

Implementation Guide

Research Service Workbench on AWS



Research Service Workbench on AWS: Implementation Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Overview	1
Features and benefits	2
Use cases	3
Concepts and definitions	3
Architecture overview	5
Architecture diagram	5
Bring Your Own Network (BYON) architecture	7
AWS Well-Architected design considerations	10
Operational excellence	10
Security	11
Reliability	11
Performance efficiency	12
Cost optimization	12
Sustainability	12
Architecture details	13
Datasets: Research study data management	13
Projects	13
RSW API call authorization	13
AWS services in this solution	14
Plan your deployment	16
Cost	16
Sample cost table	16
Security	17
IAM roles	18
Amazon Cognito	18
Amazon Virtual Private Cloud	18
Amazon CloudWatch	18
Security groups	20
Protecting availability	20
Supported AWS Regions	21
Quotas	22
Quotas for AWS services in this solution	22
AWS CloudFormation quotas	22
Resource limits within the solution	22

Deploy the solution	24
AWS CloudFormation template	24
Prerequisites	25
Deployment process overview	25
Deploy using CloudFormation template	26
Step 1: Launch the stack	26
Step 2: Deploy the hosting account	29
Deploy using CDK CLI	29
Step 1: Configure the stack	29
Step 2: Launch the main account stack	30
Step 3: Deploy the hosting account	32
Deploy the hosting account	32
Step 1: Choose a template to deploy the hosting account	32
Step 2: Deploy the CloudFormation stack for the hosting account	33
Monitoring the solution with Service Catalog AppRegistry	36
Activate CloudWatch Application Insights	38
Activate AWS Cost Explorer	40
Activate cost allocation tags associated with the solution	40
Configure AppRegistry application limits	40
Uninstall the solution	42
Prerequisite: Stop sharing SSM documents	42
Using the AWS Management Console	43
Using AWS Command Line Interface	43
Deleting the Amazon S3 buckets	43
Deleting the Amazon DynamoDB tables	44
Deleting the AWS CloudWatch logs	45
Deleting the Amazon Cognito user pool	45
Scheduling the deletion of KMS keys	46
Deleting Service Catalog resources	47
Delete portfolio	47
Delete product	48
Deleting IAM roles	48
Deleting the AppRegistry in hosting account	49
Deleting AWS CloudFormation stacks in the hosting account	50
Developer guide	51
Source code	51

Contributing to GitHub	51
Configuration guide	51
Setup Account Resources	51
Setup EnvironmentTypeConfig	54
Create new datasets	59
Environment management	60
User Management	63
Import solution collection into Postman	65
Obtain Access Token for making authenticated API requests	66
Integration Tests	66
Customization guide	72
Custom Environment Types	72
Incorporate Existing Network Resources (Bring Your Own Network)	72
Enabling AWS WAF	73
Update throttle limits with API Gateway and Lambda reserved concurrency	73
Add deny list using AWS WAF	73
API reference	75
Actions	76
Data types	169
Migration guide	195
Cost of migration	195
Migration prerequisites	195
Migration process	195
Migrating with AWS Management Console	196
Migrating with AWS CLI	199
Migration cleanup	199
Reference	200
Anonymous data collection	200
Related resources	201
Contributors	201
Notices	202
Revisions	203

Overview

Publication date: *June 2023* ([last update: September 2023](#))

Research Service Workbench on AWS is a backend API designed to empower AWS partners to more easily build self-service portals for research institutions, allowing their researchers to create and manage secure research computing environments. RSW builds on AWS services including Amazon S3, Service Catalog, and AWS Systems Manager to deliver comprehensive, cutting-edge capabilities in less time while meeting the high bar for security and governance demanded by regulators and funding agencies.

Using a custom built portal provided by a partner, built on Research Service Workbench, research IT departments can more easily work with partners to deliver a self-service portal with a comprehensive catalog of tools tailored to the needs of the researchers they support. Using a portal built using RSW, researchers can create, access, and retire environments without needing to be experts on cloud infrastructure or security. They can work individually or collaborate with others on the same data, with RSW securely orchestrating the connection of research environments to Amazon S3. Furthermore, environments are deployed in specific AWS accounts and Amazon Virtual Private Clouds (VPCs), allowing connectivity to resources on and off campus. Institutions are fully in control of their data, data flows, and AWS footprint. They can scale investments in IT and security across multiple research programs, reducing duplicate effort and freeing up resources to focus on science.

This implementation guide provides an overview of the Research Service Workbench on AWS solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the Research Service Workbench on AWS solution to the Amazon Web Services (AWS) Cloud.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution.	Cost
The estimated cost for running this solution in the US East (N.Virginia) Region is USD \$ 140.98 a month.	

If you want to . . .	Read . . .
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution. Make sure you have sufficient quota for each of the services in this solution.	Quotas
Know which AWS Regions are supported for this solution. Not all services used in this solution are available in all AWS Regions.	Supported AWS Regions
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the “stack”) for this solution.	AWS CloudFormation template

This guide is intended for solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals who want to implement Research Service Workbench in their environment.

Features and benefits

Research Service Workbench provides the following features:

Preconfigured sample research environment

Deploys Amazon SageMaker with Jupyter Notebook.

Reduce time to science

Access research environments in minutes.

Conduct research securely

Maintain consistent security, compliance, and governance.

Globally accessible

Collaborate with researchers around the world.

Scale and agility to grow

Virtually limitless tooling via Service Catalog.

Integration with [Service Catalog AppRegistry](#) and [AWS Systems Manager Application Manager](#)

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both [Service Catalog AppRegistry](#) and [AWS Systems Manager Application Manager](#). With this integration, you can centrally manage the solution's resources.

Use cases

Research is an innovation practice

With researchers needing access to tools that enable rapid experimentation, Amazon Web Service (AWS) works with higher education institutions, research labs, and researchers around the world to offer cost-effective, scalable, and secure compute, storage, and database capabilities to accelerate time to science.

Self-service portal access to approved research environments

Working with Solution Spark Partners, institutions can build simple web portals for researchers to provision cloud computing solutions without the complexity of accessing the AWS Management Console. Instead of integrating AWS Services across networking, compute, storage, and security, researchers can provision predefined, institution approved computing solutions which are governed, secure, and compliant within minutes.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

Data source

Simple Storage Service (S3) resources that hold datasets.

Environment

A term used to describe compute resources provisioned with Research Service Workbench.

Environment type

The compute options approved by IT admin users and made available to Project admins to provision.

Hosting account

Secondary AWS account which hosts compute environments and data.

Main account

The primary AWS account which deployed Research Service Workbench.

For a general reference of AWS terms, see the [AWS glossary](#) in the AWS General Reference.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.

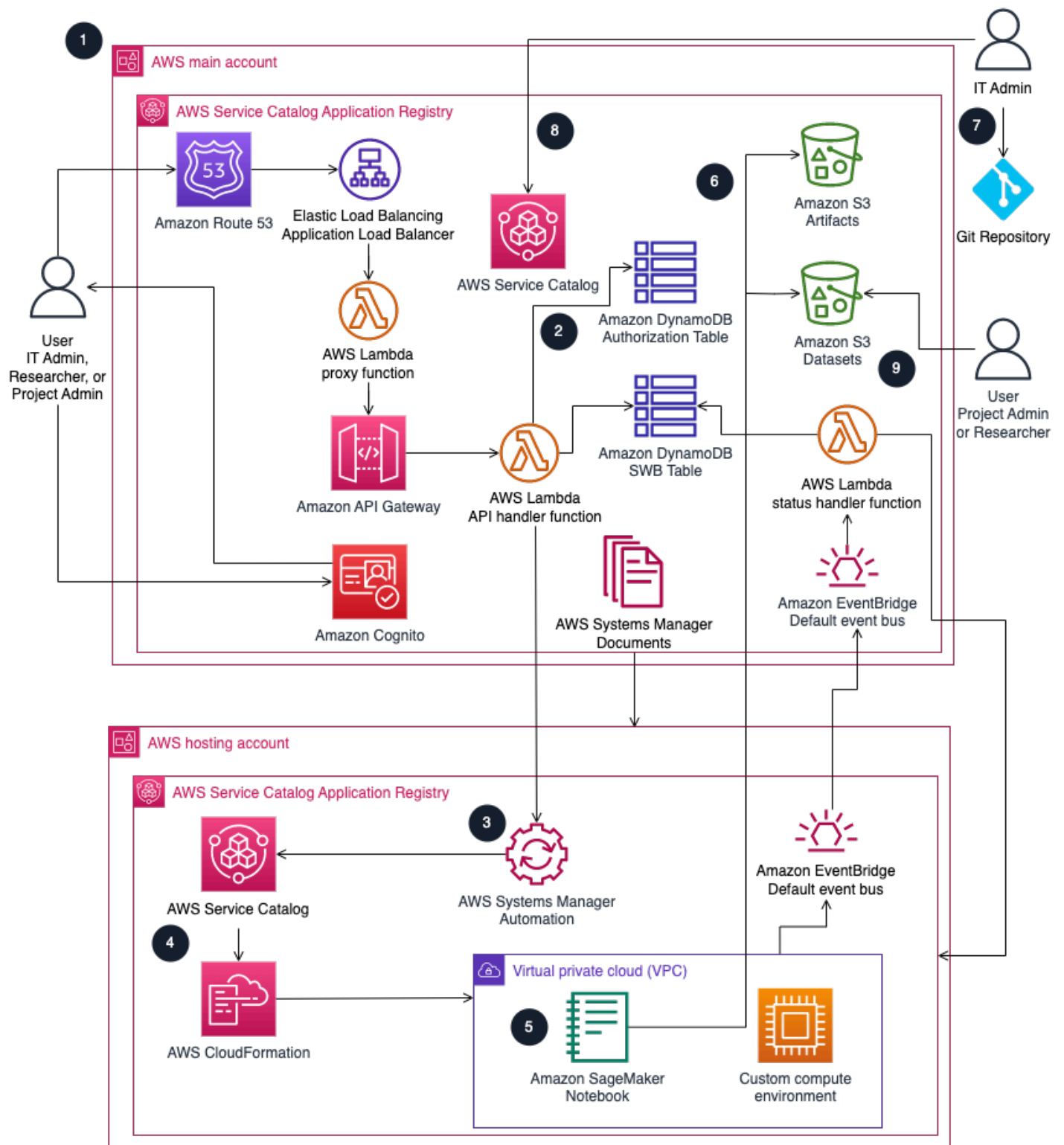


Figure 1: Research Service Workbench architecture on AWS

Note

AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) (AWS CDK) constructs.

The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. Deploy using CloudFormation template or by using the CDK CLI in the main account and hosting account subsequently.
2. Users get access based on group permissions stored in the encrypted Amazon DynamoDB authorization table.
3. Systems Manager processes the shared documents from the main account in the hosting account to launch the environment.
4. To launch an environment, AWS Service Catalog runs the AWS CloudFormation template within the product.
5. CloudFormation creates a stack from the template to launch the environment within the Virtual Private Cloud (VPC) of the Hosting Account. Environments are created within Public Subnets of the VPC. An example of these environments is an Amazon SageMaker notebook.
6. After creation, environments within the hosting account can connect to the Amazon Simple Storage Service (Amazon S3) artifacts bucket in the main account to set up custom scripts in the instance. After creation, environments created with attached datasets can access datasets within the Amazon S3 datasets bucket through Access Points in the main account.
7. IT Admins can customize the source code hosted on GitHub, including full customization of environment lifecycles (launch, terminate, start, and stop) and connection. Changes to source code require re-deployment to take effect.
8. IT Admins can publish custom environment types to Service Catalog within the main account.
9. Users can request pre-signed S3 URLs through an RSW API endpoint. With the pre-signed S3 URL, users can add data to datasets within the Datasets S3 bucket in the main account.

Bring Your Own Network (BYON) architecture

Bring Your Own Network (BYON) is an alternative architecture for Research Service Workbench on AWS where you deploy the solution into an existing network configuration. In addition to the

previous architecture details, the Application Load Balancer can be deployed into an existing VPC and a set of either private or public subnets by specifying those values in the configuration files during deployment.

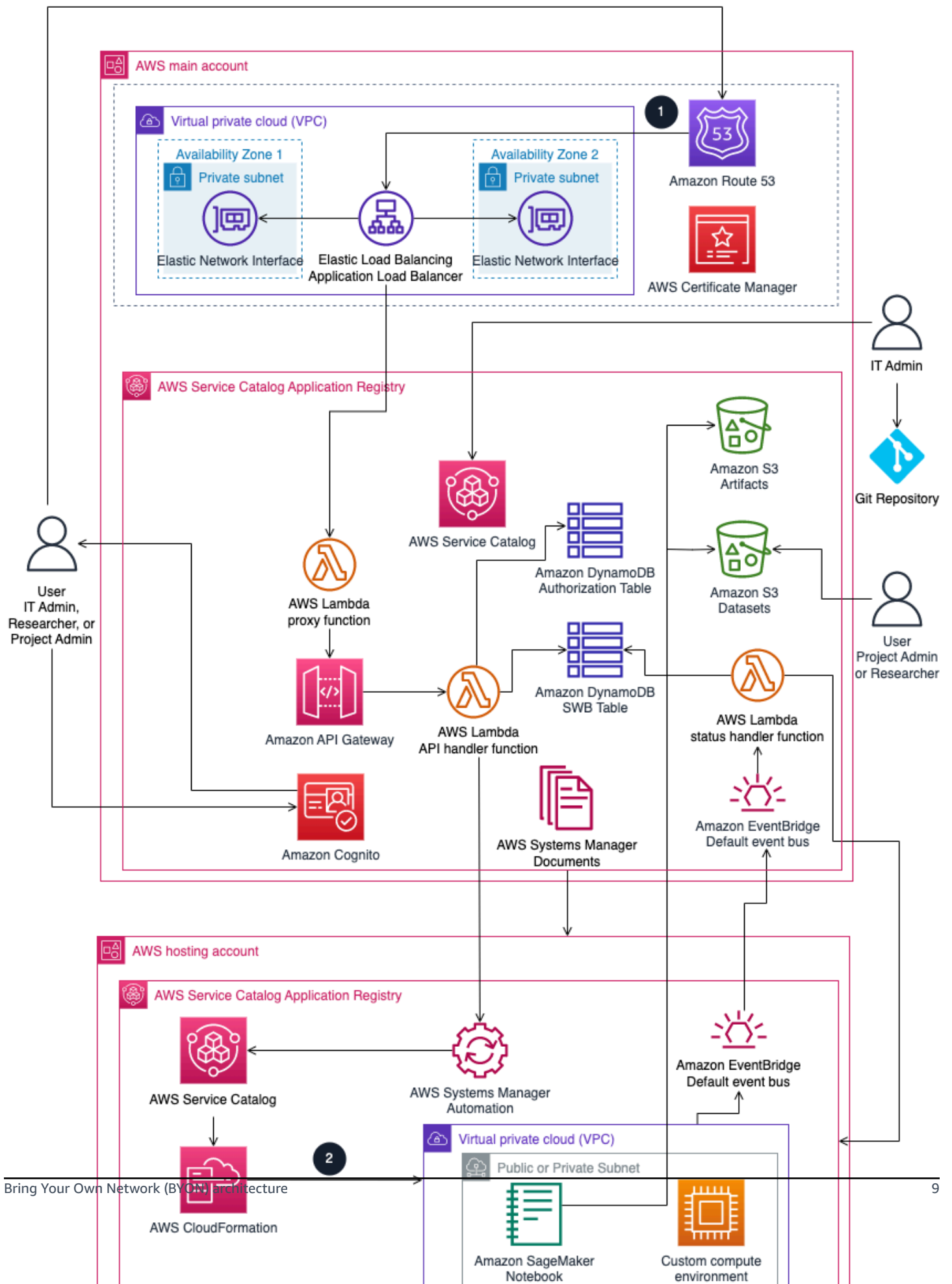


Figure 2: BYON architecture on AWS

Note

AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) (AWS CDK) constructs.

In addition to the previous architecture diagram, the high-level process flow for the BYON solution components deployed is as follows:

1. Incoming traffic is balanced using an Application Load Balancer in a VPC subnet within the main account. Depending on configuration, the subnets can be private or public and be deployed to multiple Availability Zones.
2. When creating an environment, CloudFormation launches an environment within the VPC of the hosting account using a stack from the template. Depending on your network configuration, environments deploy within either public or private subnets of the VPC. An example of these environments is a SageMaker notebook. How you [onboard a hosting account](#) affects the architecture of the subnets.

AWS Well-Architected design considerations

This solution was designed with best practices from the [AWS Well-Architected Framework](#) which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Operational excellence

This section describes how the principles and best practices of the [operational excellence pillar](#) were applied when designing this solution.

- Research Service Workbench provides metrics to Amazon CloudWatch for infrastructure visibility, and all processes handled by our compute layer are logged in CloudWatch. This includes both our API servers and our backend worker processes.

- Infrastructure deployment is managed through the AWS [CDK](#).

Security

This section describes how the principles and best practices of the [security pillar](#) were applied when designing this solution.

- Inter-service communications use AWS IAM roles.
- Multi-account communications use AWS IAM roles.
- Roles used by the solution follow least-privilege access. It only requires minimum permissions to function properly.
- Amazon Cognito handles communication between the end user and the Amazon API Gateway by generating a JWT token.
- Data storage, including Amazon S3 buckets, have encryption at rest.
- Controlled Access Points protects access to data within the Datasets S3 bucket.
- Dynamic Authorization using Amazon Cognito and DynamoDB tables ensures user access complies with the permissions within each project.

Reliability

This section describes how the principles and best practices of the [reliability pillar](#) were applied when designing this solution.

- To ensure high availability and recovery from service failure, the solution uses AWS Serverless Services wherever possible (examples include AWS Lambda, Amazon API Gateway, and Amazon S3).
- Data stored within Amazon DynamoDB has Point-in-time recovery (PITR) by default to protect from accidental write or delete operations. With point in time recovery, you don't have to worry about creating, maintaining, or scheduling on-demand backups. Amazon DynamoDB automatically scales the database capacity based on traffic.
- Data processing uses AWS Lambda functions. AWS Lambda functions run in multiple Availability Zones to ensure that it is available to process events in case of a service interruption in a single zone by default.

Performance efficiency

This section describes how the principles and best practices of the [performance efficiency pillar](#) were applied when designing this solution.

- The solution uses AWS serverless architecture throughout. This removes the operational burden of managing physical servers, and can lower transactional costs because managed services operate at cloud scale.
- The solution can launch in any region that supports AWS services used in this solution such as: AWS Lambda, Amazon API Gateway, Amazon S3, Amazon Cognito, and Service Catalog.
- The solution uses managed services throughout to reduce the operational burden of resource provisioning and management.

Cost optimization

This section describes how the principles and best practices of the [cost optimization pillar](#) were applied when designing this solution.

- Because the solution uses serverless architecture where possible, costs have been minimized.
- Amazon DynamoDB scales capacity on demand, so you only pay for the capacity you need.

Sustainability

This section describes how the principles and best practices of the [sustainability pillar](#) were applied when designing this solution.

- To minimize the environmental impact of the backend services, Research Service Workbench uses managed and serverless services. Serverless technology (such as AWS Lambda and Amazon DynamoDB) reduces the carbon footprint compared to continually operating on-premise servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Datasets: Research study data management

Research Service Workbench on AWS uses datasets to allow users to create and share research studies. The private Amazon S3 bucket created during installation hosts all studies. When the Project Admin creates a dataset for their project, the dataset's owner property becomes the project's `id` value. The S3 bucket's interaction permissions are as follows:

- Owing project researchers:

Full read and write access to data in the dataset.

- Non-owning project researchers:

No default access to datasets in other projects.

A Project Admin can share or un-share a dataset within the owning project. Once shared to the project, the dataset can be shared to another project's researchers. Access permissions can be read-only or read-write.

Projects

Projects group data, compute resources, access keys, and researchers to help define permission boundaries. Users can belong to many projects in one of two capacities: `ProjectAdmin` or `Researcher`. Researchers have the ability to access compute resources, access keys, and datasets belonging to their project and any dataset shared with them. The Project Admin role manages the project's permissions, including things like creating datasets and sharing them with other Projects.

RSW API call authorization

RSW API call authorization restricts what resources a user may access. It is built on top of [CASL](#), a popular JavaScript authorization library, and Amazon Cognito groups. Authorization enforcement

occurs at the middleware level of the application, ensuring that every request is authorized before it is processed. The authorization flow is as follows:

1. A request is made to an API route.
2. The RSW API call authorization checks to see if the action has been authorized for any of the requesting user's Amazon Cognito groups.
3. A 403 response is sent for any actions not authorized for the user's Amazon Cognito groups. Otherwise, the request is processed by the application.

AWS services in this solution

AWS service	Description
AWS Lambda	Core. Four Lambda functions: A backend API lambda function handles requests; an Account lambda function updates account information in the database; an Environment status lambda function monitors the lifecycle of an environment; a Proxy lambda function proxies incoming requests to the API Gateway.
Amazon Cognito	Core. Manages user identities and authentication. Project roles are managed by Cognito Groups.
Service Catalog	Core. Stores information about pre-created compute environments that Researchers can deploy.
AWS CloudFormation	Core. Manages the infrastructure needed to run the application including creation, deletion, and restoration of previous versions.
Amazon EC2	Core. Provides the underlying compute services to run a user's environment.

AWS service	Description
Amazon EventBridge	Supporting. Manages messaging between Main and Hosting accounts. Used in combination with the Account lambda function to keep account information updated.
AWS Systems Manager	Supporting. Stores documents for executing commands on computing environments, including startup and teardown logic for the environment.
Amazon DynamoDB	Supporting. Stores data for the application and the Dynamic Authorization Service.
Amazon SageMaker	Supporting. The service that runs on the default environment type provided with the application.
Amazon API Gateway	Supporting. Allows for custom routing based on the domain and subdomains of the request, allowing different load balancers to be targeted if desired.
Application Load Balancer	Supporting. Manages traffic to Lambda instances to support scaling.
Amazon Route 53	Supporting. Routes incoming requests to the appropriate endpoint of the application based on the type of network being used (default or Bring Your Own Network).

