Implementation Guide

Scale-Out Computing on AWS

Scale-Out Computing on AWS: Implementation Guide

Table of Contents

Deploy a multiuser environment for computationally intensive workflows, such as Computer-Aided Engineering (CAE)

Publication date: *November 2019 (last update: July 2023)*

Amazon Web Services (AWS) allows data scientists, designers, and engineers to run their scale-out workloads such as those that require parallel processing and deep learning training, without having extensive cloud experience.

The Scale-Out Computing on AWS solution helps customers easily deploy and operate a multiuser environment for computationally intensive workflows such as Computer-Aided Engineering (CAE). The solution features a large selection of compute resources, a fast network backbone, unlimited storage, and budget and cost management directly integrated within AWS. This solution also deploys a user interface (UI) with cloud workstations, file management, and automation tools that allow you to create your own queues, scheduler resources, Amazon Machine Images (AMIs), and management functions for user and group permissions.

This solution is designed to be a production-ready reference implementation you can use as a starting point for deploying an AWS environment to run scale-out workloads, allowing users to focus on running simulations designed to solve complex computational problems. For example, with the unlimited storage capacity provided by Amazon Elastic File System (Amazon EFS), users won't run out of space for project input and output files. Additionally, you can integrate your existing LDAP directory with Amazon Cognito to allow users to seamlessly authenticate and run jobs on AWS.

This implementation guide describes architectural considerations and configuration steps for deploying Scale-Out Computing on AWS in the AWS Cloud. It includes links to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting in the AWS Cloud.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the total cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **$441.27 per month**. This cost estimate includes deploying an m5.large Amazon Elastic Compute Cloud (Amazon EC2) instance, an Application Load Balancer (ALB), a highly available Amazon OpenSearch Service (OpenSearch Service) cluster, Amazon EFS, AWS Backup, and a NAT Gateway.

Note that the following factors will contribute to incremental costs for an actively used deployment:

- Compute jobs submitted and what resources they consume (EC2 instances, EBS volumes)
- Volume of data stored in persistent shared and scratch storage.

Example cost table

The following table provides an example cost breakdown for deploying this solution with the default settings in the US East (N. Virginia) Region.

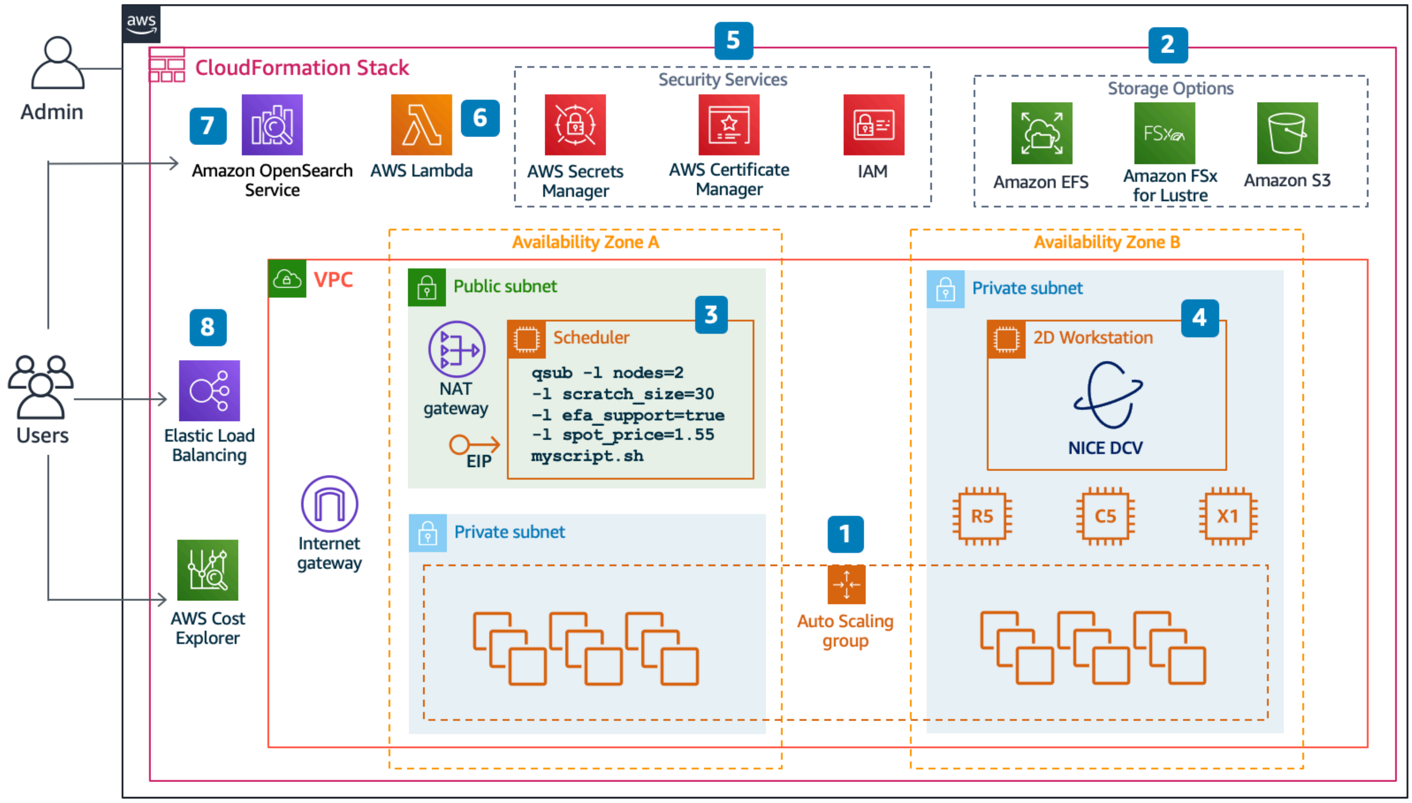
AWS service	Monthly cost
Amazon OpenSearch Service (Provides user and administrator analytics dashboard)	$231.30
Amazon EC2 (Persistent scheduler hosts the WebUI, schedular and solution scripts)	$144.73
Elastic Load Balancing (Provides accessibility)	$16.21
Amazon Elastic File System (Persistent storage for users’ home directories and applications)	$0.33
AWS Secrets Manager, AWS Backup, and other Amazon EC2 related networking	$48.70
Total:	**$441.27***

