec2:CreateAction": [
      "CreateLaunchTemplate"
     ]
    }
   }
  },
  {
   "Sid": "ConsoleFullAccess28",
   "Effect": "Allow",
   "Action": [
    "cloudformation:DescribeStacks",
    "cloudformation:ListStacks"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess29",
   "Effect": "Allow",
   "Action": [
    "s3:GetBucketLocation",
    "s3:ListAllMyBuckets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ConsoleFullAccess30",
```

```
    "Effect": "Allow",
    "Action": [
     "ssm:DescribeInstanceInformation",
     "ssm:DescribeParameters"
    ],
    "Resource": [
     "*"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess31",
    "Effect": "Allow",
    "Action": [
     "ssm:SendCommand",
     "ssm:StartAutomationExecution"
    ],
    "Resource": [
     "arn:aws:ssm:*:*:automation-definition/AWS-CreateImage:$DEFAULT",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateNetworkConnectivity",
     "arn:aws:ssm:*:*:document/AWSMigration-VerifyMountedVolumes",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateHttpResponse",
     "arn:aws:ssm:*:*:document/AWSMigration-ValidateDiskSpace",
     "arn:aws:ssm:*:*:document/AWSMigration-VerifyProcessIsRunning",
     "arn:aws:ssm:*:*:document/AWSMigration-LinuxTimeSyncSetting",
     "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess32",
```

```
    "Effect": "Allow",
    "Action": [
     "ssm:SendCommand"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": [
       "drs.amazonaws.com"
      ]
     },
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess33",
    "Effect": "Allow",
    "Action": [
     "ssm:ListDocuments",
     "ssm:ListCommandInvocations"
    ],
    "Resource": "*"
   },
   {
    "Sid": "ConsoleFullAccess34",
    "Effect": "Allow",
    "Action": [
     "ssm:GetParameter",
     "ssm:PutParameter"
    ],
    "Resource": "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*",
    "Condition": {
     "StringEquals": {
         "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess35",
    "Effect": "Allow",
```

```
    "Action": [
     "ssm:DescribeDocument",
     "ssm:GetDocument"
    ],
    "Resource": "arn:aws:ssm:*:*:document/*"
   },
   {
    "Sid": "ConsoleFullAccess36",
    "Effect": "Allow",
    "Action": [
     "ssm:GetParameters"
    ],
    "Resource": [
     "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": "ssm.amazonaws.com"
     }
    }
   },
   {
    "Sid": "ConsoleFullAccess37",
    "Effect": "Allow",
    "Action": [
     "ssm:GetAutomationExecution"
    ],
    "Resource": "arn:aws:ssm:*:*:automation-execution/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   }
  ]
}
```

Launch Actions Policy - AWSElasticDisasterRecoveryLaunchActionsPolicy

This policy allows you to use Amazon SSM and additional services required permissions to run post-launch actions in AWS Elastic Disaster Recovery (AWS DRS). Attach this policy to your IAM roles or users.

Permissions details

This policy includes the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LaunchActionsPolicy1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy2",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/*",
                "arn:aws:ssm:*:*:automation-definition/*:*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                },
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy3",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/AWS-*",
                "arn:aws:ssm:*::document/AWSCodeDeployAgent-*",
                "arn:aws:ssm:*::document/AWSConfigRemediation-*",
                "arn:aws:ssm:*::document/AWSConformancePacks-*",
                "arn:aws:ssm:*::document/AWSDisasterRecovery-*",
                "arn:aws:ssm:*::document/AWSDistroOTel-*",
                "arn:aws:ssm:*::document/AWSDocs-*",
                "arn:aws:ssm:*::document/AWSEC2-*",
                "arn:aws:ssm:*::document/AWSEC2Launch-*",
                "arn:aws:ssm:*::document/AWSFIS-*",
                "arn:aws:ssm:*::document/AWSFleetManager-*",
                "arn:aws:ssm:*::document/AWSIncidents-*",
                "arn:aws:ssm:*::document/AWSKinesisTap-*",
                "arn:aws:ssm:*::document/AWSMigration-*",
                "arn:aws:ssm:*::document/AWSNVMe-*",
                "arn:aws:ssm:*::document/AWSNitroEnclavesWindows-*",
                "arn:aws:ssm:*::document/AWSObservabilityExporter-*",
                "arn:aws:ssm:*::document/AWSPVDriver-*",
                "arn:aws:ssm:*::document/AWSQuickSetupType-*",
                "arn:aws:ssm:*::document/AWSQuickStarts-*",
                "arn:aws:ssm:*::document/AWSRefactorSpaces-*",
                "arn:aws:ssm:*::document/AWSResilienceHub-*",
                "arn:aws:ssm:*::document/AWSSAP-*",
                "arn:aws:ssm:*::document/AWSSAPTools-*",
                "arn:aws:ssm:*::document/AWSSQLServer-*",
                "arn:aws:ssm:*::document/AWSSSO-*",
                "arn:aws:ssm:*::document/AWSSupport-*",
                "arn:aws:ssm:*::document/AWSSystemsManagerSAP-*",
                "arn:aws:ssm:*::document/AmazonCloudWatch-*",
                "arn:aws:ssm:*::document/AmazonCloudWatchAgent-*",
                "arn:aws:ssm:*::document/AmazonECS-*",
                "arn:aws:ssm:*::document/AmazonEFSUtils-*",
                "arn:aws:ssm:*::document/AmazonEKS-*",
```

```
                "arn:aws:ssm:*::document/AmazonInspector-*",
                "arn:aws:ssm:*::document/AmazonInspector2-*",
                "arn:aws:ssm:*::document/AmazonInternal-*",
                "arn:aws:ssm:*::document/AwsEnaNetworkDriver-*",
                "arn:aws:ssm:*::document/AwsVssComponents-*",
                "arn:aws:ssm:*::automation-definition/AWS-*:*",
                "arn:aws:ssm:*::automation-definition/AWSCodeDeployAgent-*:*",
                "arn:aws:ssm:*::automation-definition/AWSConfigRemediation-*:*",
                "arn:aws:ssm:*::automation-definition/AWSConformancePacks-*:*",
                "arn:aws:ssm:*::automation-definition/AWSDisasterRecovery-*:*",
                "arn:aws:ssm:*::automation-definition/AWSDistroOTel-*:*",
                "arn:aws:ssm:*::automation-definition/AWSDocs-*:*",
                "arn:aws:ssm:*::automation-definition/AWSEC2-*:*",
                "arn:aws:ssm:*::automation-definition/AWSEC2Launch-*:*",
                "arn:aws:ssm:*::automation-definition/AWSFIS-*:*",
                "arn:aws:ssm:*::automation-definition/AWSFleetManager-*:*",
                "arn:aws:ssm:*::automation-definition/AWSIncidents-*:*",
                "arn:aws:ssm:*::automation-definition/AWSKinesisTap-*:*",
                "arn:aws:ssm:*::automation-definition/AWSMigration-*:*",
                "arn:aws:ssm:*::automation-definition/AWSNVMe-*:*",
                "arn:aws:ssm:*::automation-definition/AWSNitroEnclavesWindows-*:*",
                "arn:aws:ssm:*::automation-definition/AWSObservabilityExporter-*:*",
                "arn:aws:ssm:*::automation-definition/AWSPVDriver-*:*",
                "arn:aws:ssm:*::automation-definition/AWSQuickSetupType-*:*",
                "arn:aws:ssm:*::automation-definition/AWSQuickStarts-*:*",
                "arn:aws:ssm:*::automation-definition/AWSRefactorSpaces-*:*",
                "arn:aws:ssm:*::automation-definition/AWSResilienceHub-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSAP-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSAPTools-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSQLServer-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSSO-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSupport-*:*",
                "arn:aws:ssm:*::automation-definition/AWSSystemsManagerSAP-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonCloudWatch-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonCloudWatchAgent-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonECS-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonEFSUtils-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonEKS-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInspector-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInspector2-*:*",
                "arn:aws:ssm:*::automation-definition/AmazonInternal-*:*",
                "arn:aws:ssm:*::automation-definition/AwsEnaNetworkDriver-*:*",
                "arn:aws:ssm:*::automation-definition/AwsVssComponents-*:*"
            ],
```

```
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy4",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
                    ]
                },
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy5",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSDRS": "AllowLaunchingIntoThisInstance"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "drs.amazonaws.com"
```

```
                    ]
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy6",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListCommandInvocations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "LaunchActionsPolicy7",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocumentVersions",
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": "arn:aws:ssm:*:*:document/*"
        },
        {
            "Sid": "LaunchActionsPolicy8",
            "Effect": "Allow",
            "Action": [
                "ssm:GetAutomationExecution"
            ],
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy9",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/
ManagedByAWSElasticDisasterRecoveryService-*",
```

```
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy10",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/
ManagedByAWSElasticDisasterRecoveryService-*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "LaunchActionsPolicy11",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceWithLaunchActionsRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                },
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "drs.amazonaws.com"
                }
            }
        }
    ]
}
```

Console Read-Only Access Policy - AWSElasticDisasterRecoveryReadOnlyAccess

You can attach the AWSElasticDisasterRecoveryReadOnlyAccess policy to your IAM identities.

This policy provides permissions to all read-only public APIs of AWS Elastic Disaster Recovery (AWS DRS), as well as some read-only APIs of IAM, EC2 and SSM in order to list and view installed roles Recovery Instances, Source Servers and post-launch actions. Attach this policy to your users or roles.

Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DRSReadOnlyAccess1",
   "Effect": "Allow",
   "Action": [
    "drs:DescribeJobLogItems",
    "drs:DescribeJobs",
    "drs:DescribeRecoveryInstances",
    "drs:DescribeRecoverySnapshots",
    "drs:DescribeReplicationConfigurationTemplates",
    "drs:DescribeSourceServers",
    "drs:GetFailbackReplicationConfiguration",
    "drs:GetLaunchConfiguration",
    "drs:GetReplicationConfiguration",
    "drs:ListExtensibleSourceServers",
    "drs:ListStagingAccounts",
    "drs:ListTagsForResource",
    "drs:ListLaunchActions"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess2",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeSecurityGroups",
```

```
    "ec2:DescribeSubnets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess4",
   "Effect": "Allow",
   "Action": "iam:ListRoles",
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess5",
   "Effect": "Allow",
   "Action": "ssm:ListCommandInvocations",
   "Resource": "*"
  },
  {
   "Sid": "DRSReadOnlyAccess6",
   "Effect": "Allow",
   "Action": "ssm:GetParameter",
   "Resource": "arn:aws:ssm:*:*:parameter/ManagedByAWSElasticDisasterRecovery-*"
  },
  {
   "Sid": "DRSReadOnlyAccess7",
   "Effect": "Allow",
   "Action": [
    "ssm:DescribeDocument",
    "ssm:GetDocument"
   ],
   "Resource": [
    "arn:aws:ssm:*:*:document/AWS-CreateImage",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateNetworkConnectivity",
    "arn:aws:ssm:*:*:document/AWSMigration-VerifyMountedVolumes",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateHttpResponse",
    "arn:aws:ssm:*:*:document/AWSMigration-ValidateDiskSpace",
    "arn:aws:ssm:*:*:document/AWSMigration-VerifyProcessIsRunning",
    "arn:aws:ssm:*:*:document/AWSMigration-LinuxTimeSyncSetting",
    "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure"
   ]
  },
  {
   "Sid": "DRSReadOnlyAccess8",
   "Effect": "Allow",
```

```
    "Action": [
     "ssm:GetAutomationExecution"
    ],
    "Resource": "arn:aws:ssm:*:*:automation-execution/*",
    "Condition": {
     "Null": {
      "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
     }
    }
   }
  ]
}
```

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

Using service-linked roles for AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery uses AWS Identity and Access Management (IAM)service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Elastic Disaster Recovery. Service-linked roles are predefined by AWS Elastic Disaster Recovery and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS Elastic Disaster Recovery easier because you don’t have to manually add the necessary permissions. AWS Elastic Disaster Recovery defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Elastic Disaster Recovery can

assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your AWS Elastic Disaster Recovery resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery uses the service-linked role named **AWSServiceRoleForElasticDisasterRecovery**. This role includes a managed IAM policy AWSElasticDisasterRecoveryServiceRolePolicy with scoped permissions that AWS Elastic Disaster Recovery needs to run in your account.

The AWSServiceRoleForElasticDisasterRecovery service-linked role trusts the following services to assume the role: `drs.amazonaws.com`

The role permissions policy allows AWS Elastic Disaster Recovery to complete the following actions on the specified resources.

```
{
 "Version": "2012-10-17",
 "Statement": [
        {
            "Sid": "DRSServiceRolePolicy1",
            "Effect": "Allow",
            "Action": [
                "drs:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy2",
            "Effect": "Allow",
            "Action": [
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:recovery-instance/*"
```

```
        },
        {
            "Sid": "DRSServiceRolePolicy3",
            "Effect": "Allow",
            "Action": [
                "drs:CreateRecoveryInstanceForDrs",
                "drs:TagResource"
            ],
            "Resource": "arn:aws:drs:*:*:source-server/*"
        },
        {
            "Sid": "DRSServiceRolePolicy4",
            "Effect": "Allow",
            "Action": "iam:GetInstanceProfile",
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy5",
            "Effect": "Allow",
            "Action": "kms:ListRetirableGrants",
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy6",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshots",
                "ec2:DescribeSubnets",
                "ec2:DescribeVolumes",
                "ec2:DescribeVolumeAttribute",
                "ec2:GetEbsDefaultKmsKeyId",
                "ec2:GetEbsEncryptionByDefault",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeInternetGateways",
```

```
                "ec2:DescribeVpcs",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeRouteTables",
                "ec2:DescribeDhcpOptions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy7",
            "Effect": "Allow",
            "Action": [
                "ec2:RegisterImage"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DRSServiceRolePolicy8",
            "Effect": "Allow",
            "Action": [
                "ec2:DeregisterImage"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy9",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:snapshot/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy10",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy11",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteVolume",
                "ec2:ModifyVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy12",
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "ec2:ModifyInstanceAttribute",
                "ec2:GetConsoleOutput",
                "ec2:GetConsoleScreenshot"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
```

```
        },
        {
            "Sid": "DRSServiceRolePolicy13",
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy14",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy15",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy16",
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
        {
            "Sid": "DRSServiceRolePolicy17",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplate"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy18",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy19",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot"
            ],
            "Resource": "arn:aws:ec2:*:*:snapshot/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
```

```
        },
        {
            "Sid": "DRSServiceRolePolicy20",
            "Effect": "Allow",
            "Action": [
                "ec2:DetachVolume",
                "ec2:AttachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy21",
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy22",
            "Effect": "Allow",
            "Action": [
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
        },
        {
            "Sid": "DRSServiceRolePolicy23",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
```

```
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy24",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:image/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:launch-template/*"
            ]
        },
        {
            "Sid": "DRSServiceRolePolicy25",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryReplicationServerRole",
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryConversionServerRole",
                "arn:aws:iam::*:role/service-role/
AWSElasticDisasterRecoveryRecoveryInstanceRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy26",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:launch-template/*",
```

```
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:snapshot/*",
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "CreateLaunchTemplate",
                        "CreateSecurityGroup",
                        "CreateVolume",
                        "CreateSnapshot",
                        "RunInstances"
                    ]
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy27",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:image/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSElasticDisasterRecoveryManaged": "false"
                }
            }
        },
        {
            "Sid": "DRSServiceRolePolicy28",
            "Effect": "Allow",
            "Action": "cloudwatch:GetMetricData",
            "Resource": "*"
        }
    ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for AWS Elastic Disaster Recovery

You don't need to manually create a service-linked role. When you configure the Replication Configuration Template for AWS Elastic Disaster Recovery, a service-linked role is automatically created. AWS Elastic Disaster Recovery automatically creates the IAM service-linked role, which you can see in the IAM console. You don't need to manually create or configure this role.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create the first new replication configuration template in AWS Elastic Disaster Recovery, it creates the service-linked role for you again.

In the AWS CLI or the AWS API, create a service-linked role with the AWS Elastic Disaster Recovery service name. For more information, see Creating a Service-Linked Role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

Editing a service-linked role for AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery does not allow you to edit the AWSServiceRoleForElasticDisasterRecovery service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS Elastic Disaster Recovery

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If AWS Elastic Disaster Recovery is using the role when you try to delete the resources, the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To clean up AWS Elastic Disaster Recovery resources used by AWSServiceRoleForElasticDisasterRecovery

Resources can be cleaned up without stopping any AWS Elastic Disaster Recovery services. Cleaning up AWS Elastic Disaster Recovery resources will cause AWS Elastic Disaster Recovery to stop working. For more information, see Cleaning up a service-linked role in the *IAM User Guide*.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForElasticDisasterRecovery service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Supported AWS Regions for AWS Elastic Disaster Recovery service-linked roles

AWS Elastic Disaster Recovery supports using service-linked roles in all of the AWS Regions where the service is available.

Policy structure

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows:

```
{
 "Statement": [
  {
   "Effect": "effect",
   "Action": "action",
   "Resource": "arn",
   "Condition": {
    "condition": {
     "key": "value"
    }
   }
  }
 ]
}
```

There are various elements that make up a statement:

- **Effect:** The effect can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.

- **Action**: The action is the specific AWS Elastic Disaster Recovery API action for which you are granting or denying permission.
- **Resource**: The resource that's affected by the action. For AWS Elastic Disaster Recovery, you must specify "*" as the resource.
- **Condition**: Conditions are optional. They can be used to control when your policy is in effect.

Resilience in AWS Elastic Disaster Recovery

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Elastic Disaster Recovery

As a managed service, AWS Elastic Disaster Recovery is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access application recovery Service through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

All parties involved in the communication authenticate each other using TLS, IAM policies and tokens. The communication between the Agents and the replication server are based on TLS 1.2 only with the highest standard of cipher suite (PFS, ECDHE). Requests between the agent and AWS Elastic Disaster Recovery as well as between the replication server and AWS Elastic Disaster Recovery are signed using an access key ID and a secret access key that is associated with an IAM principal.

Additionally, requests must be signed using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS Elastic Disaster Recovery customers must ensure that they manually delete their access keys after installing the AWS Replication Agent and successful recovery. AWS does not delete these keys automatically. AWS Elastic Disaster Recovery does delete the keys from source servers after they are disconnected from the service. If you want your keys to automatically stop working at a certain date after you have finished using them so that you do not have to worry about manually deleting them, you can do so though the IAM permissions boundary and the aws:CurrentTime global context key.

AWS Elastic Disaster Recovery customers should use Amazon EBS encryption.

AWS Elastic Disaster Recovery customers should secure their replication servers by reducing their exposure to the public internet. This can be done through:

1. Using Security Groups to only allow permitted IP addresses to connect to the replication servers. Learn more about Security Groups.
2. Using a VPN to connect to the replication servers, such as the AWS site-to-site VPN. Learn more about the AWS Site-to-site VPN.

AWS Elastic Disaster Recovery creates and uses the "aws-replication" user within the Source server. The AWS Elastic Disaster Recovery replication server and AWS Replication Agent run under this user. Although this is not a root user, this user needs to be part of the disk group that grants this user full read and write permissions to block devices.

> **Note**
>
> AWS Elastic Disaster Recovery only uses these permissions to read from block devices.

AWS Elastic Disaster Recovery customers should only grant access to the AWS Elastic Disaster Recovery Failback Client to trusted administrators in order to prevent unauthorized entities from gaining access to your systems through the client.

AWS GovCloud

AWS GovCloud (US) are isolated AWS Regions designed to allow U.S. government agencies and customers to move sensitive workloads into the cloud.

- AWS GovCloud (US) uses FIPS 140-2 approved cryptographic modules for all AWS service API endpoints, unless otherwise indicated in the Service Endpoints section.
- AWS GovCloud (US) is appropriate for all types of Controlled Unclassified Information (CUI) and unclassified data. For more details, see Maintaining U.S. International Traffic in Arms Regulations (ITAR) Compliance .
- The AWS GovCloud (US) Regions are physically isolated and have logical network isolation from all other AWS Regions.
- AWS restricts all physical and logical access for those staff supporting AWS GovCloud (US) to US Citizens. AWS allows only vetted U.S. citizens with distinct access controls separate from other AWS Regions to administer AWS GovCloud (US). Any customer data fields that are defined as outside of the ITAR boundary (such as S3 bucket names) are explicitly documented in the service-specific section as not permitted to contain export-controlled data.
- AWS GovCloud (US) authentication is completely isolated from commercial regions.