Plan your deployment

Research Service Workbench on AWS requires at least two AWS accounts. The account which deploys Research Service Workbench will be designated as the main account. Other accounts, referred to as hosting accounts, host compute workloads or share data.

Research Service Workbench installs to a single region. When deploying the main account and onboarding hosting accounts, they must use the same AWS region. Multi-region deployments are not supported. However, even though the accounts must use the same Region, the Availability Zone in each hosting account may be different. Each hosting account supports one subnet in a single Availability Zone.

With AWS Route53 for Domain Name Services (DNS) and AWS Certificate Manager to serve SSL/TLS certificates, Research Service Workbench uses one custom domain name and its associated certificate. Domain name and associated Route53 hosted zone are prerequisites whereas the ACM certificates for this domain name is created automatically during app installation.

Research Service Workbench creates custom encryption keys using AWS Key Management Service (KMS) during deployment. Other custom KMS keys can be used for data storage shared with Research Service Workbench, but RSW must be given permission to use those keys.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of September 2023, the cost for running this solution with the default settings in the US East (N. Virginia) is approximately **\$140.98 a month**.

See the pricing webpage for each AWS service used in this solution.

We recommend creating a [budget](#) through [AWS Cost Explorer](#) to help manage costs. Prices are subject to change. For full details, see the [pricing webpage](#) for each AWS service used in this solution.

Sample cost table

The following table provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

AWS service	Dimensions	Cost [USD]
Amazon API Gateway	1,000,000 REST API calls per month	\$ 3.50
Amazon Cognito	50 unique SAML authentications	\$ 0.28
AWS Lambda	2,000,000 requests with 200 ms duration	\$ 5.28
Amazon DynamoDB	2,000,000 reads; 500,000 writes	\$ 1.13
Amazon S3	50 GB; 1,000,000 requests	\$ 6.15
KMS keys	4 customer keys; 1,000,000 decryption requests	\$ 7.00
Elastic Load Balancer	720 hours	\$19.40
Amazon EC2 – NAT Gateway [*]	3 NAT Gateways; 720 hours each	\$97.20
Service Catalog, AppRegistry	100 calls	\$ 0.14
Amazon Route 53	1,000,000 queries; 1 hosted zone	\$ 0.90

^{*}If you do not plan to use private subnets, the NAT Gateways created by the solution may be deleted.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This [shared responsibility model](#) reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit [AWS Cloud Security](#).

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions access to create Regional resources. During deployment, Research Service Workbench names the created IAM roles with the specified `SolutionName` and `StageName`.

Amazon Cognito

With Amazon Cognito, you can add user sign-up and sign-in features and control access to your web and mobile applications. Amazon Cognito provides an identity store that can scale to millions of users. This service is used for creating a User Pool that acts as the default identity provider for the application. This User Pool also manages various authorization roles that a user can be granted access to within the application. Cognito therefore serves as the authentication source and facilitates user authorization processes for the application.

Amazon Virtual Private Cloud

The [Amazon Virtual Private Cloud \(VPCs\)](#) in this solution control and isolate network traffic into research compute environments. We recommend that you review the configuration and further restrict access as needed after deploying the solution. Research Service Workbench creates network components like VPC and associated public subnets via the default installation method. However, users could enter their pre-configured network components into the application for more granular network management.

Amazon CloudWatch

Research Service Workbench on AWS enables logging in the API Gateway by default. Find your logs in Amazon CloudWatch within your log group at: `/aws/api-gateway/<name of your API>`.

Log groups created in the main account by Research Service Workbench contain the namespace (ie. `swb-<stage>-<regionShortName>`) in their names.

Audit log example:

```
{  
  "timestamp": 1679705065785,
```

```

"actor": {
  "id": "1910dae1-3b34-4356-b823-26664bda8d01",
  "roles": [
    "ITAdmin",
    "<projectIdRedacted>#ProjectAdmin"
  ]
},
"action": "isAuthorizedOnRoute",
"source": {
  "serviceName": "DynamicAuthorizationService6"
},
"requestBody": {
  "authenticatedUser": {
    "id": "<userUuidRedacted>",
    "roles": [
      "ITAdmin",
      "<projectIdRedacted>#ProjectAdmin"
    ]
  },
  "route": "/environmentTypes/<envTypeIdRedacted>/configurations",
  "method": "POST"
},
"statusCode": 200,
"logEventType": "audit"
}

```

API request log example:

```

{
  timestamp: 1679705065786,
  awsRequestId: 'sampleRequestId',
  logEventType: 'audit',
  action: 'POST /environmentTypes/<envTypeIdRedacted>/configurations',
  actor: { uid: '<userUuidRedacted>' },
  source: { ip: '<IP1Redacted>, <IP2Redacted>' },
  body: {
    type: 'sagemakerNotebook',
    description: 'Example config 1',
    name: 'config 1',
    estimatedCost: 'estimated cost',
    params: [ [Object], [Object], [Object], [Object] ]
  }
}

```


Metrics

Amazon CloudWatch stores the default metrics for AWS Lambda and Amazon API Gateway. For the full list of available metrics, see:

- [Working with AWS Lambda function metrics – AWS Lambda](#)
- [Amazon API Gateway dimensions and metrics – Amazon API Gateway](#)

Research Service Workbench on AWS does not produce any custom metrics.

Canaries

Research Service Workbench on AWS does not create any canaries by default. You can use Amazon CloudWatch Synthetics to create canaries, configurable scripts that run on a schedule, to monitor your endpoints and APIs. If desired, please refer to the user guide for [Using synthetic monitoring](#) within Amazon CloudWatch.

Security groups

The security groups created in this solution control and isolate network traffic between the Lambda functions, CSR instances, and remote VPN endpoints. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

Protecting availability

To protect the availability of Research Service Workbench (RSW) from network request overload, you can optionally configure the solution with throttling limits, a deny list, or both.

Throttling limits

Throttling your APIs protect the solution from being overwhelmed by too many requests. You can control the throttling limit with Amazon API Gateway, a fully managed service which makes it easy for developers to create, publish, maintain, monitor, and secure APIs. the solution sets the API Gateway rate limit and burst limit to throttle requests per second and concurrent requests, respectively. For more information on API Gateway throttling, see [Throttle API requests for better throughput](#).

The solution uses AWS Lambda, a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing

servers, as a compute layer. The Lambda reserved concurrency limit sets the maximum number of concurrent instances for a function. For more information on lambda reserved concurrency, see [Configuring reserved concurrency](#).

To ensure Lambda can handle the maximum API load, the API Gateway throttle limit has been set to the same as the reserved concurrency. By default, the solution sets a throttle limit of 200 requests per second. To adjust the limit in your deployment, see [Update throttle limits with API Gateway and Lambda reserved concurrency](#).

Deny list

To protect the solution from being flooded with requests from one or more IP addresses, you can configure a deny list using [AWS WAF](#), a web application firewall that controls access to your APIs. You can set up custom rules to block or allow access to the solution's APIs based on requests to your API. To block a set of IP addresses, see [Add deny list using AWS WAF](#).

Supported AWS Regions

This solution uses the AWS Lambda, Amazon API Gateway, Amazon S3, Amazon Cognito, and Service Catalog services, which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability of AWS services by Region, see the [AWS Regional Services List](#).

Research Service Workbench is supported in the following AWS Regions:

Region name	
US East (Ohio)	Canada (Central)
US East (N. Virginia)	Europe (Frankfurt)
US West (N. California)	Europe (Ireland)
US West (Oregon)	Europe (London)
Asia Pacific (Mumbai)	Europe (Milan)
Asia Pacific (Seoul)	Europe (Paris)
Asia Pacific (Singapore)	Europe (Stockholm)

Region name	
Asia Pacific (Sydney)	Middle East (Bahrain)
Asia Pacific (Tokyo)	South America (São Paulo)

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the [services implemented in this solution](#). For more information, see [AWS service quotas](#).

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the [Service endpoints and quotas](#) page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when [launching the stack](#) in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see [AWS CloudFormation quotas](#) in the *AWS CloudFormation User's Guide*.

Resource limits within the solution

Research Service Workbench on AWS imposes limits on some resources within the solution.

Users per project

Within Amazon Cognito, a user can belong to up to [100 groups](#). Users within RSW can only belong to 100 projects as either a Project Admin or Researcher. This is a hard limit.

Projects

User pools within Amazon Cognito can only contain [10,000 groups](#). Therefore, only 4,999 projects can be created in RSW. Each project creates two groups—Project Admin and Researcher groups. IT Admin is another group within the RSW deployment. This is a hard limit.

SSH keys

Within EC2, a single account can have [5,000 key pairs](#) per Region. There can be no more than 5,000 active SSH keys in RSW. Each SSH key is backed by a key pair and scoped to the Region in which RSW is deployed. This is a hard limit.

Environments

Research Service Workbench on AWS groups and adds metadata and attributes to resources using [AWS App Registry](#) applications. With App Registry, Research Service Workbench organizes resources and tracks dependencies more efficiently. RSW associates every resource to an App Registry application including all workspaces. App Registry currently has a default limit of 1000 resources per application.

If you are estimating to create more than 999 environments in your Research Service Workbench instance, a service quota increase will be needed. Follow these steps to request a quota increase for App Registry application resources:

1. Sign in to your Hosting Account in the [AWS Management Console](#).
2. Open the [Service Quotas console](#).
3. Go to **Service Catalog** and choose **Service Catalog** from the results.
4. Under **Services Quota**, choose **Resources per application**.
5. On the **Resources per application** page, choose **Request Quota Increase** under Recent quota increase requests.
6. In **Change quota**, enter your estimated number of workspaces plus one for an infrastructure resource.
7. Choose **Request**.

Hosting accounts

The solution has a limit of 403 hosting accounts.

Deploy the solution

This solution uses [AWS CloudFormation templates and stacks](#) to automate its deployment. The CloudFormation template(s) describes the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template(s).

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

[View template](#)

Research Service Workbench on AWS - Use this template to launch the solution and all associated components. The default configuration deploys Amazon Cognito, Application Load Balancers, AWS IAM roles, Amazon DynamoDB tables, Amazon S3 buckets, AWS KMS keys, Amazon VPC, subnets, Amazon API Gateway, AWS Lambda functions, and ACM certificates, but you can customize the template to meet your specific needs.

Note

AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) (AWS CDK) constructs.

This AWS CloudFormation template deploys Research Service Workbench on AWS in the AWS Cloud. Before launching the stack, see [Prerequisites](#).

Before you launch the solution, review the [cost](#), [architecture](#), [network security](#), and other considerations discussed earlier in this guide.

Important

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and

products. AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#).

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, see the [Anonymous data collection](#) section of this guide.

Prerequisites

Before installing the solution, you must have the following:

- Two AWS accounts
 - Use command line to [set your credential file](#) to have your **main account** as the default profile.
 - Set your main account's Lambda concurrent executions limit to at least 541. By default the limit should be set to 1,000. For more information, see [AWS service quotas](#).
 - Provide a separate AWS account for hosting environments, known as the **hosting account**.
 - Perform CDK [bootstrapping](#) for the region where you will deploy the solution.
- A domain name and a Route 53 hosted zone for the domain name in your main account
- Software:
 - [Rush](#) v5.62.1 or later to manage mono-repo packages
 - Node 14.x or 16.x ([compatible node versions](#))
 - An API client ([POSTMAN](#) recommended)
- (optional) To run Lambda functions locally:
 - [SAM CLI](#)
 - [Docker](#)

Deployment process overview

Before you launch the solution, review the cost, architecture, security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 30-45 minutes

Choose your deployment option:

CloudFormation template

Use [this method](#) if you do not plan to bring your own network to the main account. You may still use your own network for the hosting account.

1. [Launch the stack.](#)
2. [Deploy a hosting account.](#)

CDK CLI

Use [this method](#) if you plan to bring your own network for the main account or plan to customize the code before launching the solution.

1. [Configure the stack.](#)
2. [Launch the main account stack.](#)
3. [Deploy a hosting account.](#)

Deploy using CloudFormation template

Step 1: Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 15 minutes

1. A blue rectangular button with rounded corners containing the text "Launch solution" in white, bold, sans-serif font.

Sign in to the AWS Management Console and select the button to launch the Research Service Workbench on AWS AWS CloudFormation template.

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

Note

This solution uses the Amazon Cognito service, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Cognito is available. For the most current availability by Region, see the [AWS Regional Services List](#).

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see [IAM and STS Limits](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Cognito Domain Prefix		Provide a string for your Cognito domain name prefix. Cognito domain names must be globally unique.
Domain name		Use a custom domain owned by you. TLS certificates are generated at the time of application deployment.
Hosted Zone ID		Route 53 Hosted Zone ID linked to your domain name.
Required Stack Name	rsw-prod-release	Copy the value in the Stack name field.

Parameter	Default	Description
		<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p>⚠ Warning</p> <p>Do not change this value.</p> </div>
VPC ID		(optional) Enter an existing VPC ID for main account resources to use.
Subnet IDs		(optional) Enter existing subnet IDs for main account resources to use. Enter the IDs as comma-separated values inside the brackets.

6. Choose **Next**.

7. On the **Configure stack options** page, choose **Next**.

8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 15 minutes.

Research Service Workbench on AWS has now successfully deployed to your main account. Copy the following output values from the Outputs section of the stack for the next step:

- S3BucketArtifactsArnOutput
- AccountHandlerLambdaRoleOutput
- ApiLambdaRoleOutput
- StatusHandlerLambdaRoleOutput
- APISGatewayAPIEndpoint
- DataSetsBucketName

Note

An Application Load Balancer was created in your main account with access logging disabled by default. To enable it, see [Enable access logs for your Application Load Balancer](#).

Step 2: Deploy the hosting account

The solution must also be [deployed in a second account](#) known as the hosting account.

Deploy using CDK CLI

Step 1: Configure the stack

1. Clone the [GitHub repository](#) for the solution.
2. Go to `solutions/swb-reference`.
3. Copy `src/config/example.yaml` and rename the copied file within the config folder. Remember the name you assigned this file for this deployment. Some common values that can be used are `dev`, `beta`, and `gamma`.

Example: `<STAGE>.yaml`

4. Open your copied yaml file and uncomment the stage attribute. Provide the correct `<STAGE>` value for the attribute.

Note

The stage value should match the name of your copied file.

5. In the copied `<STAGE>.yaml` file uncomment:

a. `awsRegion`

Determines what AWS Region the solution deploys in. Must be a supported region.

b. `awsRegionShortName`

You can choose a two or three letter abbreviation for this value.

6. Uncomment `rootUserEmailParamStorePath` and provide a name for a Systems Manager parameter that will contain the main account user's email address, e.g. `/rsw/<stage>/rootUser/email`.
7. [Create a Systems Manager parameter](#) in your main account and set the name as the assigned value in `rootUserEmailParamStorePath`. Use the main account user's email address as the parameter.
8. Uncomment `allowedOrigins` and provide a list of URLs allowed to access the solution's API.
Example: `['http://localhost:3000','http://localhost:3002']`
9. Uncomment `cognitoDomain` and provide a globally unique string for the Amazon Cognito domain. This should be an alphanumeric string (hyphens allowed) that does not conflict with any other existing Amazon Cognito domains.
10. Do one of the following:
 - For a default deployment, leave `vpcId` and `albSubnetIds` commented.
 - For a bring-your-own-network (BYON) or custom network deployment, uncomment `vpcId` and `albSubnetIds` and provide your network values.
11. Do one of the following:
 - For an internet-facing ALB instance, uncomment `albInternetFacing` and set the value to `true`.
 - For a non-internet-facing ALB instance, uncomment `albInternetFacing` and set the value to `false`.
12. Uncomment `hostedZoneId` and `domainName` and provide their values from your hosted zone. If you don't have a domain configured, see [Working with hosted zones](#) to create a hosted zone.
13. Uncomment `fieldsToMaskWhenAuditing` and use the default values.
14. Save the stage file.
15. Run `chmod 777 <STAGE>.yaml` to allow the local script to read the file.

Step 2: Launch the stack in the main account

Before starting these steps, ensure you are using the main account's AWS profile in the terminal.

1. In the terminal, go to `solutions/swb-reference` within the cloned repository.
2. Compile the code base:

```
rush update && rush build && rushx compile
```

Note

If you have not already performed the CDK bootstrap, run `STAGE=<STAGE> rushx cdk bootstrap` before continuing.

3. Deploy the CDK:

```
STAGE=<STAGE> rushx cdk-deploy
```

The solution is now deployed in your account.

4. Note the AWS CloudFormation output values. You will need them in future steps.

- a. S3BucketArtifactsArnOutput
- b. AccountHandlerLambdaRoleOutput
- c. ApiLambdaRoleOutput
- d. StatusHandlerLambdaRoleOutput
- e. APIGatewayAPIEndpoint
- f. DataSetsBucketName

5. Run a post-deployment step to setup the Service Catalog portfolio and products:

```
STAGE=<STAGE> rushx run-postDeployment
```

The main account user will receive an email with a temporary password. Note this password for future API authenticated requests.

Note

We employ a [shared responsibility model](#) when you bring your own network components.

Step 3: Deploy the hosting account

The solution must also be [deployed in a second account](#) known as the hosting account.

Deploy the hosting account

Before you deploy the hosting account, you must deploy the solution in the main account:

- [Deploy using CloudFormation template](#)
- [Deploy using CDK CLI](#)

Step 1: Choose a template to deploy the hosting account

Important

You must configure Postman or another API client of your choice for this step. We recommend using Postman and provide instructions to [import the solution collection into Postman](#). After importing the collection, you will need to follow the directions to [obtain an access token](#).

Research Service Workbench hosts compute resources on the hosting account. You can choose between two options to deploy a hosting account:

1. (Default) Onboard the hosting account with a new network.

New network components will be created in the hosting account. You will onboard the hosting account without existing network components such as a VPC or subnet for the solution's resources created in the account.

2. Onboard a hosting account with existing network components (BYON).

Resources created will use existing network components. This is known as bring-your-own-network.

To choose, determine the amount of network flexibility your deployment requires. For more information, see [Onboard a hosting account template](#).

Once you have determined your preferred method:

1. In Postman or your preferred API client, locate the hosting accounts section.
2. Choose **Hosting Account Template URLs**.
3. Deploy the API call. The response will provide URLs for each of the three hosting account deployment options.
4. From your browser, sign into the AWS account for your hosting account.
5. In Postman, follow the link corresponding to your chosen network deployment option.

Step 2: Deploy the CloudFormation stack for the hosting account

After you've opened the URL in Postman, the AWS CloudFormation console opens in your browser. A new stack will have been created with the parameters completed.

1. On the **Quick Create Stack** page, review and confirm the settings.
2. Confirm the acknowledgment that the template will create AWS Identity and Access Management (IAM) resources.
3. Choose **Create stack** to deploy the stack.
4. View the status of the stack in the AWS CloudFormation console under the **Status** column. You should receive a CREATE_COMPLETE status in approximately ten minutes. The hosting account will be ready to onboard.
5. Note the following values from the **Outputs and Parameters** section of the stack.
 - a. EnvMgmtRoleArn
 - b. HostingAccountHandlerRoleArn
 - c. ExternalId

The following input parameters can be seen in the onboard account template. Not all parameters may apply to your chosen template.

Parameter	Default	Description
Namespace		An environment name that will be prefixed to resource names. Make sure this

Parameter	Default	Description
		matches your main account stack name.
Main Account Id		Your Main Account ID where RSW is deployed
External Id		A unique ID used to identify this account
Status Handler Role ARN		The arn of status handler role in the main account
API Handler Role ARN		The arn of API lambda handler role in the main account
Account Handler Role ARN		The arn of account handler role in the main account
VPC CIDR	10.0.0.0/16	Enter the IP range (CIDR notation) for the VPC created by this template
Public Subnet CIDR	10.0.0.0/19	Enter the IP range (CIDR notation) for the Subnet created by this template
Launch Constraint Role Prefix	*	Role name prefix to use when creating a launch constraint role in the on-boarded account
Launch Constraint Policy Prefix	*	Customer managed policy name prefix to use when creating a launch constraint role in the on-boarded account

Parameter	Default	Description
Enable Flow Logs		Enable flow logs on VPCs and Subnets created on this account
VPC ID		The ID of an existing VPC in which all instances will reside
Subnet ID		The ID of an existing Subnet in which all instances will reside

After you've deployed the hosting account stack, explore account resources and APIs available in the solution's [Developer Guide](#).

Monitoring the solution with Service Catalog AppRegistry

The Research Service Workbench on AWS solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both [Service Catalog AppRegistry](#) and [AWS Systems Manager Application Manager](#).

Once installed, Research Service Workbench on AWS will create two AppRegistry applications:

- Main Account

Registers the main account infrastructure stack resources such as API Lambda functions, Amazon S3 buckets, and API Gateway

- Hosting Account AppRegistry Application

Registers the hosting account infrastructure and environment resources created within Research Service Workbench

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution in the context of an application. For example, deployment status, CloudWatch alarms, resource configurations, and operational issues.

For information on how to access your AppRegistry applications, see [Viewing application details](#).

The following figure depicts an example of the application view for the Research Service Workbench on AWS stack in Application Manager.