This pricing estimate does not include ephemeral visualization or compute instances or data transfer costs. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

> **Note**
>
> This solution includes tools to manage the ephemeral usage cost. For more details, refer to the Project budgets section in this guide.

Architecture overview

Deploying this solution builds the following environment in the AWS Cloud.



```
qsub -l nodes=2
-l scratch_size=30
-l efa_support=true
-l spot_price=1.55
myscript.sh
```

Scale-Out Computing on AWS architecture

The solution uses AWS CloudFormation templates to deploy the infrastructure components, AWS services, operating systems and management software, and custom logic scripts to create a scale-out design and engineering workflow reference implementation. The following overview describes the eight components and their associated AWS services deployed with the solution. For additional details about each component, refer to the Solution components section.

1. Amazon EC2 Auto Scaling to automatically provision the resources necessary to run cluster user tasks such as scale-out compute jobs.
2. This solution also deploys Amazon Elastic File System for persistent storage, Amazon Simple Storage Service (Amazon S3) for persistent logs, and optional parallel file system FSx for Lustre.
3. At its core, the Amazon Elastic Compute Cloud (Amazon EC2) instance implements a scheduler, which dynamically provisions AWS resources required for jobs submitted by users. The scheduler

instance also hosts web interface which allows users and administrators to interact with the environment.

4. Launch a 2D or 3D Workstation that uses NICE Desktop Cloud Visualization (DCV), that can be used to submit batch jobs and run GUI tools.
5. Security services and resources that are used include AWS Secrets Manager, AWS Certificate Manager, Security Groups, and AWS Identity and Access Management(IAM).
6. AWS Lambda functions to verify the required prerequisites and create a default signed certificate for an Application Load Balancer (ALB) to manage access to DCV workstation sessions.
7. An Amazon OpenSearch Service cluster to store job and host information.
8. Elastic Load Balancing is used to ensure accessibility across Availability Zones, and Cost Allocation Tags are used with AWS Cost Explorer.

Solution components

User interface

Scale-Out Computing on AWS deploys and sets up an example web user interface (UI) with a common set of APIs that the administrator and users can use to interact with their Amazon Elastic Compute Cloud (Amazon EC2) cluster. The example UI allows users to submit jobs, manage and share their files, start/stop desktop cloud visualization (DCV) sessions, download private keys, and monitor the queue and job status in real-time. Administrators can use the UI to manage LDAP users and groups, create application profiles (for web-based job submission), and manage job queues.

Pre- and post-processing in the Cloud

This solution leverages cloud-based workstations to allow users to easily access the cluster to perform any pre- and post-processing visualization actions (such as computer-aided design). User working files persist across workstation sessions and are stored in the user home directory in Amazon Elastic File System (Amazon EFS). Administrators can create custom Linux Amazon Machine Images (AMIs) with common user applications preinstalled in the cloud workstation.

Real-time analytics

Schedulers and application logs are ingested in real-time and stored into the data lake for further processing. Node counts, job status, and metadata is automatically pushed to the Amazon OpenSearch Service (OpenSearch Service) cluster.

Custom code and automation

This solution is deployed with a collection of scripts that are customizable and can be extended to help administrators and users collect data and perform common cluster tasks. These customizations can be found in `/apps/soca/`*`<your-soca-name>`* and perform the following tasks:

- **Automatic Error Handling:** Dry run checks before provisioning Amazon EC2 capacity
- **Automatic Log Management:** Collects and backups cluster logs to Amazon S3
- **Custom job status tool:** Improves cluster status with AWS-specific information

- **Simplified LDAP user management:** Scripts to perform typical LDAP actions
- **Application License resource:** FlexLM software enabled script which calculates the number of license available for a given feature

Project budgets

This solution helps users and administrators manage their project budgets. It allows administrators to create detailed reports by users, software, teams, queues, projects, or applications using resource tagging. This solution uses AWS Cost Explorer and AWS Budgets to help users manage their expenses and forecast their budgets based on historical data. Note that if resource tagging is not activated, you must manually activate these tags for the Cost Explorer reporting platform through Cost Allocation Tags.

Customizable

This solution can be customized by users to fit their business needs. The business logic is configured using an AWS CloudFormation template and Amazon EC2 user data scripts. The solution's codebase is open-source and available on GitHub. Customization examples can be found on the official documentation.