Compliance validation for AWS Elastic Disaster Recovery

Third-party auditors assess the security and compliance of AWS Elastic Disaster Recovery as part of multiple AWS compliance programs.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program . For general information, see AWS Compliance Programs .

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact .

Your compliance responsibility when using AWS Elastic Disaster Recovery is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.

- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that AWS Elastic Disaster Recovery gives another service to the resource. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement.

The value of `aws:SourceArn` must be "arn:aws:drs:*:123456789012:source-server/*"

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (*) for the unknown portions of the ARN. For example, `arn:aws:servicename::123456789012:*` .

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in AWS Elastic Disaster Recovery to prevent the confused deputy problem.

IAM Roles that are created by AWS Elastic Disaster Recovery in your account already contain the confused deputy mitigation.

```
{
 "Version": "2012-10-17",
 "Statement": {
  "Sid": "ConfusedDeputyPreventionExamplePolicy",
  "Effect": "Allow",
  "Principal": {
   "Service": "drs.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
   "StringLike": {
    "aws:SourceArn": "arn:aws:drs:*:123456789012:source-server/*",
    "aws:SourceAccount": "123456789012"
   }
  }
 }
}
```

Monitoring

Topics

Logging AWS Elastic Disaster Recovery API calls using AWS CloudTrail

AWS Elastic Disaster Recovery is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Elastic Disaster Recovery. CloudTrail captures all API calls for AWS Elastic Disaster Recovery as events. The calls captured include calls from the AWS Elastic Disaster Recovery console and code calls to the AWS Elastic Disaster Recovery API operations. If you create a trail, you can activate continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Elastic Disaster Recovery. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Elastic Disaster Recovery, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see theAWS CloudTrail User Guide.

AWS Elastic Disaster Recovery information in CloudTrail

CloudTrail is activated on your AWS account when you create the account. When activity occurs in AWS Elastic Disaster Recovery, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, seeViewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for AWS Elastic Disaster Recovery, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

All AWS Elastic Disaster Recovery actions are logged by CloudTrail and are documented in the AWS Elastic Disaster Recovery API. For example, calls to the `DescribeSourceServers` action to generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see theCloudTrail userIdentity element.

Understanding AWS Elastic Disaster Recovery log file entries

A trail is a configuration that allows delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the DescribeSourceServers.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAAAAAAAAAAAAAAAAAA",
        "arn": "arn:aws:sts::1234567890:assumed-role/Admin/user-Isengard",
```

```
        "accountId": "1234567890",
        "accessKeyId": "BBBBBBBBBBBBBBBBBBB",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AAAAAAAAAAAAAAAAAAA",
                "arn": "arn:aws:iam::1234567890:role/Admin",
                "accountId": "1234567890",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-10-20T14:19:17Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-10-20T14:19:59Z",
    "eventSource": "drs.amazonaws.com",
    "eventName": "DescribeSourceServers",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "54.240.197.234",
    "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/94.0.4606.81
    Safari/537.36",
    "requestParameters": {
        "maxResults": 1000,
        "filters": {}
    },
    "responseElements": null,
    "requestID": "d7618669-db08-4b53-bf6e-8a2cd57a677d",
    "eventID": "436c17a7-3a54-4f4e-815d-4d980339744e",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "1234567890",
    "eventCategory": "Management"
}
```


CloudWatch Metrics for DRS

The following are CloudWatch metrics for DRS:

- **TotalSourceServerCount** - number of source servers
- **LagDuration** - the age of the latest consistent snapshot, in seconds
- **Backlog** - the amount of data yet to be synced, in bytes.
- **DurationSinceLastSuccessfulRecoveryLaunch** - the amount of time that has passed since the last Drill or Recovery instance launch in seconds.
- **ElapsedReplicationDuration** - the cumulative amount of time this server has been replicating for in seconds.

Alarm events and EventBridge

Sample events for Elastic Disaster Recovery

The following are sample events for Elastic Disaster Recovery:

Source server data replication status

These events are triggered when source servers' data replication state changes from Stalled (replication not functioning properly) and not stalled (replication is functioning as expected).

STALLED

```
{
 "version": "0",
 "id": "9da9af57-9253-4406-87cb-7cc400e43465",
 "detail-type": "DRS Source Server Data Replication Stalled Change",
 "source": "aws.drs",
 "account": "111122223333",
 "time": "2016-08-22T20:12:19Z",
 "region": "us-west-2",
 "resources": [
  "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
 ],
 "detail": {
  "state": "STALLED"
 }
}
```

NOT_STALLED

```
{
 "version": "0",
 "id": "9da9af57-9253-4406-87cb-7cc400e43465",
 "detail-type": "DRS Source Server Data Replication Stalled Change",
 "source": "aws.drs",
 "account": "111122223333",
 "time": "2016-08-22T20:12:19Z",
 "region": "us-west-2",
 "resources": [
  "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
 ],
 "detail": {
  "state": "NOT_STALLED"
 }
}
```

Source server launch result

These events are triggered when a drill or recovery instance is launched for a source server and indicate whether the launch succeeded or failed.

RECOVERY_LAUNCH_SUCCEEDED

```
{
    "version": "0",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "detail-type": "DRS Source Server Launch Result",
    "source": "aws.drs",
    "account": "111122223333",
    "time": "2016-08-22T20:12:19Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
    ],
    "detail": {
        "state": "RECOVERY_LAUNCH_SUCCEEDED",
        "job-id": "drsjob-04ca7d0d3fb6afa3e",
        "is-drill": "FALSE"
    }
}
```

RECOVERY_LAUNCH_FAILED

```
{
    "version": "0",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "detail-type": "DRS Source Server Launch Result",
    "source": "aws.drs",
    "account": "111122223333",
    "time": "2016-08-22T20:12:19Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
    ],
    "detail": {
        "state": "RECOVERY_LAUNCH_FAILED",
        "job-id": "drsjob-04ca7d0d3fb6afa3e",
        "is-drill": "FALSE"
    }
}
```

Recovery instance failback State Change

These events are triggered as part of the failback process and indicate if failback is in progress, completed or failed.

FAILBACK_IN_PROGRESS

```
{
    "version": "0",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "detail-type": "DRS Recovery Instance Failback State Change",
    "source": "aws.drs",
    "account": "111122223333",
    "time": "2016-08-22T20:12:19Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:recovery-instance/ri-12345678901234567"
    ],
```

```
    "detail": {
    "state": "FAILBACK_IN_PROGRESS"
    }
}
```

FAILBACK_COMPLETED

```
{
    "version": "0",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "detail-type": "DRS Recovery Instance Failback State Change",
    "source": "aws.drs",
    "account": "111122223333",
    "time": "2016-08-22T20:12:19Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:recovery-instance/ri-12345678901234567"
    ],
    "detail": {
        "state": "FAILBACK_COMPLETED"
    }
}
```

FAILBACK_ERROR

```
{
    "version": "0",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "detail-type": "DRS Recovery Instance Failback State Change",
    "source": "aws.drs",
    "account": "111122223333",
    "time": "2016-08-22T20:12:19Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:recovery-instance/ri-12345678901234567"
    ],
    "detail": {
    "state": "FAILBACK_ERROR"
    }
```

```
}
```

PIT Snapshot Taken

This event is triggered whenever a point in time snapshot is taken and includes its identifiers.

PIT Snapshot Taken

```
{
    "account": "111122223333",
    "detail": {
        "DrsSnapshotID": "112233",
        "EbsSnapshotIDs": "445566,778899"
    },
    "detail-type": "DRS PIT Snapshot Taken",
    "id": "9da9af57-9253-4406-87cb-7cc400e43465",
    "region": "us-west-2",
    "resources": [
        "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
    ],
    "source": "aws.drs",
    "time": "2016-08-22T20:12:19Z",
    "version": "0"
}
```

Registering event rules

You create EventBridge rules that capture events coming from your Elastic Disaster Recovery resources.

> **Note**
>
> When you use the AWS Management Console to create an event rule, the console automatically adds the IAM permissions necessary to grant EventBridge Event permissions to call your desired target type. If you are creating an event rule using the AWS CLI, you must grant permissions explicitly. For more information, see Event Patterns in the*Amazon EventBridge User Guide*.

To create Amazon EventBridge rules

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Using the following values, create an EventBridge rule that captures events coming from Elastic Disaster Recovery resources:

   - For **Rule type**, choose **Rule with an event pattern**.
   - For **Event source**, choose **Other**.
   - For **Event pattern**, choose **Custom patterns (JSON editor)**, and paste one of the following event pattern examples into the text area:
     - To catch all Elastic Disaster Recovery events:

```
{
 "source": [
  "aws.drs"
 ]
}
```

     - To catch all Recovery instance failback state changes:

```
{
 "detail-type": [
  "DRS Recovery Instance Failback State Change"
 ],
 "source": [
  "aws.drs"
 ]
}
```

     - To catch all events relating to a given Source server:

```
{
 "source": [
  "aws.drs"
 ],
 "resources": [
  "arn:aws:drs:us-west-2:111122223333:source-server/s-12345678901234567"
 ]
}
```

- For **Target types**, choose**AWS service**, and for **Select a target** choose your desired target.

For details about creating rules, see Creating Amazon EventBridge rules that react to events in the**Amazon EventBridge User Guide**.

Troubleshooting

Topics

Troubleshooting Failback Errors

Topics

Error – Could not associate failback client to recovery instances

If you see the "Could not associate failback client to recovery instances" error when using the Failback Client, that may mean that you associated the incorrect credentials with your User. Ensure that you attach the **AWSElasticDisasterRecoveryFailbackInstallationPolicy** policy to the user or role and restart the failback process. Learn more about Failback Client credentials.

Error – Could not verify recovery instance connectivity to DRS

If you see the "Could not verify recovery instance connectivity to Elastic Disaster Recovery" error when using the Failback Client, you should troubleshoot potential connectivity issues:

1. Make sure that the agent on the recovery instance is activated and running.
2. A public IP must be set on the recovery instance in Amazon EC2.
3. TCP Port 443 outbound must be open on the recovery instance for the pairing to succeed.

4. Make sure that you don't have this error in your agent logs: Error – driver was compiled for a different kernel not loading.

Error message: AWS Replication agent is not connected to DRS. Verify the agent is installed and running, and that it has connectivity to the service

In certain cases, following an attempt to perform a reverse replication action, you will receive an error message indicating that the AWS Replication agent is not connected to AWS Elastic Disaster Recovery. In this case, verify that:

1. The agent is installed and running
2. The server is connected to the internet or the NAT gateway

If after performing the steps above you did not identify any agent or connectivity issues, reinstall the agent as recovery instance and try again.

Error message: botocore.exceptions.CredentialRetrievalError: Error when retrieving credentials from cert

The Failback Client uses Amazon Linux 2 (AL2) and leverages certificate-based authentication to AWS Elastic Disaster Recovery endpoints for certain actions. AL2 assumes that the hardware clock time provided from the underlying hardware or hypervisor is UTC, which can result in time skew if it is not. Ensure that the time configured within the BIOS or EFI Shell of the failback target is set to UTC, and not LocalTime.

Troubleshooting Communication Errors

Topics

- Solving Communication Problems over TCP Port 443 between the staging area and the Elastic Disaster Recovery Service Manager
- Calculating the required bandwidth for TCP Port 1500
- Verifying Communication over Port 1500
- Solving Communication Problems over Port 1500

Solving Communication Problems over TCP Port 443 between the staging area and the Elastic Disaster Recovery Service Manager

- **DHCP** – Check the DHCP options set of the VPC of the staging area.

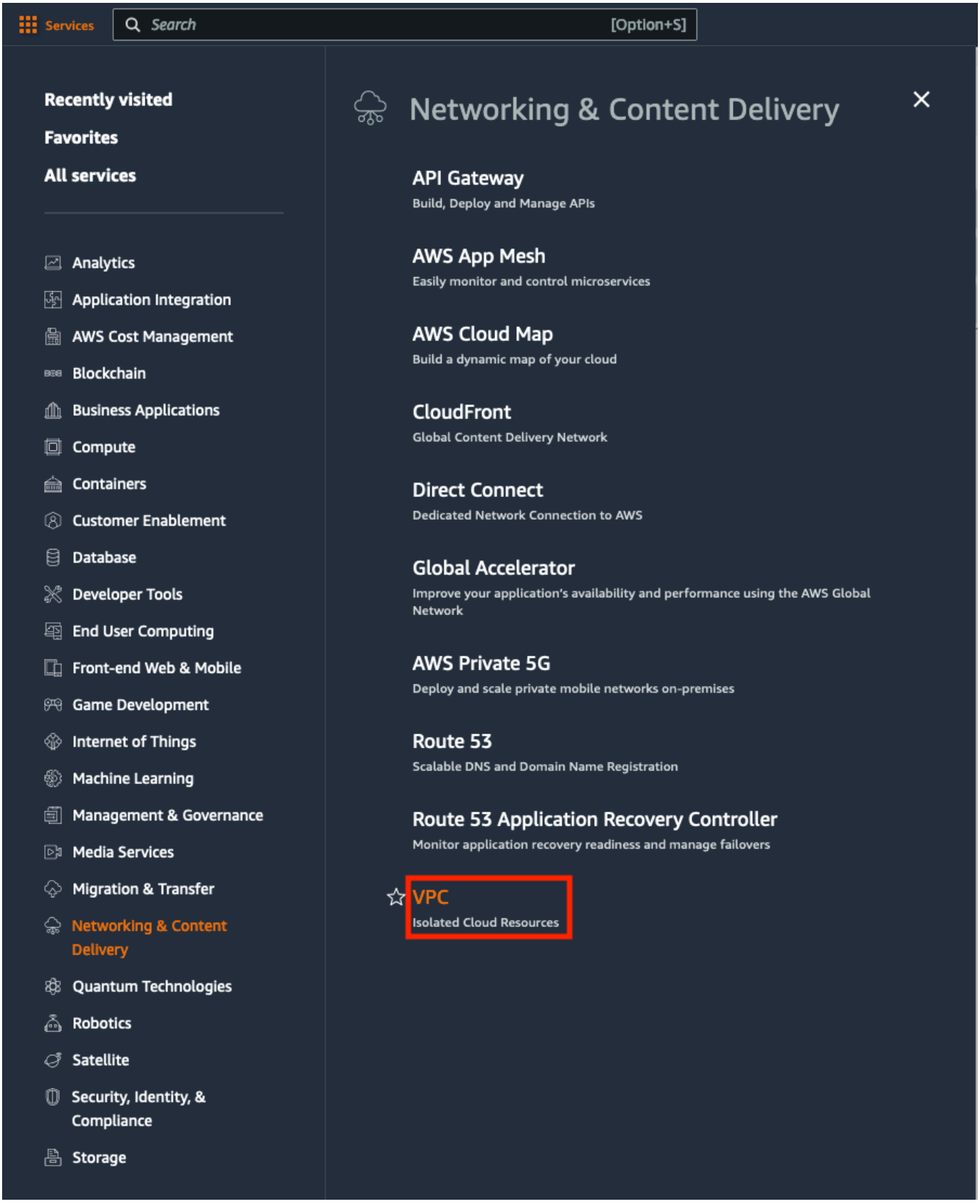
  Ensure that the IPv4 CIDR, the DHCP options set, the Route table, and the Network ACL are correct.

- **DNS** – Ensure that you are allowing outbound DNS resolution and connectivity over TCP Port 443.

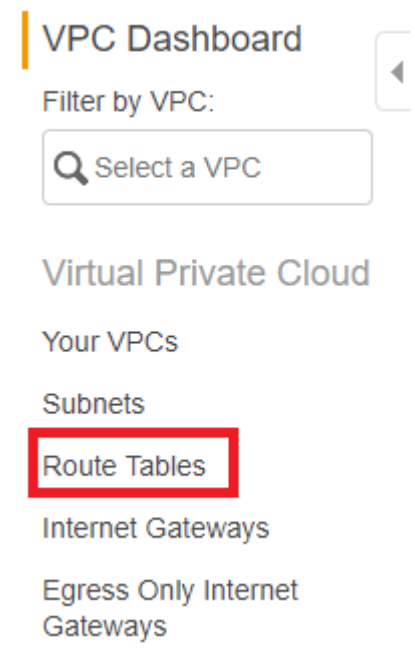
- **Route Rules** – the Route Rules on the Staging Area subnet may be inaccurately set. The Route Rules should allow outbound traffic to the Internet.

  To check and set the Route Rules on the staging area subnet:

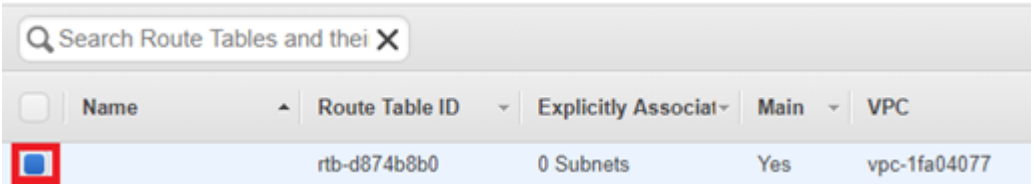
  1. Sign in to AWS console, click on **Services** and select **VPC** under **Networking & Content Delivery**.



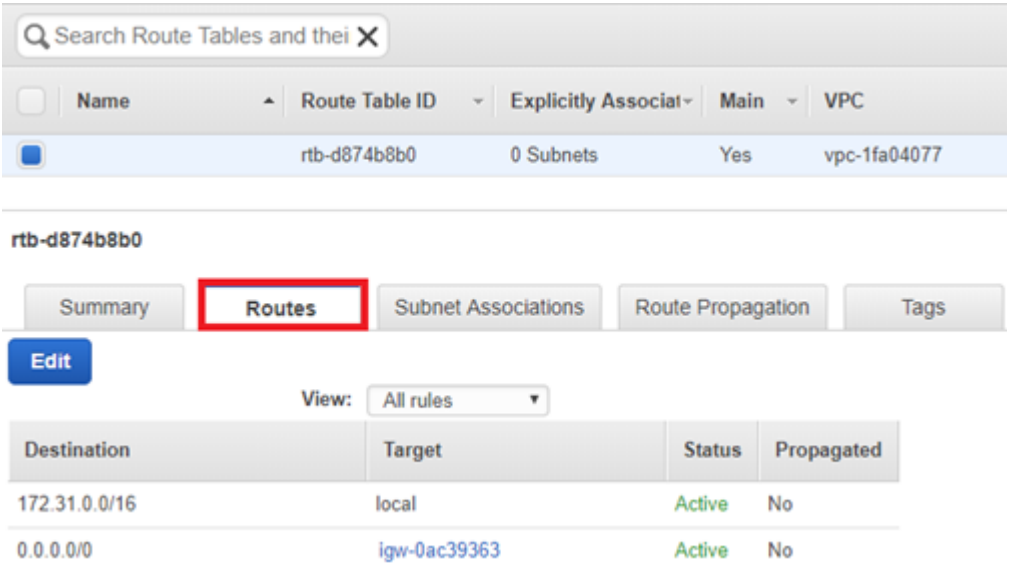
2. On the **VPC Dashboard** toolbar, select the **Route Tables** option.



3. On **Route Tables** page, check the box of the Route Table of your staging area.



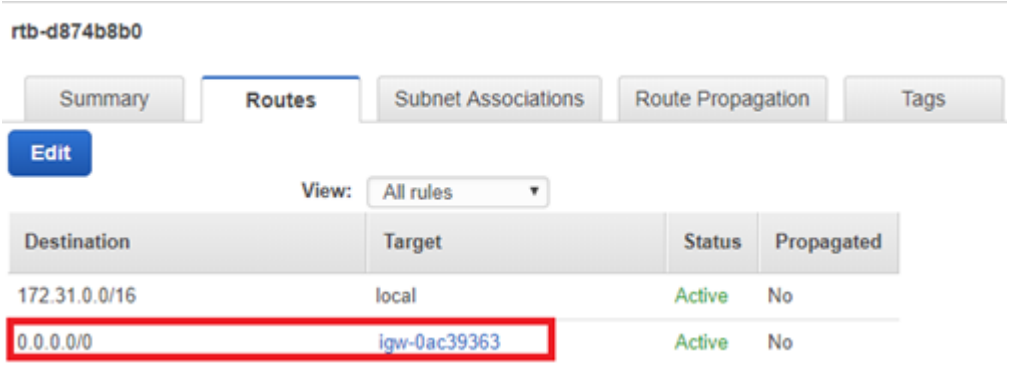
4. This will open the details for your Route Table. Navigate to the **Routes** tab.



5. Within the **Target** column of the **Routes** tab, find the route you are using for the outbound communication to the Internet (either **igw** – Internet Gateway, vgw – **VPN** or **i** – EC2 instance).

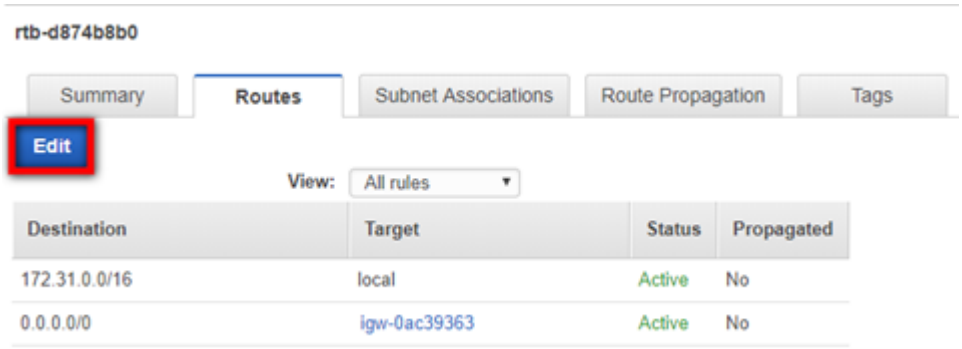
Verify that the address space in the Destination column is covering the AWS Elastic Disaster Recovery IPs and URLs.

Note: AWS Elastic Disaster Recovery AWS-specific IPs and URLs include: 52.72.172.158, 52.53.92.136, s3.amazonaws.com, s3.us-west-1.amazonaws.com, s3.eu-west-1.amazonaws.com and outbound access to the Amazon EC2 endpoint of the AWS Region.



6. If the address is not **0.0.0.0/0**, you will need change it to **0.0.0.0/0.**

   Click the **Edit** button.



7. Input **0.0.0.0/0** into the Destination field for the correct **Target**. Click **Save**.

   Note: If you are using VPN, enter a specific IP address range in the **Destination** column.

- **Network ACL** – The network ACL on the staging area subnet may block the traffic. Verify that the ephemeral ports are open.

Calculating the required bandwidth for TCP Port 1500

The required bandwidth for transferring the replicated data over TCP Port 1500 should be based on the write speed of the participating Source machines. The recommended bandwidth should be at least the sum of the average write speed of all replicated source machines.

Minimal bandwidth = the sum of the write speed of all Source machines

For example, suppose you are replicating two Source machines. One has a write speed of 5 MBps (meaning it 5 megabytes of data every second), while the other has 7 MBps. In this case, the recommended bandwidth should be at least 12 MBps.

Finding the Write Speed of Your source servers

To calculate the required bandwidth for transferring replicated data over TCP Port 1500, you need to know the write speed of your source machines. Use the following tools to find the write speed of your source servers:

Linux

Use the iostat command-line utility, located in the systat package. The iostat utility monitors system input/output device loading and generates statistical reports.

The iostat utility is installed with yum (RHEL/CentOS), via apt-get (Ubuntu), and via zypper (SUSE).

To use iostat for checking the write speed of a Source machine, enter the following: iostat -x <interval>

- -x - displays extended statistics.
- <interval> – the number of seconds iostat waits between each report. Each subsequent report covers the time since the previous report.

For example, to check the write speed of a machine every 3 seconds, enter the following command:

iostat -x 3

We recommend that you run the iostat utility for at least 24 hours, since the write speed to the disk changes during the day, and it will take 24 hours of runtime to identify the average running speed.

Windows

Install and use the DiskMon application. DiskMon logs and displays all hard disk activity on a Windows system.

Installing DiskMon

DiskMon presents read and write offsets are presented in terms of sectors (512 bytes). Events can be either timed for their duration (in microseconds), or stamped with the absolute time that they were initiated.

Verifying Communication over Port 1500

If there is a connection problem from the Source server to the Replication Servers or the Staging Area, use the following methods to check the connection.

To verify the integrity of the connection from a Source server to the Staging Area over TCP Port 1500:

1. Launch a new Linux machine in the Staging Area subnet.
2. On the new Linux machine, run the following command to open a listener in the Staging Area subnet:

   nc -l 1500
3. On the Source machine, run the following command to check connectivity:

   telnet <new machine ip> 1500

Solving Communication Problems over Port 1500

To solve connectivity problems between Source server and the staging area, check the following:

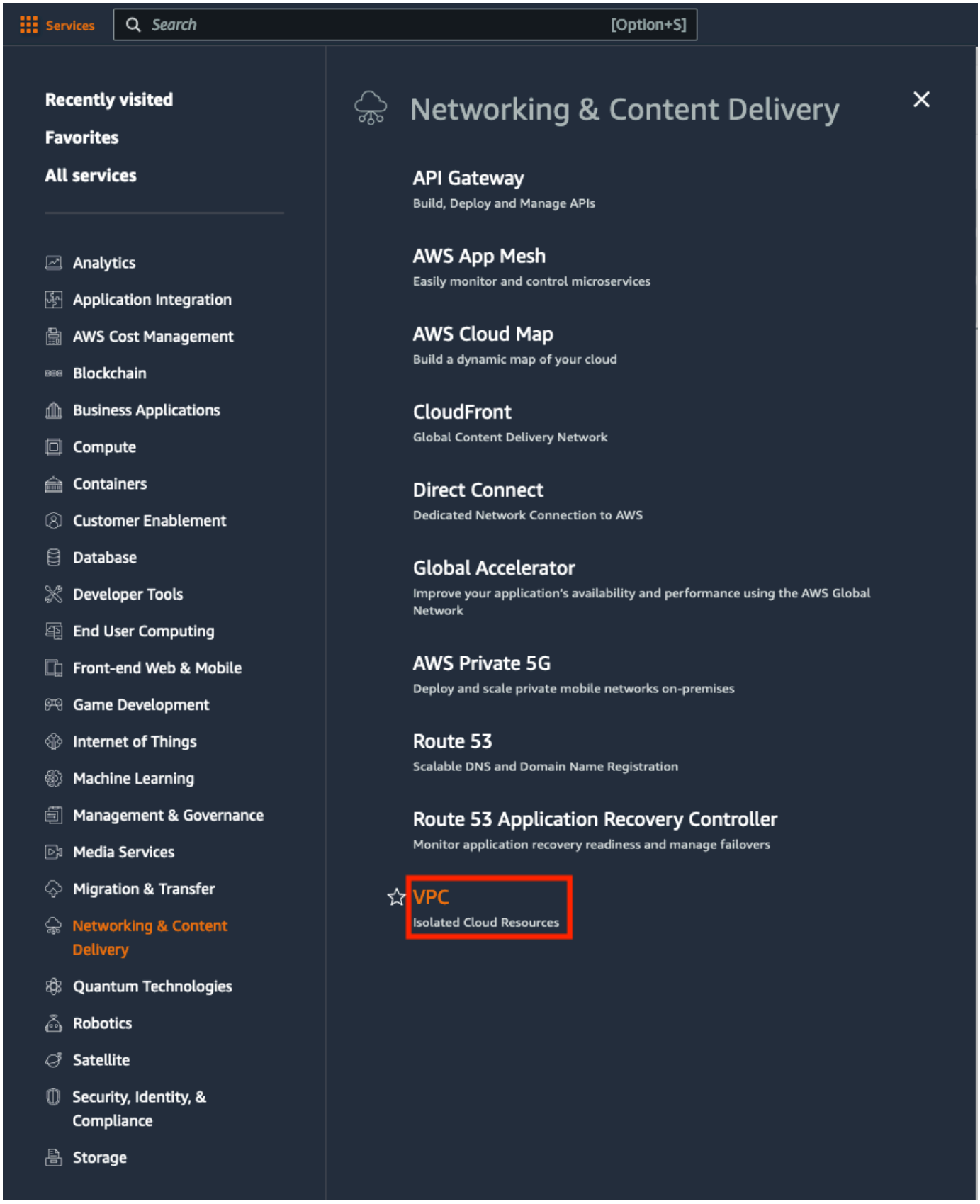
- The Network ACL on the Staging Area subnet may deny the traffic.
- Route Rules on the staging area subnet may be inaccurately set.
- The firewall, both internal and external, in the Source machine/infrastructure may block communication.
- The **Use VPN...**checkbox in the Elastic Disaster Recovery Console may not be set correctly.

Enabling the Network ACL

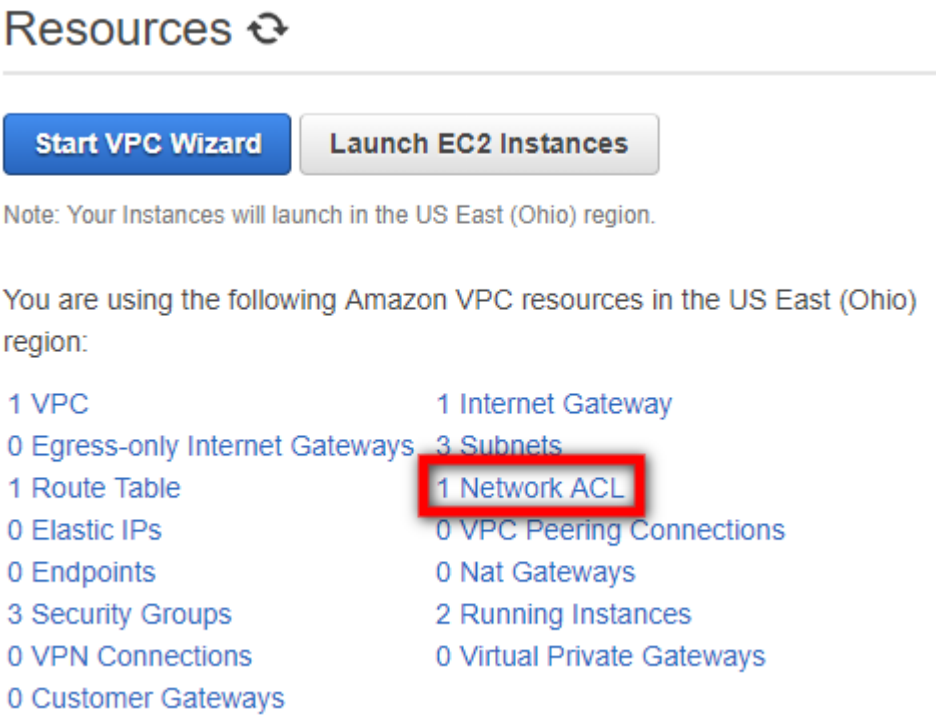
The Network ACL on the staging area subnet may block connectivity. By default, the Network ACL allows connectivity. However, if the ACL setting was changed to deny traffic, you need to change it back.

To check and activate the network ACL on the staging area subnet:

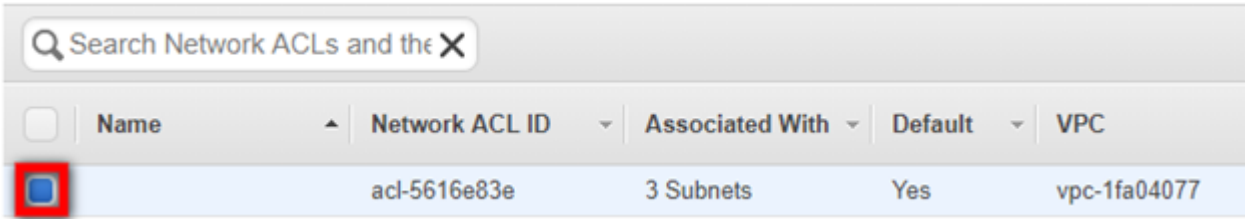
1. Sign in to the AWS console, click on **Services** and select **VPC** under **Networking & Content Delivery.**



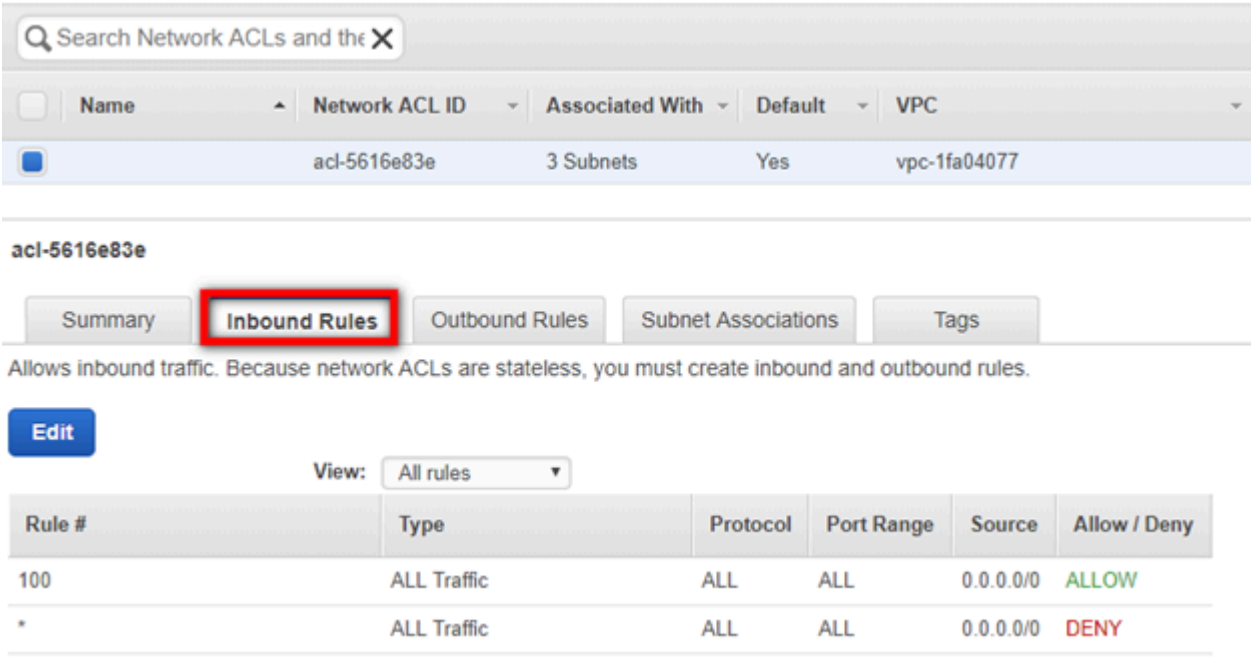
2. On the **Resources** list, select the **Network ACL** option:



3. On **Network ACL** page, select the check box next to the Network ACL of your staging area.

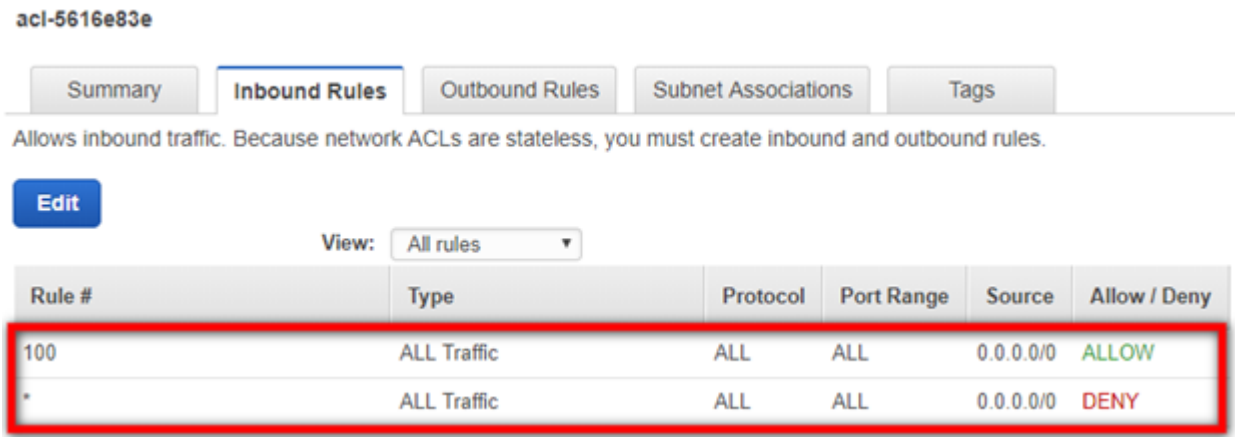


4. On the details table of the selected **Network ACL**, select the **Inbound Rules** tab.

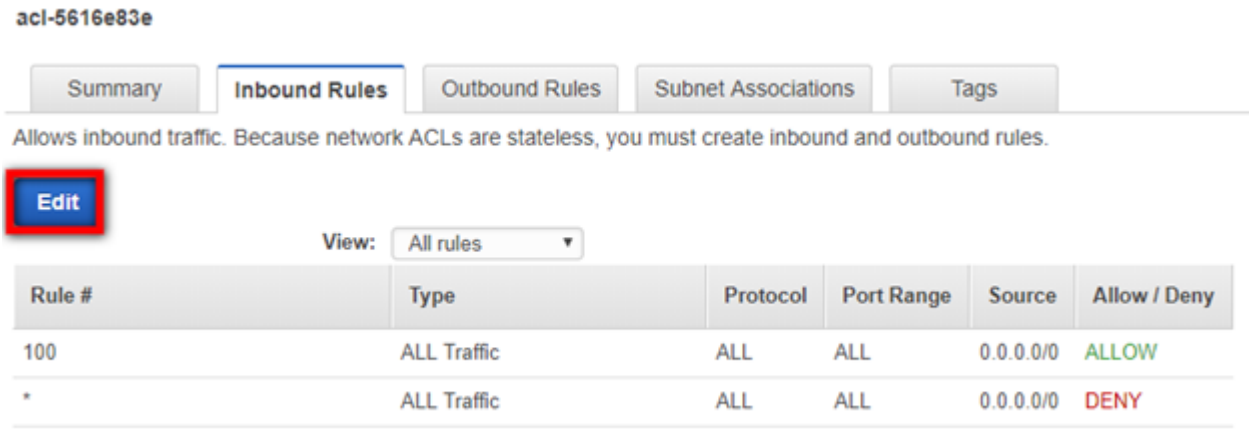


5. On the **Inbound Rules** tab, verify that the Rule that determines the traffic to replication server subnet set to **Allow**.

   Note: The Target should allow traffic on TCP Port 1500 from the address space of the Source environment. The Network ACL does not necessarily need to be open to all Port Ranges, as in the screenshot below.

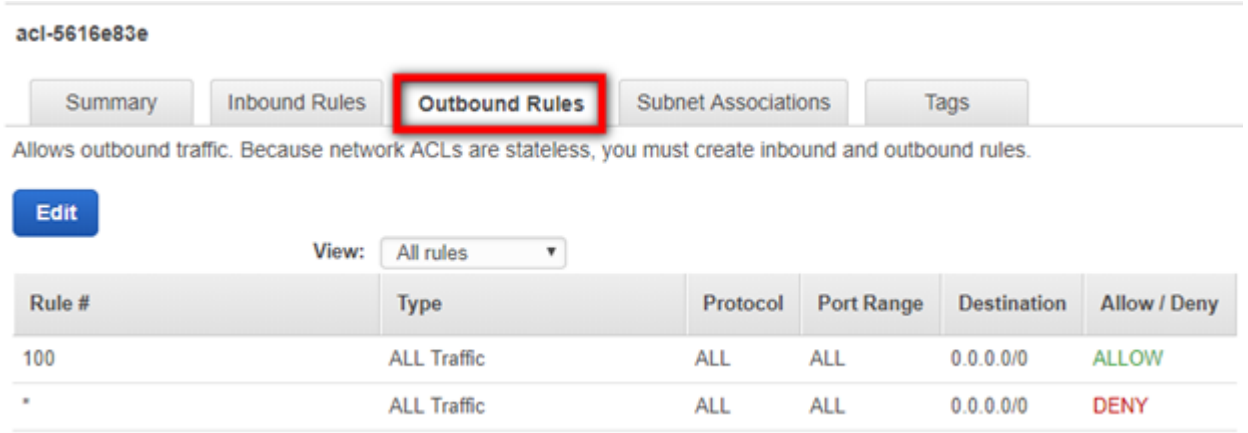


6. If the rule is set to **Deny**, click **Edit**.

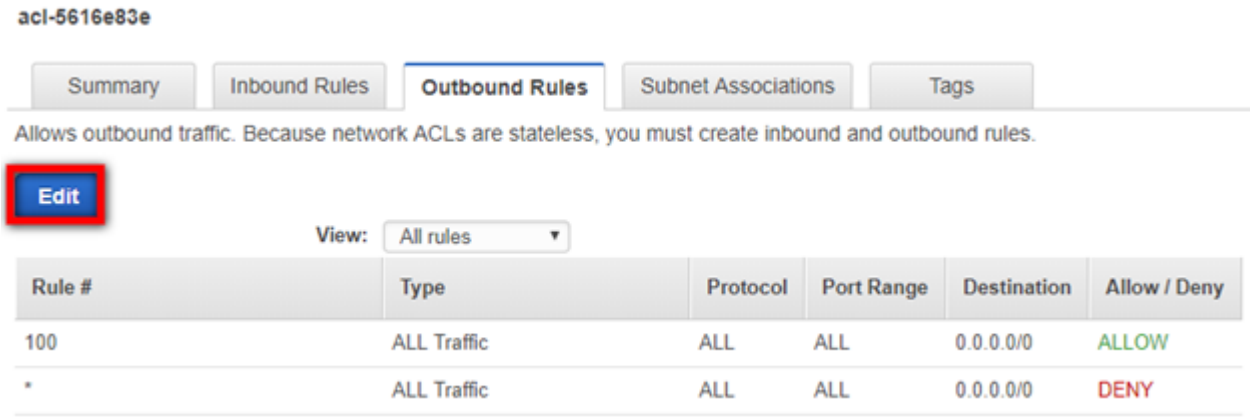


7. Click the dropdown under **Allow/Deny** and select **Allow**. Click **Save**.

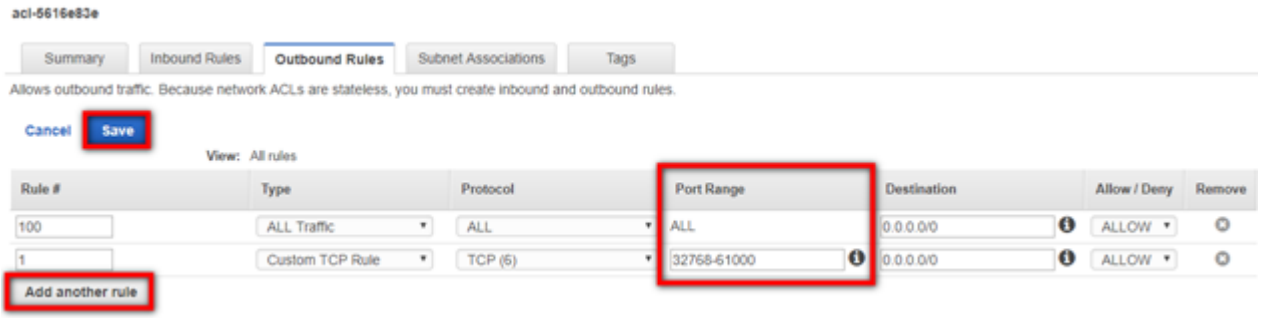
8. You will also need to check the **Ephemeral Ports** on the **Outbound Rules** tab. Within the same **Network ACL**, navigate to the **Outbound Rules** tab.



9. You will need to ensure that you are allowing the correct **Ephemeral Port range** for your particular client. Ephemeral Port range varies based on each client's operating system. Click the Edit button to edit your **Ephemeral Port's Port Range** category.



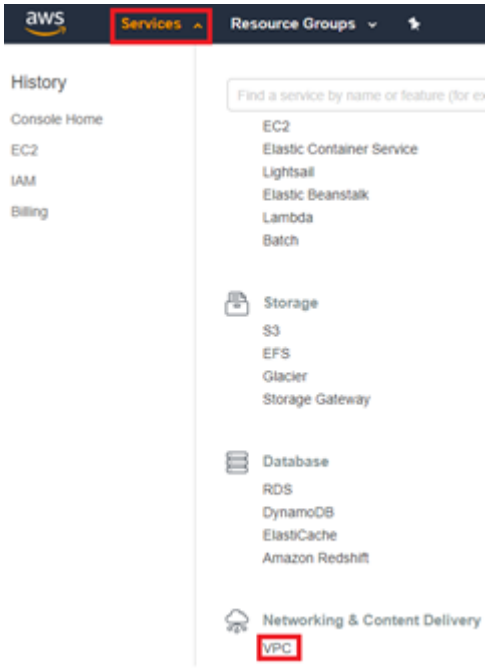
10. Edit the **Port Range** and click **Save**. You may have to create a new Rule by clicking the **Add another rule** button.



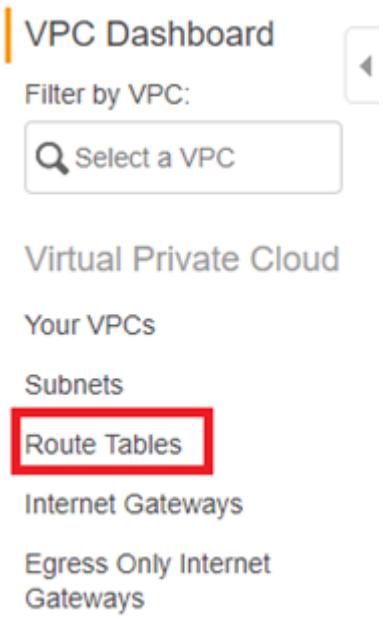
Setting Route Rules on the Staging Area Subnet

To check and set the Route Rules on the staging area subnet in AWS:

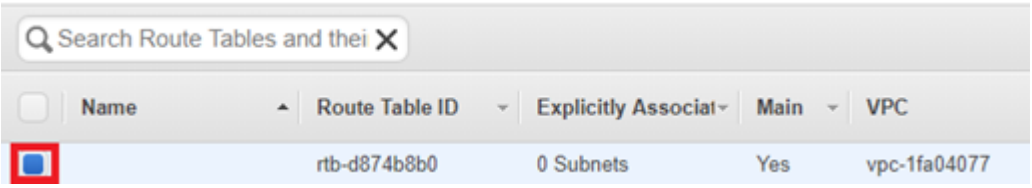
1. Sign in to AWS console, click on **Services** and select **VPC** under **Networking & Content Delivery**.



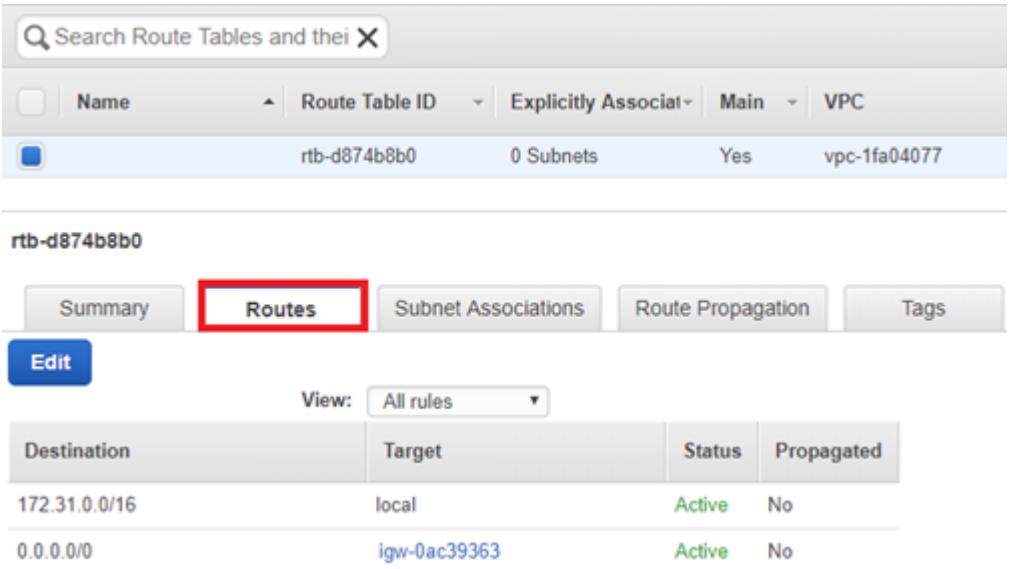
2. On the **VPC Dashboard** toolbar, select the **Route Tables** option.



3. On the **Route Tables** page, check the box of the Route Table of your staging network.

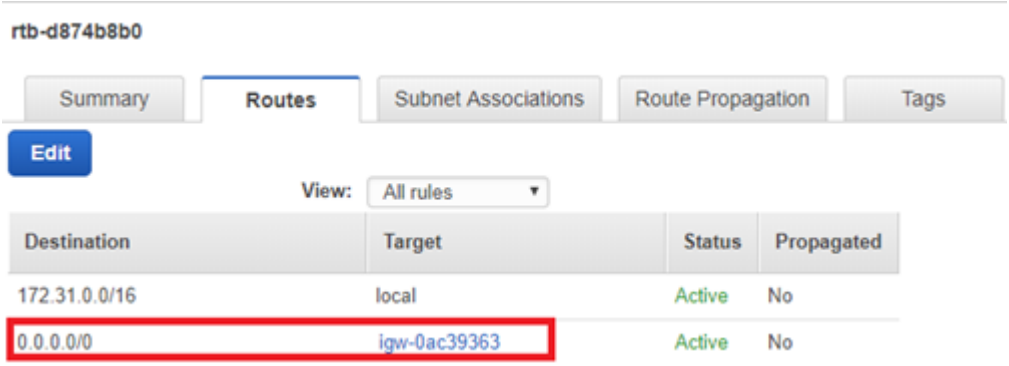


4. This will open the details for your Route Table. Navigate to the **Routes** tab.



5. Within the **Target** column of the **Routes** tab, find the route you are using for the inbound traffic from the **Source** on TCP Port 1500 (either **igw** – Internet Gateway, **vgw** – VPN, or **i** – EC2 instance). Verify that the **Destination address** is **0.0.0.0/0.**

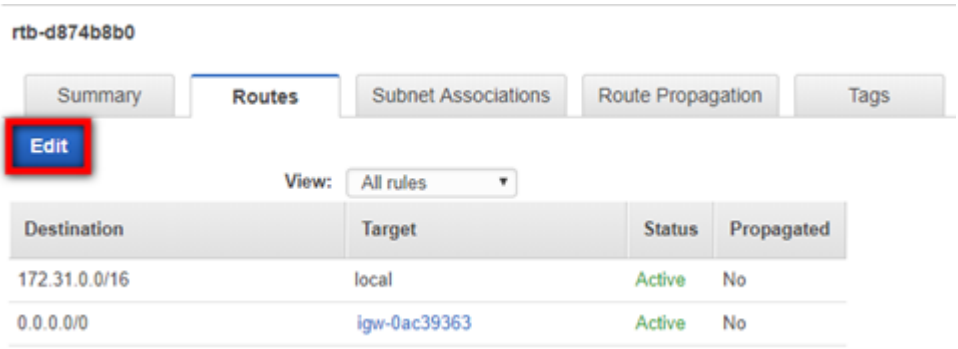
   Note: The Rule may be specific to the address space of the source machines.



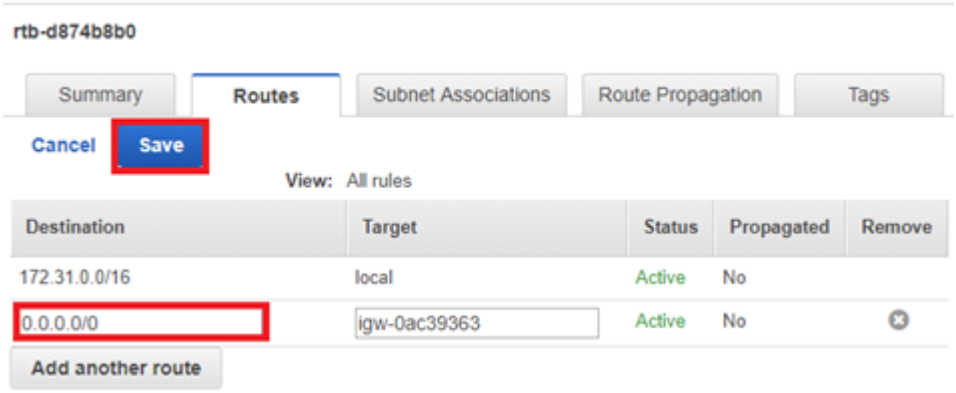
6. If the address is not 0.0.0.0/0, you will need change it to 0.0.0.0/0.

   Note: The Rule may be specific to the address space of the source machines.

   1. Click the Edit button.



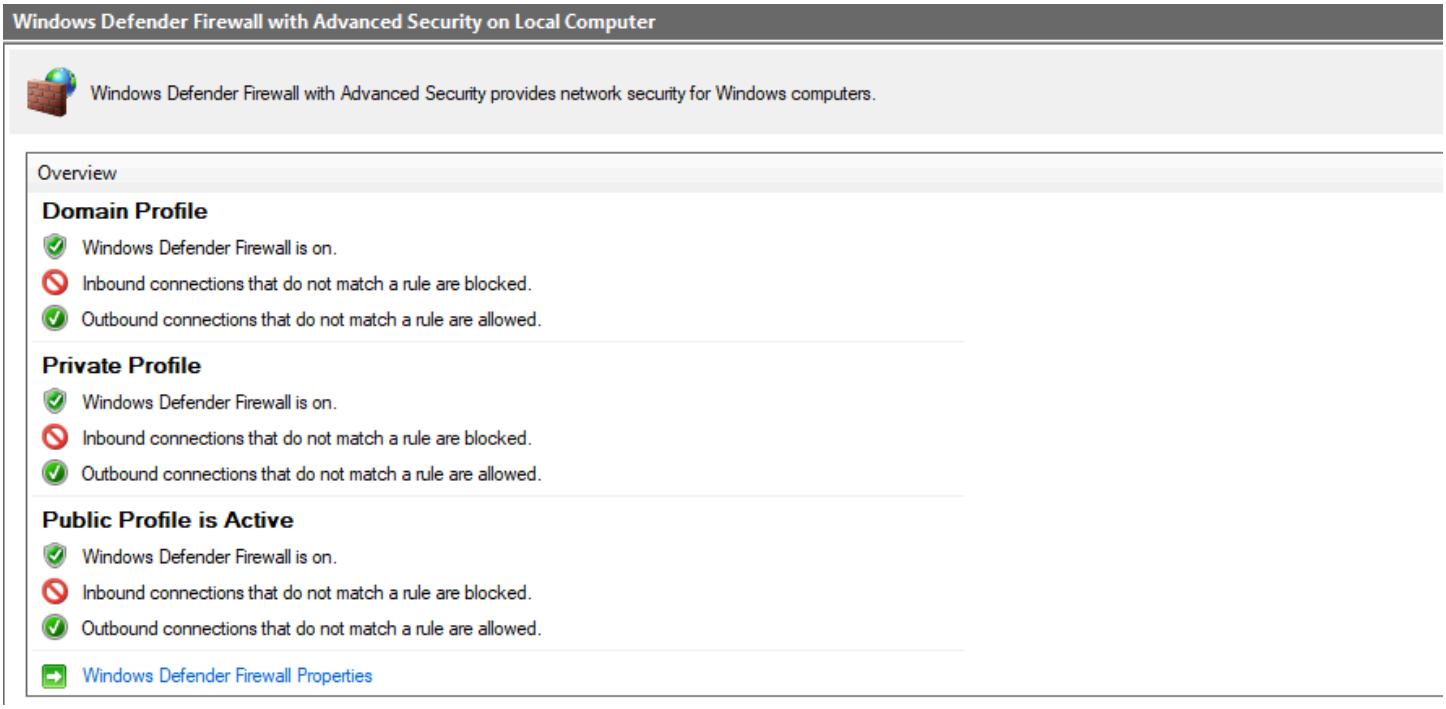
2. Input **0.0.0.0/0**into the **Destination** field for the correct Target. Click **Save**.



Note: If you are using VPN, enter a specific IP address range in the **Destination** column.

Firewall (both internal and external) in the Source server / infrastructure.

Firewall issues may have several causes. Check the following if you experience any firewall issues, such as Windows Firewall connection issues:



- Ensure that the subnet you assigned for the Replication Servers still exists.

In Linux, run `sudo systemctl stop firewalld` on the Recovery Instance to troubleshoot firewall issues.

Troubleshooting Agent Issues

Topics

Error: Installation Failed

When the installation of the AWS Replication Agent on a source server fails during the running of the Installer file, you will receive an error message.

This type of error means that the Agent was not installed on the source server, and therefore the server will not appear on the AWS Elastic Disaster Recovery Console. After you fix the issue that caused the installation to fail, you need to rerun the Agent Installer file to install the Agent.

This app cant run on your PC error – Windows

If you encounter the following error "This app can't run on your PC", when trying to install the AWS Replication Agent on your Windows 10 source machine, try the following.

This error is indicative that your particular version of Windows 10 is likely the 32-bit version. To verify this, you can

1. Use the Windows key + I keyboard shortcut to open the Settings app.

2. Click System.

3. Click About.

4. Under System type, you will see two pieces of information: if it says 32-bit operating system, x64-based processor, then it means that your PC is running a 32-bit version of Windows 10 on a 64-bit processor.

If it says 32-bit operating system, x86-based processor, then your computer doesn't support Windows 10 (64-bit).

At the moment, only 64 bit operating systems are supported for Elastic Disaster Recovery Service.

If your OS is indeed 64-bit, then there may be other elements blocking the installation of your agent. The block is actually coming from the Windows Operating System itself. You would need to identify what the cause is, (for example, broken registry key),

Is having a mounted '/tmp' directory a requirement for the Agent?

The simple requirement is just to have enough free space. There is no need for this to be a separate mount. The need for the '/tmp' requirement is actually only if '/tmp' is a separate mount. If '/tmp' is not a separate mount, then it would fall under '/', for which we have the 2 GiB free requirement. This allows for the '/tmp' to fall into this requirement.

Installation Failed – Old Agent

Installation may fail due to an old AWS Replication Agent. Ensure that you are attempting to install the latest version of the AWS Replication Agent. You can learn how to download the Agent here.

Installation Failed on Linux Machine

If the installation failed on a Linux source server, check the following:

1. **Free Disk Space**

   Free disk space on the root directory – verify that you have at least 3 GB of free disk on the root directory (/) of your Source machine. To check the available disk space on the root directory, run the following command: df -h /

   Free disk space on the /tmp directory – for the duration of the installation process only, verify that you have at least 500 MB of free disk on the /tmp directory. To check the available disk space on the /tmp directory run the following command: df -h /tmp

   After you have entered the above commands for checking the available disk space, the results will be displayed as follows:

```
ubuntu@Linux-1:~$ df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.4G  6.0G  19% /
ubuntu@Linux-1:~$ df -h /tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.4G  6.0G  19% /tmp
```

2. **The format of the list of disks to replicate**

   During the installation, when you are asked to enter the disks you want to replicate, do NOT use apostrophes, brackets, or disk paths that do not exit. Type only existing disk paths, and separate them with a comma, as follows:

   /dev/xvdal,/dev/xvda2.

3. **Version of the Kernel headers package**

   Verify that you have kernel-devel/linux-headers installed that are exactly of the same version as the kernel you are running.

   The version number of the kernel headers should be completely identical to the version number of the kernel. To handle this issue, follow these steps:

   a. **Identify the version of your running kernel.**

      To identify the version of your running kernel, run the following command:

      uname -r

```
[root@ip-172-31-1-164 ~]# uname -r
4.4.41-36.55.amzn1.x86_64
[root@ip-172-31-1-164 ~]#
```

The 'uname -r' output version should match the version of one of the installed kernel headers packages (kernel-devel-<version number> / linux-headers-<version number>).

b. **Identify the version of your kernel-devel/linux-headers.**

To identify the version of your running kernel, run the following command:

On RHEL/CENTOS/Oracle/SUSE:

rpm -qa | grep kernel

```
[root@ip-172-31-1-164 ~]# rpm -qa | grep kernel
kernel-tools-4.4.41-36.55.amzn1.x86_64
kernel-4.4.41-36.55.amzn1.x86_64
kernel-headers-4.4.41-36.55.amzn1.x86_64
[root@ip-172-31-1-164 ~]#
```

Note: This command looks for kernel-devel.

On Debian/Ubuntu: apt-cache search linux-headers

```
ubuntu@Linux-1:~$ apt-cache search linux-headers
linux-headers-3.13.0-24 - Header files related to Linux kernel version
 3.13.0
linux-headers-3.13.0-24-generic - Linux kernel headers for version 3.1
3.0 on 64 bit x86 SMP
linux-headers-3.13.0-24-lowlatency - Linux kernel headers for version
3.13.0 on 64 bit x86 SMP
```

c. **Verifying that the folder that contains the kernel-devel/linux-headers is not a symbolic link.**

Sometimes, the content of the kernel-devel/linux-headers, which match the version of the kernel, is actually a symbolic link. In this case, you will need to remove the link before installing the required package.

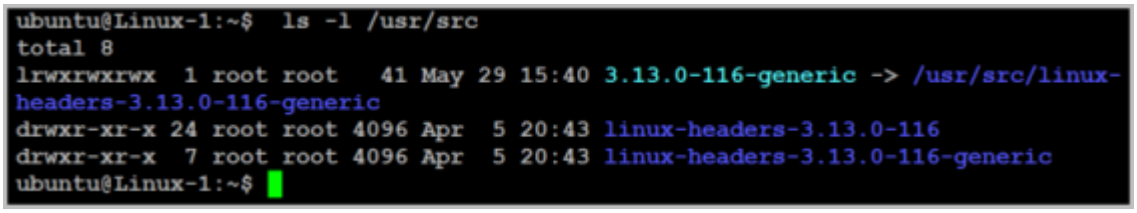
To verify that the folder that contains the kernel-devel/linux-headers is not a symbolic link, run the following command:

On RHEL/CENTOS/Oracle/SUSE:

ls -l /usr/src/kernels

On Debian/Ubuntu:

ls -l /usr/src



In the above example, the results show that the linux-headers are not a symbolic link.

d. **[If a symbolic link exists] Delete the symbolic link.**

If you found that the content of the kernel-devel/linux-headers, which match the version of the kernel, is actually a symbolic link, you need to delete the link. Run the following command:

rm /usr/src/<LINK NAME>

For example: rm /usr/src/linux-headers-4.4.1

e. **Install the correct kernel-devel/linux-headers from the repositories.**

If none of the already installed kernel-devel/linux-headers packages match your running kernel version, you need to install the matching package.

Note: You can have several kernel headers versions simultaneously on your OS, and you can therefore safely install new kernel headers packages in addition to your existing ones (without uninstalling the other versions of the package.) A new kernel headers package does not impact the kernel, and does not overwrite older versions of the kernel headers.

Note: For everything to work, you need to install a kernel headers package with the exact same version number of the running kernel.

To install the correct kernel-devel/linux-headers, run the following command:

On RHEL/CENTOS/Oracle/SUSE:

sudo yum install kernel-devel-`uname -r`

On Debian/Ubuntu:

sudo apt-get install linux-headers-`uname -r`

f. **[If no matching package was found] Download the matching kernel-devel/linux-headers package.**

If no matching package was found on the repositories configured on your machine, you can download it manually from the Internet and then install it.

To download the matching kernel-devel/linux-headers package, navigate to the following sites:

- RHEL, CENTOS, Oracle, and SUSE package directory
- Debian package directory
- Ubuntu package directory

4. **The make, openssl, wget, curl, gcc and build-essential packages**

Note: Usually, the existence of these packages is not required for Agent installation. However, in some cases where the installation fails, installing these packages will solve the problem.

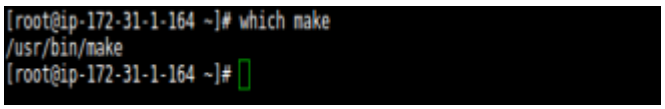
If the installation failed, the make, openssl, wget, curl, gcc, and build-essential packages should be installed and stored in your current path.

To verify the existence and location of the required packages, run the following command:

which <package>

For example, to locate the make package:

which make



5. **Error: urlopen error [Errno 110] Connection times out**

This error occurs when outbound traffic is not allowed over TCP Port 443. Port 443 needs to be open outbound to the AWS Elastic Disaster Recovery Manager.

```
root@ubuntu:~# python installer_linux.py
CloudEndure Installer Downloader started!
Downloading and running installer for a 64 bit system...
Error downloading installer! Please contact support@cloudendure.com
Error details:  <urlopen error [Errno 110] Connection timed out>
root@ubuntu:~#
```

6. **Powerpath support**

   powermt check

   ```
   [root@localhost ~]# multipath -l
   mpathh (3600c0ff0001187d84537fe5101000000) dm-0 HP,P2000 G3 FC
   size=93G features='1 queue_if_no_path' hwhandler='0' wp=rw
   `-+- policy='round-robin 0' prio=0 status=active
     |- 2:0:0:1   sda 8:160 active undef running
     `- 2:0:1:1   sdb 8:176 active undef running
   ```

   If so, contact AWS Support for instructions on how to install the AWS Replication Agent on such machines.

7. **Error: You need to have root privileges to run this script**

   ```
   adam@adam ~> python installer_linux.py
   CloudEndure Installer Downloader started!
   Downloading and running installer for a 64 bit system...
   You need to have root privileges to run this script.
   ```

   Make sure you run the installer either as root or by adding sudo at the beginning:

   sudo python installer_linux.py

Installation Failed on Windows Machine

If the installation failed on a Windows Source server, check the following:

1. **.NET Framework**

   Verify that .NET Framework version 3.5 or above is installed on your Windows Source servers.

2. **Free disk space**

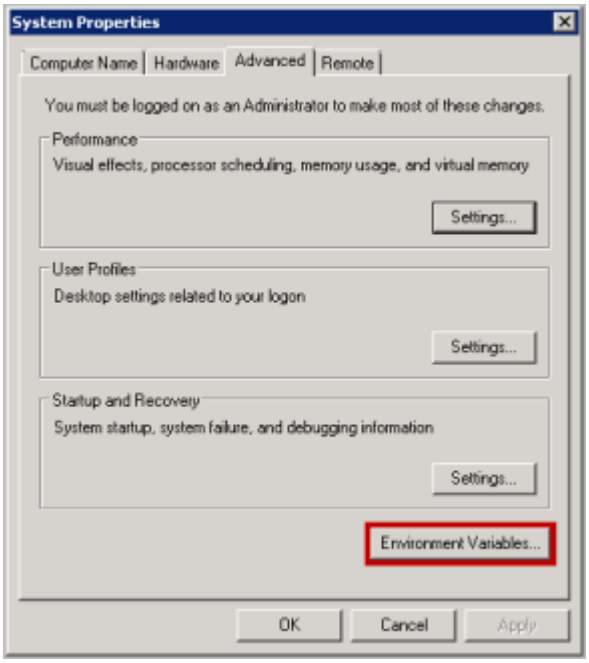
   Verify that there is at least 1 GB of free disk space on the root directory (C:\) of your Source servers for the installation.

3. **net.exe and sc.exe location**

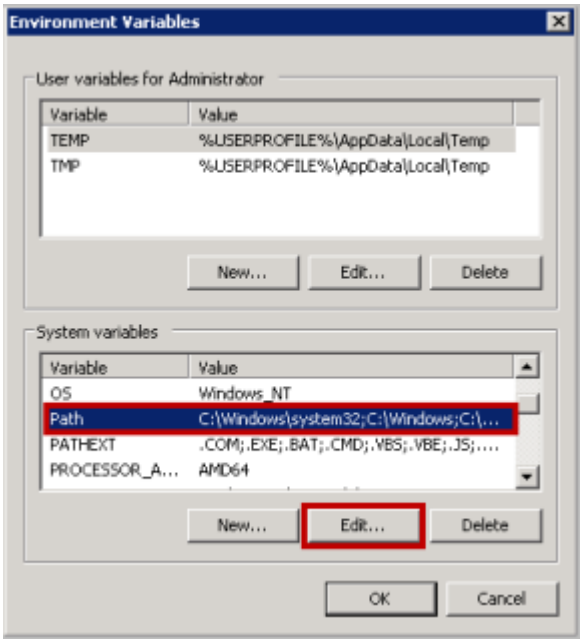
   Verify that the net.exe and/or sc.exe files, located by default in the C:\Windows\System32 folder, are included in the **PATH Environment Variable**.

   a. Navigate to **Control Panel >System and Security >System >Advanced system settings**.

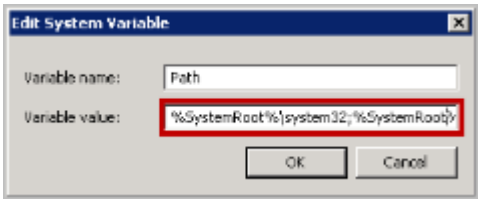
   b. On the **System Properties** dialog box **Advanced** tab, click the **Environment Variables** button.



   c. On the **System Variables** section of the **Environment Variables** pane, select the **Path** variable. Then, click the **Edit** button to view its contents.



d. On the **Edit System Variable** pane, review the defined paths in the **Variable value** field. If the path of the net.exe and/or sc.exe files does not appear there, manually add it to the **Variable value** field, and click **OK**.



Windows – Installation Failed - Request Signature

If the AWS Replication Agent installation fails on Windows with the following error:

```
botocore.exceptions.ClientError: An error occurred (InvalidSignatureException) when
                    calling the GetAgentInstallationAssetsElastic Disaster
 RecoveryInternal operation: {"message":"The
                    request signature we calculated does not match the signature you
 provided. Check your AWS Secret
                    Access Key and signing method. Consult the service documentation
 for details.
```

Attempt to rerun the installer with power shell instead of CMD. At times, when the installer is ran in CMD, the AWS Secret Key does not get pasted properly into the installer and causes installation to fail.

Error – driver was compiled for a different kernel not loading

This error may manifest if a significant amount of time has passed between when you performed a failover and when you are performing a failback.

This error may occur on the source server or on the recovery instance. You can identify this error by looking at the agent log in /var/lib/aws-replication-agent/agent.log.0

To fix this issue on a recovery instance, reboot the recovery instance and reinstall the AWS Replication Agent as recovery instance.

To fix this issue on a source server, reboot the source server and then reinstall the AWS Replication Agent.

Error – certificate verify failed

This error (CERTIFICATE_VERIFY_FAILED) may indicate that the OS does not trust the certification authority used by our endpoints. To resolve this issue, try the following steps:

1. Open Microsoft Edge or Internet Explorer to update the operating system trusted root certificates. This will work if the operating system does not have restrictions to download the certificates.
2. If the first step does not resolve the issue, download and install the Amazon Root Certificates manually.

Common replication errors

This section describes common replication errors and possible explanations and potential mitigations.

Replication errors

- Agent not seen
- Not converging
- Failback client not seen
- Snapshot failure

- Unstable network
- Failed to download replication software to failback client
- Failed to configure replication software
- Failed to establish communication with recovery instance
- Failed to connect AWS replication Agent to replication software
- Failed to establish communication with replication software
- Failed to create firewall rules
- Failed to authenticate with service
- Failed to create staging disks
- Failed to pair the replication agent with replication server
- Unknown data replication error

Agent not seen

- If this message appears on the source server dashboard, ensure that:
  - The source machine has access to the AWS Elastic Disaster Recovery service.
  - The replication agent is in running state. For Windows, use Windows services management console (services.msc) or command line (for example, get-services PowerShell). For Linux, use the systemctl status command.

  If the agent is indeed in running state, verify that the connectivity to the Regional AWS DRS endpoint on TCP Port 443. Learn more about verifying connectivity to AWS DRS regional endpoints.
- If this message appears on your recovery dashboard, ensure that:
  - You have connectivity, as previously discussed.
  - The required EC2 profile is associated with the recovery instance.

Not converging

This error message (NOT_CONVERGING) could indicate an inadequate replication speed.

- Follow the instructions on calculating the required bandwidth.
- Verify network bandwidth.

- Verify replicator EBS volumes (associated with the source server) performance. If required, modify EBS volume type from the AWS DRS console: Go to the specific source server page and select the **Disk settings** tab.

Failback client not seen

This error message (FAILBACK_CLIENT_NOT_SEEN) could indicate that there's a network connectivity issue and that the Failback Client is unable to communicate with the AWS DRS endpoint. Check network connectivity.

Snapshot failure

This error message (SNAPSHOTS_FAILURE) indicates that the service is unable to take a consistent snapshot.

This can be caused by:

- Inadequate IAM permissions – Ensure that you have the required IAM permissions (attached to the required IAM roles).
- API throttling – Check if you have activated throttling. If throttling is not activated, check your CloudTrail logs for throttling errors.

Unstable network

This error message (UNSTABLE_NETWORK) may indicate that there are network issues. Check your connectivity, then run the network bandwidth test.

Failed to download replication software to failback client

This error message (FAILED_TO_DOWNLOAD_REPLICATION_SOFTWARE_TO_FAILBACK_CLIENT) may indicate that there are connectivity issues. Check your connectivity to the S3 endpoint and try again.

If the issue persists, you might have a proxy or a network security appliance filtering your traffic and blocking the software download.

Failed to configure replication software

This error message (FAILED_TO_CONFIGURE_REPLICATION_SOFTWARE) may appear for multiple reasons. Try again and if the issue persists, contact AWS support.

Failed to establish communication with recovery instance

This message (FAILED_TO_ESTABLISH_RECOVERY_INSTANCE_COMMUNICATION) could indicate communication issues. Ensure that the Failback Client is able to communicate with the recovery instance.

If you are utilizing public network, (no VPN, no direct connect, and more), ensure that your recovery instance has a public IP. By default, AWS DRS launch template deactivates public IP, and recovery instances are only launched with private IPs.

Failed to connect AWS replication Agent to replication software

This error message (FAILED_TO_PAIR_AGENT_WITH_REPLICATION_SOFTWARE) may indicate a pairing issue. AWS DRS needs to provide the replication server and agent with information to allow them to communicate. Make sure there is network connectivity between the agent, replication server, and the AWS DRS endpoint.

If the issue persists, contact support.

Failed to establish communication with replication software

This error message (FAILED_TO_ESTABLISH_AGENT_REPLICATOR_SOFTWARE_COMMUNICATION) may suggest that there are network connectivity issues. Make sure you have network connectivity between the agent, replication server and the AWS DRS endpoint.

If this message appears during failback, ensure that TCP port 1500 is opened inbound on the recovery instance.

Failed to create firewall rules

This error message (Firewall rules creation failed) can be caused by several reasons.

1. Ensure that the IAM permission prerequisites are met.
2. Review the replication settings of the associated source server.

Failed to authenticate with service

This error message (Failed to authenticate the replication server with the service) may indicate a communication issue between the replication server and the DRS endpoint on TCP Port 443. Check the subnet you selected and ensure that TCP Port 443 is open from your replication server.

To verify the connection:

- Launch a test Ubuntu machine in the same subnet that was selected in the replication settings.
- On the machine, run the following command:

```
wget <enter_DRS_regional_endpoint>
```

- If the command fails, there is a connectivity problem.

Failed to create staging disks

This error message (Failed to create staging disks) may indicate that your AWS account is configured to encrypted EBS disks but the IAM user does not have the required permissions to encrypt using the selected KMS key. Ensure that the IAM prerequisites are met.

Failed to pair the replication agent with replication server

This error message (Failed to pair replication agent with replication server) may be caused by multiple reasons. Make sure that you have connectivity between the replication agent, the replication server, and the DRS endpoint. If the issue persists, contact Support.

Unknown data replication error

Unknown errors (unknown_error) can occur for any number of reasons. There are several steps you can take to attempt to mitigate the issue:

- Check connectivity.
- Check throttling.
- Check performance issue on the replication server.
- Check the network bandwidth between the agent and the replication server.
- Check the replication agent logs.

Other toubleshooting topics

Topics

Windows License activation – AWS

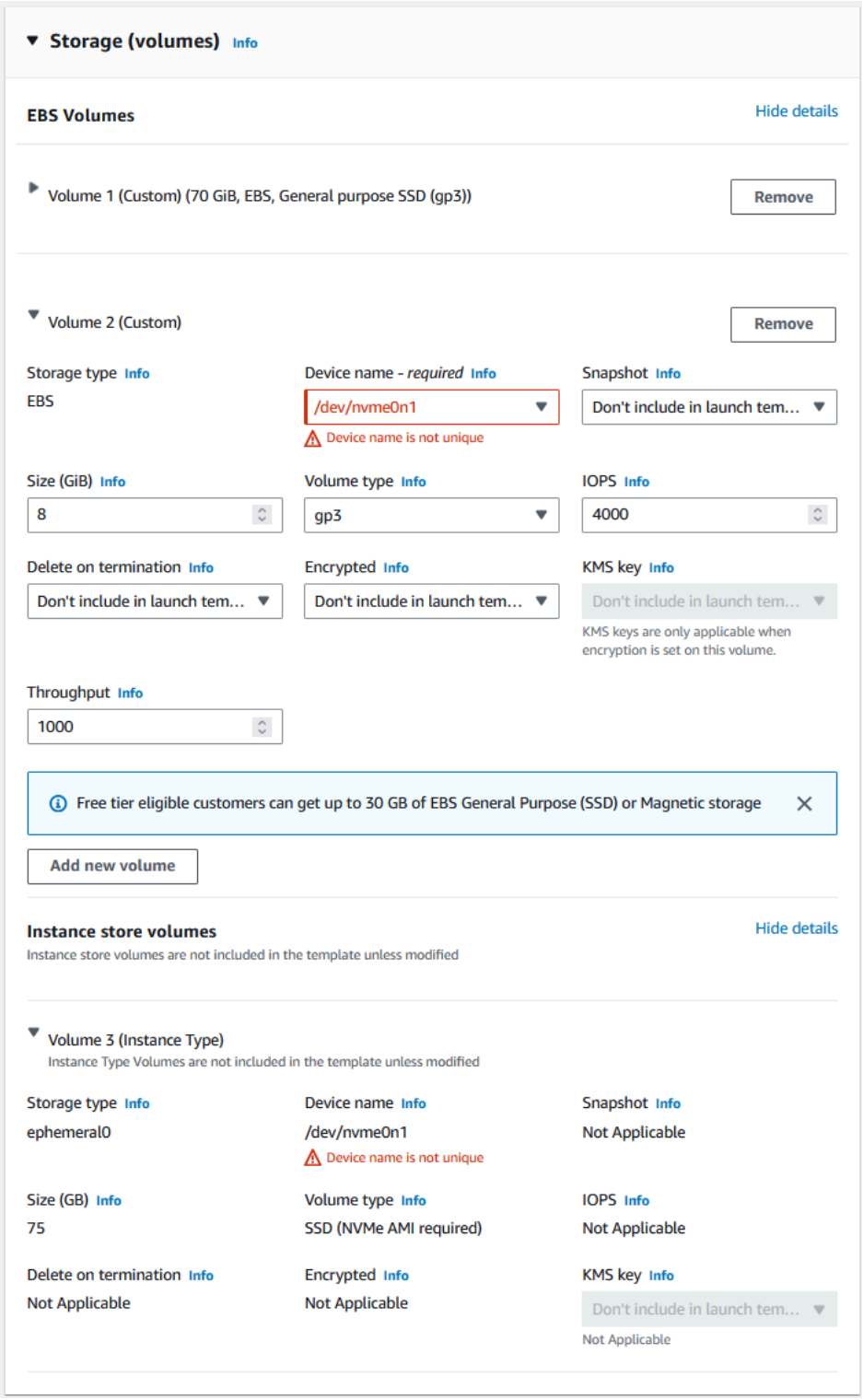
AWS Elastic Disaster Recovery converts the Windows OS licenses to AWS Windows licenses and activates them against the AWS KMS.

If license activation failed, follow this AWS guide to resolve the issue.

> **Important**
>
> When performing a failback, AWS DRS does not have access to the Customer licenses and therefore cannot activate the licenses. After failback is complete, you can activate the licenses manually or using post-launch scripts.

Replicating Instance Store Volumes

When installing the DRS agent on an EC2 Instance with Instance Store volumes attached, device name conflicts can arise in the Recovery Instance's EC2 Launch Template if the template also specifies Instance Store volumes.



You can resolve this error in one of two ways:

- If you require protection of the data on the Source Server's Instance Store Volume, ensure the Recovery Instance's EC2 Launch Template is reconfigured to provide a unique Device Name that

will not collide with the default Instance Store mappings. For example, the "Device Name" for the EBS volume can be changed to /dev/xvdc1.

- If you do not require protection of the data on the Source Server's Instance Store volume, ensure instance store volumes are excluded from replication via the --devices installation parameter. The DRS agent will not populate any volumes excluded from replication in the EC2 Launch Template.

Replication lag issues

Potential solutions:

- Make sure that the source server is up and running.
- Make sure that AWS Elastic Disaster Recovery services are up and running.
- Make sure that TCP Port 1500 is not blocked outbound from the Source server to the replication server.
- If the MAC address of the Source had changed, that would require a reinstallation of the AWS Replication Agent.
- If the source machine was rebooted recently or the AWS Elastic Disaster Recovery services were restarted, the disks are reread after this and until it's finished, the lag will grow.
- If the source machine had a spike of write operations, the lag will grow until AWS Elastic Disaster Recovery service manages to flush all the written data to the drill or recovery instance replication server.

Turning driver signing off in Windows 2003

When installing on Windows 2003, ensure that **Driver Signing** is **Off**.

1. Right-click **My Computer**.

2. Select **Properties** to open **System Properties**.

3. In the **System Properties** dialog box, select the **Hardware** tab.

4. Click on the **Driver Signing** button.

5. Select **Ignore – Install the software anyway and don't ask for my approval**.

6. Select **Make this action the system default**.

Windows Drive changes

Users may see changes in Windows drive letter assignments (for example, Drive D changed to E) on Target machines launched by AWS Elastic Disaster Recovery.

This happens because Windows sometimes reconfigures the drive letters when a machine comes up on a new infrastructure, for example, if the source server had a drive letter mapped to a disk that was not replicated (such as a network drive). You can solve this issue by remapping the drive letters on the drill or recovery instance correctly after it has been launched.

Error: Failed to connect using HTTP channel

This error mostly occurs when the Conversion server is unable to communicate with the necessary AWS Endpoints for staging area communication.

- Check if any network changes were made in the staging area that could affect the Conversion server reaching the AWS Endpoints (Firewall settings, DNS settings, Security Group settings, Route table settings, and Access Control List settings).
- Test TCP Port 443 connectivity with a test instance from the staging area subnet, to the required endpoints.
- If the issue persists after confirming network connectivity please create a case with AWS Premium Support for further investigation.

Windows Dynamic Disk troubleshooting

Moving a Windows Dynamic Disk from a local computer to another computer may change the disk status to "Foreign", resulting in a disruption in replication. The solution is to import the foreign disk, as discussed in this Microsoft troubleshooting article.

FAQ

Topics

Elastic Disaster Recovery Concepts

Topics

What is the Recovery Time Objective (RTO) of Elastic Disaster Recovery?

The Recovery Time Objective (RTO) of Elastic Disaster Recovery is typically measured in minutes. The RTO is highly dependent on the OS boot time.

What is the Recovery Point Objective (RPO) of Elastic Disaster Recovery?

The Recovery Point Objective (RPO) of Elastic Disaster Recovery is typically in the sub-second range.

General questions

Topics

- Can AWS Elastic Disaster Recovery protect physical servers?
- What data is stored on and transmitted through AWS Elastic Disaster Recovery servers?
- What is the Recovery Time Objective (RTO) of AWS Elastic Disaster Recovery?
- What is the Recovery Point Objective (RPO) of AWS Elastic Disaster Recovery?
- What to consider when replicating Active Directory
- Does AWS Elastic Disaster Recovery work with LVM and RAID configurations?
- What is there to note regarding SAN/NAS Support?
- Does AWS Elastic Disaster Recovery support Windows License Migration?
- Can you perform an OS (Operating System) upgrade with AWS Elastic Disaster Recovery?
- What are the private APIs used by AWS DRS to define actions in the IAM Policy?
- What post-launch scripts does AWS Elastic Disaster Recovery support?
- Is BitLocker encryption supported?
- Can I set instance metadata on my launched instance to support IMDSv2 only?
- Upgrading from CEDR to AWS DRS - Manual instructions

What source infrastructure does AWS Elastic Disaster Recovery support?

With AWS Elastic Disaster Recovery, you can recover your applications on AWS from any source infrastructure on you can install the AWS Replication Agent, and on which you can run the DRS Failback Client. This includes physical infrastructure, virtual machines on hypervisors by VMware, Microsoft, and others, and cloud infrastructure from other cloud providers.

How do I upgrade from CloudEndure Disaster Recovery to AWS Elastic Disaster Recovery?

You can use the CEDR to DRS Upgrade Assessment Tool and the Server Upgrade Tool and to move your source servers from CloudEndure Disaster Recovery (CEDR) to AWS Elastic Disaster Recovery (DRS). Learn more in the CloudEndure documentation.

AWS Elastic Disaster Recovery (Elastic Disaster Recovery) is the next generation of CloudEndure Disaster Recovery (CEDR) and is the recommended service to use for Disaster Recovery to AWS. All customers are encouraged to transition from CEDR to Elastic Disaster Recovery, as soon as this is feasible for them.

Prior to upgrading, learn more about the differences between the two services, and make sure that DRS is right for you.

For manual upgrading instructions, refer to this section.

Can AWS Elastic Disaster Recovery protect physical servers?

Because AWS Elastic Disaster Recovery works at the OS layer it can protect not only virtual servers but physical ones as well.

What data is stored on and transmitted through AWS Elastic Disaster Recovery servers?

AWS Elastic Disaster Recovery store only configuration and log data on the AWS Elastic Disaster Recovery Console's encrypted database. Replicated data is always stored on the customer's own cloud VPC. The replicated data is encrypted in transit.

What is the Recovery Time Objective (RTO) of AWS Elastic Disaster Recovery?

The Recovery Time Objective (RTO) of Elastic Disaster Recovery is typically measured in minutes. The RTO is highly dependent on the OS boot time.

What is the Recovery Point Objective (RPO) of AWS Elastic Disaster Recovery?

The Recovery Point Objective (RPO) of AWS Elastic Disaster Recovery is typically in the sub-second range.

What to consider when replicating Active Directory

There are two main approaches when it comes to migrating Active Directory or domain controllers from a disaster:

1. Replicating the entire environment, including the AD server(s) - in this approach it is recommended to launch the drill or recovery AD servers first, wait until it's up and running and then launch the other drill or recovery instances, to make sure the AD servers are ready to authenticate them.

2. Leaving the AD server(s) in the Source environment - in this approach, the drill or recovery instances will communicate back to the AD server in the source environment and will take the source server's place in the AD automatically.

   In this case, it is important to conduct any drills using an isolated subnet in the AWS cloud, so to avoid having the drill or recovery instances communicate into the source AD server outside of a recovery.

Does AWS Elastic Disaster Recovery work with LVM and RAID configurations?

AWS Elastic Disaster Recovery works with any hardware RAID configuration and LVM configuration.

> **Note**
>
> Boot partitions that span or mirror, using software over multiple physical disks, are not supported by AWS Elastic Disaster Recovery and are not recommended for use in EC2. If your source server's configuration contains mirrored boot partition (e.g. Windows Mirrored Disks), we recommend installing the agent to replicate only one physical disk of the mirrored boot partition using the `--devices` parameter.
>
> - Windows EC2 RAID Documentation.
> - Linux EC2 RAID Documentation.

What is there to note regarding SAN/NAS Support?

If the disks are represented as block devices on the machine, as most SAN are, Elastic Disaster Recovery will replicate them transparently, just like actual local disks.

If the disks are mounted over the network, such as an NFS share, as most NAS implementations are, the AWS Replication Agent would need to be installed on the actual NFS server in order to replicate the disk.

Does AWS Elastic Disaster Recovery support Windows License Migration?

AWS Elastic Disaster Recovery conforms to the Microsoft Licensing on AWS guidelines.

Can you perform an OS (Operating System) upgrade with AWS Elastic Disaster Recovery?

No. AWS Elastic Disaster Recovery copies the entire machine as-is. However, you can copy the data disks exclusively and attach them to a new machine with an upgraded OS.

What are the private APIs used by AWS DRS to define actions in the IAM Policy?

AWS Elastic Disaster Recovery (AWS DRS)utilizes the following private API resources as actions in the IAM Policy. Learn more about actions, resources, and condition keys for Elastic Disaster Recovery.

- BatchCreateVolumeSnapshotGroupForDRS – Grants permission to create volume snapshot group.
- BatchDeleteSnapshotRequestForDRS – Grants permission to batch delete snapshot request.
- DescribeReplicationServerAssociationsForDRS – Grants permission to describe replication server associations.
- DescribeSnapshotRequestsForDRS – Grants permission to describe snapshots requests.
- GetAgentCommandForDRS – Grants permission to get agent command.
- GetAgentConfirmedResumeInfoForDRS – Grants permission to get agent confirmed resume info.
- GetAgentInstallationAssetsForDRS – Grants permission to get agent installation assets.
- GetAgentReplicationInfoForDRS – Grants permission to get agent replication info.
- GetAgentRuntimeConfigurationForDRS – Grants permission to get agent runtime configuration.
- GetAgentSnapshotCreditsForDRS – Grants permission to get agent snapshots credits.
- GetChannelCommandsForDRS – Grants permission to get channel commands.
- NotifyAgentAuthenticationForDRS – Grants permission to notify agent authentication.
- NotifyAgentConnectedForDRS – Grants permission to notify agent is connected.
- NotifyAgentDisconnectedForDRS – Grants permission to notify agent is disconnected
- NotifyAgentReplicationProgressForDRS – Grants permission to notify agent replication progress.
- RegisterAgentForDRS – Grants permission to register agent.
- SendAgentLogsForDRS – Grants permission to send agent logs.

- SendAgentMetricsForDRS – Grants permission to send agent metrics.
- SendChannelCommandResultForDRS – Grants permission to send channel command result.
- SendClientLogsForDRS – Grants permission to send client logs.
- SendClientMetricsForDRS – Grants permission to send client metrics.
- UpdateAgentBacklogForDRS – Grants permission to update agent backlog.
- UpdateAgentConversionInfoForDRS – Grants permission to update agent conversion info.
- UpdateAgentReplicationInfoForDRS – Grants permission to update agent replication info.
- UpdateAgentSourcePropertiesForDRS – Grants permission to update agent source properties.
- IssueAgentCertificateForDr – Grants permission to issue an agent certificate.
- CreateConvertedSnapshotForDrs – Grants permission to create converted snapshot.

What post-launch scripts does AWS Elastic Disaster Recovery support?

DRS can run scripts on a launched drill or recovery instance. This is done by creating the following folder on the source server and placing the scripts within that folder.

Linux: /boot/post_launch (any files that are marked as executable)

Windows: C:\Program Files (x86)\AWS Replication Agent\post_launch\ (any .exe, .cmd, or .bat files)

Once you put these scripts in the above folders on the source server, the folder will be replicated to the drill or recovery instance and be executed once after the instance boots for the first time.

> **Note**
>
> Post-launch scripts on Windows run under the Local System context. Post-launch scripts on Linux run under the 'root' user.

Is BitLocker encryption supported?

DRS does not support OS-based disk encryption features such as BitLocker. These should be deactivated before using AWS Elastic Disaster Recovery.

Can I set instance metadata on my launched instance to support IMDSv2 only?

You can easily set Instance Metadata Service Version 2 (IMDSv2) on your recovery instances using the EC2 launch template associated with your DRS source server.

Follow the instructions on the EC2 launch template page.

When you are redirected to the EC2 console to modify your template, take the following steps

- Click **Advanced details > Metadata version**.
- Select **V2 only (token required)**.

You can then set this launch template as your default version.

Upgrading from CEDR to AWS DRS - Manual instructions

> **Important**
>
> You can now use the CEDR to DRS Upgrade Assessment Tool and the Server Upgrade Tool and to move your source servers from CloudEndure Disaster Recovery (CEDR) to AWS Elastic Disaster Recovery (AWS DRS). Learn more in the CloudEndure documentation.

AWS Elastic Disaster Recovery (AWS DRS) is the next generation of CloudEndure Disaster Recovery (CEDR) and is the recommended service to use for Disaster Recovery to AWS. All customers are encouraged to transition from CEDR to AWS DRS, as soon as this is feasible for them.

Prior to upgrading, learn more about the differences between the two services, and make sure that DRS is right for you.

The following are the manual instructions for upgrading:

1. Follow the DRS getting started procedure to initialize AWS DRS in the AWS Region you want to replicate to.
2. Launch a recovery instance (target machine) using CloudEndure, and make sure that it works as expected. Once you have verified that everything works as expected, terminate the launched instance using the CloudEndure Console by choosing the "Delete Target Machines" option. If

you want to keep the instance, activate EC2 termination protection before removing the source machine from the CloudEndure service.

Until the server is ready on DRS, CloudEndure will still be your way to launch Recovery instances should you need them. That is why you must make sure that recovery using CloudEndure is working as expected for the server/s you are about to transition to DRS.

3. Pause data replicationfor this server in CloudEndure.
4. Manually uninstall the CloudEndure agent from your source servers.

> **Important**
>
> Do ***not*** do use the **Remove from console** option available from the CloudEndure user console. By keeping this server's records in CloudEndure, you also maintain it's Point In Time recovery points, allowing you to launch a recovery instance using CloudEndure, should you need such a recovery instance before this server is ready on Elastic Disaster Recovery.

5. Install the AWS Replication Agent on your source server.
6. Configure Replication settings and Launch settings for this server in AWS Elastic Disaster Recovery (AWS DRS).
7. Wait for initial sync to be complete until your source server's data replication status has reached the **Healthy** state in the AWS DRS console.
8. Use DRS tolaunch a drill instance for your source server and make sure it works as desired.
9. Wait for the number of recovery days you want to have Points In Time for to pass. For example, if you have CloudEndure and AWS DRS configured to retain 10 daily recovery Points In Time, then wait for 10 full days after the server has achieved the **Healthy** state in AWS DRS before removing it from CloudEndure.

> **Important**
>
> Remove your source servers from the CloudEndure console.
> This action will cause all replication resources created for this server in AWS to be terminated. Until you do this, these resources continue to cost you money.
> If you have a launched a target instance in AWS using CEDR, consider whether you want to keep it or not.

If you experience a DR event during or before the server reaches the **Healthy** state in AWS DRS, navigate to the CloudEndure console and launch a Target instance from there. This will launch the Target instance from the last PIT the system created before you removed the CloudEndure agent from the source servers. The CloudEndure console UI will show you the PIT from when this will launch.

> **Note**
>
> During some of the time it takes to transition from CloudEndure to DRS you will not have the same level of protection: While replication in CloudEndure is paused and the server has not yet completed the initial scan, you will not be able to launch instances in DRS, and only be able to launch instances in CloudEndure with data prior to the pause action. This applies both to launching from latest snapshot and to launching from point-in-time.

> **Note**
>
> Once you install the AWS Replication Agent on the source server, and until you remove that source server from the CloudEndure user console, you will be paying for the two services in parallel, and nearly twice for replication resources such as EBS, snapshots, and more.

Agent related

Topics

What does the AWS Replication Agent do?

The AWS Replication Agent performs an initial block-level read of the content of any volume attached to the server and replicates it to the replication server. The Agent then acts as an OS-level read filter to capture writes and synchronizes any block level modifications to the Elastic Disaster Recovery replication server, ensuring near-zero RPO.

What kind of data is transferred between the Agent and the AWS Elastic Disaster Recovery Service Manager?

The AWS Replication Agent sends the following types of information to the AWS Elastic Disaster Recovery Service Manager:

- Monitoring metrics of the Agent itself
- Replication status (started, stalled, resumed)