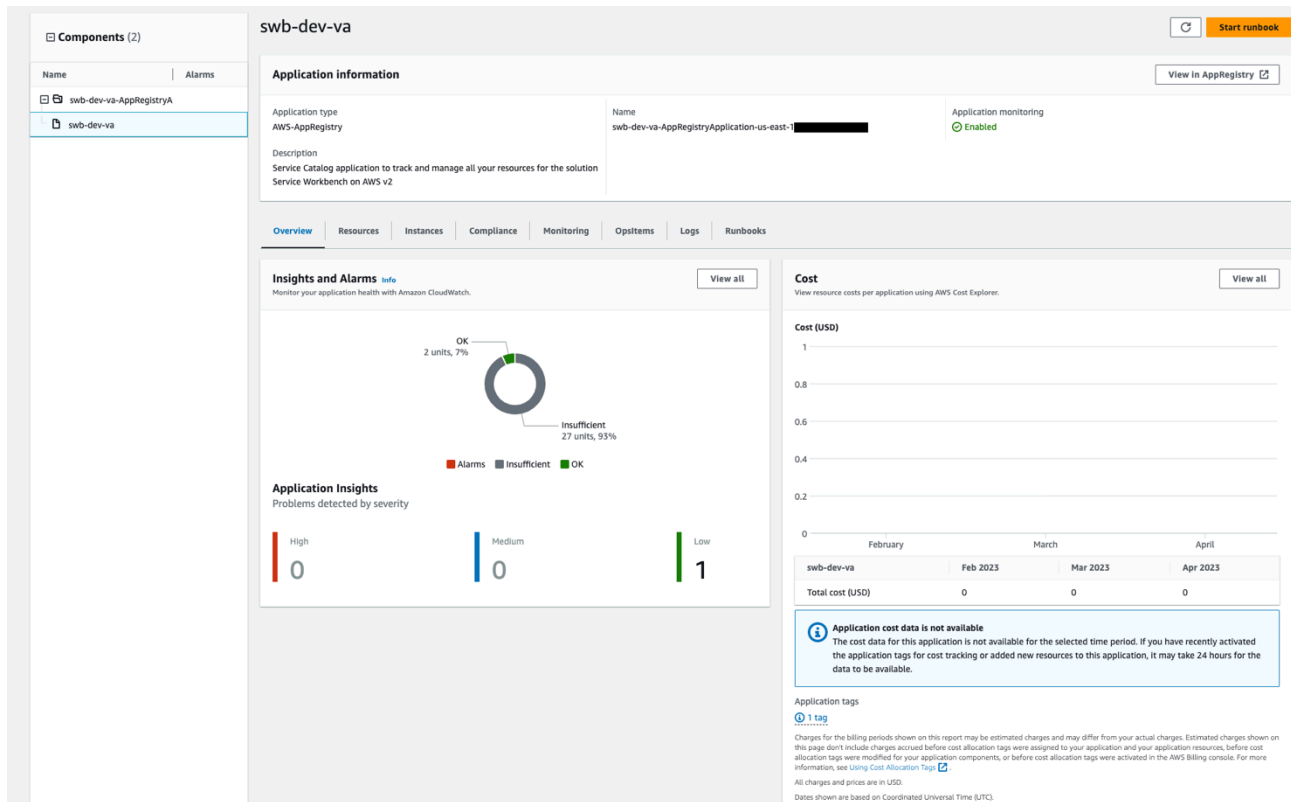


Figure 1: Research Service Workbench on AWS main account infrastructure stack in Application Manager

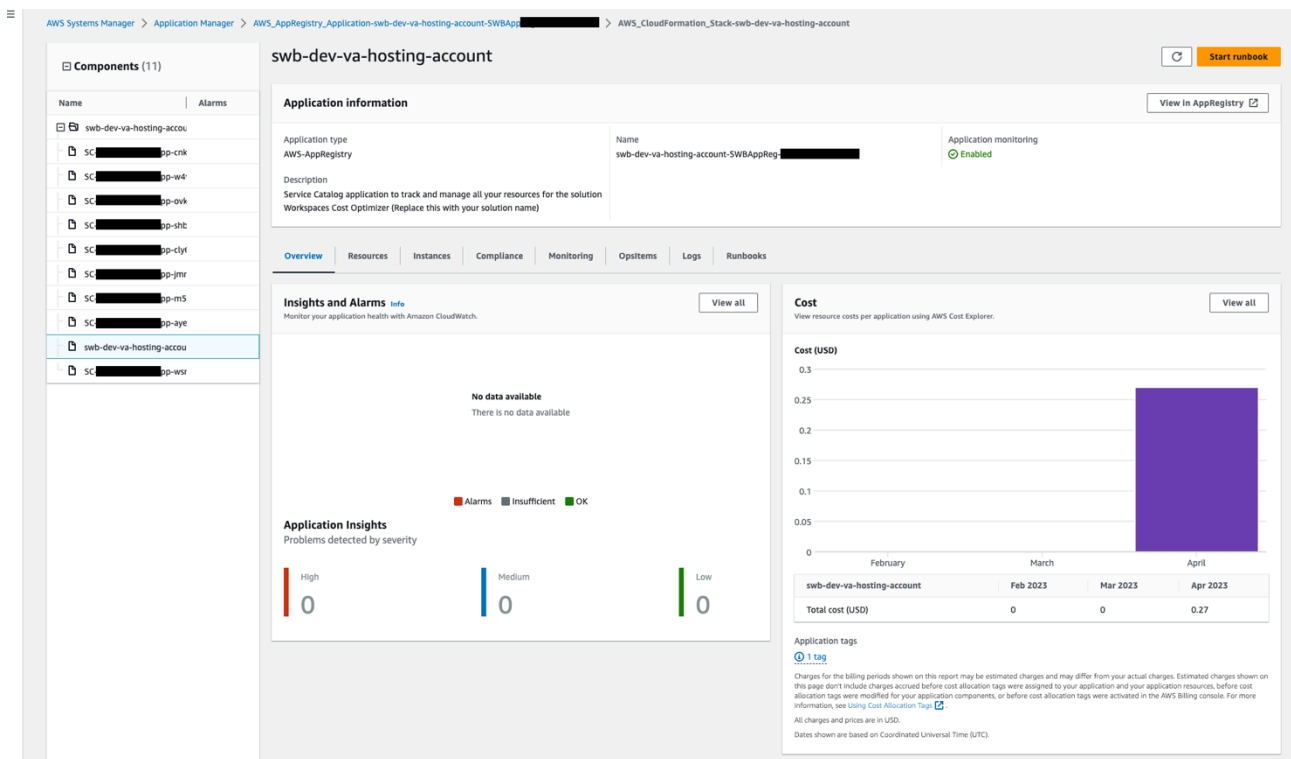


Figure 2: Research Service Workbench on AWS hosting account infrastructure stack in Application Manager

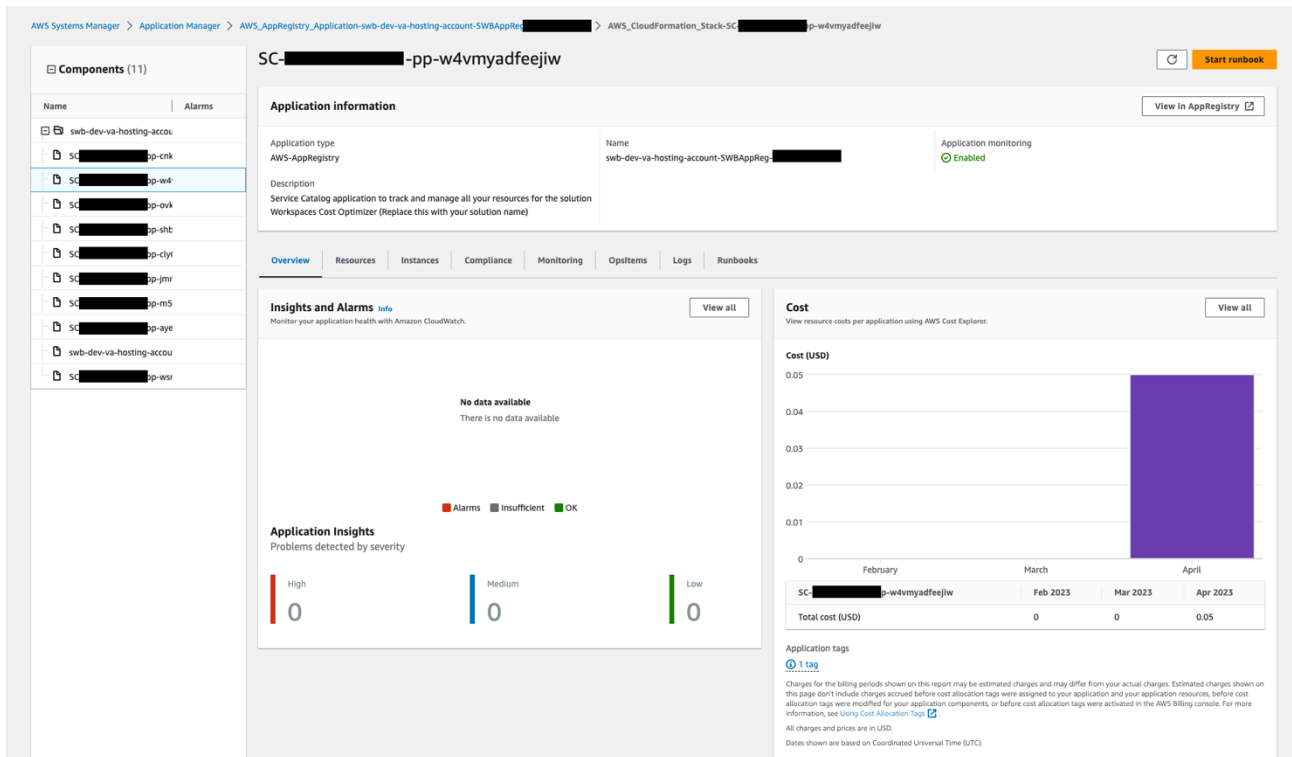


Figure 3: Research Service Workbench on AWS hosting account environment stack in Application Manager

Note

You must activate CloudWatch Application Insights, AWS Cost Explorer, and cost allocation tags associated with this solution. They are not activated by default.

Activate CloudWatch Application Insights

1. Sign in to the [Systems Manager console](#).
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.

5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.

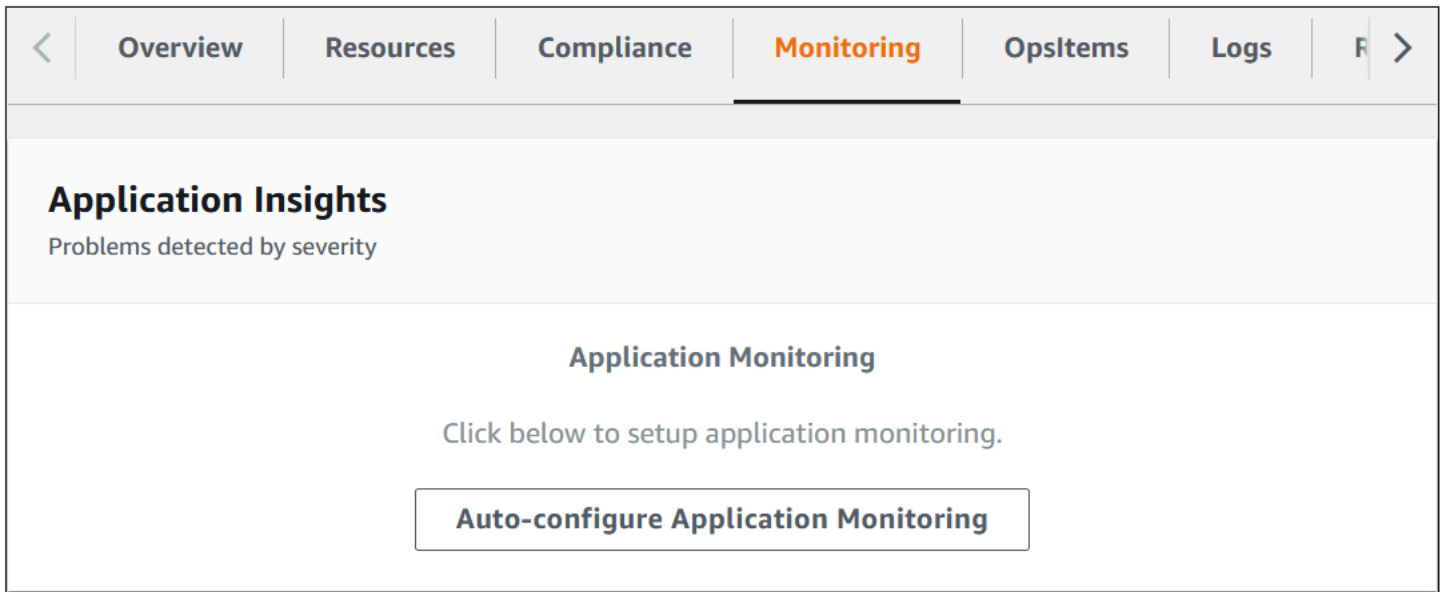


Figure 4: Auto-configure Application Monitoring

Monitoring for your applications is now activated and the following status box appears:

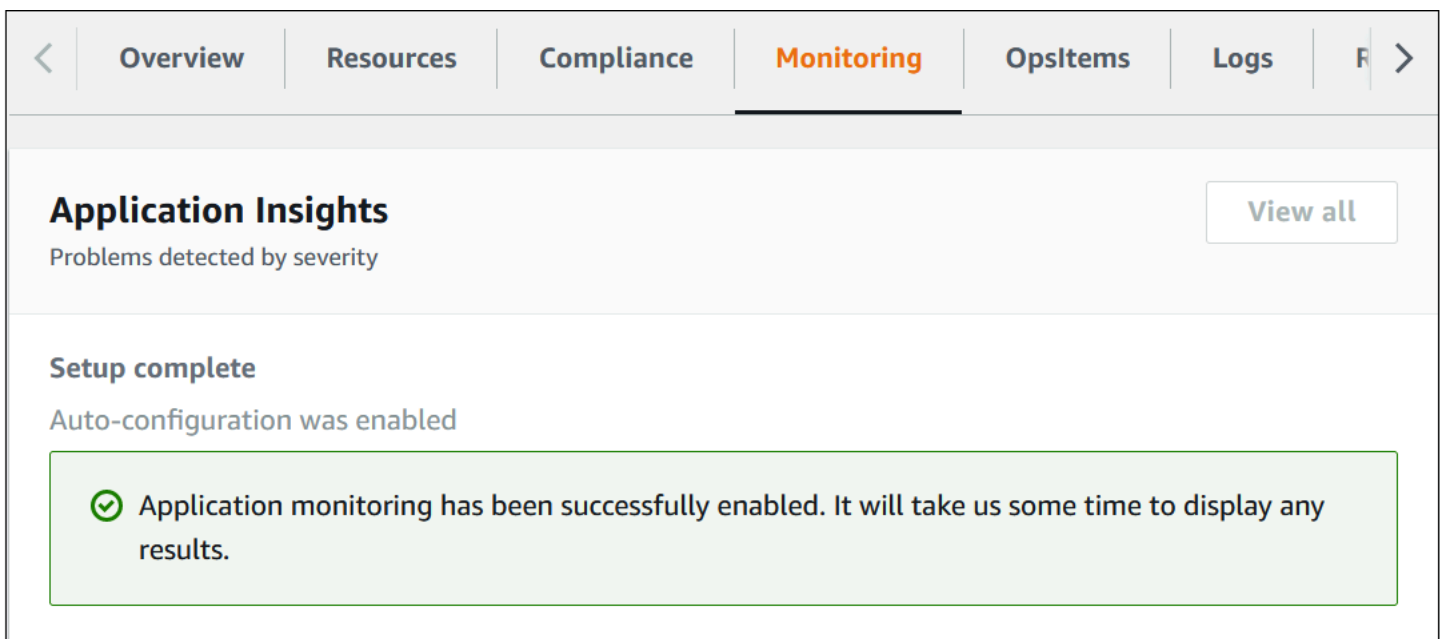


Figure 5: Application monitoring status message

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the [AWS Cost Management console](#).
2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.

The activation process can take up to 24 hours to complete. Once activated, you can open the Cost Explorer user interface to further analyze cost data for the solution.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate the cost allocation tags associated with this solution to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the [AWS Billing and Cost Management and Cost Management console](#).
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the AppManagerCFNStackKey tag, then select the tag from the results shown.
4. Choose **Activate**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Configure AppRegistry application limits

Research Service Workbench on AWS uses AWS App Registry applications to group and add metadata and attributes to created resources.

By using App Registry, Research Service Workbench is able to organize its resources and track their dependencies more efficiently. Every resource created in Research Service Workbench is associated to an App Registry application including all workspaces. App Registry currently has a default limit of 1000 resources per application.

If you are estimating to have more than 999 Workspaces created in your Service Workbench instance a service quota increase will be needed. Follow these steps to request a quota increase for App Registry application resources:

1. Sign in to your **Hosting Account** in the [AWS Management Console](#).
2. Open the [Service Quotas console](#).
3. Under **Service Quotas**, choose **Resources per application**.
4. On the **Resources per application** page, choose **Request Quota Increase** under **Recent quota increase requests**.
5. In **Change quota**, enter your estimated number of workspaces plus 1 (infrastructure resource).
6. Choose **Request**.

Uninstall the solution

You can uninstall the Research Service Workbench on AWS solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the following resources created by this solution. AWS Solutions Implementations do not automatically delete these resources in case you have stored data to retain. If you do not delete these resources, you may continue to incur costs.

Prerequisite: Stop sharing SSM documents

The main and hosting accounts share SSM documents. Before deleting the solution's AWS CloudFormation stack, stop sharing the SSM documents:

- SagemakerNotebookLaunch
- SagemakerNotebookTerminate

Follow these steps to stop sharing SSM documents using the AWS Management Console:

1. Open the AWS Systems Manager [console](#).
2. In the navigation pane, choose **Documents**.
3. In the documents list, choose the document you want to stop sharing.
4. Choose **View details**.
5. On the Permissions tab, verify that you are the document owner. Only a document owner can stop sharing a document.
6. Choose **Edit**.
7. Choose **X** to remove all AWS hosting account IDs.
8. Choose **Save**.

To stop sharing SSM documents using AWS CLI, run the following command:

```
aws ssm modify-document-permission --name <document-name>  
-permission-type Share --account-ids-to-remove 'AWS hosting  
account ID'
```

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, see [What is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name  
<installation-stack-name>
```

Deleting the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 bucket (for deploying in an opt-in Region) if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data.

- dataset bucket: begins with <stack-name>-s3datasets
- artifacts bucket: begins with <stack-name>-s3artifacts
- access log bucket: begins with <stack-name>-s3accesslogs

Follow these steps to delete each Amazon S3 bucket:

1. Sign in to the [Amazon S3 console](#).
2. Choose **Buckets** from the navigation.
3. Select the S3 bucket, then choose **Empty**. You will need to repeat these steps for each bucket.
4. Enter *permanently delete* to confirm, then choose **Empty**.
5. For the dataset bucket, you will need to delete all access points:
 - a. Under **Access Points** section, choose the access point, then choose **Delete**.

- b. Confirm the deletion by entering the access point name. Repeat for all access points.
6. Select the S3 bucket, then choose **Delete**.
7. Enter bucket name to confirm, then choose **Delete bucket**.

Because the S3 buckets created from this solution are version enabled, we recommend using the console instead of AWS CLI to delete them.

Deleting the Amazon DynamoDB tables

The DynamoDB tables created by this solution:

- application table:

<stack-name>

- dynamicAuth table:

begins with <stack-name>-swbStageRegionShortNamedynamicauth

Follow these steps to delete the Service Workbench DynamoDB tables using the AWS Management Console:

1. Sign in to the Amazon DynamoDB [console](#).
2. Go to the **Tables** section.
3. Select the application and dynamicAuth tables.
4. Choose **Delete**.
5. To delete the table using AWS CLI, run the following command:

```
aws dynamodb delete-table --table-name <tablename>
```

Deleting the AWS CloudWatch logs

This solution retains the CloudWatch Logs to prevent accidental data loss if you decide to delete the AWS CloudFormation stack. After uninstalling the solution, you can manually delete these logs if you do not need to retain the data:

- VPC Flow Log: begins with `<stack-name>-SWBVpcVpcFlowLogGroup`
- APIGateway Access Log: begins with `<stack-name>-APIGatewayAccessLogs`

Follow these steps to delete the CloudWatch Logs:

1. Sign in to the Amazon CloudWatch [console](#).
2. Choose **Log Groups** from the navigation.
3. Locate the log groups created by the solution
4. Select one of the log groups.
5. Choose **Actions**.
6. Choose **Delete**.

Repeat the steps until you have deleted all the solution log groups.

To delete the CloudWatch logs using AWS CLI, run the following command:

```
aws logs delete-log-group --log-group-name <value>
```

Deleting the Amazon Cognito user pool

Follow these steps to delete the user pool using the AWS Management Console:

1. Go to the Amazon Cognito [console](#).
2. From the navigation, choose **User Pools**.
3. Select the user pool `[code] swb-userpool-<stage>-<awsRegionShortName>`.
4. Choose **Delete**.
5. Enter the name of the user pool to confirm.

To delete the user pool using AWS CLI, do the following:

1. Find the `user-pool-id` of the user pool `swb-userpool-<stage>-<awsRegionShortName>` using:

```
aws cognito-idp list-user-pools
```

2. Note the resulting Id.
3. Delete the user pool by specifying the `user-pool-id` from the previous Id.