Persistent and unlimited storage

This solution deploys two unlimited Amazon Elastic File System (Amazon EFS) storage locations (`/apps and /data`). You can also deploy high-speed Amazon EBS SSD-backed disks or FSx for Lustre that can be used as a scratch location on your compute nodes.

Centralized user mangement

Customers can create unlimited LDAP users and groups. By default, this solution deploys a default LDAP account and a Sudoers LDAP group which manages the SUDO permission on the cluster.

Scheduler instance

This solution deploys an Amazon EC2 instance running the open source PBS Professional (PBSPRO) 20.0.1 job scheduling software. This solution has an AGPLv3 licensing component. For more information, refer to Notices.

Application programming interface (API)

This solution provides an HTTP REST API for administrators and users to interact with the cluster programmatically. Through the API you can create users, groups, and queues; submit jobs; and view and change job status using either bash or python. You can also manage remote desktop (DCV) sessions through the API.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

> **Note**
>
> To maintain security best practices on AWS, we recommend reviewing the steps below after you implement this solution.
>
> 1. Ensure OpenSearch Service domain logging is activated.
> 2. Ensure Elastic Load Balancing (ELB) logging is activated.

Security groups

The security groups created in this solution are designed to control and isolate network traffic between the Amazon Virtual Private Cloud's (Amazon VPC) for the scheduler and compute components. We recommend that you review the security groups and further restrict access as needed.

Upload an SSL certificate for the user interface

The deployed UI uses HTTPS with an Application Load Balancer endpoint. To upload a signed SSL certificate, refer to the Upload your SSL certificate to ACM section in the GitHub repository.

Integrate with existing LDAP directory

With Amazon Cognito, your users can sign in to the web user interface automatically (without the need for a password). They can use social identity providers, such as Google, Facebook, and Amazon, or enterprise identity providers, such as Microsoft Active Directory using SAML. For instructions to activate SSO authentication to the web interface, refer to Enable Oauth2 authentication with Cognito in the *Scale-Out Computing on AWS Knowledge Base*.

Note

By default, this solution uses an OpenLDAP service. We recommend activating a connection to an external LDAP directory with encryption (LDAPS).

AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of Scale-Out Computing on AWS in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

View template

scale-out-computing-on-aws.template: Use this template to launch the solution and all associated components. The default configuration deploys Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 Auto Scaling, AWS Lambda, Amazon Elastic File System, AWS Secrets Manager, AWS Identity and Access Management, and Amazon Elastic Compute Cloud, but you can also customize the template based on your specific needs.

This template in turn launches the following nested stacks:

- **scale-out-computing-on-aws-network.template:** This template deploys the Amazon Virtual Private Cloud (Amazon VPC), route tables, Internet Gateway, Elastic IP address, and NAT Gateway components of the solution.
- **scale-out-computing-on-aws-security.template:** This template deploys the AWS Security Groups, and AWS Identity and Access Management (IAM) role components of the solution.
- **scale-out-computing-on-aws-storage.template:** This template deploys the Amazon Elastic File System (Amazon EFS) component of the solution.
- **scale-out-computing-on-aws-scheduler.template:** This template deploys the Amazon Elastic Compute Cloud (Amazon EC2) component of the solution.
- **scale-out-computing-on-aws-analytics.template:** This template deploys the Amazon OpenSearch Service (OpenSearch Service) component of the solution.
- **scale-out-computing-on-aws-viewer.template:** This template deploys the Desktop Cloud Visualization (DCV), Application Load Balancer (ALB), IAM role, and AWS Certificate Manager components of the solution.
- **scale-out-computing-on-aws-configuration.template:** This template deploys the AWS Secrets Manager and AWS Backup components of the solution.

Automated deployment

Before you launch the automated deployment, please review the architecture, prerequisites, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the Scale-Out Computing on AWS solution into your account.

Time to deploy: Approximately 35 minutes

Prerequisites

Subscribe to Amazon Machine Images

This solution uses an Amazon Machine Image (AMI) as the host operating system for the scheduler instance, user desktop instances, and compute node instances. By default, you must select the base AMI to use for all three instances in the **Linux Distribution** template parameter or specify a **Custom AMI**. As of this release, this solution supports the following AMIs for the scheduler instance:

- Red Hat Enterprise Linux 7
- CentOS 7
- Amazon Linux 2

> **Note**
>
> If you choose to use the CentOS 7 image, you must subscribe to CentOS 7 in the AWS Marketplace, to allow the installer to access the AMI during installation.
> This solution supports a heterogeneous environment. After installation, administrators and users can specify a custom AMI per job and queue.

Service-linked roles for Amazon EC2 Auto Scaling

This solution deploys Amazon EC2 Auto Scaling to scale out multi-instance, user-submitted jobs. Verify that AWS Identity and Access Management (AWS IAM) roles have the appropriate permissions supporting Amazon EC2 Auto Scaling. For more information, refer to Auto Scaling Service-Linked Roles.

Select your deployment

This solution can be deployed using a default set of parameters in the AWS CloudFormation template, or you can customize the solution by building your own custom installer by cloning the GitHub repository.

> ⚠ **Important**
>
> Deploying the AWS CloudFormation template with the default parameters is recommended for testing and proof of concept. However, if you are using this solution in a production environment, we recommend deploying this solution using a custom installer in your own hosted repository to reduce costs, and maintain customization and extensibility. If you choose to deploy this solution using a custom installer, refer to Deployment using custom installer.