- Backlog information
- OS and hardware information.

When an Agent is installed on a source server, it collects the following information on the machine:

- Host name and ID.
- List of CPUs including models and number of cores
- Amount of RAM
- Hardware and OS information.
- Number of disks and their size – in Windows, disk letters; in Linux, block device names.
- Installed applications (Windows)
- Installed Packages (Linux)
- Running services.
- Machine's Private IP address.

Can a proxy server be used between the source server and the Elastic Disaster Recovery Console?

Yes. You can configure the proxy either by using an environment variable prior to the installation (Linux and Windows), or by using the `--proxy-address` flag in the Linux installer.

Using the installer: `./aws-replication-installer-init --proxy-address https://PROXY:PORT/`

Using environment variable: `export https_proxy=https://PROXY:PORT/; ./aws-replication-installer-init`

Make sure the proxy has a trailing forward slash (/).

What are the pre-requisites needed to install the AWS Replication Agent?

The installation requirements for source server depend on the type of OS that the server runs – either Linux or Windows.

View the prerequisites.

What ports does the AWS Replication Agent utilize?

The Agent utilizes TCP Port 443 to communicate to the Elastic Disaster Recovery Service Manager and TCP Port 1500 for replication to AWS.

What kind of resources does the AWS Replication Agent utilize?

The AWS Replication Agent is lightweight and nondisruptive. The agent utilizes approximately 5% CPU and 300 MB of RAM.

Can Elastic Disaster Recovery migrate containers?

Elastic Disaster Recovery only supports the replication of full servers. Nevertheless, Elastic Disaster Recovery replicates on a server level and therefore any containers within the selected servers will be replicated.

Does the AWS Replication Agent cache any data to disk?

Elastic Disaster Recovery does not write any cache or do any sort of journalling to disk. The Agent holds a buffer which is large enough to map all volume's blocks ~250 MB in memory.

The Agent then acts as a sort of write filter and will replicate changed blocks directly from memory to the replication server. In cases where the data no longer in memory, the Agent will read the block from the volume directly. This is the case where you may see backlog in the Elastic Disaster Recovery Console. The cause of this is the volume of change is greater than the bandwidth available.

How is communication between the AWS Replication Agent and the Elastic Disaster Recovery Service Manager secured?

All communication is encrypted using SSL. In addition, each Agent is assigned a key during installation which is used to encrypt all traffic. All keys are unique and are not shared across multiple Agents.

Is it possible to change the port the AWS Replication Agent utilizes from TCP Port 1500 to a different port?

No. The Elastic Disaster Recovery Agent can only utilize TCP Port 1500 for replication.

How do I manually uninstall the Elastic Disaster Recovery Agent from a server?

Please refer to:the section called “Uninstalling the agent”.

When do I need to reinstall the Agent?

Agent re-installations are required in these cases:

- After adding new volumes if the Automatic replication of new disks option is not activated for the source server where the volume is added.
- Windows OS upgrades (ex. Windows Server 2012 to Windows Server 2016)
- Some new features require a re-installation to apply. In this case, the feature documentation will specifically state that this is a requirement for the feature to be activated.

How much bandwidth does the AWS Replication Agent consume?

The AWS Replication Agent opens up to five connections and will attempt to maximize available bandwidth.

Throttling can be activated by selecting the specific server and selecting the **Settings** page in the Elastic Disaster Recovery Console.

How many disks can the AWS Replication Agent replicate?

The Agent can replicate up to 50 disks from a single server.

Is it possible to add a disk to replication without a complete resync of any disks that have already been replicated??

When you add a disk to a source server, AWS Elastic Disaster Recovery will automatically identify it and add it to the **Disk settings** tab in the console.

This feature is activated automatically for newly added servers. Learn how to deactivate or reactivated this feature.

Which Windows and Linux OSs support no-rescan upon reboot?

A shutdown (from the OS menu or CLI) of any supported Linux or Windows source server no longer causes a rescan in DRS once the source server is restarted. A rescan means that the agent on the source server rereads all blocks on all replicated disks and transmits blocks that are different from the previously replicated data. A rescan is similar to the initial sync but is faster because only blocks that are different need to be transmitted.

Rescans can still happen following a hard reboot, crashes, or when you add or remove disks to or from the source server. In addition, a rescan will occur if the underline Storage types do not use static DUIDs (such as 3PARdata). Supported OSs include:

Windows Server

- 2012r1
- 2012r2
- 2016
- 2019
- 2022

Linux

- CentOS 6–8
- Oracle 6–8
- RHEL 6–9
- Rocky 8 and 9
- SLES 12 and 15
- Debian 9–11
- Ubuntu 16, 18, 20, and 22
- Amazon Linux 2

Note

For Linux, no-rescan on reboot is supported only on environments that use initramfs.

Important

A rescan duration may impact your RPO

- While a rescan is conducted, point of time recovery cannot be made.

- If a disaster occurs during the rescan, you will only be able to restore point of time from before the rescan began. This could affect your ability to meet your RPO.

How do temporary credentials work?

The temporary credential mechanism was developed specifically to provide an easy and secure way to install AWS DRS Agents. The main flow of the temporary credentials' creation process relies on generating a x509 certificate per agent and then using this x509 certificate to receive temporary IAM credentials. This process utilizes a similar mechanism to the one used by IAM Roles Anywhere.

Where can I find the AWS DRS Replication Agent logs

The AWS DRS agent logs are stored in agent.log.0:

- **Linux:** /var/lib/aws-replication-agent/agent.log.0
- **Windows 64 bit:** C:\Program Files (x86)\AWS Replication Agent\agent.log.0

In addition, you can review the installation log located in: <install_path>\aws_replication_agent_installer.log

Replication related

Topics

- Why is 0.0.0.0:1500 added to inbound rules in the Staging Area?
- How long does a rescan take?
- Is the Elastic Disaster Recovery replication crash consistent?
- How can I perform an SSL connectivity and bandwidth test?

What do Lag and Backlog mean during replication?

During replication you may see a server falls out of Continuous Data Protection (CDP) mode. This may occur for various reasons, typically related to the network throughput or interruption.

- **Lag** – The amount of time since the server was last in CDP mode.
- **Backlog** – The amount of data that was written to the disk and still needs to be replicated in order to reach CDP mode.
- **ETA** – The estimated time remaining to return to CDP.

Is the replicated data encrypted?

Elastic Disaster Recovery encrypts all the data in transit.

How is the replication server provisioned and managed in the Staging Area?

Elastic Disaster Recovery provisions the replication server(s) and automatically manages the addition and removal of the servers as necessary.

What type of replication server is utilized in the Elastic Disaster Recovery Staging Area?

AWS Elastic Disaster Recovery provisions a t3.small server by default. The typical ratio of volumes to replication servers is 15:1.

Does AWS Elastic Disaster Recovery compress data during replication?

Yes, AWS Elastic Disaster Recovery utilizes LZ4 compression during transit resulting in 60-70% compression depending on the type of data.

Are events that are generated by the AWS Elastic Disaster Recovery servers logged in Cloudtrail in AWS?

Yes, AWS Elastic Disaster Recovery is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Elastic Disaster Recovery (AWS DRS). CloudTrail captures all API calls for AWS DRS as events. The calls captured include calls from the AWS DRS console and code calls to the AWS DRS API operations. Learn more about AWS DRS and Cloudtrail.

How many snapshots does Elastic Disaster Recovery create?

Point in Time (PIT) is a disaster recovery feature which allows launching an instance from a snapshot captured at a specific Point In Time. As source servers are replicated, Point in Time states are chronicled over time, while a retention policy will determine which Points in Time are not required after a defined duration.

Elastic Disaster Recovery has the following PIT state schedule:

- Every 10 minutes for the last hour
- Once an hour for the last 24 hours
- Once a day for the last 7 days (or a different retention period, as configured)

You can increase or decrease the default 7 day snapshot retention rate from anywhere between 1 day and 365 days in the replication settings. Learn more about managing Point in Time retention.

Does Elastic Disaster Recovery delete snapshots?

AWS Elastic Disaster Recovery automatically deletes snapshots that are no longer used (such as those left over after source servers have been removed from the Elastic Disaster Recovery Console) or those that are past the designated retention setting.

How much capacity is allocated to the staging area?

A volume is created for each volume in the source infrastructure of the same size. The EBS volumes will be a 1:1 match for the source machines provisioned size.

Why is 0.0.0.0:1500 added to inbound rules in the Staging Area?

AWS Elastic Disaster Recovery uses TCP Port 1500 for replication between the source agents and the replication server. The connection is open for all IPs and can be managed by ACLs or networks controls to limit inbound IPs.

How long does a rescan take?

A rescan may occur after a reboot of the source server. The rescan time will vary depending on the size of the source disks. The time depends on the performance of the disks (linear read), staging area disk performance, and the rate of write operations on the source server (which are sent in parallel with the rescan). The rescan is functioning normally as long as its moving forward and is not "stuck".

Is the Elastic Disaster Recovery replication crash consistent?

Yes, AWS Elastic Disaster Recovery's replication is crash consistent.

How can I perform an SSL connectivity and bandwidth test?

> **Note**
>
> This tool is designed for AWS only.

You can use our SSL bandwidth tool to check for replication bandwidth availability.

1. In your target region, launch a c5.large test server using the public AMI named CE-ssl-speedtest.
2. Select the same subnet as the subnet used in the replication settings of your source machine.
3. Make sure that the security group allows TCP Port 1500 inbound access.
4. On the source machine, browse to: https://{test_server_ip}:1500/speedtest
5. Click **Start**.

> **Note**
>
> - Browse to the web page using the test server **public** or **private** IP according to what you defined in your replication settings.

- The following are the AMI details per region.
  - ami-00b38c08ab3506ea7 – US East (N. Virginia)
  - ami-0bd8423a4d80563fc – US East (Ohio)
  - ami-00b7159e9c985a8da – US West (N. California)
  - ami-033a4924b13126a7b – US West (Oregon)
  - ami-0bf60b09675c8d9b6 – Africa (Cape Town)
  - ami-0f01375b50763621b – Asia Pacific (Hong Kong)
  - ami-0b1aeb50834102c18 – Asia Pacific (Mumbai)
  - ami-0b1aeb50834102c18 – Asia Pacific (Hyderabad)
  - ami-044fa8034a31d7578 – Asia Pacific (Tokyo)
  - ami-08b042df0d4c458ea – Asia Pacific (Seoul)
  - ami-0971e46306691cd68 – Asia Pacific (Osaka)
  - ami-0afd42552b236f9dd – Asia Pacific (Singapore)
  - ami-04e7cc6b5d9e8ffa1 – Asia Pacific (Sydney)
  - ami-02f31943dfd88549d – Asia Pacific (Jakarta)
  - ami-033db317ada5abd55 – Asia Pacific (Melbourne)
  - ami-01c24408802db503d – Canada (Central)
  - ami-0b8643189a66159c9 – Europe (Stockholm)
  - ami-0dd5a09d2ae8f46b3 – Europe (Ireland)
  - ami-097fb47f3a1c2bf7e – Europe (London)
  - ami-0a3f9008725d0b4d1 – Europe (Paris)
  - ami-0c65965703bb0e541 – Europe (Milan)
  - ami-01b6fcc2337f6420d – Europe (Spain)
  - ami-07b7defb87a46bb48 – Europe (Frankfurt)
  - ami-01b3e93b3ac0e1340 – Europe (Zurich)
  - ami-016edc078b48f370b – Israel (Tel Aviv)
  - ami-0c90e298af7a2e563 – Middle East (Bahrain)
  - ami-0f7c14e62ef760768 – Middle East (UAE)
  - ami-0edd5ecfc56804583 – South America (São Paulo)

- Ensure that the security groups are configured to permit connectivity on inbound port 1500.

AWS related

Topics

What does the Elastic Disaster Recovery machine conversion server do?

The machine conversion server converts the disks to boot and run on AWS.

Specifically, it makes bootloader changes, injects hypervisor drivers, and installs cloud tools.

How do I change the server AMI on AWS after recovery?

After the machine has been launched by AWS Elastic Disaster Recovery, switching the AMI can be done by launching a vanilla machine from the required AMI, stopping that machine, detaching all the disks (including the root) and then attaching the disks from the drill or recovery instance created by Elastic Disaster Recovery.

Which AWS services are automatically installed when launching a drill or recovery instance?

AWS Elastic Disaster Recovery (AWS DRS) automatically installs EC2Config. After installation, EC2Config automatically installs the SSM EC2 Configuration Service.

CloudWatch, AWS Powershell or CLI are not automatically installed. This can be done by combining the AWS DRS APIs and the AWS APIs - you can use the AWS DRS APIs to determine the EC2 instance IDs of the machines and then use AWS API/CLI to turn on the detailed monitoring. An alternative approach would be to do it via AWS API only based on the tags you associate with the machine. A third approach would be to do so from the post-launch script.

AWS DRS installs EC2Launch (Windows 2016 only). Customers need to configure EC2Launch based on the specific requirements explained here. This configuration step needs to be performed post Recovery using the wizard in C:\Program Data\Amazon\EC2-Windows\Launch\Settings \Ec2LaunchSettings.exe on the drill or recovery instance.

How long does it take to copy a disk from the AWS Elastic Disaster Recovery staging area to production?

AWS Elastic Disaster Recovery uses internal cloud provider snapshots. This process typically takes less than a minute and the size of the volume does not impact the time.

What are the differences between conversion servers and replication servers?

Replication servers run on Linux and conversion servers (for Windows machines) run on Windows.

The conversion is done by AWS Elastic Disaster Recovery automatically bringing up a vanilla Windows conversion server machines in the same subnet with the replication servers as part of the launch job.

Both conversion and replication servers have Public IPs

The conversion servers will use the same security groups as the replication server.

The conversion servers must be able to access the AWS Elastic Disaster Recovery Service Manager.

The conversion servers machines, just like the Replication servers are managed automatically by Elastic Disaster Recovery. Any attempt to disrupt their automated functionality will result in failed conversions.

Can I prevent Elastic Disaster Recovery from cleaning up drill instance resources in AWS?

AWS Elastic Disaster Recovery will, by default, remove any resources created during the drill process either when requested by the user or when a new drill instance is launched.

To prevent this in AWS, you can Activate Termination Protection for the drill or recovery instance, and the resources will not be removed upon a new instance launch.

Why are my Windows Server disks read-only after launching the drill or recovery instance?

When launching drill or recovery instances Windows Server may boot with all the disks as read-only.

This is a common issue that occurs when detaching and attaching data disks. It can be resolved by following the steps in this Microsoft TechNet article.

What impacts the conversion and boot time of drill and recovery instances?

Prior to launching the drill or recovery instance, AWS Elastic Disaster Recovery goes through a machine conversion server process on the boot volume. The conversion process is fairly quick.

While the actual conversion process itself is quick, the time to boot the drill or recovery instance varies depending on many factors unrelated to any Elastic Disaster Recovery processes. Some of these are controllable and should be taken into account when drill or recovery times are of importance.

- Operating system - The amount of time required to boot the operating system is dependent on the OS itself. While Linux servers typically boot quickly, Windows servers may take additional time, due to the nature of the Windows OS. If opportunity permits, drill the boot time of the source server. If Linux OS takes a long time to boot ensure to check that dhclient (Dynamic Host Configuration Protocol Client) is installed and the system so it can pull an IP.

- Scheduled Windows Updates - If the Windows server has pending patches, ensure those are installed prior to launching the drill or recovery instance. If pending patches remain, the boot time in the cloud may be severely impacted as the patch process may commence upon the initial boot.
- Boot volume type - Depending on services/applications, boot time may be impacted by disk performance. It is recommended that boot volumes be drilled with a higher performance SSD and even by provisioning IOPs to ensure throughput. This may be more critical during the first initial boot of the server in the cloud, as all initial settings are applied. In many cases, the boot volume type may be scaled back after the initial boot and should be drilled.

> **Note**
>
> The first boot of Windows machines on AWS may take up to 45 minutes due to Windows adjusting to the AWS virtual hardware.

How is the AWS Licensing Model Tenancy chosen for Elastic Disaster Recovery?

Elastic Disaster Recovery conforms to the Microsoft Licensing on AWS guidelines.

How does Elastic Disaster Recovery interact with interface VPC endpoints?

If you use Amazon Virtual Private Cloud (Amazon VPC) to host your AWS resources, you can establish a private connection between your Amazon VPC and AWS Elastic Disaster Recovery. You can use this connection to allow AWS Elastic Disaster Recovery to communicate with your resources on your VPC without going through the public internet.

Amazon VPC is an AWS service that you can use to launch AWS resources in a virtual network that you define. With a VPC, you have control over your network settings, such the IP address range, subnets, route tables, and network gateways. With VPC endpoints, the routing between the Amazon VPC and AWS services is handled by the AWS network, and you can use IAM policies to control access to service resources.

To connect your VPC to Elastic Disaster Recovery, you define an *interface VPC endpoint* for Elastic Disaster Recovery. An interface endpoint is an elastic network interface with a private IP address

that serves as an entry point for traffic destined to a supported AWS service. The endpoint provides reliable, scalable connectivity to Elastic Disaster Recovery without requiring an internet gateway, network address translation (NAT) instance, or VPN connection. For more information, see What is Amazon VPC in the *Amazon VPC User Guide*.

Interface VPC endpoints are powered by AWS PrivateLink, an AWS technology that facilitates private communication between AWS services using an elastic network interface with private IP addresses. For more information, see AWS PrivateLink.

For more information, see Getting Started in the *Amazon VPC User Guide*.

If the AWS replication agents are installed with a principal using AWSElasticDisasterRecoveryAgentInstallationPolicy and a VPCE policy is used (to scope down access), add the following statement to your policy:

```
{
                "Effect": "Allow",
                "Principal": "*",
                "Action": "execute-api:Invoke",
                "Resource": "arn:aws:execute-api:<region>::*/POST/CreateSessionForDrs"
                }
```

Will AWS Elastic Disaster Recovery reserve EC2 capacity for recovery?

AWS Elastic Disaster Recovery relies on Amazon EC2 On-Demand pools by default. If a specific Amazon EC2 instance type is unavailable to support your recovery, DRS will automatically attempt scale up the instance repeatedly until an available instance type is found, but in extreme circumstances, instances may not always be available. To ensure the availability of the required instance types you need for your most critical applications, you may purchase EC2 Capacity Reservations. You can specifically designate which applications you want to use the EC2 Capacity Reservations for by using launch templates.

Advanced FAQ

Topics

Does AWS DRS support Nutanix?

Nutanix hypervisor is supported along with other hypervisor vendors. The AWS Replication Agent is installed on the virtual machine (VM) and performs block level replication. In addition, the client ISO is booted for failback on the same VM itself.

Does DRS AWS support VMWare vSphere?

VMware vSphere is supported (both on-premises as well as VMware on AWS). Examples of detailed walkthroughs: Disaster recovery for VMware Cloud on AWS using AWS Elastic Disaster Recovery. Performing a failback with the DRS Mass Failback Automation client.

Does AWS DRS support Microsoft Hyper-V?

Both Hyper-V and Microsoft Azure are supported. The AWS Replication Agent installation and replication follows same process described in Adding source servers. For failback to Azure, review the Building a disaster recovery site on AWS for workloads on Microsoft Azure blog post.

Release Notes

AWS Elastic Disaster Recovery performs regular Service and Client releases of new features and capabilities.

Service Release Notes

Client Release Notes

AWS Elastic Disaster Recovery Service Release Notes

May 2024

- AWS Elastic Disaster Recovery now supports protecting Source Servers with up to 60 volumes.

April 2024

- AWS Elastic Disaster Recovery now supports AWS Outposts. For more information see: Working with AWS DRS and Outposts.
- Source Networks – Added support for replicating Security Groups with references to other Security Groups.

January 2024

- AWS managed policy update – Updated AWSElasticDisasterRecoveryServiceRolePolicy and AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policies to support replicating marketplace licenses to launched instances.
- Source Networks – Added support for replicating Security Groups with Prefix Lists.

November 2023

- AWS Elastic Disaster Recovery is now generally available in the AWS GovCloud (US) Regions. This launch gives customers in both the public and commercial sectors, as well as their partners, access to AWS DRS capabilities in the AWS GovCloud (US) Regions.

- Introduced DR drill validation automation for AWS Elastic Disaster Recovery, this allows you to automate validations when launching EC2 instances for recovery and drills.
- AWS managed policy update – updated AWSElasticDisasterRecoveryReadOnlyAccess to support describing additional post-launch actions.
- New AWS managed policy – Added new policy: AWSElasticDisasterRecoveryConsoleFullAccess_v2.
- AWS managed policy updates – Created new revisions to support DRS in AWS GovCloud and added Statement ID (SID) to managed policy statements. The following managed policies were updated:

  AWSElasticDisasterRecoveryAgentPolicy

  AWSElasticDisasterRecoveryAgentInstallationPolicy

  AWSElasticDisasterRecoveryEc2InstancePolicy

  AWSElasticDisasterRecoveryConsoleFullAccess

  AWSElasticDisasterRecoveryLaunchActionsPolicy

  AWSElasticDisasterRecoveryNetworkReplicationPolicy

  AWSElasticDisasterRecoveryRecoveryInstancePolicy

  AWSElasticDisasterRecoveryServiceRolePolicy

  AWSElasticDisasterRecoveryConversionServerPolicy

  AWSElasticDisasterRecoveryFailbackPolicy

  AWSElasticDisasterRecoveryFailbackInstallationPolicy

  AWSElasticDisasterRecoveryStagingAccountPolicy_v2

  AWSElasticDisasterRecoveryStagingAccountPolicy

  AWSElasticDisasterRecoveryReplicationServerPolicy

- New revision of AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy to support DRS in GovCloud

October 2023

- Introduced a new feature: Recover into existing instance, allowing you to set an existing EC2 instance as the target of a drill, recovery or failback launch, instead of launching a new instance.
- AWS managed policy update – Updated policies AWSElasticDisasterRecoveryConsoleFullAccess and AWSElasticDisasterRecoveryLaunchActionsPolicy to support launching into existing instance.

September 2023

- Introduced a new feature: Post-launch actions framework for automating any action needed to be performed on recovery instances after launch.
- Service launch in Israel (Tel Aviv) Region.
- AWS managed policy update – added policies AWSElasticDisasterRecoveryRecoveryInstancePolicy and AWSElasticDisasterRecoveryLaunchActionsPolicy to support post-launch actions.

August 2023

- Added support for Amazon Linux 2023.
- Source Networks – Added support for replicating Route Tables.

July 2023

- Service launch in the following regions: Europe (Zurich), Europe (Spain), Asia Pacific (Hyderabad), Australia (Melbourne), and Middle East (UAE) regions.
- Introduced a new feature: In-AWS Right Sizing, allowing you to easily replicate your EC2 instance and EBS volume types between AWS regions.

June 2023

- Introduced a new feature: Trusted accounts, allowing to quickly create roles for multiple accounts and providing visibility into existing permissions.
- AWS managed policy update – updated AWSElasticDisasterRecoveryAgentInstallationPolicy to support network replication and recovery.

May 2023

- Introduced a new feature: Network replication configurations, allowing you to easily replicate your existing source network configurations, saving time and resources and preventing security risks.

- New AWS managed policy – Added new policies: AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy and AWSElasticDisasterRecoveryNetworkReplicationPolicy policy.
- AWS managed policy updates – Updated the AWSElasticDisasterRecoveryRecoveryInstancePolicy policy, the AWSElasticDisasterRecoveryEc2InstancePolicy policy, the AWSElasticDisasterRecoveryAgentPolicy policy, the AWSElasticDisasterRecoveryServiceRolePolicy policy, and the AWSElasticDisasterRecoveryConsoleFullAccess policy.

April 2023

- Introducing a new feature: Launch settings management, allowing to configure default launch settings that apply to newly add source servers and the ability to update multiple servers' settings.
- AWS managed policy updates – AWSElasticDisasterRecoveryAgentPolicy and AWSElasticDisasterRecoveryConsoleFullAccess.

March 2023

- Introduced a new feature: automated replication of new disks Introduced a new feature: support for Oracle ASM Filter Driver

February 2023

- Introduced a new feature: MAP 2.0 Auto Tagging

December 2022

- New AWS managed policy – Added the AWSElasticDisasterRecoveryStagingAccountPolicy_v2 policy.

November 2022

- Added support for cross-Region failback and cross-Availability-Zone recovery. Learn more about cross-Region failback and cross-Availability-Zone recovery.

- AWS managed policy update – updated AWSElasticDisasterRecoveryAgentInstallationPolicy for Replication Agent reinstallation on recovery instance.

October 2022

- AWS managed policy update – AWSElasticDisasterRecoveryRecoveryInstancePolicy.

September 2022

- Service launch in Asia Pacific (Jakarta) Region.

June 2022

- Service launch in the following regions: US West (N. California), Africa (Cape Town), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), Asia Pacific (Osaka), Asia Pacific (Seoul), Canada (Central), Europe (Milan), Europe (Paris), Europe (Stockholm), Middle East (Bahrain), and South America (São Paulo).
- AWS managed policy update – Updated several policies: AWSElasticDisasterRecoveryAgentInstallationPolicy, AWSElasticDisasterRecoveryFailbackInstallationPolicy, AWSElasticDisasterRecoveryServiceRolePolicy, and AWSElasticDisasterRecoveryReplicationServerPolicy.

May 2022

- AWS managed policy update – Updates the AWSElasticDisasterRecoveryConsoleFullAccess policy and the AWSElasticDisasterRecoveryReadOnlyAccess policy.

April 2022

- AWS managed policy update – Updates the AWSElasticDisasterRecoveryAgentPolicy policy.

March 2022

- Added support for no rescan for all Windows operating systems and certain Linux operating systems. Learn more about the no-rescan feature.

February 2022

- New AWS managed policy – Added the AWSElasticDisasterRecoveryStagingAccountPolicy.

January 2022

- Added support for failback automation.

November 2021

- New AWS managed policy – Added several policies:
  - AWSElasticDisasterRecoveryStagingAccountPolicy
  - AWSElasticDisasterRecoveryAgentPolicy
  - AWSElasticDisasterRecoveryConversionServerPolicy
  - AWSElasticDisasterRecoveryFailbackPolicy
  - AWSElasticDisasterRecoveryFailbackInstallationPolicy
  - AWSElasticDisasterRecoveryConsoleFullAccess
  - AWSElasticDisasterRecoveryReplicationServerPolicy
  - AWSElasticDisasterRecoveryRecoveryInstancePolicy
  - AWSElasticDisasterRecoveryServiceRolePolicy

AWS Elastic Disaster Recovery Client Release Notes

AWS Elastic Disaster Recovery regularly releases new versions of the Replication Agent, Failback Client, and DRS Mass Failback Automation Client with bug fixes and feature enhancements. Releases are staggered, and may not be available in all supported regions simultaneously.

What's in a Release?

AWS Elastic Disaster Recovery releases can contain changes of varying scale, impact, and severity.

- **miscellaneous bug fixes** - contain non-material bug fixes or enhancements within the software components.
- **miscellaneous security enhancements** - provide tangible security enhancements that address vulnerabilities within the software package or subcomponents/dependencies.
- **miscellaneous performance enhancements** - provide non-material improvements to component responsiveness, reliability, and/or reslience.

Agent Version History

The following table describes the released versions of the AWS Elastic Disaster Recovery Agent. See the agent reinstallation instructions on how to upgrade the Replication Agent to the latest version.

What's in a Release?

Agent Version	Details	Release date
6.15.0	• Miscellaneous security enhancements. • Miscellaneous performance enhancements.	23 June 2024
6.14.0	• Miscellaneous security enhancements.	19 June 2024
6.11.0	• Miscellaneous security enhancements.	15 June 2024
6.10.0	• Fixed an issue preventing the installation of the AWS Replication Agent on Linux Source Servers when over 100 storage devices were present. • Miscellaneous security enhancements.	8 June 2024
6.8.0	• Added support for installing the AWS Replication Agent on Ubuntu 24.04 with kernel 6.8. • Miscellaneous security enhancements.	2 June 2024

Agent Version	Details	Release date
6.7.0	• Fixed an issue preventing failback to on-premises when the Recovery Instance was using a Linux Operating System. • Miscellaneous bug fixes.	29 May 2024
6.6.0	• Added support for protecting up to 60 volumes on a Source Server.	18 May 2024
6.5.0	• Fixed issue preventing Replication Agent installation on Linux Source Servers when a debug kernel is present.	14 May 2024
6.3.0	• Miscellaneous security enhancements.	8 May 2024
6.2.0	• (Re-Release) Fixed issue preventing in-place agent upgrades from installations prior to v5.31.	24 April 2024
6.1.0	• Miscellaneous bug fixes. • Miscellaneous security enhancements.	18 April 2024
6.0.0	• Miscellaneous bug fixes. • Fixed an issue preventing some services from correctly starting after a drill in CentOS 6.	12 April 2024
5.37.0	• Fixed issue preventing in-place agent upgrades from installat ions prior to v5.31.	30 March 2024
5.32.0	• Miscellaneous bug fixes.	11 March 2024

Agent Version	Details	Release date
5.31.0	• Miscellaneous security enhancements. • Fixed issue that would lead to a BSOD in Windows Server if the Source Server was migrated to AWS with Application Migration Service. • Removed support for Windows Server 2008 R2 with the `AwsReplicationWindowsInstaller.exe` installer. **Important** New agent installations and existing agent upgrades for Windows Server 2008 R2 should use `AwsReplicationWindowsLegacyInstaller.exe` going forward.	8 March 2024
5.28.0	• Fixed an issue preventing the unmounting of LVM devices on source servers after the AWS Replication Agent was installed.	27 February 2024
5.26.1	• Fixed issue preventing installation when CIFS mounts contained spaces on source servers.	21 February 2024
5.26.0	• Miscellaneous bug fixes.	18 February 2024
5.25.0	• Miscellaneous security enhancements.	12 February 2024
5.24.0	• Miscellaneous bug fixes.	7 February 2024

Agent Version	Details	Release date
5.22.0	• Added support for protecting EC2 Instances with Marketplace Licenses. • Added support for Linux Kernel 6.6 and 6.7. • Fixed issue preventing installation on Linux when using UEFI boot with no grub.cfg present.	28 January 2024
5.20.0	• Miscellaneous bug fixes.	8 January 2024
5.19.0	• Fixed an issue preventing AWS-to-AWS failback from starting when using both `--devices` and `--force-volumes` flags.	27 December 2023
5.18.0	• Fixed issue preventing Agent installation when over 50 disks were attached to the Source Server. • Note: DRS can only replicate up to 50 disks.	19 December 2023
5.17.0	• Minor bug fixes.	18 December 2023
5.15.0	• Added support for replicating to GovCloud Regions. • Fixed issue preventing Copy private IP from functioning in certain Windows Server Portuguese localizations. • Fixed issue that would result in a timeout starting post-launch services on certain Linux configurations. • Minor bug fixes.	10 December 2023
5.14.1	• Fixed issue preventing a Source Server from entering a Healthy data replication status after a reboot.	22 November 2023
5.14.0	• Fixed issue preventing in-AWS failback from succeeding in multi-account scenarios. • Fixed issue preventing SuSE 15 SP3 from successfully launching in Xen-based EC2 Instance Types.	18 November 2023

Agent Version	Details	Release date
5.13.0	• Fixed issue preventing no-rescan-on-reboot from functioning in Ubuntu 22.04.	18 November 2023
5.12.0	• Fixed issue preventing post boot conversion scripts from executing in RHEL 6 installations. • Fixed issue resulting in multiple initrd files being present on Source Servers.	8 November 2023
5.11.0	• Miscellaneous bug fixes.	2 November 2023
5.10.0	• Fixed issue preventing in-place Linux agent upgrades when proxy settings are present.	24 October 2023
5.9.0	• Replicating EC2 instances that were launched with marketplace product codes is no longer supported. • Fixed issue impacting initramfs generation in some Linux versions.	22 October 2023
5.8.0	• Miscellaneous bug fixes.	8 October 2023
5.7.0	• Miscellaneous bug fixes.	27 September 2023
5.6.0	• Installation fixes for SLES15 with UEFI. • Added support for no-rescan-on-reboot on Amazon Linux 2023.	14 September 2023
5.5.0	• Added support for Amazon Linux 2023 (See Supported Operating Systems). • Miscellaneous bug fixes.	7 September 2023

Agent Version	Details	Release date
5.4.0	• Fixed a rare issue preventing OS boot if a reboot was performed immediately after a kernel update. • Added support for Oracle Linux 8.7/8.8 (See Supported Operating Systems). • Miscellaneous bug fixes.	7 September 2023
5.2.0	• Fixed rare issue that could result in a kernel panic in older Linux Kernels.	2 August 2023
5.1.0	• Fixed issue where multipath devices may not be properly detected. • Miscellaneous bug fixes.	26 July 2023
5.0.0	• Fixed issue preventing replication after a kernel upgrade was performed. • Miscellaneous bug fixes and security enhancements.	26 July 2023
4.11.0	• Enhancements for operating systems with btrfs. • Installer Enhancements.	22 June 2023
4.10.0	• Added support for kernel upgrades. • Fixed issue preventing replication after reinstallation on certain older operating systems. • Miscellaneous bug fixes.	19 May 2023
4.8.0	• Added support for no-rescan-on-reboot for btrfs. • Miscellaneous bug fixes. • Installer Enhancements.	22 April 2023
4.7.0	• Introduced auto detection of new disks. • Miscellaneous bug fixes.	22 March 2023

Agent Version	Details	Release date
4.6.0	• Introduced support for Oracle ASM Filter Driver. • Miscellaneous bug fixes and security enhancements.	22 March 2023
4.5.0	• Miscellaneous bug fixes and security enhancements.	22 February 2023
4.4.0	• Added support for RHEL 8.7 (See Supported Operating Systems).	16 February 2023
4.3.0	• Miscellaneous performance enhancements.	12 February 2023
4.1.0	• Fixed issue preventing installation on RHEL 6 Operating Systems when using UEFI. • Fixed issue with replication stalling in Windows Servers with teamed network adapters. • Fixed issue with multipath drive detection in Ubuntu 20.04 and 22.04.	24 January 2023
3.7.0	• Miscellaneous bug fixes and performance enhancements.	28 December 2022
3.6.0	• Miscellaneous bug fixes and security enhancements.	13 December 2022

Agent Version	Details	Release date
3.4.0	• Fixed issue preventing syncing Source Servers in some foreign language Windows Server installation. • Fixed issue allowing installation without necessary flags present. • Fixed issue preventing installation via SSM. • Fixed issue preventing installation on SLES12 SP1. • Added support for no-rescan-on-reboot for RHEL 9. • Fixed issue resulting in occasional rescan after agent reinstall ation on CentOS 6. • Fixed issue with no-rescan-on-reboot on CentOS 6.	20 October 2022
3.3.0	• Miscellaneous performance enhancements.	19 September 2022
3.2.0	• Added support for RHEL 5 and CentOS 5 (See Supported Operating Systems). • Miscellaneous bug fies and performance enhancements.	6 September 2022
3.1.0	• Fixed issue preventing agent startup on CentOS. • Miscellaneous bug fies and performance enhancements.	6 September 2022
3.0.0	• Added support for legacy Windows Server operating systems (See Supported Operating Systems). • Added proxy support for Linux. • Miscellaneous bug fies and performance enhancements.	6 September 2022

Failback Client Version History

The following table describes the released versions of the AWS Elastic Disaster Recovery Failback Client. The latest version of the Failback Client can be downloaded by following the failback instructions.

What's in a Release?

Client Version	Details	Release date
5.21.0	• Fixed issue preventing custom DRS Endpoints from being used during failback.	13 January 2024
5.20.0	• Fixed an issue preventing failback from starting when using static IP configurations and specifying an S3 Endpoint. • Fixed an issue where the failback client may return `ERROR: Manifest download timed out.` during initial configuration.	8 January 2024
5.13.0	• Fixed issue preventing failback client from starting when configured with 4GB of memory.	18 November 2023
5.8.0	• Fixed issue preventing downloading replicator software to the failback client.	8 October 2023
5.5.0	• Fixed issues preventing specifying S3 endpoints while using static IP addresses.	7 September 2023
5.3.0	• Miscellaneous bug fixes.	7 September 2023
5.2.0	• Fixed issue preventing automatic disk mappings during failback. • Miscellaneous manual disk mapping improvements.	2 August 2023
4.12.0	• Fixed issue preventing failback to volumes of different sizes.	6 July 2023
4.11.0	• Fixed issue preventing the failback client from detecting some physical disk configurations. • Fixed issue preventing failback to volumes larger than the replica. • Miscellaneous manual disk mapping improvements.	22 June 2023

Client Version	Details	Release date
4.5.0	• Miscellaneous bug fixes.	22 February 2023
4.1.0	• Fixed issue preventing VMWare tools from being enabled on failback.	24 January 2023
3.6.0	• Miscellaneous bug fixes.	13 December 2022

DRSFA Version History

The following table describes the released versions of the DRSFA Client. The latest version of the DRSFA Client can be downloaded by following the DRSFA installation instructions.

What's in a Release?

Client Version	Details	Release date
5.2.0	• Fixed issue preventing automatic disk mappings during failback. • Miscellaneous manual disk mapping improvements.	2 August 2023
3.6.0	• Miscellaneous performance enhancements.	13 December 2022
3.1.0	• Fixed issue with failback where VMWare VMs would not eject ISOs via cdrom. • Fixed issue with failback when a VMWare VM had an ISO alredy mounted via cdrom.	20 August 2022

CEDR Upgrade Tool Version History

The following table describes the released versions of the CloudEndure to DRS Upgrade Tool. The latest version of the CloudEndure to DRS Upgrade Tool can be downloaded by following the Upgrade Tool Guide.

What's in a Release?

Client Version	Details	Release date
5.28.0	• Fixed an issue preventing upgrades when source machines had very large PIT EBS Snapshots.	27 February 2024
5.20.0	• Fixed an issue preventing upgrades from non-Nitro instances running Windows Server 2022.	8 January 2024
5.14.0	• Fixed issue preventing upgrades when unencrypted EBS volumes were used for replication. • Fixed issue preventing upgrades when specifying devices using the `--drives` flag. • Fixed issue preventing upgrades in GovCloud when KMS keys were specified.	18 November 2023
5.12.0	• Fixed issue preventing DRS service tags from being applied when `--import-blueprint` was used.	12 November 2023
5.3.1	• Fixed issue preventing certain operating sytems from installing the DRS agent after upgrade.	13 August 2023
5.2.0	• Added support for legacy operating systems.	2 August 2023
3.7.0	• Fixed issue preventing upgrade on CloudEndure servers with large disks.	28 December 2022

Document history for the AWS Elastic Disaster Recovery User Guide

The following are the latest documentation updates for AWS Elastic Disaster Recovery. We update the documentation frequently to address the feedback that you send us.

Change	Description	Date
Updated AWS managed policies	Created new revisions of managed policies to support additional parameter types in SSM Parameters Store for post-launch actions. The following managed policies were updated: AWSElasticDisasterRecoveryConsoleFullAccess_v2 AWSElasticDisasterRecoveryLaunchActionsPolicy	May 19, 2024
AWS Elastic Disaster Recovery Support for AWS Outposts	AWS Elastic Disaster Recovery now supports AWS Outposts. For more information see: Working with AWS DRS and Outposts.	April 18, 2024
Updated AWS managed policy	Created a new revision of the AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy, to support	January 28, 2024

Change	Description	Date
	replicating marketplace licenses to launched instances.	
Updated AWS managed policy	Created new revision of the AWSElasticDisasterRecoveryServiceRolePolicy policy, to support replicating marketplace licenses to launched instances.	January 28, 2024
Updated AWS managed policies	Updated policies to support managed prefix lists for network replication and recovery. The following managed policies were updated: AWSElasticDisasterRecoveryNetworkReplicationPolicy AWSElasticDisasterRecoveryServiceRolePolicy	January 3rd, 2024

Change	Description	Date
Updated AWS managed policies	Created new revisions of managed policies to support DRS to GovCloud and added Sid to statements in managed policies. The following managed policies were updated: AWSElasticDisasterRecoveryAgentPolicy AWSElasticDisasterRecoveryAgentInstallationPolicy AWSElasticDisasterRecoveryEc2InstancePolicy AWSElasticDisasterRecoveryConsoleFullAccess AWSElasticDisasterRecoveryLaunchActionsPolicy AWSElasticDisasterRecoveryNetworkReplicationPolicy AWSElasticDisasterRecoveryRecoveryInstancePolicy AWSElasticDisasterRecoveryServiceRolePolicy	November 27, 2023

Change	Description	Date
	AWSElasticDisasterRecoveryConversionServerPolicy AWSElasticDisasterRecoveryFailbackPolicy AWSElasticDisasterRecoveryFailbackInstallationPolicy AWSElasticDisasterRecoveryStagingAccountPolicy_v2 AWSElasticDisasterRecoveryStagingAccountPolicy AWSElasticDisasterRecoveryReplicationServerPolicy	
Updated AWS managed policy	Created new revision of AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy to support DRS in GovCloud	November 27, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryReadOnlyAccess policy to support describing additional post-launch actions.	November 27, 2023

Change	Description	Date
New AWS managed policy	Added the AWSElasticDisasterRecoveryConsoleFullAccess_v2 policy to support running predefined post-launch actions from the console.	November 27, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryServiceRolePolicy policy to support managed prefix lists for network replication and recovery.	October 15, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryConsoleFullAccess and AWSElasticDisasterRecoveryLaunchActionsPolicy policies to support recovery into existing instance.	October 15, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryEc2InstancePolicy policy to allow sending installation result metrics to AWS Elastic Disaster Recovery.	October 10, 2023

Change	Description	Date
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryAgentInstallationPolicy policy to allow sending installation result metrics to AWS Elastic Disaster Recovery.	October 10, 2023
New AWS managed policy	Added the AWSElasticDisasterRecoveryLaunchActionsPolicy policy to support post-launch actions.	September 13, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryReadOnlyAccess policy to support post-launch actions.	September 13, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryEc2InstancePolicy policy to support network replication and recovery.	June 13, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryAgentInstallationPolicy policy to support network replication and recovery.	June 13, 2023

Change	Description	Date
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryConsoleFullAccess policy to support network replication and recovery.	June 13, 2023
New AWS managed policy	Updated the AWSElasticDisasterRecoveryNetworkReplicationPolicy policy to support network replication and recovery.	June 13, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryServiceRolePolicy policy to support network replication and recovery.	June 13, 2023
New AWS managed policy	Added the AWSElasticDisasterRecoveryCrossAccountReplicationPolicy policy to support cross-account failback.	May 7, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryRecoveryInstancePolicy policy to support cross-account failback by the agent after reverse replication.	May 7, 2023

Change	Description	Date
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryEc2InstancePolicy policy to support cross-account replication by the agent.	May 7, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryConsoleFullAccess policy to support default EC2 launch templates and bulk editing of source server EC2 launch templates.	April 19, 2023
Updated AWS managed policy	Updated the AWSElasticDisasterRecoveryAgentPolicy policy to support the kernel upgrade feature.	April 1, 2023
New AWS managed policy	Added the AWSElasticDisasterRecoveryStagingAccountPolicy_v2 policy to support the recover of source servers into a separate target account and to allow failing back.	December 11, 2022
Cross-Region failback and cross-Availability-Zone recovery	Added support for cross-Region failback and cross-Availability-Zone recovery.	November 27, 2022