```
aws cognito-idp delete-user-pool --user-pool-id <value>
```

Scheduling the deletion of KMS keys

Warning

Deleting a KMS key can potentially cause unintentional data loss. When a KMS key is deleted, all data that was encrypted under the KMS key is unrecoverable. (The only exception is a multi-Region replica key.) To prevent the use of a KMS key without deleting it, use `DisableKey`.

KMS keys used by S3 buckets and DynamoDB tables are also retained since S3 buckets and DynamoDB tables are retained. After deleting the S3 buckets or DynamoDB tables, the corresponding KMS keys may be deleted. There are 4 keys created by this solution:

- key to S3 datasets bucket
- key to S3 artifacts bucket
- key to DynamicDB application table
- key to DynamicDB DynamicAuth table

The physical ID of the key can be found in the CloudFormation [console](#) under the deleted stack resources.

Because deleting a KMS key may cause unintentional data loss, AWS KMS requires you to set a waiting period of 7 – 30 days. The default waiting period is 30 days.

Follow these steps to schedule deletion of KMS keys using the AWS Management Console:

1. Go to the Amazon KMS [console](#).
2. From the navigation, choose **Customer managed keys**.
3. Select the KMS keys.
4. Under **Key actions**, choose **Schedule key deletion**.
5. Enter the **Waiting period** (in days).
6. Choose **Schedule deletion**.

To delete the KMS key using AWS CLI, run the following command:

```
aws kms
schedule-key-deletion --key-id <value> --pending-window-in-days
<value>
```

Deleting Service Catalog resources

Delete portfolio

Before you can delete a portfolio, you must remove all its products, constraints, and shares:

1. Open a portfolio to display Portfolio details.
2. Note the **Product name** (e.g. sagemakerNotebook) and its **Product ID** as it's going to be needed to delete the product.
3. Choose the tab to remove them.

You can then proceed to delete the portfolio:

4. Go to the Amazon Service Catalog [console](#).
5. In the navigation, choose **Portfolios**.
6. Select the portfolio <stack-name> .

7. Choose **Actions**.
8. Choose **Delete**.
9. In the confirmation window, choose **Delete**.

Since the portfolio created by this solution has associated products, constraints and shared accounts, we recommend using the console instead of AWS CLI to delete it.

Delete product

To delete a product using the Service Catalog console:

1. Go to the Amazon Service Catalog [console](#).
2. Navigate to the **Product list**.
3. Choose the product removed from previous portfolio.
4. Choose **Actions**.
5. Select **Delete product**.
6. Enter *delete* to confirm, then choose **Delete**.

To delete the Service Catalog product using AWS CLI, run the following command using the id value from the previous step's **Product ID**:

```
aws servicecatalog delete-product --id <value>
```

Deleting IAM roles

The IAM roles used by the API Gateway CloudWatch log is retained. To delete this IAM role using the console:

1. Sign in to the AWS Management Console and open the IAM [console](#).
2. In the navigation pane, choose **Roles**.
3. Select the role name beginning with [code] stack-name-APIGatewayAPICloudWatchRole.
4. Choose **Delete**.
5. In the confirmation dialog box, review the last accessed information, which shows when the selected role last accessed an AWS service. This helps you to confirm if the role is

currently active. If you want to proceed, enter the name of the role in the text input field and choose **Delete**.

To delete IAM role using AWS CLI, run the following command:

```
aws iam delete-role --role-name role-name
```

Deleting the AppRegistry in hosting account

Use the AWS Management Console:

1. Disassociate all resources from the Application:
2. Sign in to the [Amazon Service Catalog console](#).
3. In the navigation, choose AppRegistry and then choose Applications.
4. In Applications, open <stack-name> Application.
5. Choose Resource collections.
6. In Resource collections associated to application, select all resources.
7. Choose **Disassociate**.
8. Choose **Ok** to complete the process.
9. Delete Application:
 - a. Locate the <stack-name> Application.
 - b. Choose **Delete**.

To delete AppRegistry using AWS CLI, do the following:

1. Disassociate all resources from the Application:

```
aws servicecatalog-appregistry disassociate-resource --application  
<value> --resource-type CFN_STACK --resource <value>
```

2. Delete Application:

```
aws servicecatalog-appregistry delete-application --application  
<value>
```

Deleting AWS CloudFormation stacks in the hosting account

There are stacks in the hosting account to provision environment resources:

- `swb-<stage>-<awsRegionShortName>-hosting-account` provisions resources necessary to use this hosting account with Research Service Workbench
- Additional stacks to provision environments such as SageMaker Notebook Instances, stack name begins with `SC-<hostingAccountID>`

To delete each stack using the AWS Management Console:

1. Sign in to the [AWS CloudFormation console](#).
2. On the Stacks page, select the stack.
3. Choose **Delete**.

To delete stack using AWS CLI, run the following command:

```
aws cloudformation delete-stack --stack-name <stack-name>
```

Developer guide

Source code

Visit our [GitHub repository](#) to download the source files for this solution and to share your customizations with others.

The Research Service Workbench on AWS templates are generated using the [AWS Cloud Development Kit \(AWS CDK\) \(AWS CDK\)](#). See the [README.md file](#) for additional information.

Contributing to GitHub

To contribute changes to Research Service Workbench on AWS, create a branch from develop and commit changes to your branch. Then, open a pull request on the develop branch of our repo.

Configuration guide

Topics

- [Setup Account Resources](#)
- [Setup EnvironmentTypeConfig](#)
- [Create new datasets](#)
- [Environment management](#)
- [User Management](#)
- [Import solution collection into Postman](#)
- [Obtain Access Token for making authenticated API requests](#)
- [Integration Tests](#)

Setup Account Resources

Onboard hosting account

1. In the **RSW Official** Postman Collection under the **hosting account** folder, choose the **Create Hosting Account** API to onboard a hosting account. Remember to fill in the correct values for your account. Custom values you need to provide will be in chevrons: `<example>`.

2. In the body tab, set `envMgmtRoleArn` parameter to the `EnvMgmtRoleArn` value from [Deploy the Hosting Account](#).
3. In the body tab, set `hostingAccountHandlerRoleArn` parameter to the `HostingAccountHandlerRoleArn` value from [Deploy the Hosting Account step](#).
4. Follow these steps to assign a value to `projectAdmin1UserNameParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file:
 - a. Uncomment `projectAdmin1UserNameParamStorePath` and provide a name for a SSM parameter that will contain a Project Admin user's email address, e.g. `/rsw/<STAGE>/PA/email`.
 - b. Follow instructions in [Create User Step](#) to create a new Project Admin user for the integration tests
 - c. Follow instructions in [to create a SSM Parameter](#) in your main account and set the name as the assigned value in `projectAdmin1UserNameParamStorePath` and the value as the created Project Admin's email.
5. Send a **Create Hosting Account** request.

```
POST {{API_URL}}/aws-accounts
```

```
{
  "name": "<Unique account name>",
  "awsAccountId": "<Hosting Account 12 Digit ID>",
  "envMgmtRoleArn": "<CFN_OUTPUT.EnvMgmtRoleArn>",
  "hostingAccountHandlerRoleArn": "<CFN_OUTPUT.HostingAccountHandlerRoleArn>",
  "externalId": "workbench"
}
```

6. Once the request execute successfully a response with the following format will be displayed:

```
{
  "id": "acc-#####-####-####-####-#####",
  "name": "<Unique account name>",
  "awsAccountId": "<Hosting Account 12 Digit ID>",
  "envMgmtRoleArn": "<CFN_OUTPUT.EnvMgmtRoleArn>",
  "hostingAccountHandlerRoleArn": "<CFN_OUTPUT.HostingAccountHandlerRoleArn>",
  "externalId": "workbench",
  "status": "PENDING",
  "updatedAt": "2023-03-17T13:45:46.195Z",
  "createdAt": "2023-03-07T22:31:40.783Z"
}
```

```
}

```

7. Take note of the id that was returned. We'll need it for the next step. We'll refer to this value as `ACCOUNT_ID`.
8. Wait for account handler to run. It runs once every 5 minutes. You'll know that it's completed when the account status is listed as `CURRENT`.
9. To monitor the account status, in **RSW Official** Postman Collection under **hosting account** folder choose **Get Hosting Account API**.
10. In the params tab set `accountId` parameter to the `ACCOUNT_ID` value from previous step and send request.
11. A response with the status property will be displayed:

```
{
  "id": "acc-#####-####-####-####-#####",
  "name": "<Unique account name>",
  "awsAccountId": "<Hosting Account 12 Digit ID>",
  "status": "PENDING"
}
```

You can also find cloudwatch logs for the account handler in the Main account at `aws/lambda/rsw-<stage>-<awsRegionShortName>-accountHandlerLambda`.

Setup Cost Center

1. In the **RSW Official** Postman Collection under **costCenters** folder choose **Create Cost Center API**.
2. In the body tab set `accountId` parameter to the `ACCOUNT_ID` value from [Onboard hosting account step](#).
3. Send a **Create Cost Center** request

```
POST {{API_URL}}/costCenters/
```

```
{
  "name": "<cost center name>",
  "accountId": "<ACCOUNT_ID>",
  "description": "<cost center description>"
}
```

```
}
```

4. Take note of the id from the **Create Cost Center** response. We'll need it for the next step. We'll refer to this value as `COST_CENTER_ID`.

Setup Project

1. In **RSW Official Postman Collection** under **projects** folder choose **Create project API**.
2. In the body tab set `costCenterId` parameter to the `COST_CENTER_ID` value from [Retrieve Environment Type Id step](#).
3. Send a Create project request

```
POST {{API_URL}}/projects
```

```
{
  "name": "<project name>",
  "description": "<project description>",
  "costCenterId": "<COST_CENTER_ID>",
}
```

In the response take note of the id that was returned. We'll refer to this value as `PROJECT_ID`.

Setup EnvironmentTypeConfig

Retrieve Environment Type Id

1. In **RSW Official** Postman Collection under **envType** folder choose **List envTypes** API and send request. If there aren't any environment types displaying in the response, check whether the post deployment step ran correctly.
2. Running the **List envTypes** request in postman should return a json with the following format:

```
{
  "data": [
    {
      "id": " et-<productId>,<provisioningArtifactId> ",
      "productId": " <productId> ",
      "provisioningArtifactId": " <provisioningArtifactId> ",
      "description": " description ",
    }
  ]
}
```

```

"name": " name ",
"type": " sagemakerNotebook ",
"status": " NOT_APPROVED ",
"createdAt": " 2022-07-21T21:24:57.171Z ",
"updatedAt": " 2022-07-21T21:24:57.171Z ",
"params":
  {
    "DatasetsBucketArn": {
      "Description": " Name of the datasets bucket in the main
account ",
      "Type": " String "
    },
    "EncryptionKeyArn": {
      "Description": " The ARN of the KMS encryption Key used to
encrypt data in the notebook ",
      "Type": " String "
    },
    "AccessFromCIDRBlock": {
      "Default": " 10.0.0.0/19 ",
      "Description": " The CIDR used to access sagemaker
notebook ",
      "Type": " String "
    },
    "VPC": {
      "Description": " VPC for Sagemaker Notebook ",
      "Type": " AWS::EC2::VPC::Id "
    },
    "S3Mounts": {
      "Description": " A JSON array of objects with name, bucket
and prefix properties used to mount data ",
      "Type": " String "
    },
    "Namespace": {
      "Description": " An environment name that will be prefixed
to resource names ",
      "Type": " String "
    },
    "MainAccountId": {
      "Description": " The Main Account ID where application is
deployed ",
      "Type": " String "
    },
    "MainAccountKeyArn": {

```

```

        "Description": " The ARN of main account bucket encryption
key ",
        "Type": " String "
    },
    "IamPolicyDocument": {
        "Description": " The IAM policy to be associated with the
launched workstation ",
        "Type": " String "
    },
    "EnvironmentInstanceFiles": {
        "Description": " An S3 URI (starting with \" s3:// \" )
that specifies the location of files to be copied to the environment instance,
including any bootstrap scripts ",
        "Type": " String "
    },
    "MainAccountRegion": {
        "Description": " The region of application deployment in
main account ",
        "Type": " String "
    },
    "InstanceType": {
        "Default": " ml.t3.xlarge ",
        "Description": " EC2 instance type to launch ",
        "Type": " String "
    },
    "Subnet": {
        "Description": " Subnet for Sagemaker Notebook, from the
VPC selected above ",
        "Type": " AWS::EC2::Subnet::Id "
    },
    "AutoStopIdleTimeInMinutes": {
        "Description": " Number of idle minutes for auto stop to
shutdown the instance (0 to disable auto-stop) ",
        "Type": " Number "
    }
}
]
}

```

In the response, take note of the environment type id you are looking for. The name will have the format <product name>-<provisioning artifact name>, e.g. sagemakerNotebook-v1. We'll need it for the next step. We'll refer to this id value as ENV_TYPE_ID.

Approve Environment Type

1. In **RSW Official** Postman Collection under **envType** folder choose **Update envType** API to change the status of environment type.
2. In the params tab set id parameter to the ENV_TYPE_ID value from the [Retrieve Environment Type Id step](#).
3. Send an **Update envType** request

```
PATCH {{API_URL}}/environmentTypes/:id
```

```
{
  "status": "APPROVED"
}
```

Create Environment Type Config

1. In **RSW Official** Postman Collection under **envTypeConfig** folder choose **Create envTypeConfig** API.
2. In the params tab set envTypeId parameter to the ENV_TYPE_ID value from the [Retrieve Environment Type Id step](#).
3. Send a **Create envTypeConfig** request:

```
POST {{API_URL}}/environmentTypes/:envTypeId/configurations
```

```
{
  "type": "sagemakerNotebook",
  "description": "<description>",
  "name": "<environment type config name>",
  "estimatedCost": "<estimated cost>",
  "params": [
    {
      "key": "IamPolicyDocument",
      "value": "${iamPolicyDocument}"
    }
  ]
}
```

```
    },
    {
      "key": "InstanceType",
      "value": "ml.t3.medium"
    },
    {
      "key": "AutoStopIdleTimeInMinutes",
      "value": "0"
    },
    {
      "key": "CIDR",
      "value": "0.0.0.0/0"
    }
  ]
}
```

4. In the response take note of the id that was returned. We'll refer to this value as ENV_TYPE_CONFIG_ID.

If you would like to launch a sagemaker notebook instance with a different instance type than ml.t3.medium, you can replace that value in the JSON above.

Associate Project to Environment Type Configuration

1. In **RSW Official** Postman Collection under **project** folder choose **Associate project with EnvTypeConfig** API.
2. In the params tab set projectId parameter to the PROJECT_ID value from the [Setup Project step](#).
3. In the params tab set envTypeId parameter to the ENV_TYPE_ID value from the [Retrieve Environment Type Id step](#).
4. In the params tab set envTypeConfigId parameter to the ENV_TYPE_CONFIG_ID value from the [Create Environment Type Config step](#).
5. Send **Associate project with EnvTypeConfig** request.

```
PUT {{API_URL}}/projects/:projectId/environmentTypes/:envTypeId/
configurations/:envTypeConfigId/relationships
```

Create new datasets

1. In the **RSW Official** Postman Collection under **datasets** folder, choose the **Create Internal DataSet** API.
2. In the params tab, set the `projectId` parameter to the `PROJECT_ID` value from the [Setup Project step](#).
3. In the body tab, set the `Region` parameter to the `awsRegion` value from [Setup Config File Step](#).
4. In the body tab, set the `storageName` parameter to the `DataSetsBucketName` value from the [Deploy The Code Step](#).

Note

Only Researchers and Project Admins can access this API, the user calling this API needs to have access permissions to the project assigned in the request, for more information see [Assign Project to User](#).

```
POST {{API_URL}}/projects/:projectId/datasets/
```

```
{
  "name": "<Enter a unique DataSet name>",
  "region": "<awsRegion>",
  "storageName": "<Enter the main account DataSets bucket name>",
  "path": "<Folder name to be created for this in the bucket>",
  "awsAccountId": "<Main account ID>",
  "type": "internal"
}
```

At this point you'll receive a JSON response. That response will have an `id` value. You could use that `id` value in the `datasetIds` array while launching an environment. Once registered a `DataSet` using this API, you could also upload files to its bucket folder directly so they're available at environment boot time.

To mount a dataset to your Amazon SageMaker instance, connect to the instance, open a terminal window, and run:


```
/usr/local/bin/mount_s3.sh
```

Environment management

Launch Sagemaker Notebook Instance

1. In **RSW Official** Postman Collection under **environments** folder choose **Launch Environment API**.
2. In the params tab set `projectId` parameter to the `PROJECT_ID` value from [Setup Project step](#).
3. In the body tab set `envTypeId` parameter to the `ENV_TYPE_ID` value from [Retrieve Environment Type Id step](#).
4. In the body tab set `envTypeConfigId` parameter to the `ENV_TYPE_CONFIG_ID` value from [Create Environment Type Config step](#).

Note

Only Researchers and Project Admins can access this API. The user calling this API and the environment type config in the request need to have access permissions to the project assigned in the request. For more information see [Assign Project to User](#) and [Associate Project to Environment Type Configuration](#).

5. Send **Launch Environment** request.

```
POST {{API_URL}}/projects/:projectId/environments
```

```
{
  "description": "<description>",
  "name": "<environment name>",
  "envTypeId": "<ENV_TYPE_ID>",
  "envTypeConfigId": "<ENV_TYPE_CONFIG_ID>",
  "datasetIds": [],
  "envType": "sagemakerNotebook"
}
```

6. In the response take note of the `id` and `projectId` that were returned. We'll refer to the `id` value as `ENV_ID` and the `projectId` value as `ENV_PROJECT_ID`.

Check Environment Status

1. In **RSW Official** Postman Collection under **environments** folder choose **Get Environment API**.
2. In the params tab set id parameter to the ENV_ID value from [Launch Sagemaker Notebook Instance step](#).
3. In the params tab set projectId parameter to the ENV_PROJECT_ID value from [Launch Sagemaker Notebook Instance step](#).

Note

The user calling this API needs to have access permissions to the project assigned in the request. For more information see [Assign Project to User](#).

4. Send **Get Environment** request.

```
GET {{API_URL}}/projects/:projectId/environments/:id
```

In the response you'll see the status of the environment. PENDING means the environment is being provisioned. COMPLETED means the environment is ready to be used.

Connect to Environment

1. In the **RSW Official** Postman Collection under **environments** folder, choose **Get Connection API**.
2. In the params tab set id parameter to the ENV_ID value from [Launch Sagemaker Notebook Instance step](#).
3. In the params tab set projectId parameter to the ENV_PROJECT_ID value from [Launch Sagemaker Notebook Instance step](#).

Note

Only Researchers and Project Admins can access this API. The user calling this API needs to have access permissions to the project assigned in the request. For more information see [Assign Project to User](#).

4. Send **Get Connection** request.

```
GET {{API_URL}}/projects/:projectId/environments/:id/connections
```

5. In the response you'll find a url. Copy and paste that url into the browser to view your Sagemaker Notebook instance.

Stop an Environment

1. In **RSW Official** Postman Collection under **environments** folder choose **Stop Environment** API.
2. In the params tab set id parameter to the ENV_ID value from [Launch Sagemaker Notebook Instance step](#).
3. In the params tab, set projectId parameter to the ENV_PROJECT_ID value from the [Launch Sagemaker Notebook Instance step](#).

Note

The user calling this API needs to have access permissions to the project assigned in the request. For more information see [Assign Project to User](#).

4. Send **Stop Environment** request.

```
PUT {{API_URL}}/projects/:projectId/environments/:id/stop
```

Start an Environment

1. In **RSW Official** Postman Collection under **environments** folder choose **Start Environment** API.
2. In the params tab, set id parameter to the ENV_ID value from [Launch Sagemaker Notebook Instance step](#).
3. In the params tab, set projectId parameter to the ENV_PROJECT_ID value from [Launch Sagemaker Notebook Instance step](#).

Note

The user calling this API needs to have access permissions to the project assigned in the request, for more information see [Assign Project to User](#).

4. Send **Start Environment** request.

```
PUT {{API_URL}}/projects/:projectId/environments/:id/start
```

Terminate the Environment

1. In **RSW Official** Postman Collection under **environments** folder choose **Terminate Environment** API.
2. In the params tab set id parameter to the ENV_ID value from [Launch Sagemaker Notebook Instance step](#).
3. In the params tab, set projectId parameter to the ENV_PROJECT_ID value from [Launch Sagemaker Notebook Instance step](#).

Note

The user calling this API needs to have access permissions to the project assigned in the request, for more information see [Assign Project to User](#).

4. Send Terminate **Environment** request.

```
PUT {{API_URL}}/projects/:projectId/environments/:id/terminate
```

User Management

Create new admin user

1. You must go to the Cognito console in your AWS Management Console.
2. Under **User pools**, look for and click on rsw-userpool-<stage>-<abbreviation>.
3. Under the **Users tab**, choose **Create user**.
4. Once the user is created, click on the username and under **Group memberships**, choose **Add user to group** to add the user to the ITAdmin group.

Create Users

1. In **RSW Official** Postman Collection under **users** folder choose **Create User** API.

2. Send **Create User** request.