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameter: **Stack Name**, **User Name**, **Password**
- Review the other template parameters, and adjust if necessary.

Step 2. Access the Amazon EC2 cluster

- Access the Amazon EC2 cluster through the UI or SSH.

Step 3. Manage your budget and monitor costs

- Set up cost allocation and budgets

Step 1. Launch the stack

Important

This solution includes an option to send anonymized 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 though 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 template and deploy the solution. For more information, refer to the Collection of operational metrics section in this guide.

This automated AWS CloudFormation template deploys Scale-Out Computing on AWS in the AWS Cloud. Verify that you have completed the prerequisites before you launch the stack.

Note

You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `scale-out-computing-on-aws` AWS CloudFormation template.

   Launch solution

   You can also download the template as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar. Note that if you choose to launch in a different Region, we recommend using a custom installer in your own Amazon S3 bucket to avoid data transfer costs. For more information, refer to Deployment using custom installer.

> **Note**
>
> This solution uses three Availability Zones to maximize the resources users have for submitting jobs. Therefore, you must launch this solution in an AWS Region that has at least three Availability Zones.
>
> This solution uses the AWS Fargate service, which is currently available in specific AWS Regions only. Therefore, you must deploy this solution in a Region where AWS Fargate is available.
>
> For more information, refer to Global Infrastructure.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack.

> **Note**
>
> This solution deploys an analytics stack that limits your stack name to 24 lowercase characters. The template automatically adds the prefix soca- to your solution stack name.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Install Location		
Installer S3 Bucket	`solutions-reference`	The default AWS bucket name. Do not change this parameter unless you are using a custom installer.
Installer Folder	`scale-out-computing-on-aws/latest/`	The default AWS folder name. Do not change this parameter unless you are using a custom installer.

Parameter	Default	Description
Linux Distribution		
Linux Distribution	`AmazonLinux2`	Select the preferred Linux distribution for the scheduler and compute instances.
Custom AMI	<Optional input>	If using a customized Amazon Machine Image, enter the ID. **Note** If you are using your own AMI, you still have to specify the base Linux Operating System. For more information, refer to the section called “Subscribe to Amazon Machine Images”.
Network and Security		
EC2 Instance Type for Scheduler	`m5.large`	Select the instance type for the scheduler.
VPC CLuster CIDR	`10.0.0.0/16`	Choose the CIDR (/16) block for the VPC. This is the internal network over which your cluster will communica te.

Parameter	Default	Description
IP Address	*<Requires input>*	Identifies the default IP(s) allowed to directly SSH into the scheduler and access Amazon ES. To restrict access, use IP/subnet (x.x.x.x/32 for your own IP or x.x.x.x/24 for a range. Replace x.x.x.x with the PUBLIC IP. To identify the public IP, use a tool, such as https://ifconfig.co/. **Note** 0.0.0.0/0 means ALL INTERNET access and is not recommended.
Key Pair Name	*<Requires input>*	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
Default LDAP user		
User Name	*<Requires input>*	The username for your default LDAP user.

Parameter	Default	Description
Password	*<Requires input>*	The password for your default LDAP user. **Note** 5 characters minimum. Must start with a letter. Accepted special characters include: ! # @ + _ ^ { } [ ] ( )

6. Choose **Next**.
7. On the **Configure Stack Options** page, choose **Next**. Optionally, you can specify tags to associate with the entire stack and specify an IAM role that will be used for all of the stack creation functions.
8. On the **Review** page, review and confirm the settings. Be sure to 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 see a status of **CREATE_COMPLETE** in approximately 35 minutes. Note that the custom installer deployment takes approximately 60 minutes.

Step 2. Access the Amazon EC2 cluster

After the AWS CloudFormation template has successfully launched you can access the Amazon EC2 cluster via HTTPS through a web portal or SSH.

Access the cluster through the web portal

Use the following procedure to access the cluster through the web portal:

1. After the solution has deployed, navigate to the stack **Outputs** tab
2. Copy the value for **WebUserInterface**, and paste the link into a web browser.

Note

It can take up to 15 minutes for the UI to be installed after the AWS CloudFormation template is completed.
To open the webpage, you must authorize the web browser to trust the self-signed security certificate (add an exception) or Upload a Server Certificate to the Elastic Load Balancer endpoint. Note that access to the web UI will be restricted to the subnet specified in the **IP Address** template parameter.

Access the cluster through SSH

Administrator SSH

By default, this solution installs the key pair that you provided in the **Key Pair Name** template parameter. Note that all SSH sessions are required to use public key encryption.

Additionally, this solution provides an admin connection string for operating systems with native SSH clients (Mac/Linux) in the **Outputs** section of the CloudFormation stack.

Note

It can take up to 15 minutes for final installation procedures to complete. During this time, attempts to login via SSH will generate a message indicating that the installation is still running.

User SSH

Users who access the cluster through SSH must download their PEM key. For more information on downloading your PEM/PPK key and setting up the SSH client, refer to How to access the solution.

Adding users to the cluster

Scale-Out Computing on AWS uses open LDAP for directory services. Administrators can interact with their directory using LDAP directly. This solution installs a custom wrapper that can be found in `/apps/soca/cluster_manager/ldap_manager.py`. This wrapper can be run by administrators logged in via SSH, or for quick actions via the web portal. For example, adding/ deleting users, resetting user passwords, and granting and revoking administrator privileges.

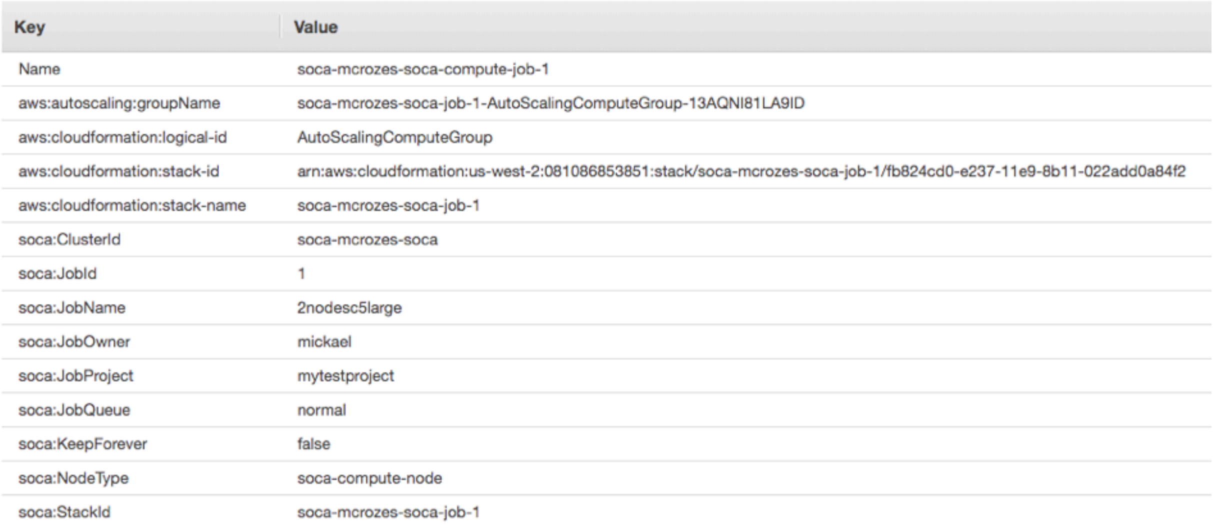
Use the following procedure to create or delete a user through the user interface:

1. In Admin, navigate to **Users Management**.
2. Add and delete users.

> **Note**
>
> Deleting users will prevent user access to the cluster but will not remove associated **$HOME** directory and data.

Step 3. Manage your budget and monitor costs

By default, this solution implements comprehensive tagging of cluster resources and allows admin-defined tags during cluster deployment. All Amazon Elastic Compute Cloud (Amazon EC2) resources launched by this solution come with Amazon EC2 tags that can be used to get detailed information about your cluster usage. You can modify and add tags based on your business needs.



Key	Value
Name	soca-mcrozes-soca-compute-job-1
aws:autoscaling:groupName	soca-mcrozes-soca-job-1-AutoScalingComputeGroup-13AQNI81LA9ID
aws:cloudformation:logical-id	AutoScalingComputeGroup
aws:cloudformation:stack-id	arn:aws:cloudformation:us-west-2:081086853851:stack/soca-mcrozes-soca-job-1/fb824cd0-e237-11e9-8b11-022add0a84f2
aws:cloudformation:stack-name	soca-mcrozes-soca-job-1
soca:ClusterId	soca-mcrozes-soca
soca:JobId	1
soca:JobName	2nodesc5large
soca:JobOwner	mickael
soca:JobProject	mytestproject
soca:JobQueue	normal
soca:KeepForever	false
soca:NodeType	soca-compute-node
soca:StackId	soca-mcrozes-soca-job-1

Scale-Out Computing on AWS default tags

AWS cost explorer

Use the following procedures to setup cost allocation and budgets to track the costs associated with solution resources running in your account.

Activate cost allocation tags

1. In the AWS Cost Management console, select your **account name**, then select **My Billing Dashboard**
2. In the left-hand navigation pane, select **Cost allocation tags**
3. Search all tags, then select **Activate**

> **Note**
>
> Tags may take up to 24 hours to activate.