```
POST {{API_URL}}/users
```

```
{
  "firstName": "<first name>",
  "lastName": "<last name>",
  "email": "<email address>"
}
```

3. In the response take note of the id that was returned. We'll refer to this value as **USER_ID**.

Reset User Password

1. Go to the [Amazon Cognito console](#) in your main account. If prompted, enter your AWS credentials.
2. Choose **User Pools**.
3. Choose your RSW user pool with name `rsw-userpool-<STAGE>-<Region>`.
4. Choose the **App integration** tab.
5. Under **App client list** choose RSW app client with name `rsw-client-<STAGE>-<Region>`.
6. Under **Hosted UI** choose **View Hosted UI**.
7. If the user has a temporary password, login with your user credentials and you will be prompted to set a new password.
8. If the user already has a non temporary password follow instructions [here](#) to reset password.

Assign Project to User

Note

Only Researchers and ProjectAdmin require project association.

1. In RSW Official Postman Collection under projects folder choose Add User To Project API.
2. In the params tab set `userId` parameter to the `USER_ID` value from [Create User step](#).
3. In the params tab set `projectId` parameter to the `PROJECT_ID` value from [Setup Project step](#).

4. In the body tab set role parameter to the role the user is going to be assigned for the provided project(ProjectAdmin/Researcher).
5. Send **Add User To Project** request.

POST {{API_URL}}/projects/:projectId/users/:userId/relationships

```
{
  "role": "Researcher"
}
```

Import solution collection into Postman

You need to configure Postman or another API client of your choice to onboard a hosting account. Follow these instructions to import the solution's collection into Postman.

1. Create a [Postman environment](#).
2. Name your environment variables:
 - a. API_URL

The value should be the APIGatewayAPIEndpoint.
 - b. ACCESS_TOKEN

The value should be the accessToken.
 - c. CSRF_COOKIE

The value should be the csrfCookie.
 - d. CSRF_TOKEN

The value should be the csrfToken.
3. Import the solution's [POSTMAN collection](#). For more information, see [Importing data into Postman](#).
4. Assign a value to the terminatedEnvId parameter in the ./integration-tests/config/<STAGE>.yaml file.
 - a. Create a [new Amazon SageMaker environment](#) and wait for COMPLETED status.
 - b. Note the id value from the response.

- c. Use the new environment's id to [stop the environment](#) and wait for STOPPED status.
- d. Use the new environment's id to [terminate the environment](#).
- e. In the `./integration-tests/config` directory, assign the id of the new environment to the `terminatedEnvId` in the `<stage>.yaml` file.

Obtain Access Token for making authenticated API requests

Note

If you are trying to obtain a token for a user still using their temporary password, you must first [reset their password](#).

1. Go to the `rsw-reference/scripts` folder.
2. Run `rushx build` to retrieve all required dependencies.
3. Run `STAGE=<STAGE> node generateCognitoTokens.js <userName> '<password>'`. Choose a user that has been created for your RSW deployment. The quotes around `<password>` is necessary for the script to correctly parse passwords which include symbols.
4. In the console output, use the `accessToken`, `csrfCookie`, and `csrfToken` provided to make authenticated API requests.

Integration Tests

The `rsw-reference` package contains integration tests that run API tests against RSW APIs, they can be configured to run automatically as part of a GitHub workflow or CI/CD pipeline.

Note

Integration tests create resources in the environment they are executed against.

Setup Integration Test Config File

1. In `./integration-tests/config` make a copy of `example.yaml` and name it `<STAGE>.yaml`. Uncomment the attributes and provide the appropriate config values.

2. For envTypeid and envType, follow instructions in [Retrieve environment type id step](#) and choose the Environment Type that integration test will use as default when creating any Environment, copy the values from properties id and type from request and assign id value to envTypeid property and type value to envType property in ./integration-tests/config/<STAGE>.yaml file
3. Follow these steps to assign a value to the envTypeConfigId parameter in ./integration-tests/config/<STAGE>.yaml file
 - a. In **RSW Official** Postman Collection under **envTypeConfig** folder choose **List envTypeConfigs** API.
 - b. In the params tab set envTypeid parameter to the envTypeid value from your ./integration-tests/config/<STAGE>.yaml.
 - c. Send **List envTypeConfigs** request. If there are no environment type configs displayed please follow instructions in [Setup Environment Type Config step](#) to create a new environment type config for selected environment type.
 - d. Choose the Environment Type Config that integration test will use as default when creating any Environment and copy the id value from the request.
 - e. In ./integration-tests/config directory assign value copied to envTypeConfigId property in <STAGE>.yaml file
4. Follow these steps to assign a value to projectId parameter in ./integration-tests/config/<STAGE>.yaml file
 - a. In **RSW Official** Postman Collection under **projects** folder choose **List projects** API.
 - b. Send **List projects** request. If there are no projects displayed please follow instructions in [Setup Project step](#) to create a new project.
 - c. Choose the Project that integration test will use as default when creating any Environment and testing project functionality and copy the id value from the response.
 - d. In ./integration-tests/config directory assign id to projectId in <STAGE>.yaml file
5. Follow these steps to assign a value to terminatedEnvId parameter in ./integration-tests/config/<STAGE>.yaml file
 - a. Follow instructions in [Launch Sagemaker Notebook Environment](#) to create a new Sagemaker environment and wait for COMPLETED status.
 - b. Copy the id value from the response
 - c. Use the id of the new environment to [Stop the environment](#) and wait for STOPPED status.

- d. Use the id of the new environment to [Terminate the environment](#).
 - e. Use new instance information to [Terminate](#)
 - f. In `./integration-tests/config` directory assign the id of the new environment to the `terminatedEnvId` in `<STAGE>.yaml` file
6. Follow these steps to assign a value to `rootUserNameParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
 - a. Uncomment `rootUserNameParamStorePath` and provide a name for a SSM parameter that will contain the main account user's email address, e.g. `/rsw/<STAGE>/rootUser/email`.
 - b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `rootUserNameParamStorePath` and the value as the main account user's email address.
 7. Follow these steps to assign a value to `rootPasswordParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
 - a. Uncomment `rootPasswordParamStorePath` and provide a name for a SSM parameter that will contain the main account user's email address, e.g. `/rsw/<STAGE>/rootUser/password`.
 - b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `rootPasswordParamStorePath` and the value as the main account user's password from [Reset User Password Step](#).
 8. Follow these steps to assign a value to `projectAdmin1UserNameParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
 - a. Uncomment `projectAdmin1UserNameParamStorePath` and provide a name for a SSM parameter that will contain a Project Admin user's email address, e.g. `/rsw/<STAGE>/PA/email`.
 - b. Follow instructions in [Create User Step](#) to create a new Project Admin user for the integration tests
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `projectAdmin1UserNameParamStorePath` and the value as the created Project Admin's email.
 9. Follow these steps to assign a value to `projectAdmin1PasswordParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file

- a. Uncomment `projectAdmin1PasswordParamStorePath` and provide a name for a SSM parameter that will contain the Project Admin's password, e.g. `/rsw/<STAGE>/PA/password`.
 - b. Follow instructions in [Reset User Password Step](#) to assign a password to the Project Admin assigned to `projectAdmin1UserNameParamStorePath`.
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `projectAdmin1PasswordParamStorePath` and the value as the Project Admin's new password.
10. Follow these steps to assign a value to `projectAdmin2UserNameParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file:
- a. Uncomment `projectAdmin2UserNameParamStorePath` and provide a name for a SSM parameter that will contain a second Project Admin user's email address, e.g. `/rsw/<STAGE>/PA2/email`.
 - b. Follow instructions in [Create User Step](#) to create a new Project Admin user for the integration tests
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `projectAdmin2UserNameParamStorePath` and the value as the created second Project Admin's email.
11. Follow these steps to assign a value to `projectAdmin2PasswordParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
- a. Uncomment `projectAdmin2PasswordParamStorePath` and provide a name for a SSM parameter that will contain the second Project Admin's password, e.g. `/rsw/<STAGE>/PA2/password`.
 - b. Follow instructions in [Reset User Password Step](#) to assign a password to the second Project Admin assigned to `projectAdmin2UserNameParamStorePath`.
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `projectAdmin2PasswordParamStorePath` and the value as the second Project Admin's new password.
12. Follow these steps to assign a value to `researcher1UserNameParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file

- a. Uncomment researcher1UserNameParamStorePath and provide a name for a SSM parameter that will contain a Researcher's email address, e.g. /rsw/<STAGE>/Researcher/email.
 - b. Follow instructions in [Create User Step](#) to create a new Researcher user for the integration tests
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in researcher1UserNameParamStorePath and the value as the created Researcher's email.
13. Follow these steps to assign a value to researcher1PasswordParamStorePath parameter in ./integration-tests/config/<STAGE>.yaml file
- a. Uncomment researcher1PasswordParamStorePath and provide a name for a SSM parameter that will contain the Researcher's password, e.g. /rsw/<STAGE>/Researcher/password.
 - b. Follow instructions in [Reset User Password Step](#) to assign a password to the Researcher assigned to researcher1UserNameParamStorePath.
 - c. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in researcher1PasswordParamStorePath and the value as the Researcher's new password.
14. Follow these steps to assign a value to hostAwsAccountIdParamStorePath parameter in ./integration-tests/config/<STAGE>.yaml file
- a. Uncomment researcher1PasswordParamStorePath and provide a name for a SSM parameter that will contain the 12 Digit Hosting Account Id, e.g. /rsw/<STAGE>/accountsTest/awsAccountId.
 - b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in hostAwsAccountIdParamStorePath and the value as the the 12 Digit Hosting Account Id.
15. Follow these steps to assign a value to hostingAccountHandlerRoleArnParamStorePath parameter in ./integration-tests/config/<STAGE>.yaml file
- a. Uncomment hostingAccountHandlerRoleArnParamStorePath and provide a name for a SSM parameter that will contain the hosting account handler role ARN, e.g. /rsw/<STAGE>/accountsTest/hostingAccountHandlerRoleArn.

- b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `hostingAccountHandlerRoleArnParamStorePath` and the value as the `HostingAccountHandlerRoleArn` parameter from [Deploy Hosting Account Step](#).
16. Follow these steps to assign a value to `envMgmtRoleArnParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
 - a. Uncomment `envMgmtRoleArnParamStorePath` and provide a name for a SSM parameter that will contain the hosting account event management role ARN, e.g. `/rsw/<STAGE>/accountsTest/envMgmtRoleArn`.
 - b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `envMgmtRoleArnParamStorePath` and the value as the `EnvMgmtRoleArn` parameter from [Deploy Hosting Account Step](#).
17. Follow these steps to assign a value to `encryptionKeyArnParamStorePath` parameter in `./integration-tests/config/<STAGE>.yaml` file
 - a. Uncomment `encryptionKeyArnParamStorePath` and provide a name for a SSM parameter that will contain the hosting account encryption key ARN, e.g. `/rsw/<STAGE>/accountsTest/encryptionKeyArn`.
 - b. Follow instructions to [create a SSM Parameter](#) in your main account and set the name as the assigned value in `encryptionKeyArnParamStorePath` and the value as the `EncryptionKeyArn` parameter from [Deploy Hosting Account Step](#).
18. Uncomment `defaultHostingAccountId` and in `./integration-tests/config/<STAGE>.yaml` file and assign value `ACCOUNT_ID` from [Onboard Hosting Account Step](#).
19. In this root directory run `STAGE=<STAGE> rushx integration-tests`

Implement Integration tests

To use the framework for calling the RSW API, create a `ClientSession` and then use the `resources` attribute to call the CRUD commands.

Example code for creating new user:

```
const setup: Setup = new Setup();
const adminSession = await setup.getDefaultAdminSession();
const { data } = await adminSession.resources.users.create({
  firstName: '<first name>',
  lastName: '<Last name>',
```

```
    email: `<email address>`  
  });
```

Example code for GET one user:

```
const setup: Setup = new Setup();  
const adminSession = await setup.createAdminSession();  
const userId = 'userId';  
const { data: user } = await adminSession.resources.users.user(userId).get();
```

Example code for GETTING all users:

```
const setup: Setup = new Setup();  
const adminSession = await setup.createAdminSession();  
const { data: response } = await adminSession.resources.users.get();
```

Customization guide

Topics

- [Custom Environment Types](#)
- [Incorporate Existing Network Resources \(Bring Your Own Network\)](#)
- [Enabling AWS WAF](#)
- [Update throttle limits with API Gateway and Lambda reserved concurrency](#)
- [Add deny list using AWS WAF](#)

Custom Environment Types

Research Service Workbench allows users to upload and provision custom AWS compute environments and manage them via RSW APIs. Custom Environment Types are used to define those compute environments. Each custom Environment Type needs an associated CloudFormation template that defines the compute requirements. To learn more about making a custom compute environment via a custom Environment Type, [check out this guide](#).

Incorporate Existing Network Resources (Bring Your Own Network)

Research Service Workbench allows you to use existing network resources when launching your compute environments. In addition to using the standard network that gets created in a non-

custom installation, you can bring your own network (BYON) and onboard a Hosting account with existing network components (like VPC, Subnets). If you plan to bring your own network components, we suggest you follow [Security best practices for your VPC](#).

More information about using existing network resources with Research Service Workbench can be found in the [Deployment Guide](#).

Enabling AWS WAF

Research Service Workbench on AWS allows configuration of AWS Web Application Firewall (WAF) to protect the RSW-created Amazon API Gateway. Please see this documentation for more information on how you can protect your deployment:

- <https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html>
- <https://repost.aws/knowledge-center/waf-apply-rate-limit>

Update throttle limits with API Gateway and Lambda reserved concurrency

Follow the instructions below to update RSW's API Gateway throttle limit and AWS Lambda reserved concurrency limit.

Step 1: Update the API throttle limit

1. Open the file `solutions/rsw-reference/RSWStack.ts` in the [GitHub repository](#).
2. Search for `private _API_LIMIT: number = 200;`. By default, the API limit is set at 200.
3. Update the `_API_LIMIT` to fit the intended use case.

Step 2: Update the stack

Deploy the updated stack by following the directions to [Launch the stack in the main account](#).

Add deny list using AWS WAF

AWS WAF allows setting rate limits and regular rules based on IP addresses. Follow these instructions to deny a set of IP addresses from accessing the solution.

Note

Use the main account to add a deny list.

Step 1: Create web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at <https://console.aws.amazon.com/wafv2/>.
2. In the navigation pane, under **AWS WAF**, choose **Web ACLs**.
3. Choose **Create Web ACL**.
4. For **Name**, enter the name that you will use to identify this web ACL.
5. *(Optional)* For **Description - optional**, enter a longer description for the web ACL.
6. For **CloudWatch metric name**, change the default name if applicable. Follow the console guidance for valid characters. The name cannot contain special characters, white spaces, or metric names reserved for AWS WAF, including All and Default_Action.
7. For **Resource type**, choose **Regional resources**. Select the region that RSW is deployed in.
8. For **Associated AWS resources**, choose **Add AWS resources**.
9. In the **Add AWS resources** dialog box, choose **Application Load Balancer**.
10. Select the Application Load Balancer deployed as part of the solution's CloudFormation stack. The name of the Application Load Balancer can be found by searching for the key `MainAccountLoadBalancerArnOutput` in the RSW Cloudformation stack Outputs.
11. Choose **Next**.
12. In **Add rules and rule groups**, choose **Next**.
13. In **Set rule priority**, choose **Next**.
14. In **Configure metrics**, choose **Next**.
15. In **Review and create web ACL**, choose **Create web ACL**.

Step 2: Create a deny list IP set

1. Sign in to the AWS Management Console and open the AWS WAF console at <https://console.aws.amazon.com/wafv2/>.
2. In the navigation pane, under AWS WAF, choose **IP sets**. and then **Create IP set**.

3. Choose **Create IP set**.
4. For **IP set name**, enter DenyList.
5. For **Region**, choose the region where you deployed the solution.
6. For **IP addresses**, add the IP addresses you would like to deny.
7. Choose **Create IP set**.

Step 3: Add a deny list rule

1. Open the web ACL created previously.
2. Choose the **Rules** tab.
3. From **Add rules**, choose **Add my rules and rule groups**.
4. For **Rule type**, choose **Rule builder**.
5. For **Rule builder**, choose **Rule visual editor**.
6. For **Name**, enter DenyListRule.
7. For **Type**, choose **Regular rule**.
8. For **If a request**, choose **matches the statement**.
9. For **Inspect** under **Statement**, choose **Originate from an IP address in**.
10. For **IP set**, choose **DenyList**.
11. For **Action**, choose **Block**.
12. Choose **Add rule**.

Additional rules can be added to block or limit access to the solution's APIs as required for compliance or user needs. For more information, see [AWS WAF rules](#).

API reference

Topics

- [Actions](#)
- [Data types](#)

Actions

Topics

- [User APIs](#)
- [Project APIs](#)
- [Account APIs](#)
- [Cost Center APIs](#)
- [Environment type APIs](#)
- [Dataset APIs](#)
- [Environment APIs](#)
- [SSH Key APIs](#)
- [Authentication APIs](#)

User APIs

Topics

- [Create user](#)
- [Update user](#)
- [Get user](#)
- [List users](#)
- [List project users by role](#)
- [Add user to project](#)
- [Remove user from project](#)
- [Delete user](#)

Create user

Creates a user in the Cognito User Pool. Only users who are IT Admins can access this API. The route to invoke is POST /users.

Request parameters

There are no request parameters for this API.

Request body

The following body elements are for this specific action:

First Name

The first name of the user to add.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

Last Name

The last name of the user to add.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

Email

The email of the user to add. This will be used as their username to login. This must be unique for each user added.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID of the newly created user.

Type: String

First Name

The first name of the newly created user.

Type: String

Last Name

The last name of the newly created user.

Type: String

Email

The email of the newly created user. This will be used as their username to login. This is unique to this user.

Type: String

Status

1 if the user is active and 0 if the user is inactive.

Type: Number

Valid Values: 0 | 1

Roles

List of the current roles the newly created user is assigned to. On creation, this value will be empty. You can add a user to roles by using the Update User API or Add User to Project API.

Type: Array of strings

Update user

Updates values of a user in the Cognito User Pool. Only users who are IT Admins can access this API. The route to invoke is PATCH /users/:id where :id is the UUID of a created user to update.

Request parameters

There are no request parameters for this API.

Request body

The following body elements are for this specific action.

First Name

The first name to update the user's first name to. Only specify this value if you want to update the First Name of the user.

Type: String

Required: No

Last Name

The last name to update the user's last name to. Only specify this value if you want to update the Last Name of the user.

Type: String

Required: No

Email

The email to update the user's email to. Only specify this value if you want to update the Email of the user. This will be used as their username to login. This must be unique for each user added.

Type: String

Required: No

Status

The status to update the user's status to. Only specify this value if you want to update the Status of the user.

Type: String

Valid Values: ACTIVE | INACTIVE

Required: No

Roles

The roles to add to the user. Only specify this value if you want to add a role to the user.

Type: Array of strings

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID of the updated user.

Type: String

First Name

The first name of updated user.

Type: String

Last Name

The last name of the updated user.

Type: String

Email

The email of the updated user. This will be used as their username to login. This is unique to this user.

Type: String

Status

1 if the user is active and 0 if the user is inactive.

Type: Number

Valid Values: 0 | 1

Roles

List of the current roles the updated user is assigned to.

Type: Array of strings

Get user

Gets a single user. Only IT Admins and Project Admins can access this API. The route to invoke is GET /users/:userId where :userId is the ID of the User to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID which uniquely identifies to dataset.

Type: String

First Name

The first name of the user.

Type: String

Required: Yes

Last Name

The last name of the user.

Type: String

Required: Yes

Email

The email of the user. This is used as their username to login. This must be unique for each user added.

Type: String

Required: Yes

Status

The status of the user. 1 means active and 0 means inactive.

Type: Number

Valid Values: 0| 1

Required: Yes

Roles

The roles assigned to the user. Can be empty, contain "ITAdmin", or contain a combination of project associations as Project Admin or Researcher like "<projectId>#ProjectAdmin" and "<projectId>#Researcher", respectively.

Type: Array of strings

Required: Yes

List users

Lists users from the Cognito User Pool. Only users who are IT Admins or Project Admins can access this API. If this is called by the IT Admin, all users are listed. If this is called by a Project Admin, only users who are in the project owned by the Project Admin are listed. The route to invoke is GET/users.

Request parameters

Page Size

The number of users you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing users to get the next page of results. This is an opaque string.

Type: String

Required: No

There are no request parameters for this API.

Request body

There are no body elements for this API.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of users from the Cognito User Pool.

Type: Array of [User](#) objects

Pagination Token

A pagination token to use to get the next page of users from the Cognito User Pool. If the Pagination Token is not present, there are no more users to list from the Cognito User Pool.

Type: String

List project users by role

Lists users who are a certain role of a project. Only IT Admins or Project Admins assigned to the project can access this API. The route to invoke is GET /project/:projectId/users where :projectId is the ID of the project to list users for.

Request parameters

The following parameters are for this specific action.

Role

The Role type of the user to list

Type: String

Valid Values: ProjectAdmin | Researcher

Required: Yes

Page Size

The number of users you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing users to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this API.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of users from the Cognito User Pool.

Type: Array of [User](#) objects

Pagination Token

A pagination token to use to get the next page of users from the Cognito User Pool. If the Pagination Token is not present, there are no more users to list from the Cognito User Pool.

Type: String

Add user to project

Adds a user to a project as a Project Admin or Researcher. Only IT Admins or Project Admins assigned to the project can access this API. The route to invoke is POST /projects/:projectId/users/:userId/relationships where :projectId is the ID of the project to add the user to and :userId is the ID of the user to add to that project.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Role

The role within the project to add the user to.

Type: String

Valid Values: ProjectAdmin | Researcher

Required: Yes

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Remove user from project

Removes a user from a project. Only IT Admins or Project Admins assigned to the project can access this API. The route to invoke is DELETE /projects/:projectId/users/:userId/relationships where :projectId is the ID of the project to remove the user from and :userId is the ID of the user to remove from that project.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Delete user

Deletes a user from the Cognito User Pool. Only IT Admins can access this API. The route to invoke is DELETE /users/:id/purge where :id is the ID of the user to delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Project APIs

Topics

- [Create project](#)
- [Get project](#)
- [List projects](#)
- [Update project](#)
- [Soft delete project](#)

Create project

Creates a project. Only IT Admins can access this API. The route to invoke is POST /projects.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name of the project to create. This must be unique.

Type: String

Required: Yes

Description

The description of the project to create.

Type: String

Required: Yes

Cost Center ID

The ID of the cost center to link the project to. This cost center must already exist before you create a project in it.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The ID of the newly created project

Type: String

Name

The Name of the newly created project

Type: String

Description

The Description of the newly created project

Type: String

Cost Center ID

The ID of the cost center the newly created project is in.

Type: String

Status

The Status of the newly created project.

Type: String

Valid Values: AVAILABLE | SUSPENDED | DELETED

Created At

The DateTime the newly created project was created at.

Type: String

Updated At

The DateTime the newly created project was updated at. Immediately after creation, this date will be the same as the Created At value.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center of the newly created project is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center of the newly created project is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center of the newly created project is in. This value is likely `workbench`.

Type: String

Account ID

The UUID of the AWS Account that the cost center of the newly created project is in.

Type: String

Get project

Gets the information of a single project. IT Admins can access this API. Users who are Project Admins or Researchers of the project being read can access this API. The route to invoke is GET / projects/:projectId where :projectId is the project being read.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The ID of the read project

Type: String

Name

The Name of the read project

Type: String

Description

The Description of the read project

Type: String

Cost Center ID

The ID of the cost center the read project is in.

Type: String

Status

The Status of the read project.

Type: String

Valid Values: AVAILABLE | SUSPENDED | DELETED

Created At

The DateTime the read project was created at.

Type: String

Updated At

The DateTime the read project was last updated at.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center of the read project is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center of the read project is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center of the read project is in. This value is likely workbench.

Type: String

Account ID

The UUID of the AWS Account that the cost center of the read project is in.

Type: String

List projects

List projects. If this is called by an IT Admin, all projects are listed. If this is called by a user who is Project Admin and/or Researcher in at least one project, all projects that that user is a member of will be listed. The route to invoke is GET /projects.

Request parameters

The following parameters are for this specific action.

Page Size

The number of projects you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing project to get the next page of results. This is an opaque string.

Type: String

Required: No

Filter

This is a complex query parameter of the form `filter[<attributeName>][<comparisonOperator>]` where valid attribute names are `createdAt` (to filter by created at attribute of the projects), `dependency` (to filter by cost center ID), `name` (to filter by name of the projects), and `status` (to filter on the status of the projects) and valid comparison operators are `eq` (equal to value), `lt` (alphanumerically less than value), `lte` (alphanumerically less than or equal to value), `gt` (alphanumerically greater than value), `gte` (alphanumerically greater than or equal to value), `between` (in which case you have to pass two filters with `between` with two different values like `filter[<attributeName>][between][value1]=<string>&filter[<attributeName>][between][value1]=<string>` to filter alphanumeric values between `value1` and `value2` values), and `begins` (begins with the substring of the given value). You may only filter by one attribute at a time. You may combine this with sorting on the same attribute you filtered on.

Type: String

Required: No

Sort

This is a complex query parameter of the form `sort[<attributeName>]` where valid attribute names are `createdAt` (to filter by created at attribute of the projects), `dependency` (to filter by

cost center ID), name (to filter by name of the projects), and status (to filter on the status of the projects). You may only sort by one attribute at a time. You may combine this with filtering on the same attribute you sorted on.

Type: String

Valid Values: asc | desc

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of projects.

Type: Array of [Project](#) objects

Pagination Token

A pagination token to use to get the next page of projects. If the Pagination Token is not present, there are no more projects to list.

Type: String

Update project

Update values of a project. Only IT Admins and users who are Project Admins can access this API. IT Admins can update any project. Users who are Project Admins can update projects they are Project Admins of. The route to invoke is PATCH /projects/:projectId where :projectId is the ID of the project to update.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action. At least one value must be defined and non-empty.

Name

The name to update the project's name to. Only specify this value if you want to update the Name of the project.

Type: String

Constraints: Minimum length of 1 character

Required: No

Description

The description to update the project's description to. Only specify this value if you want to update the Description of the project.

Type: String

Constraints: Minimum length of 1 character

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The ID of the updated project

Type: String

Name

The Name of the updated project

Type: String

Description

The Description of the updated project Type: String

Cost Center ID

The ID of the cost center the updated project is in.

Type: String

Status

The Status of the updated project.

Type: String

Valid Values: AVAILABLE | SUSPENDED | DELETED

Created At

The DateTime the updated project was created at.

Type: String

Updated At

The DateTime the updated project was last updated at.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center of the updated project is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center of the updated project is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center of the updated project is in. This value is likely `workbench`.

Type: String

Account ID

The UUID of the AWS Account that the cost center of the updated project is in.

Type: String

Soft delete project

Performs a soft delete of the specified project. This means the record of the project will remain, but it will be marked as deleted and can no longer be used. Only IT Admins and users who are Project Admins can access this API. IT Admins can soft delete any project. Users who are Project

Admins can soft delete projects they are Project Admins of. The route to invoke is DELETE / projects/:projectId where :projectId is the ID of the project to soft delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Account APIs

Topics

- [Create account](#)
- [Retrieve hosting account template URLs](#)
- [Get account](#)
- [List accounts](#)
- [Update account](#)

Create account

Creates a record of an onboarded AWS Account as a hosting account to be used to host environments within Service Workbench on AWS. Only IT Admins can access this API. The route to invoke is POST /awsAccounts.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name for the onboarded hosting account to be identified as. This value must be unique.

Type: String

Required: Yes

AWS Account ID

Hosting account 12 digit ID. This value must be unique.

Type: String

Required: Yes

Env Mgmt Role ARN

The EnvMgmtRole ARN from the output of the CloudFormation stack of deploying to the hosting account.

Type: String

Required: Yes

Hosting Account Handler Role ARN

The HostingAccountHandlerRole ARN from the output of the CloudFormation stack of deploying to the hosting account.

Type: String

Required: Yes

External ID

The external ID for the onboarded hosting account to be identified with. For the use case of Research the solution, we recommend using the value workbench.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID to identify the onboarded hosting account.

Type: String

Name

The name of the onboarded hosting account.

Type: String

AWS Account ID

The 12 digit ID of the onboarded hosting account.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded hosting account.

Type: String

Error

If there is an error associated with the onboarded hosting account, it is detailed here. Type: An [Error](#) object

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded hosting account.

Type: String

VPC ID

If there is a VPC associated with the onboarded hosting account, it is detailed here.

Type: String

Subnet ID

If there is a subnet associated with the onboarded hosting account, it is detailed here.

Type: String

Encryption Key ARN

If there is an encryption key associated with the onboarded hosting account, it is detailed here.

Type: String

External ID

The external ID of the onboarded hosting account. This value is likely workbench. Type: String

Environment Instance Files

If there are environment instance files associated with the onboarded hosting account, the S3 URL where environment instance files are stored is detailed here.

Type: String

Stack Name

The CloudFormation stack name of the onboarded hosting account.

Type: String

Status

The status of the onboarded hosting account. It is likely in PENDING status immediately after account record creation.

Type: String

Valid Values: CURRENT | NEEDS_UPDATE | NEEDS_ONBOARD | PENDING | ERRORED | UNKNOWN

Created At

The DateTime the newly created account record was created at.

Type: String

Updated At

The DateTime the newly created account record was updated at. Immediately after creation, this date will be the same as the Created At value.

Type: String

Retrieve hosting account template URLs

Retrieves URLs to help with onboarding or updating hosting account CloudFormation stacks. Only IT Admins can access this API. The route to invoke is POST /awsAccountTemplateUrls.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

External ID

The external ID of the onboarded hosting account. This value for Service Workbench hosted accounts is likely `workbench`.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

onboard-account

Contains the `TemplateResponse` for the [default deployment](#) of the solution to use if you want to onboard or update an AWS account using the default networking capacity.

Type: A [TemplateResponse](#) object

onboard-account-tgw

Contains the `TemplateResponse` for the Transit Gateway-enabled deployment of the solution to use if you want to onboard or update an AWS account using the Transit Gateway capacity.

Type: A [TemplateResponse](#) object

onboard-account-byon

Contains the `TemplateResponse` for the Bring-Your-Own-Network-enabled deployment of the solution to use if you want to onboard or update an AWS account using the custom networking capacity.

Type: A [TemplateResponse](#) object

Get account

Gets the information for an onboarded hosting account. Only IT Admins can access this API. The route to invoke is POST `/awsAccounts/:id` where `:id` is the UUID of the hosting account after you created a record for it.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID to identify the onboarded hosting account.

Type: String

Name

The name of the onboarded hosting account.

Type: String

AWS Account ID

The 12 digit ID of the onboarded hosting account.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded hosting account.

Type: String

Error

If there is an error associated with the onboarded hosting account, it is detailed here.

Type: An [Error](#) object

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded hosting account.

Type: String

VPC ID

If there is a VPC associated with the onboarded hosting account, it is detailed here.

Type: String

Subnet ID

If there is a subnet associated with the onboarded hosting account, it is detailed here.

Type: String

Encryption Key ARN

If there is an encryption key associated with the onboarded hosting account, it is detailed here.

Type: String

External ID

The external ID of the onboarded hosting account. This value is likely “workbench”. Type: String

Environment Instance Files

If there are environment instance files associated with the onboarded hosting account, the S3 URL where environment instance files are stored is detailed here

Type: String

Stack Name

The CloudFormation stack name of the onboarded hosting account.

Type: String

Status

The status of the onboarded hosting account.

Type: String

Valid Values: CURRENT | NEEDS_UPDATE | NEEDS_ONBOARD | PENDING | ERRORED | UNKNOWN

Created At

The DateTime the account record was created at.

Type: String

Updated At

The DateTime the account record was last updated at.

Type: String

List accounts

Lists information about onboarded hosting accounts. Only IT Admins can access this API. The route to invoke is GET /awsAccounts.

Request parameters

The following parameters are for this specific action.

Page Size

The number of projects you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing project to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of onboarded hosting accounts.

Type: Array of [Account](#) objects

Pagination Token

A pagination token to use to get the next page of hosting accounts. If the Pagination Token is not present, there are no more hosting account to list.

Type: String

Update account

Updates values of an onboarded hosting account record. Only IT Admins can access this API. The route to invoke is PATCH /awsAccounts/:id where :id is the UUID of the onboarded hosting account to update.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name to update the hosting account's name to. Only specify this value if you want to update the Name of the hosting account. The name of a hosting account must be unique.

Type: String

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID to identify the onboarded hosting account.

Type: String

Name

The name of the onboarded hosting account.

Type: String

AWS Account ID

The 12 digit ID of the onboarded hosting account.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded hosting account.

Type: String

Error

If there is an error associated with the onboarded hosting account, it is detailed here.

Type: An [Error](#) object

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded hosting account. Type: String

VPC ID

If there is a VPC associated with the onboarded hosting account, it is detailed here.

Type: String

Subnet ID

If there is a subnet associated with the onboarded hosting account, it is detailed here.

Type: String

Encryption Key ARN

If there is an encryption key associated with the onboarded hosting account, it is detailed here.

Type: String

External ID

The external ID of the onboarded hosting account. This value is likely “workbench”.

Type: String

Environment Instance Files

If there are environment instance files associated with the onboarded hosting account, the S3 URL where environment instance files are stored is detailed here

Type: String

Stack Name

The CloudFormation stack name of the onboarded hosting account.

Type: String

Status

The status of the onboarded hosting account.

Type: String

Valid Values: CURRENT | NEEDS_UPDATE | NEEDS_ONBOARD | PENDING | ERRORED | UNKNOWN

Created At

The DateTime the account record was created at.

Type: String

Updated At

The DateTime the account record was last updated at.

Type: String

Cost Center APIs

Topics

- [Create cost center](#)
- [Get cost center](#)
- [List cost centers](#)
- [Update cost center](#)
- [Soft delete cost center](#)

Create cost center

Creates a cost center. Only IT Admins can access this API. The route to invoke is POST /costCenters.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name of the cost center to create. This must be unique.

Type: String

Required: Yes

Account ID

The ID of the hosting account to link the new cost center to. This hosting account must already be onboarded and have a record created with status CURRENT before you create a cost center in it.

Type: String

Required: Yes

Description

The description of the cost center to create.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID to identify the newly created cost center.

Type: String

Name

The name of the newly created cost center.

Type: String

Account ID

The UUID of the onboarded hosting account the newly created cost center is in.

Type: String

Description

The description of the newly created cost center.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the newly created cost center is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the newly created cost center is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the newly created cost center is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the newly created cost center is in. This value is likely workbench.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the newly created cost center is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the newly created cost center is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the newly created cost center is in.

Type: String

AWS Account ID

The ID of the AWS Account that the newly created cost center is in.

Type: String

Status

The status of the newly created cost center.

Type: String

Valid Values: AVAILABLE | DELETED

Created At

The DateTime the newly created cost center was created at.

Type: String

Updated At

The DateTime the newly created cost center was updated at. Immediately after creation, this date will be the same as the Created At value.

Type: String

Get cost center

Gets the information of a single cost center. Only IT Admins can access this API. The route to invoke is GET /costCenters/:costCenterId where :costCenterId is the ID of the cost center to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID to identify the cost center.

Type: String

Name

The name of the cost center.

Type: String

Account ID

The UUID of the onboarded hosting account the cost center is in.

Type: String

Description

The description of the cost center.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center is in. This value is likely workbench.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center is in.

Type: String

Status

The status of the cost center.

Type: String

Valid Values: AVAILABLE | DELETED

Created At

The DateTime the cost center was created at.

Type: String

Updated At

The DateTime the cost center was last updated at.

Type: String

List cost centers

List cost centers. Only IT Admins can access this API. The route to invoke is GET /costCenters.

Request parameters

The following parameters are for this specific action.

Page Size

The number of cost centers you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing cost centers to get the next page of results. This is an opaque string.

Type: String

Required: No

Filter

This is a complex query parameter of the form `filter[<attributeName>][<comparisonOperator>]` where the only valid attribute name is name (to filter by name of the cost centers) and valid comparison operators are eq (equal to value), lt (alphanumerically less than value), lte (alphanumerically less than or equal to value), gt (alphanumerically greater than value), gte (alphanumerically greater than or equal to value), between (in which case you have to pass two filters with between with two different values like `filter[<attributeName>][between][value1]=<string>&filter[<attributeName>][between][value1]=<string>` to filter alphanumeric values between value1 and value2 values), and begins (begins with the substring of the given value). You may combine this with sorting on the same attribute you filtered on.

Type: String

Required: No

Sort

This is a complex query parameter of the form `sort[<attributeName>]` where the only valid attribute name is name (to sort by name of the cost centers). You may combine this with filtering on the same attribute you sorted on.

Type: String

Valid Values: asc | desc

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of cost centers.

Type: Array of [Cost Center](#) objects

Pagination Token

A pagination token to use to get the next page of cost centers. If the Pagination Token is not present, there are no more cost centers to list.

Type: String

Update cost center

Updates values of a single cost center. Only IT Admins can access this API. The route to invoke is PATCH/costCenters/:costCenterId where :costCenterId is the ID of the cost center to update.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name to update the cost center's name to. Only specify this value if you want to update the Name of the cost center.

Type: String

Required: No

Description

The description to update the cost center's description to. Only specify this value if you want to update the Description of the cost center.

Type: String

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID to identify the cost center.

Type: String

Name

The name of the cost center.

Type: String

Account ID

The UUID of the onboarded hosting account the cost center is in.

Type: String

Description

The description of the cost center.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center is in. This value is likely workbench.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center is in.

Type: String

Status

The status of the cost center.

Type: String

Valid Values: AVAILABLE | DELETED

Created At

The DateTime the cost center was created at.

Type: String

Updated At

The DateTime the cost center was last updated at.

Type: String

Soft delete cost center

Performs a soft delete of the specified cost center. This means the record of the cost center will remain, but it will be marked as deleted and can no longer be used. Only IT Admins can access this API. The route to invoke is DELETE /costCenters/:costCenterId where :costCenterId is the ID of the cost center to soft delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Environment type APIs

Topics

- [Get environment type](#)

- [List environment types](#)
- [Update environment type](#)
- [Create environment type configuration](#)
- [Get environment type configuration](#)
- [Get environment type configuration for project](#)
- [List environment type configurations](#)
- [List environment type configurations for project](#)
- [List projects for environment type configuration](#)
- [Associate environment type configuration with project](#)
- [Remove environment type configuration from project](#)
- [Update environment type configuration](#)
- [Soft delete environment type configuration](#)

Get environment type

Gets the information for a single environment type. Users who are IT Admins, Project Admins, or Researchers can access this API. The route to invoke is GET /environmentTypes/:id where :id is the ID of the environment type to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The ID of the environment type. This is composed of the product ID and the provisioning artifact ID from Service Catalog.

Type: String

Name

The name of the environment type.

Type: String

Description

The description of the environment type.

Type: String

Status

The status of the environment type.

Type: String

Valid Values: APPROVED | NOT_APPROVED

Type

The type of the environment. This is equivalent to the productName in the Service Catalog Portfolio.

Type: String

Params

Parameters required by the Service Catalog Product CloudFormation template. The keys of these records are the parameter name and the value is a description of the parameter.

Type: Record of [CloudFormation Template Parameter](#) objects

Product ID

The product ID of the environment type.

Type: String

Provisioning Artifact ID

The provisioning artifact ID of the environment type.

Type: String

Created At

The DateTime the environment type was created at.

Type: String

Updated At

The DateTime the environment type was last updated at.

Type: String

List environment types

List environment types. Users who are IT Admins, Project Admins, or Researchers can access this API. The route to invoke is GET /environmentTypes.

Request parameters

The following parameters are for this specific action.

Page Size

The number of environment types you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing environment types to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of environment types.

Type: Array of [Environment Type](#) objects

Pagination Token

A pagination token to use to get the next page of environment types. If the Pagination Token is not present, there are no more environment types to list.

Type: String

Update environment type

Update values of an environment type. Only IT Admins can access this API. The route to invoke is PATCH /environmentTypes/:id where :id is the ID of the environment type to update.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action. At least one value must be defined and non-empty.

Name

The name to update the environment types's name to. Only specify this value if you want to update the Name of the environment type.

Type: String

Required: No

Description

The description to update the environment type's description to. Only specify this value if you want to update the Description of the environment type.

Type: String

Required: No

Status

The status to update the environment type's status to. Only specify this value if you want to update the Status of the environment type.

Type: String

Valid Values: APPROVED | NOT_APPROVED

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The ID of the environment type. This is composed of the product ID and the provisioning artifact ID from Service Catalog.

Type: String

Name

The name of the environment type.

Type: String

Description

The description of the environment type.

Type: String

Status

The status of the environment type.

Type: String

Valid Values: APPROVED | NOT_APPROVED

Type

The type of the environment. This is equivalent to the `productName` in the Service Catalog Portfolio.

Type: String

Params

Parameters required by the Service Catalog Product CloudFormation template. The keys of these records are the parameter name and the value is a description of the parameter.

Type: Record of [CloudFormation Template Parameter](#) objects

Product ID

The product ID of the environment type.

Type: String

Provisioning Artifact ID

The provisioning artifact ID of the environment type.

Type: String

Created At

The DateTime the environment type was created at.

Type: String

Updated At

The DateTime the environment type was last updated at.

Type: String

Create environment type configuration

Creates an environment type configuration. Only IT Admins can access this API. The route to invoke is `POST /environmentTypes/:envTypeId/configurations` where `:envTypeId` is the ID of the environment type to create a configuration for.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name of the environment type configuration to create.

Type: String

Required: Yes

Type

The type of the environment type to create a configuration for.

Type: String

Required: Yes

Description

The description of the environment type configuration to create.

Type: String

Required: Yes

Estimated Cost

The user-supplied estimated cost for the environment type configuration to create.

Type: String

Required: No

Params

The parameters for the environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID of the newly created environment type configuration.

Type: String

Name

The name of the newly created environment type configuration.

Type: String

Type

The type of the newly created environment type configuration.

Type: String

Description

The description of the newly created environment type configuration.

Type: String

Estimated Cost

The user-supplied estimated cost for the newly created environment type configuration.

Type: String

Status

The status of the newly created environment type configuration.

Type: String

Valid Values: AVAILABLE | DELETED

Params

The parameters for the newly created environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was updated at. Immediately after creation, this date will be the same as the Created At value.

Type: String

Get environment type configuration

Gets information for a single environment type configuration. Only IT Admins can access this API. The route to invoke is GET /environmentTypes/:envTypeId/configurations/:configId where :envTypeId is the ID of the environment type of the environment type configuration and :configId is the ID of the environment type configuration to get.

Request parameters

There are no parameters for this specific action.

Request body

Request Body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID of the environment type configuration.

Type: String

Name

The name of the environment type configuration.

Type: String

Type

The type of the environment type configuration.

Type: String

Description

The description of the environment type configuration.

Type: String

Estimated Cost

The user-supplied estimated cost for the environment type configuration.

Type: String

Status

The status of the environment type configuration.

Type: String

Valid Values: AVAILABLE | DELETED

Params

The parameters for the environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was last updated at.

Type: String

Get environment type configuration for project

Gets information for a single environment type configuration within a project. IT Admins and users who are Project Admins or Researchers within the specified project can access this API. The route to invoke is GET /projects/:projId/environmentTypes/:envTypeId/

configurations/:configId where :projId is the project to look for the specified environment type configuration, :envTypeId is the ID of the environment type of the environment type configuration, and :configId is the ID of the environment type configuration to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID of the environment type configuration.

Type: String

Name

The name of the environment type configuration.

Type: String

Type

The type of the environment type configuration.

Type: String

Description

The description of the environment type configuration.

Type: String

Estimated Cost

The user-supplied estimated cost for the environment type configuration.

Type: String

Status

The status of the environment type configuration.

Type: String

Valid Values: AVAILABLE | DELETED

Params

The parameters for the environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was last updated at.

Type: String

List environment type configurations

List environment type configurations. Only IT Admins can access this API. The route to invoke is GET /environmentTypes/:envTypeId/configurations where :envTypeId is the ID of the environment type to get configurations for.

Request parameters

The following parameters are for this specific action.

Page Size

The number of environment type configurations you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing environment type configurations to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of environment type configurations.

Type: Array of [Environment Type Configuration](#) objects

Pagination Token

A pagination token to use to get the next page of environment type configurations. If the Pagination Token is not present, there are no more environment type configurations to list.

Type: String

List environment type configurations for project

List environment type configurations for a specific project. IT Admins and users who are Project Admins or Researchers within the specified project can access this API. The route to invoke is GET / projects/:projectId/environmentTypes/:envTypeId/configurations where :projectId is the ID of the project to list environment type configurations for and :envTypeId is the ID of the environment type to get configurations for.

Request parameters

Request Parameters

The following parameters are for this specific action.

Page Size

The number of environment type configurations you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing environment type configurations to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of environment type configurations.

Type: Array of [Environment Type Configuration](#) objects

Pagination Token

A pagination token to use to get the next page of environment type configurations. If the Pagination Token is not present, there are no more environment type configurations to list.

Type: String

List projects for environment type configuration

List projects associated with a specific environment type configuration. Only IT Admins can access this API. The route to invoke is GET /environmentTypes/:envTypeId/configurations/:envTypeConfigId/projects where :envTypeId is the ID of the environment type of the specific configuration and :envTypeConfigId is the ID of the environment type configuration to get projects for.

Request parameters

The following parameters are for this specific action.

Page Size

The number of projects you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing project to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of projects.

Type: Array of Project objects

Pagination Token

A pagination token to use to get the next page of projects. If the Pagination Token is not present, there are no more projects to list.

Type: String

Associate environment type configuration with project

Associates an environment type configuration with a project so users can launch environments using this environment type configuration within the project. Only IT Admins can access this API. The route to invoke is PUT /projects/:projId/environmentTypes/:etId/configurations/:configId/relationships where :projId is the ID of the project to associate the environment type configuration with, :etId is the ID of the environment type of the specific configuration and :configId is the ID of the environment type configuration to associate with the project.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Remove environment type configuration from project

Removes an environment type configuration from a project so users can not launch environments using this environment type configuration within the project. Only IT Admins can access this API. The route to invoke is DELETE /projects/:projId/environmentTypes/:etId/configurations/:configId/relationships where :projId is the ID of the project to remove the environment type configuration from, :etId is the ID of the environment type of the specific configuration and :configId is the ID of the environment type configuration to remove from the project.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Update environment type configuration

Update values of an environment type configuration. Only IT Admins can access this API. The route to invoke is PATCH /environmentTypes/:envTypeId/configurations/:configId where :envTypeId is the ID of the environment type of the specific configuration and :configId is the environment type configuration to update.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Description

The description to update the environment type configuration's description to. Only specify this value if you want to update the Description of the environment type configuration.

Type: String

Required: No

Estimated Cost

The estimated cost to update the environment type configuration's name to. Only specify this value if you want to update the Estimated Cost of the environment type configuration.

Type: String

Required: No

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID of the environment type configuration.

Type: String

Name

The name of the environment type configuration.

Type: String

Type

The type of the environment type configuration.

Type: String

Description

The description of the environment type configuration.

Type: String

Estimated Cost

The user-supplied estimated cost for the environment type configuration.

Type: String

Status

The status of the environment type configuration.