Activate cost explorer

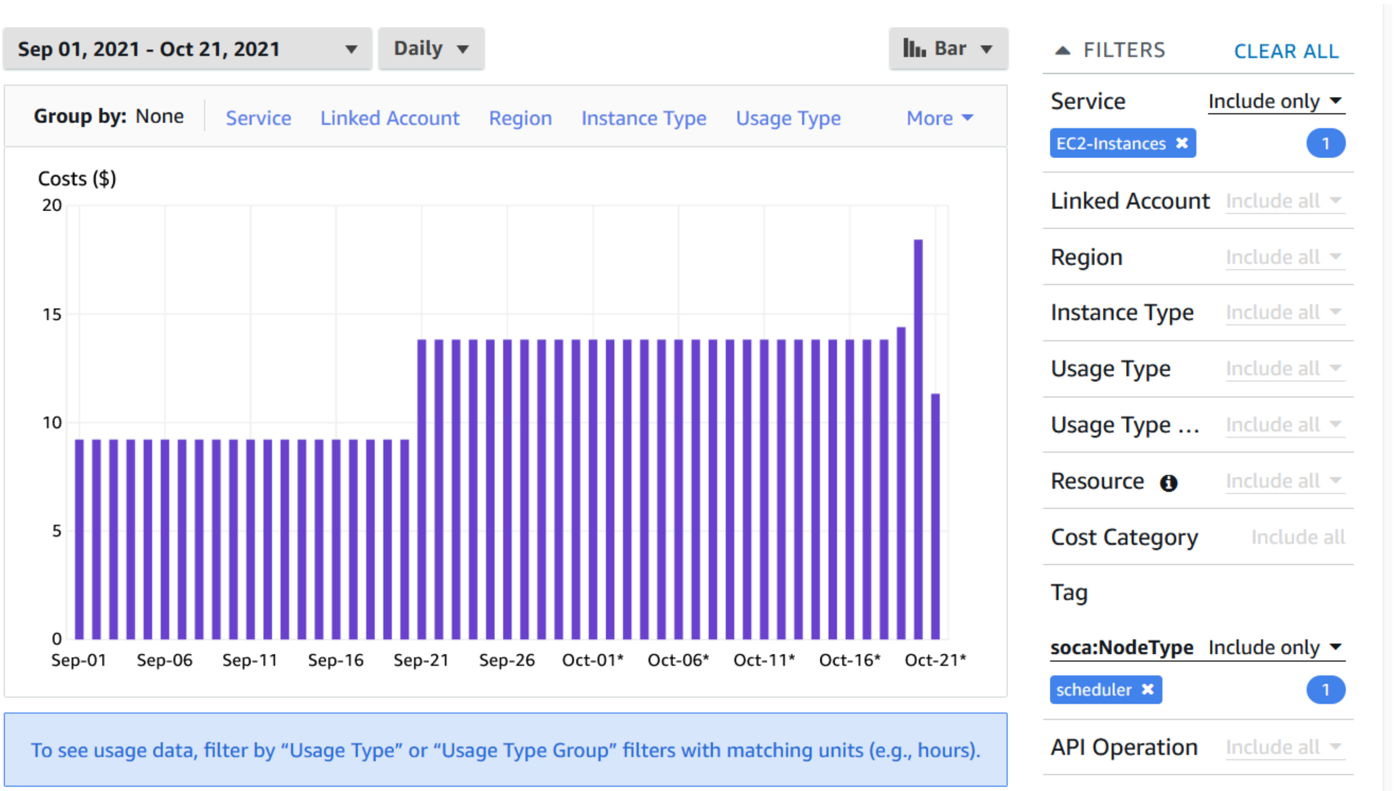
1. In the AWS Cost Management console, select **My Billing Dashboard**.
2. Select **Cost Explorer**, then select **Enable Cost Explorer**.

> **Note**
>
> Cost Explorer may take up to 24 hours to be activated.

Query cost explorer

1. Select **Cost Explorer** then specify your filters.

   The following example shows the **EC2-Instance** cost group by the day for the node type scheduler.



Sep 01, 2021 - Oct 21, 2021

Daily

Bar

Group by: None

FILTERS

Service: Include only — EC2-Instances

Linked Account: Include all

Region: Include all

Instance Type: Include all

Usage Type: Include all

Usage Type ...: Include all

Resource: Include all

Cost Category: Include all

Tag

soca:NodeType: Include only — scheduler

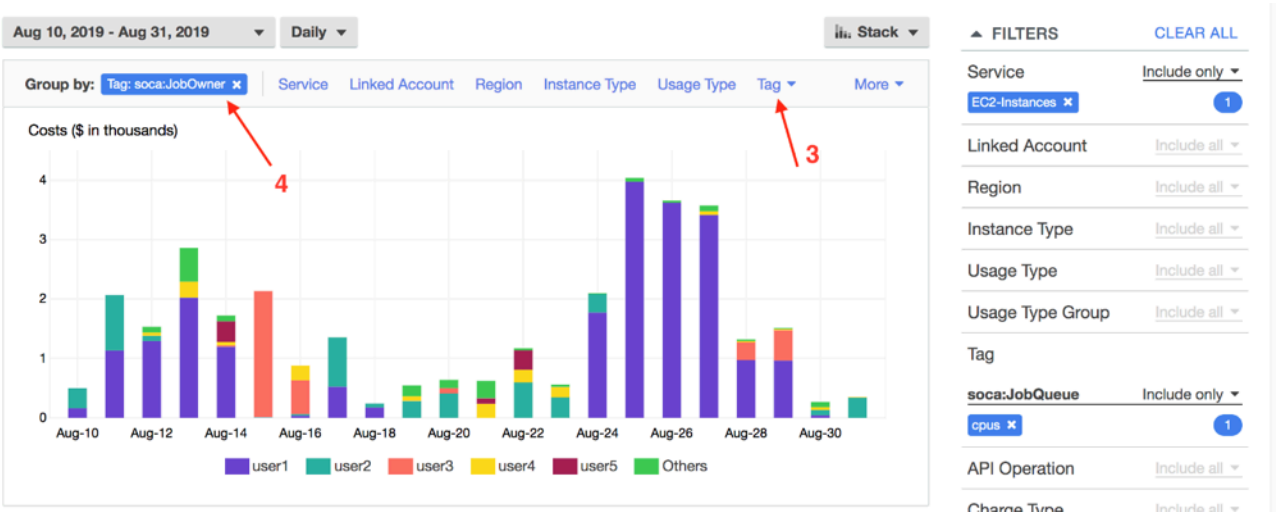
API Operation: Include all

To see usage data, filter by “Usage Type” or “Usage Type Group” filters with matching units (e.g., hours).

EC2-Instance cost group by day

2. For more detailed information, select **Group By** and apply any additional filters.
3. Select **Tag** and select the tag **soca:JobOwner**. The graph will automatically be updated with a cost breakdown by users for the **cpus** queue.

   The following example shows user level information for the **cpus** queue.



Aug 10, 2019 - Aug 31, 2019

Daily

Stack

Group by: Tag: soca:JobOwner

FILTERS

Service: Include only — EC2-Instances

Linked Account: Include all

Region: Include all

Instance Type: Include all

Usage Type: Include all

Usage Type Group: Include all

Tag

soca:JobQueue: Include only — cpus

API Operation: Include all

Charge Type: Include all

User level information for cpus queue

AWS budgets

1. In the AWS Budgets console, select your **Billing Dashboard**.
2. In the navigation pane, select **Budget**, then select **Create a budget**.
3. Specify your **Cost Budget**, then apply the tag(s) you want to monitor (i.e. cost center, team, user, and queue or project), and the budget limit you do not want to exceed.

Additional resources

AWS services

- AWS CloudFormation
- AWS Lambda
- Amazon EC2 Auto Scaling
- AWS Cost Explorer
- AWS Budgets
- AWS Secrets Manager
- AWS Identity and Access Management
- Amazon EC2
- Application Load Balancer
- AWS Backup
- Amazon S3
- Amazon OpenSearch Service

Other documentation

- Scale-Out Computing on AWS Knowledge Base

Deployment using custom installer

For customers who want to leverage existing AWS resources required for the solution while maintaining customization and extensibility, this solution allows you to deploy the AWS CloudFormation template using the AWS Cloud Development Kit (AWS CDK) based custom installer in your hosted repository for production environments. The CDK based installer also allows customization to all resources created during the installation. For example, you can choose how many NAT Gateways to deploy (default to 1), the KMS encryption to use for your file systems (default to aws/key), the instance type (default to m5.large) to provision for the scheduler, and other options that are specified with the CloudFormation implementation.

Note

These instructions are applicable to versions 2.7.0 or later of this AWS Solution. For older versions, refer to the legacy documentation.

This solution is a collection of AWS CloudFormation templates, Amazon Elastic Compute Cloud (Amazon EC2) user data bootstrap scripts, and Python scripts. Before deploying, create a custom installer based on the build ID you want to use.

Automated deployment

Before you launch the automated deployment, please review the architecture, prerequisites, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to deploy this solution into your account using a custom installer.

Time to deploy: Approximately 60 minutes

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

Prerequisites

- Set up an Amazon S3 bucket.

[Step 1. Create and confirm required IAM policies]

- Apply IAM policies required to deploy the solution

[Step 2. Download the solution template]

- Download the solution from the Github repository

[Step 3. Create and run the installer]

- Create and upload the build
- Review solution parameters, and adjust if necessary
- Launch the solution installer

Prerequisites

Set up Amazon S3 bucket

This solution uses an Amazon S3 bucket for storing data. Before deploying this solution, you must create a new Amazon S3 bucket in your AWS account. Or, you can use an existing Amazon S3 bucket.

Step 1. Create and confirm required IAM policies

> **Note**
>
> Following step 2, you can find the list of all required IAM policies to install the solution via `installer/SOCAInstallerIamPolicy.json`

If needed, you can create an IAM policy and assign it to the use or Role you are planning to use to install this solution.

1. Navigate to **IAM** in the AWS console, select **Policies** in the left sidebar menu then choose **Create Policy**.
2. Select **JSON** and copy and paste the content of `installer/SOCAInstallerIamPolicy.json`

3. Select the user or Role you are using to install Scale-Out Computing on AWS.
4. Choose **Add Permissions** and attach the policy you just created.

Step 2. Download the solution template

This solution is open-source and hosted on GitHub. Use the following procedure to download the solution template:

1. Navigate to the GitHub repository, select **Clone or download**.
2. Select **Download Zip**.
3. When the zip file is downloaded, extract the file on your local machine.

Step 3. Create and run the installer

After you have cloned your repository, run the `installer/soca_installer.sh` script. The installer will perform the following tasks:

> **Note**
>
> For the build and installer environment, Linux (Amazon Linux 2, RHEL, Centos) or MacOS host operating system are supported to run the solution installer.

- Check if Python3 is available on your system
- Create a custom Python virtual-environment and install required libraries
- Install NodeJS, NPM, AWS CDK and AWS Command Line Interface if needed
- Set up your Scale-Out Computing on AWS cluster

The installer is built with AWS CDK. To learn more, refer to AWS CDK.

> **Note**
>
> You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

1. Run the soca_install.sh script located in the installer folder. Assuming your current working directory is the root level of the solution stack.

   ```
   $ ./installer/soca_install.sh
   ```

2. The installer script will prompt for your cluster parameters. Follow the instructions and choose a Amazon S3 bucket you own, the name of your cluster, the SSH keypair to use and other cluster parameters.

   > **Note**
   >
   > You can pass all parameters via arguments to automate the installation process. Run `soca_installer.sh --help` to see all options available.

Once all the parameters are specified, installer will run the `cdk bootstrap` command. This action will create a staging S3 bucket and store all assets generated by AWS CDK. No actions will be performed if you already have your environment activated for AWS CDK.