Type: String

Valid Values: AVAILABLE | DELETED

Params

The parameters for the environment type configuration.

Type: Array of Environment Type Configuration Parameter objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was last updated at.

Type: String

Soft delete environment type configuration

Performs a soft delete of the specified environment type configuration. This means the record of the environment type configuration will remain, but it will be marked as deleted and can no longer be used. Only IT Admins can access this API. The route to invoke is DELETE / environmentTypes/:envTypeId/configurations/:etcdId where :envTypeId is the ID of the environment type of the specific configuration and :etcdId is the ID of the environment type configuration to soft delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Dataset APIs

Topics

- [Create internal dataset](#)
- [Get dataset for project](#)
- [Get dataset](#)
- [Soft delete dataset](#)
- [Get dataset permissions](#)
- [Upload file to dataset](#)
- [Associate dataset with project](#)
- [Remove dataset from project](#)

- [List datasets for project](#)

Create internal dataset

Creates a dataset within a project. Only users who are Project Admins of the specified project can access this API. This dataset will then be managed by the project specified. The route to invoke is POST /projects/:projId/datasets where :projId is the ID of the project to create the dataset in.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name of the dataset to create. This name must be unique.

Type: String

Required: Yes

Region

The AWS region the solution is deployed in.

Type: String

Required: Yes

Storage Name

The main account datasets bucket name. This should be set to the same value of the DataSetsBucketName from the CloudFormation outputs from deploying the solution to the main account.

Type: String

Required: Yes

Path

The name of the folder to be created for this dataset in the main account datasets bucket.

Type: String

Required: Yes

AWS Account ID

The AWS Account ID of the main account where the solution is deployed.

Type: String

Required: Yes

Type

The type of dataset to create. This should be internal.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID which uniquely identifies to dataset.

Type: String

Path

The storage path where the dataset files can be found at the location within the main account dataset bucket.

Type: String

AWS Account ID

If present, this is the AWS Account ID of the main account where the datasets bucket is stored.

Type: String

Storage Type

The storage type of the dataset. This value is likely S3.

Type: String

Region

If present, this is the AWS region of the dataset storage.

Type: String

Name

The name of the dataset.

Type: String

Description

If present, this is the description of the dataset.

Type: String

External Endpoints

If present, this is a list of the ID(s) of any endpoint(s) through which the dataset is accessible.

Type: Array of strings

Storage Name

The name that identifies the storage location such as the URL to an S3 bucket where this dataset is stored.

Type: String

Type

If present, this is the type of the dataset. This value is likely internal.

Type: String

Owner

If present, this is the owner of the dataset.

Type: String

Created At

The DateTime the newly created dataset was created at.

Type: String

Get dataset for project

Gets a single dataset within a project. IT Admins and users who are Project Admins or Researchers in the specified project can access this API. The route to invoke is GET /projects/:projectId/datasets/:datasetId where :projectId is the ID of the project to look for the dataset in and :datasetId is the ID of the dataset to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID which uniquely identifies to dataset.

Type: String

Path

The storage path where the dataset files can be found at the location within the main account dataset bucket.

Type: String

AWS Account ID

If present, this is the AWS Account ID of the main account where the datasets bucket is stored.

Type: String

Storage Type

The storage type of the dataset. This value is likely S3.

Type: String

Region

If present, this is the AWS region of the dataset storage.

Type: String

Name

The name of the dataset.

Type: String

Description

If present, this is the description of the dataset.

Type: String

External Endpoints

If present, this is a list of the ID(s) of any endpoint(s) through which the dataset is accessible.

Type: Array of strings

Storage Name

The name that identifies the storage location such as the URL to an S3 bucket where this dataset is stored.

Type: String

Type

If present, this is the type of the dataset. This value is likely internal.

Type: String

Owner

If present, this is the owner of the dataset.

Type: String

Created At

The DateTime the dataset was created at.

Type: String

Get dataset

Gets a single dataset. Only IT Admins can access this API. The route to invoke is GET / datasets/:datasetId where :datasetId is the ID of the dataset to get.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID which uniquely identifies to dataset.

Type: String

Path

The storage path where the dataset files can be found at the location within the main account dataset bucket.

Type: String

AWS Account ID

If present, this is the AWS Account ID of the main account where the datasets bucket is stored.

Type: String

Storage Type

The storage type of the dataset. This value is likely S3.

Type: String

Region

If present, this is the AWS region of the dataset storage.

Type: String

Name

The name of the dataset.

Type: String

Description

If present, this is the description of the dataset.

Type: String

External Endpoints

If present, this is a list of the ID(s) of any endpoint(s) through which the dataset is accessible.

Type: Array of strings

Storage Name

The name that identifies the storage location such as the URL to an S3 bucket where this dataset is stored.

Type: String

Type

If present, this is the type of the dataset. This value is likely internal.

Type: String

Owner

If present, this is the owner of the dataset.

Type: String

Created At

The DateTime the dataset was created at.

Type: String

Soft delete dataset

Performs a soft delete of the specified dataset owned by the specified project. This means the record of the dataset and the corresponding S3 folder (with its current contents) will remain,

but it will be marked as deleted and can no longer be used within any projects. Only users who are Project Admins of the specified project can access this API. The specified project must be the owning project of the dataset (i.e. the project the dataset was created in). The route to invoke is DELETE /projects/:projectId/datasets/:datasetId where :projectId is the ID of the project that owns the dataset and :datasetId is the ID of the dataset to soft delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Get dataset permissions

Gets the permission information for a dataset. Only users who are Project Admins of the specified project can access this API. The specified project must be the owning project of the dataset (i.e. the project the dataset was created in). The route to invoke is GET /projects/:projectId/datasets/:datasetId/permissions where :projectId is the ID of the project that owns the dataset and :datasetId is the ID of the dataset to get permissions for.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

An object with a list of dataset permissions for the given dataset.

Type: [Dataset Permissions](#) Object

Pagination Token

A pagination token to use to get the next page of dataset permissions. If the Pagination Token is not present, there are no more dataset permissions to list.

Type: String

Upload file to dataset

Gets pre-signed URL(s) to upload file(s) to a dataset. Only users who are Project Admins of the specified project can access this API. The specified project must be the owning project of the dataset (i.e. the project the dataset was created in). The route to invoke is GET /projects/:projectId/datasets/:datasetId/upload-requests where :projectId is the ID of the project that owns the dataset and :datasetId is the ID of the dataset to upload files to.

Request parameters

The following parameters are for this specific action.

File Names

The file name(s) to upload files for to the dataset. This is what the file(s) will be saved as within the dataset. To upload a single file, set this value to that file name. To upload multiple files, set this value to be an array of the file names as strings.

Type: String or array of strings

Required: Yes

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

URLs

A list of the URL(s) for the pre-signed link(s) to upload the file(s)

Type: Array of strings

Associate dataset with project

Associates a dataset with a project. Only users who are a Project Admin of the project that owns the dataset can access this API. The route to invoke is PUT /projects/:projId/datasets/:datasetId/relationships where :projId is the ID of the project to associate the dataset with and :datasetId is the ID of the dataset to create permissions for. Sequential calls will overwrite previous accessLevel values.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Access Level

The access level to grant the project with for this dataset.

Type: String

Valid Values: read-write | read-only

Required: Yes

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Remove dataset from project

Deletes a dataset from a project. Only users who are a Project Admin of the project that owns the dataset can access this API. You cannot delete a dataset from the owning project. The route to invoke is DELETE /projects/:projId/datasets/:datasetId/relationships where :projId is the ID of the project to delete the dataset from and :datasetId is the ID of the dataset to delete permissions for.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

List datasets for project

Lists datasets for a specific project. Users who are Project Admins of or Researchers of the specified project can access this API. This will list datasets owned by or associated to the project. The route to invoke is GET /projects/:projId/datasets where :projId is the ID of the project to get datasets for.

Request parameters

The following parameters are for this specific action.

Page Size

The number of datasets you want returned at a single time. If no value is sent, the default page size is 1000 items.

Type: Number

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing datasets to get the next page of results. This is an opaque string.

Type: String

Required: No

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of datasets.

Type: Array of [Dataset](#) objects

Pagination Token

A pagination token to use to get the next page of datasets. If the Pagination Token is not present, there are no more datasets to list.

Type: String

Environment APIs

Topics

- [List environments for project](#)
- [Create environment](#)
- [Get environment](#)
- [List environments](#)
- [Terminate environment](#)
- [Start environment](#)
- [Stop environment](#)
- [Get environment connection](#)

List environments for project

Lists the environment within a specified project. IT Admins or users who are Project Admins or Researchers of the specified project can access this API. The route to invoke is GET / projects/:projectId/environments where :projectId is the ID of the project to get environments for.

Request parameters

The following parameters are for this specific action.

Page Size

The number of environments you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing environments to get the next page of results. This is an opaque string.

Type: String

Required: No

Sort

The order to display the environments based on creation date

Valid Values: asc | desc

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of environments.

Type: Array of [Environment](#) objects

Pagination Token

A pagination token to use to get the next page of environments. If the Pagination Token is not present, there are no more environments to list.

Type: String

Create environment

Creates an environment within a project. Users who are Project Admins or Researchers within the specified project can access this API. The route to invoke is POST /projects/:projectId/environments where :projectId is the ID of the project to create the environment in.

Request parameters

There are no parameters for this specific action.

Request body

The following body elements are for this specific action.

Name

The name of the environment.

Type: String

Required: Yes

Description

The description of the environment.

Type: String

Required: Yes

Env Type ID

The ID of the environment type to use to create the environment.

Type: String

Required: Yes

Env Type Config ID

The ID of the environment type configuration to use to create the environment.

Type: String

Required: Yes

Datasets IDs

The ID(s), if any, of datasets to mount on the environment. This can be an empty list for mounting no datasets.

Type: Array of strings

Required: No

Env Type

The Type of the environment type to use to create the environment.

Type: String

Required: Yes

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

ID

The UUID of the newly created environment.

Type: String

Instance ID

The ID of the instance backing the newly created environment.

Type: String

Project ID

The ID of the project the newly created environment is in.

Type: String

Name

The name of the newly created environment.

Type: String

Description

The description of the newly created environment.

Type: String

Status

The status of the newly created environment. Environments can be connected to when they are in COMPLETED status.

Type: String

Valid Values: PENDING | COMPLETED | STARTING | STOPPING | STOPPED | TERMINATING | TERMINATED | FAILED | TERMINATING_FAILED | STARTING_FAILED | STOPPING_FAILED

Env Type Config ID

The UUID of the environment type configuration for the newly created environment.

Type: String

Provisioned Product ID

The ID of the Provisioning Product from Service Catalog for the newly created environment.

Type: String

Owner

The owner of the newly created environment.

Type: String

Created At

The DateTime the environment was created at.

Type: String

Updated At

The DateTime the environment was last updated at. Immediately after creation, this value should be the same as the Created At.

Type: String

CIDR

The CIDR range for the newly created environment.

Type: String

Datasets

The Dataset metadata of the newly created environment.

Type: Array of [Dataset Metadata](#) objects

Endpoints

The Endpoint metadata of the newly created environment.

Type: Array of [Endpoint Metadata](#) objects

ETC

The environment type configuration metadata of the newly created environment.

Type: [Environment Type Configuration Metadata](#) object

INID

The Instance metadata of the newly created environment.

Type: [Instance Metadata](#) object

PROJ

The Project metadata of the newly created environment.

Type: [Project Metadata](#) object

Get environment

Gets a single environment. IT Admins, Project Admins, and Researchers can access this API. The route to invoke is GET /projects/:projectId/environments/:environmentId where :projectId is the ID of the project that owns the environment and :environmentID is the ID of the Environment.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

ID

The UUID of the newly created environment.

Type: String

Instance ID

The ID of the instance backing the newly created environment.

Type: String

Project ID

The ID of the project the newly created environment is in.

Type: String

Name

The name of the newly created environment.

Type: String

Description

The description of the newly created environment.

Type: String

Status

The status of the newly created environment. Environments can be connected to when they are in COMPLETED status.

Type: String

Valid Values: PENDING | COMPLETED | STARTING | STOPPING | STOPPED | TERMINATING | TERMINATED | FAILED | TERMINATING_FAILED | STARTING_FAILED | STOPPING_FAILED

Env Type Config ID

The UUID of the environment type configuration for the newly created environment.

Type: String

Provisioned Product ID

The ID of the Provisioning Product from Service Catalog for the newly created environment.

Type: String

Owner

The owner of the newly created environment.

Type: String

Created At

The DateTime the environment was created at.

Type: String

Updated At

The DateTime the environment was last updated at. Immediately after creation, this value should be the same as the Created At.

Type: String

CIDR

The CIDR range for the newly created environment.

Type: String

Datasets

The Dataset metadata of the newly created environment.

Type: Array of [Dataset Metadata](#) objects

Endpoints

The Endpoint metadata of the newly created environment.

Type: Array of [Endpoint Metadata](#) objects

ETC

The environment type configuration metadata of the newly created environment.

Type: [Environment Type Configuration Metadata](#) object

INID

The Instance metadata of the newly created environment.

Type: [Instance Metadata](#) object

PROJ

The Project metadata of the newly created environment.

Type: [Project Metadata](#) object

List environments

Lists all environments Only users who are IT Admins can access this API. The route to invoke is GET /environments.

Request parameters

Page Size

The number of projects you want returned at a single time. If no value is sent, the default page size is 50 items.

Type: Number

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Pagination Token

The token returned in the last response that indicates where to continue listing project to get the next page of results. This is an opaque string.

Type: String

Required: No

Filter

This is a complex query parameter of the form filter[<attributeName>] [<comparisonOperator>] where valid attribute names are createdAt (to filter by created at attribute of the environments), projectId (to filter by project assigned to environments), owner (to filter by the user creating the environment) , name (to filter by name of the environments), and status (to filter on the status of the environments) and valid comparison operators are eq (equal to value), begins (starts with value), lt (alphanumerically less than value), lte (alphanumerically less than or equal to value), gt (alphanumerically greater than value), gte (alphanumerically greater than or equal to value), between (in which case you

have to pass two filters with between with two different values like filter[<attributeName>][between][value1]=<string>&filter[<attributeName>][between][value1]=<string> to filter alphanumeric values between value1 and value2 values), and begins (begins with the substring of the given value). You may only filter by one attribute at a time. You may combine this with sorting on the same attribute you filtered on.

Type: String

Required: No

Sort

This is a complex query parameter of the form sort[<attributeName>] where valid attribute names are createdAt (to sort by created at attribute of the environments), projectId (to sort by project assigned to environments), owner (to sort by the user creating the environment) , name (to sort by name of the environments), and status (to sort by the status of the environments). You may only sort by one attribute at a time. You may combine this with filtering on the same attribute you sorted on.

Type: String

Valid Values: asc | desc

Required: No

Request body

There are no body elements for this API.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Data

A list of environments.

Type: Array of [Environment](#) objects

Pagination Token

A pagination token to use to get the next page of environments. If the Pagination Token is not present, there are no more environments to list.

Type: String

Terminate environment

Terminates an environment. An environment must be in a STOPPED state to be terminated. IT Admins, Project Admins, and Researchers can access this API. The route to invoke is PUT / projects/:projectId/environments/:envId/terminate .

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

No elements are returned by this API. A 204 HTTP Status Code indicates success.

Start environment

Start an environment. IT Admins, Project Admins, and Researchers can access this API. The route to invoke is PUT /projects/:projectId/environments/:envId/start .

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

No elements are returned by this API. A 204 HTTP Status Code indicates success.

Stop environment

Stop an environment. An environment must be in the COMPLETED state to stop it. IT Admins, Project Admins, and Researchers can access this API. The route to invoke is PUT / projects/:projectId/environments/:envId/stop .

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

No elements are returned by this API. A 204 HTTP Status Code indicates success.

Get environment connection

Gets the connection information for the environment. Project Admins and Researchers can access this API. The route to invoke is GET /projects/:projectId/environments/:id/connections.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

AuthCredResponse

An object containing the url for the connection

Type: Object

InstructionResponse

The instructions on how to use the connection URL.

Type: String

SSH Key APIs

Topics

- [Create SSH key](#)
- [List user SSH keys for project](#)
- [Delete SSH key](#)
- [Send SSH key public key](#)

Create SSH key

Creates an SSH Key for the requesting user in the specified project. Users who are Project Admins or Researchers in the specified project can access this API. The route to invoke is POST / projects/:projectId/sshKeys where :projectId is the ID of the project to create the SSH Key in

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 201 HTTP Status Code indicates success.

Project ID

The ID of the project that the newly create SSH Key is in.

Type: String

Private Key

The private key contents of the key pair of the newly created SSH Key. Save the contents of this value as a .pem file to be able to use the private key to connect to environments. This value can not be retrieved after this moment.

Type: String

SSH Key ID

The ID of the newly created SSH Key. This corresponds to the Name of the EC2 Key Pair backing the key.

Type: String

Owner

The UUID of the user who created and owns the newly created SSH Key.

Type: String

List user SSH keys for project

Lists the SSH Keys within a specific project for a user. Currently, there should only ever be at most one key per user per project within the solution. Users who are Project Admins of or Researchers in the specified project can access this API. The route to invoke is GET /projects/:projectId/sshKeys where :projectId is the ID of the project to get SSH Keys for.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

SSH Keys

An array of SSH Keys for the requesting user in the project. Type: Array of [SSH Key](#) objects

Delete SSH key

Deletes a SSH Key owned by a user within a project. This is a hard delete and will remove any record of the SSH Key existing. The route to invoke is DELETE /projects/:projectId/sshKeys/:sshKeyId/purge where :projectId is the ID of the project the SSH Key is in and :sshKeyId is the ID of the SSH Key to delete.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

There are no elements returned by the API. A 204 HTTP Status Code indicates success.

Send SSH key public key

Sends the SSH Key Public Key to an environment so a user can connect to it. User who are Project Admins of or Researchers in the project the environment being connected to is in can access this API. The route to invoke is GET /projects/:projectId/environments/:environmentId/sshKeys where :projectId is the ID of the project in which the environment belongs and :environmentId is the ID of the environment to connect to.

Request parameters

There are no parameters for this specific action.

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

Public DNS Name

The public DNS name of the environment to use to connect to.

Type: String

Public IP

The public IP of the environment to use to connect to.

Type: String

Private DNS Name

The private DNS name of the environment to use to connect to.

Type: String

Private IP

The private IP of the environment to use to connect to.

Type: String

Authentication APIs

Topics

- [Get login URL](#)
- [Get tokens from authorization code](#)
- [Get logout URL](#)
- [Refresh token](#)
- [Check logged in user](#)

Get login URL

Used to get the url to the authentication hosted UI. The `stateVerifier` and `codeChallenge` request query parameters are temporary values passed in by the client. The client will replace these values later in order to keep them a client secret. The route to invoke is GET /login .

Request parameters

stateVerifier

Holds a temporary state value that will be passed to the client. The client will replace this value later in order to keep them a client secret.

Required: Yes

codeChallenge

Holds a temporary state value that will be passed to the client. The client will replace this value later in order to keep them a client secret.

Required: Yes

Request body

There are no body elements for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

signInUrl

The URL to use for signing in.

Type: String

csrfToken

A security token used during the login process.

Type: String

Get tokens from authorization code

Used to authenticate a user from its access token. The route to invoke is POST /token.

Request parameters

There are no parameters for this specific action.

Request body

code

The authorization code

Required: Yes

codeVerifier

A PKCE code verifier value

Required: Yes

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

access_token Cookie

The access token used for authorizing requests.

Type: String

refresh_token Cookie

The refresh token used to get a new access token once the current access token has expired.

Type: String

idToken

The authentication token used for authenticating requests.

Type: String

Get logout URL

Used to get the URL that will log out the user. The route to invoke is POST /logout.

Request parameters

There are no parameters for this specific action.

Request body

There are no body for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

logoutUrl

The URL used to log the user out

Type: String

Refresh token

Used to refresh the user's authenticated id token stored in the refresh_token cookie header. The route to invoke is GET/refresh .

Request parameters

There are no parameters for this specific action.

Request body

There are no body for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

access_token Cookie

The access token used for authorizing requests.

Type: String

idToken

The authentication token used for authenticating requests.

Type: String

Check logged in user

Used to check if there is a logged in user. If there is valid refresh_token cookie present, the function will set a new access_token cookie and return a new idToken as well as the logged in status in the response body. The route to invoke is GET /loggedIn .

Request parameters

There are no parameters for this specific action.

Request body

There are no body for this specific action.

Response elements

The following elements are returned by the API. A 200 HTTP Status Code indicates success.

access_token Cookie

The access token used for authorizing requests.

Type: String

idToken

The authentication token used for authenticating requests.

Type: String

loggedIn

Whether the user is logged in or not.

Type: Boolean

csrfToken

A new CSRF token

Type: Optional String

Data types

Topics

- [User](#)
- [Project](#)
- [Error](#)
- [Template response](#)
- [Account](#)
- [Cost Center](#)
- [Environment type](#)
- [Environment type configuration parameter](#)
- [Environment type configuration](#)
- [Dataset permissions](#)
- [Permission](#)
- [Dataset](#)
- [SSH key](#)
- [Environment](#)
- [Dataset metadata](#)

- [Endpoint metadata](#)
- [Environment type configuration metadata](#)
- [Instance metadata](#)
- [Project metadata](#)

User

Describes a user within the solution.

Contents

First Name

The first name of the user.

Type: String

Required: Yes

Last Name

The last name of the user.

Type: String

Required: Yes

Email

The email of the user. This is used as their username to login. This must be unique for each user added.

Type: String

Required: Yes

Status

The status of the user. 1 means active and 0 means inactive.

Type: Number

Valid Values: 0 | 1

Required: Yes

Roles

The roles assigned to the user. Can be empty, contain "ITAdmin", or contain a combination of project associations as Project Admin or Researcher like <projectId>#ProjectAdmin and <projectId>#Researcher, respectively.

Type: Array of strings

Required: Yes

Project

Describes a project within the solution

Contents

ID

The ID of the project

Type: String

Name

The Name of the project

Type: String

Description

The Description of the project

Type: String

Cost Center ID

The ID of the cost center the project is in.

Type: String

Status

The Status of the project.

Type: String

Valid Values: AVAILABLE | SUSPENDED | DELETED

Created At

The DateTime the project was created at.

Type: String

Updated At

The DateTime the project was last updated at.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center of the project is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center of the project is in.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center of the project is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center of the project is in.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center of the project is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center of the project is in.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center of the project is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center of the project is in. This value is likely workbench.

Type: String

Account ID

The UUID of the AWS Account that the cost center of the project is in.

Type: String

Error

Details an error from an onboarded hosting account in Service Workbench on AWS.

Contents

Type

The type of the error.

Type: String

Value

The message of the error.

Type: String

Template response

Details a template response for onboarding or updating a hosting account in the solution.

Contents

Create URL

The URL to use if you are onboarding an AWS account as a hosting account for the first time.

Type: String

Required: Yes

Update URL

The URL to use if you are updating an already-onboarded AWS hosting account.

Type: String

Required: Yes

Account

Details an onboarded hosting account within Service Workbench on AWS.

Contents

ID

The UUID to identify the onboarded hosting account.

Type: String

Required: Yes

Name

The name of the onboarded hosting account.

Type: String

Required: Yes

AWS Account ID

The 12 digit ID of the onboarded hosting account.

Type: String

Required: Yes

Env Mgmt Role ARN

The environment management role ARN of the onboarded hosting account.

Type: String

Required: Yes

Error

If there is an error associated with the onboarded hosting account, it is detailed here.

Type: An [Error](#) object

Required: No

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded hosting account.

Type: String

Required: Yes

VPC ID

If there is a VPC associated with the onboarded hosting account, it is detailed here.

Type: String

Required: No

Subnet ID

If there is a subnet associated with the onboarded hosting account, it is detailed here.

Type: String

Required: No

Encryption Key ARN

If there is an encryption key associated with the onboarded hosting account, it is detailed here.

Type: String

Required: No

External ID

The external ID of the onboarded hosting account. This value is likely "workbench".

Type: String

Required: Yes

Environment Instance Files

If there are environment instance files associated with the onboarded hosting account, the S3 URL where environment instance files are stored is detailed here

Type: String

Required: No

Stack Name

The CloudFormation stack name of the onboarded hosting account.

Type: String

Required: Yes

Status

The status of the onboarded hosting account. If the status is CURRENT, the account is ready to be used to create computing resources in. If the status is NEEDS_UPDATE, the onboarding stack needs to be updated with the most up-to-date onboarding template before the account can be used to create computing resources. If the status is NEEDS_ONBOARD, the account needs to be onboarded as a hosting account. If the status is PENDING, the account is between states and waiting for the account handler lambda to update the status after an update. If the status is ERRORED, please inspect the Error attribute of this object to determine the issue. If the status is UNKNOWN, something went wrong.

Type: String

Valid Values: CURRENT | NEEDS_UPDATE | NEEDS_ONBOARD | PENDING | ERRORED | UNKNOWN

Required: Yes

Created At

The DateTime the account record was created at.

Type: String

Required: Yes

Updated At

The DateTime the account record was last updated at.

Type: String

Required: Yes

Cost Center

Details a cost center within the solution.

Contents

ID

The UUID to identify the cost center.

Type: String

Required: Yes

Name

The name of the cost center.

Type: String

Required: Yes

Account ID

The UUID of the onboarded hosting account the cost center is in.

Type: String

Required: Yes

Description

The description of the cost center.

Type: String

Required: Yes

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

External ID

The external ID of the onboarded AWS hosting account that the cost center is in. This value is likely workbench.

Type: String

Required: Yes

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center is in.

Type: String

Required: Yes

AWS Account ID

The ID of the AWS Account that the cost center is in.

Type: String

Required: Yes

Status

The status of the cost center.

Type: String

Valid Values: AVAILABLE | DELETED

Required: Yes

Created At

The DateTime the cost center was created at.

Type: String

Required: Yes

Updated At

The DateTime the cost center was last updated at.

Type: String

Required: Yes

Environment type

Details an environment type within the solution.

Contents**ID**

The ID of the environment type. This is composed of the product ID and the provisioning artifact ID from Service Catalog.

Type: String

Name

The name of the environment type.

Type: String

Description

The description of the environment type.

Type: String

Status

The status of the environment type.

Type: String

Valid Values: APPROVED | NOT_APPROVED

Type

The type of the environment. This is equivalent to the productName in the Service Catalog Portfolio.

Type: String

Params

Parameters required by the Service Catalog Product CloudFormation template. The keys of these records are the parameter name and the value is a description of the parameter.

Type: Record of [CloudFormation Template Parameter](#) objects

Product ID

The product ID of the environment type.

Type: String

Provisioning Artifact ID

The provisioning artifact ID of the environment type.

Type: String

Created At

The DateTime the environment type was created at.

Type: String

Updated At

The DateTime the environment type was last updated at.

Type: String

Environment type configuration parameter

Details an environment type configuration parameter within the solution.

Contents

Key

The key of the parameter.

Type: String

Required: Yes

Value

The value of the parameter. If this is `SYSTEM_SELECTED`, the API pulls this value. If this is `USER_SELECTED`, the researcher will select an option from a list of choices and the value is not set at the time of environment type configuration creation. If the value is `USER_SELECTED`, you must provide a value for Allowed Values for this parameter. If the value is a string, that is the preset value of the parameter.

Type: String

Required: Yes

Environment type configuration

Details an environment type configuration within Service Workbench on AWS.

Contents

ID

The UUID of the environment type configuration.

Type: String

Name

The name of the environment type configuration.

Type: String

Type

The type of the environment type configuration.

Type: String

Description

The description of the environment type configuration.

Type: String

Estimated Cost

The user-supplied estimated cost for the environment type configuration.

Type: String

Status

The status of the environment type configuration. AVAILABLE means this environment type configuration can be used to create new environments. DELETED means this environment type configuration can not be used to create new environments.

Type: String

Valid Values: AVAILABLE | DELETED

Default Value: AVAILABLE

Params

The parameters for the environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was last updated at.

Type: String

Dataset permissions

Details a dataset permissions for a dataset within Service Workbench on AWS.

Contents

Dataset ID

The ID of the dataset.

Type: String

Required: Yes

Permissions

The permissions of the dataset.

Type: Array of [Permission](#) Objects

Required: Yes

Permission

Details a permission within the solution.

Contents

Identity Type

The identity type of the permission.

Type: String

Valid Values: USER | GROUP

Required: Yes

Identity

The identity of the permission. This is either a user UUID (if Identity Type is USER) or a group ID (if Identity Type is GROUP).

Type: String

Required: Yes

Access Level

The access level of the permission.

Type: String

Valid Values: read-only | read-write

Required: Yes

Data

A list of projects.

Type: Array of [Project](#) objects

Pagination Token

A pagination token to use to get the next page of projects. If the Pagination Token is not present, there are no more projects to list.

Type: String

Dataset

Details a dataset within the solution.

Contents

ID

The UUID which uniquely identifies to dataset.

Type: String

Required: Yes

Path

The storage path where the dataset files can be found at the location within the main account dataset bucket.

Type: String

Required: Yes

AWS Account ID

If present, this is the AWS Account ID of the main account where the datasets bucket is stored.

Type: String

Required: No

Storage Type

The storage type of the dataset. This value is likely S3.

Type: String

Required: Yes

Region

If present, this is the AWS region of the dataset storage.

Type: String

Required: No

Name

The name of the dataset.

Type: String

Required: Yes

Description

If present, this is the description of the dataset.

Type: String

Required: No

External Endpoints

If present, this is a list of the ID(s) of any endpoint(s) through which the dataset is accessible.

Type: Array of strings

Required: No

Storage Name

The name that identifies the storage location such as the URL to an S3 bucket where this dataset is stored.

Type: String

Required: Yes

Type

If present, this is the type of the dataset. This value is likely internal.

Type: String

Required: No

Owner

If present, this is the owner of the dataset.

Type: String

Required: No

Created At

The DateTime the dataset was created at.

Type: String

Required: Yes

SSH key

Details an SSH Key within the solution.

Contents

Project ID

The ID of the project the SSH Key is in.

Type: String

Required: Yes

Public Key

The public key contents of the key pair of the SSH Key.

Type: String

Required: Yes

SSH Key ID

The ID of the SSH Key. This corresponds to the Name of the EC2 Key Pair backing the key.

Type: String

Required: Yes

Owner

The UUID of the user who created and owns the SSH Key.

Type: String

Required: Yes

Create Time

The DateTime the SSH Key was created at.

Type: String

Required: Yes

Environment

Details an environment within the solution.

Contents

ID

The UUID of the environment.

Type: String

Instance ID

The ID of the instance backing the environment.

Type: String

Project ID

The ID of the project the environment is in.

Type: String

Name

The name of the environment.

Type: String

Description

The description of the environment.

Type: String

Status

The status of the environment. Environments can be connected to when they are in COMPLETED status.

Type: String

Valid Values: PENDING | COMPLETED | STARTING | STOPPING | STOPPED | TERMINATING | TERMINATED | FAILED | TERMINATING_FAILED | STARTING_FAILED | STOPPING_FAILED

Env Type Config ID

The UUID of the environment type configuration for the environment.

Type: String

Provisioned Product ID

The ID of the Provisioning Product from Service Catalog for the environment.

Type: String

Owner

The owner of the environment.

Type: String

Created At

The DateTime the environment was created at.

Type: String

Updated At

The DateTime the environment was last updated at.

Type: String

Dataset metadata

Details dataset metadata for an environment within Service Workbench on AWS.

Contents

ID

The ID of the dataset attached to the environment.

Type: String

Created At

The DateTime the dataset metadata record was created at.

Type: String

Updated At

The DateTime the dataset metadata record was last updated at.

Type: String

Endpoint metadata

Details endpoint metadata for an environment within Service Workbench on AWS.

Contents

ID

The ID of the endpoint attached to the environment.

Type: String

Dataset ID

The ID of the dataset corresponding to endpoint.

Type: String

Created At

The DateTime the endpoint metadata record was created at.

Type: String

Updated At

The DateTime the endpoint metadata record was last updated at.

Type: String

Environment type configuration metadata

Details instance metadata for an environment within Service Workbench on AWS.

Contents**ID**

The UUID of the environment type configuration.

Type: String

Type

The type of the environment type configuration.

Type: String

Product ID

The product ID of the environment type.

Type: String

Provisioning Artifact ID

The provisioning artifact ID of the environment type.

Type: String

Params

The parameters for the environment type configuration.

Type: Array of [Environment Type Configuration Parameter](#) objects

Created At

The DateTime the environment type configuration was created at.

Type: String

Updated At

The DateTime the environment type configuration was last updated at.

Type: String

Instance metadata

Details instance metadata for an environment within Service Workbench on AWS.

Contents

Instance ARN

The ARN of the instance backing the environment.

Type: String

ID

The ID of the environment for this instance.

Type: String

Created At

The DateTime the instance metadata record was created at.

Type: String

Updated At

The DateTime the instance metadata record was last updated at.

Type: String

Project metadata

Describes project metadata for an environment within Service Workbench on AWS.

Contents

ID

The ID of the project attached to the environment.

Type: String

Name

The name of the project attached to the environment.

Type: String

Subnet ID

The subnet ID of the subnet in the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

AWS Account ID

The ID of the AWS Account that the cost center of the project attached to the environment is in.

Type: String

Environment Instance Files

The S3 URL where environment instance files are stored in the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

VPC ID

The VPC ID of the VPC in the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

Env Mgmt Role ARN

The environment management role ARN of the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

Encryption Key ARN

The encryption key ARN of the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

External ID

The external ID of the onboarded AWS hosting account that the cost center of the project attached to the environment is in. This value is likely workbench.

Type: String

Created At

The DateTime the project metadata was created at.

Type: String

Updated At

The DateTime the project metadata was last updated at.

Type: String

Hosting Account Handler Role ARN

The hosting account handler role ARN of the onboarded AWS hosting account that the cost center of the project attached to the environment is in.

Type: String

Migration guide

You can migrate studies from Service Workbench on AWS to datasets in Research Service Workbench on AWS (RSW). To migrate your studies from SWB to datasets in RSW, you will need to create projects and datasets in RSW. With Amazon S3's copy APIs, you can transfer data from the study S3 buckets to the dataset S3 bucket.

Cost of migration

If you are copying S3 data to the same AWS Region, there are no fees. When copying to a different region, fees are incurred. For more information, see [Amazon S3 pricing](#).

Migration prerequisites

After deploying Research Service Workbench on AWS, IT Admins need to create each project the datasets will belong to. You can use the Create Project API in RSW.

Once all projects are created, the IT Admin will need to create a Project Admin user with an email address different from the one used for the IT Admin role. If your email provider supports plus addressing, we recommend using your existing address with +pa. For example, myemail+pa@amazon.com. This allows registration emails to come to your standard email address.

After creating the Project Admin user, IT Admins should add that user to each project that owns a dataset. Use the Add User to Project API, specifying ProjectAdmin as the role in the request body.

When the Project Admin user has been added to all projects, the IT Admin should login as the Project Admin user and create datasets under each project using the Create Internal Dataset API. This creates a folder in the dataset S3 bucket, allowing IT Admins to copy data from studies into the datasets.

Migration process

IT Admins need an IAM user and role with access to both the study S3 bucket(s) and the dataset S3 bucket. The RSW dataset S3 bucket will be named `rsw-<stage>-<region-abbreviation>-s3datasets<uuid>`.

You need the following IAM permissions:

On the study S3 bucket:

- S3:ListBucket
- S3:GetObject

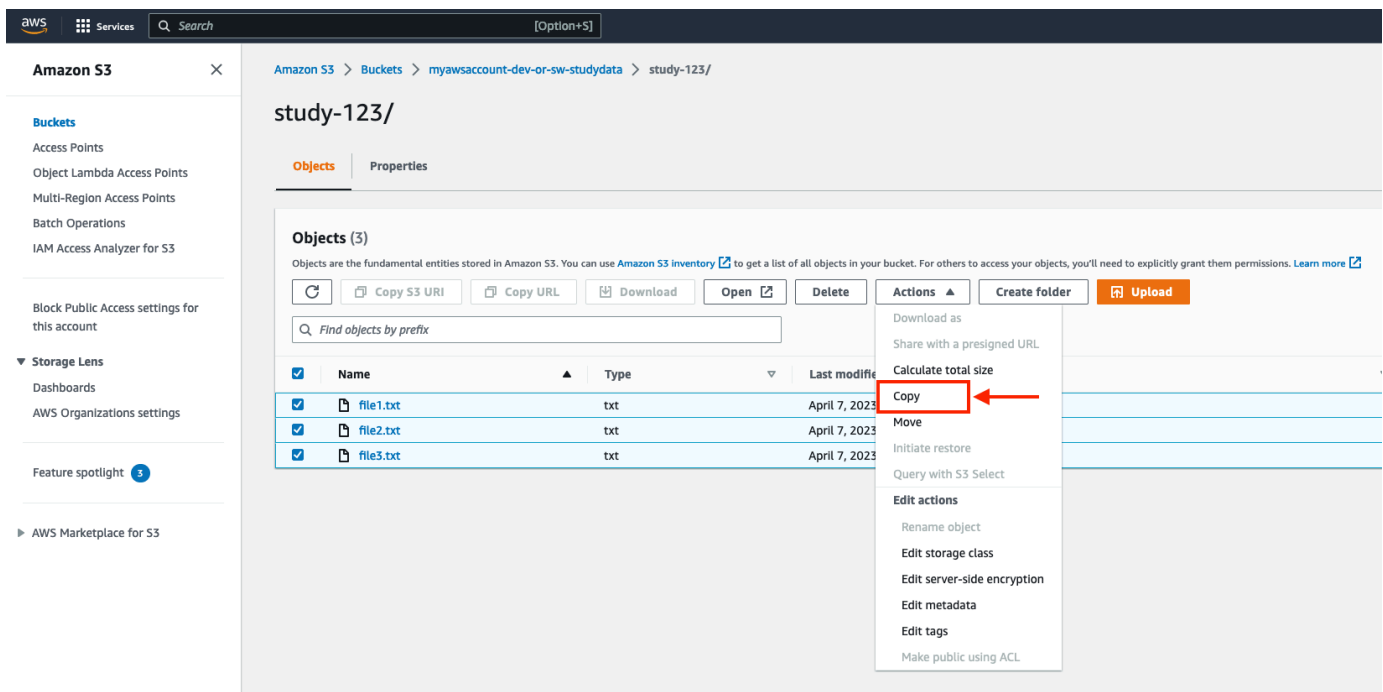
On the datasets S3 bucket:

- S3:ListBucket
- S3:PutObject

Migration can be performed through AWS Management Console or AWS CLI.

Migrating with AWS Management Console

1. Sign in to the AWS Management Console with the account that has permissions of the study S3 bucket(s) and the datasets S3 bucket.
2. Select the study S3 bucket and choose the files from the study you want to migrate into the dataset.
3. Choose **Actions**.
4. Choose **Copy**.



The screenshot shows the AWS Management Console interface for an Amazon S3 bucket named 'study-123/'. The left sidebar displays navigation options for Amazon S3, including Buckets, Access Points, and Storage Lens. The main content area shows the bucket's contents, with three objects selected: file1.txt, file2.txt, and file3.txt. The 'Actions' menu is open, and the 'Copy' option is highlighted with a red box and a red arrow.

<input checked="" type="checkbox"/>	Name	Type	Last modified
<input checked="" type="checkbox"/>	file1.txt	txt	April 7, 2023
<input checked="" type="checkbox"/>	file2.txt	txt	April 7, 2023
<input checked="" type="checkbox"/>	file3.txt	txt	April 7, 2023

Figure 1: Copy the S3 files

5. Provide the S3 bucket URL for the dataset S3 bucket and folder.
6. Choose **Copy**.

aws Services Search [Option+S]


other AWS accounts and groups have been removed.
[Learn more](#)

Destination

Destination type

Bucket
 Access Point

Destination

s3://swb-dev-uw2-s3datasetsuud/study-123/  [View](#) [Browse S3](#)

Format: s3://(bucket-name)/(prefix)

Destination bucket name
-

Destination prefix
study-123/

Destination details

The following bucket settings impact new objects stored in the specified destination.

Bucket Versioning When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended user actions and application failures. Learn more	Default encryption key type Bucket settings for default encryption are automatically applied to any specified object that is unencrypted. -	Object Lock When enabled, objects in this bucket might be prevented from being deleted or overwritten for a fixed amount of time or indefinitely. Learn more
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Specified objects

Find objects by name

Name	Type	Last modified	Size
file1.txt	txt	April 7, 2023, 03:50:00 (UTC-07:00)	0 B
file2.txt	txt	April 7, 2023, 03:50:00 (UTC-07:00)	0 B
file3.txt	txt	April 7, 2023, 03:50:00 (UTC-07:00)	0 B

Additional checksums

Checksum functions are used for additional data integrity verification of new objects. [Learn more](#)

Additional checksums

Copy existing checksum functions
 Apply the same checksum function to the destination object. The checksum value may change compared to when it was added. [Learn more](#)

Replace with a new checksum function

Checksum function for additional data integrity validation which will overwrite any existing data integrity settings.

Cancel [Copy](#)

Figure 2: Provide the S3 bucket URL

7. Verify you receive a success message: **Successfully copied objects**. You may also verify success by checking the datasets S3 bucket.

Migrating with AWS CLI

Run the following command for each folder that needs to migrate from a SWB study to a RSW dataset:

```
aws s3 cp
s3://myawsaccount-stage-regionabbreviation-sw-studydata/study-123
s3://swb-stage-regionabbreviation-s3datasetsuuid/study-123
--recursive
```

You will receive a copy confirmation for each file in the folder that migrates to the S3 datasets bucket.

Migration cleanup

After all studies migrate to RSW datasets, the IT Admin should remove the Project Admin user created for migration from all projects and delete the user. Use the Remove User from Project API. After removing the user, use the Delete User API.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, and a list of builders who contributed to this solution.

Anonymous data collection

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When invoked, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each Research Service Workbench on AWS deployment
- **Timestamp** - Data-collection timestamp
- **Example: Instance Data** - Count of the state and type of instances that are managed by the EC2 Scheduler in each AWS Region

Example data:

Running: {t2.micro: 2}, {m3.large:2}

Stopped: {t2.large: 1}, {m3.xlarge:3}

AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#). To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
AnonymousData:  
SendAnonymousData:  
Data: Yes
```

to:

```
AnonymousData:  
SendAnonymousData:  
Data: No
```

4. Sign in to the [AWS CloudFormation console](#).
5. Select Create stack.
6. On the Create stack page, Specify template section, select Upload a template file.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in [Launch the stack](#) in the Deploy the solution section of this guide.

Related resources

- The Service Workbench on AWS solution is a similar to Research Service Workbench on AWS.

Contributors

- Or Glazer
- Marianna Ghirardelli
- Kevin Park
- Tyler Mikev
- Sanket Dharwadkar
- Jane Yu
- Fernando Aranda Carrillo
- Travis Berkley
- Greg Grieff
- Brandi Hopkins

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Research Service Workbench is licensed under the terms of the of the Apache License Version 2.0 available at [The Apache Software Foundation](#).

Revisions

For more information, see the [CHANGELOG.md](#) file in the GitHub repository.

Date	Change
June 2023	Initial release
September 2023	Release version 2.0.1 - For more information, refer to the CHANGELOG.md file in the GitHub repository.