The installer will upload the scripts (<100 MB) required to configure the scheduler to the Amazon S3 bucket you specified.

Next, the installer will trigger a `cdk deploy` command and the deployment will start. This will create a new CloudFormation stack on your AWS account.

Once the CloudFormation stack is created, the installer will verify if the solution is configured correctly. The installer will exit once the solution is fully configured and reachable.

Troubleshooting

If the Scale-Out Computing on AWS solution stack fails when being deployed into your account, verify the following:

- You received a region resource approval email. When deploying Amazon Elastic Compute Cloud (Amazon EC2) resources for the first-time automated approval is required. If you did not receive the email, wait five minutes for the email and try again.
- Verify that you have the correct inbound restrictions set for the scheduler instance security group.
- Verify the **Stack Name** parameter is less than 24 characters and does not include any capital letters.
- You launched this solution in an AWS Region that has at least three Availability Zones. This solution uses three Availability Zones to maximize the resources users have for submitting jobs.

> **Note**
>
> To help make it easier to troubleshoot, we recommend disabling the **rollback on failure feature** in the AWS CloudFormation template.
> You can also find additional troubleshooting help in the What is Scale-Out Computing on AWS? documentation.

Collection of operational metrics

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When activated, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Base Operating System:** The operating system selected for the solution deployment
- **Unique ID (UUID):** Randomly generated, unique identifier for each solution deployment
- **Timestamp:** Data-collection timestamp
- **Instance Data:** Type or count of the state and type of instances that are provided for by the Amazon EC2 scheduler instance for each job in each AWS Region
- **Keep Forever:** If instances are running when no job is running
- **EFA Support:** If EFA support was selected
- **Spot Support:** If Spot support was invoked for new auto-scaling stacks
- **Stack Creation Version:** The version of the stack that is created or deleted
- **Status:** The status of the stack (`stack_created` or `stack_deleted`)
- **Scratch Disk Size:** The size of the scratch disk selected for each solution deployment
- **Region:** The region where the stack is deployed
- **FSxLustre:** If the job is using FSx for Lustre

Note that AWS will own the data gathered via this survey. Data collection will be subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

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:

   ```
   "Send" : {
   "AnonymousUsage" : { "Data" : "Yes" }
   },
   ```

   to:

```
"Send" : {
"AnonymousUsage" : { "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 Automated Deployment section of this guide.

Source code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Contributors

The following individuals contributed to this document:

- Mickael Crozes
- Walker Stemple
- Ahmed Elzeftawi

Revisions

Date	Change
November 2019	Initial release
April 2020	Integrated AWS Backup for solution durabilit y and AWS Cognito customization option to activate AWS IAM Identity Center (successor to AWS Single Sign-On) for end users. For more information, refer to the CHANGELOG.md file in the GitHub repository.
May 2020	Fixed a bug with the Amazon Linux 2 Amazon Machine Image (AMI) which caused desktop cloud visualization (DCV) to be configured incorrectly.
July 2020	Added REST API, ability to restrict jobs to reserved instances, web-based file system explorer with application specific job submission templates, and highly available Amazon Elasticsearch Service clusters. For more information, refer to the CHANGELOG .md file in the GitHub repository.
March 2021	Release v2.6.1: For a detailed description of the changes to version 2.6.1, refer to the CHANGELOG.md file in the GitHub repository.
October 2021	Release v2.7.0: Added job-shared queue allowing multiple jobs to run on the same Amazon EC2 instance for jobs with similar requirements. Desktop cloud visualization (DCV) session management is now available via REST API. Desktops sessions are now tracked on OpenSearch via soca_desktops

Date	Change
	index. Scale-Out Computing on AWS installer is managed by CDK. For more information, refer to the CHANGELOG.md file in the GitHub repository.
March 2022	Release v2.7.1: Minor update and bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
June 2022	Release v2.7.2: Bug fixes. For more informati on, refer to the CHANGELOG.md file in the GitHub repository.
August 2022	Release v2.7.3: Bug fix. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2023	Release v2.7.4: Minor updates and bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Notices

The Scale-Out Computing on AWS solution retrieves a number of third-party software packages (such as open source packages) from third-party servers at install-time or build-time ("External Dependencies"). The External Dependencies are subject to license terms that you must accept in order to use this solution, including an Affero GPL license. If you do not accept all of the applicable license terms, you should not use this solution. We recommend that you consult your company’s open source approval policy before proceeding.

Provided below is a list of the External Dependencies and the applicable license terms as indicated by the documentation associated with the External Dependencies as of Amazon's most recent review of such documentation.

This information is provided for convenience only. Amazon does not promise that the list or the applicable terms and conditions are complete, accurate, or up-to-date, and Amazon will have no liability for any inaccuracies. You should consult the download sites for the External Dependencies for the most complete and up-to-date licensing information.

Your use of the external dependencies is at your sole risk. In no event will Amazon be liable for any damages, including without limitation any direct, indirect, consequential, special, incidental, or punitive damages (including for any loss of goodwill, business interruption, lost profits or data, or computer failure or malfunction) arising from or relating to the External Dependencies, however caused and regardless of the theory of liability, even if Amazon has been advised of the possibility of such damages. These limitations and disclaimers apply except to the extent prohibited by applicable law.

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 current AWS 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. The responsibilities and liabilities of AWS 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.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.