Scale-Out Computing on AWS

AWS Implementation Guide

Mickael Crozes

November 2019

Last updated: July 2020 (see revisions)

Contents

About This Guide

This implementation guide discusses architectural considerations and configuration steps for deploying Scale-Out Computing in the Amazon Web Services (AWS) Cloud. It includes links to AWS CloudFormation templates that launch, configure, and run 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.

Overview

Amazon Web Services (AWS) enables data scientists and engineers to manage their scale-out workloads such as high-performance computing (HPC) 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 enable 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, enabling 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 enable users to seamlessly authenticate and run jobs on AWS.

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **$370 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 Elasticsearch Service (Amazon ES) cluster, Amazon EFS, AWS Backup, and a NAT Gateway.

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

Architecture Overview

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

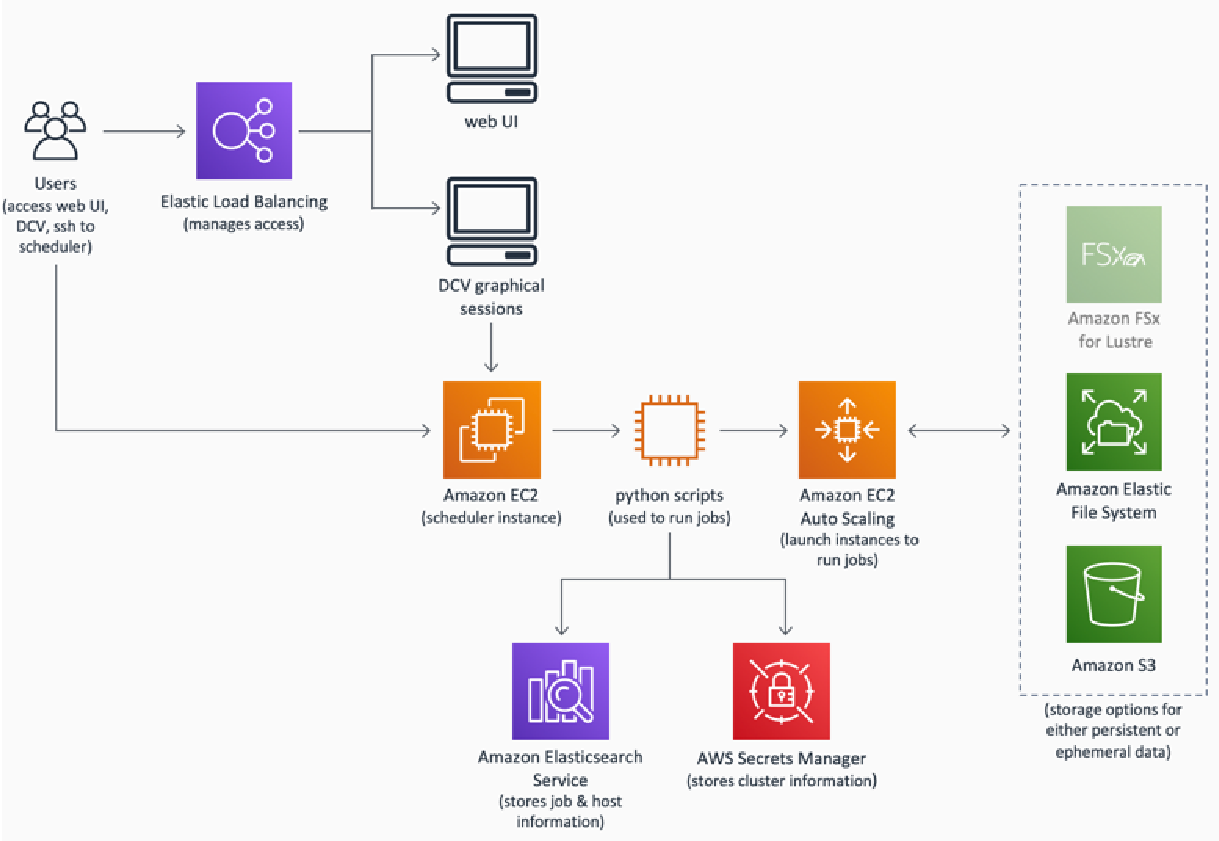

Users
(access web UI, DCV, ssh to scheduler)
Elastic Load Balancing
(manages access)
web UI
DCV graphical sessions
Amazon EC2
(scheduler instance)
python scripts
(used to run jobs)
Amazon EC2 Auto Scaling
(launch instances to run jobs)
Amazon FSx for Lustre
Amazon Elastic File System
Amazon S3
(storage options for either persistent or ephemeral data)
Amazon Elasticsearch Service
(stores job & host information)
AWS Secrets Manager
(stores cluster information)

Figure 1: Scale-Out Computing on AWS architecture

Scale-Out Computing on AWS 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.

At its core, this solution implements a `scheduler` Amazon Elastic Compute Cloud (Amazon EC2) instance, which leverages AWS CloudFormation and Amazon EC2 Auto Scaling to automatically provision the resources necessary to execute cluster user tasks such as scale-out compute jobs and remote visualization sessions.

This solution also deploys Amazon Elastic File System (Amazon EFS) for persistent storage; AWS Lambda functions to verify the required prerequisites and create a default signed certificate for an Application Load Balancer (ALB) to manage access to Desktop Cloud Visualization (DCV) workstation sessions; an Amazon Elasticsearch Service (Amazon ES) cluster to store job and host information; and AWS Secrets Manager to store the solution configuration files. This solution also leverages AWS Identity and Access Management (IAM) roles to enforce least privileged access.

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 enable 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 Elasticsearch Service (Amazon ES) 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 execute 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

High-Performance Computing (HPC) Budgets

This solution helps users and administrators easily manage their HPC budgets. It generates 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 enabled, you must manually enable 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. This 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 files (`/apps` and `/data`). You can also deploy high-speed Amazon EBS SSD-backed disks or Amazon FSx for Lustre that can be used as a scratch location on your compute nodes.

Centralized User Management

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) 18.1.4 job scheduling software. This solution has an AGPLv3 licensing component. For more information, see 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.

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, and AWS Identity and Access Management, 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 Elasticsearch Service (Amazon ES) 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 Backups 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 Images (AMI) to 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 Autoscaling. For more information, see 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.

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, maintain customization and extensibility. If you choose to deploy this solution using a custom installer, see Appendix A.

What We'll Cover

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 parameters: **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

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

> **Note**: You are responsible for the cost of the AWS services used while running this solution. See the section for more details. For full details, see 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 to the right 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, see Appendix A.

> **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. For more information, see 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.
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, see 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 communicate.
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/.

Parameter	Default	Description
		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.
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, see 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_id>/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.

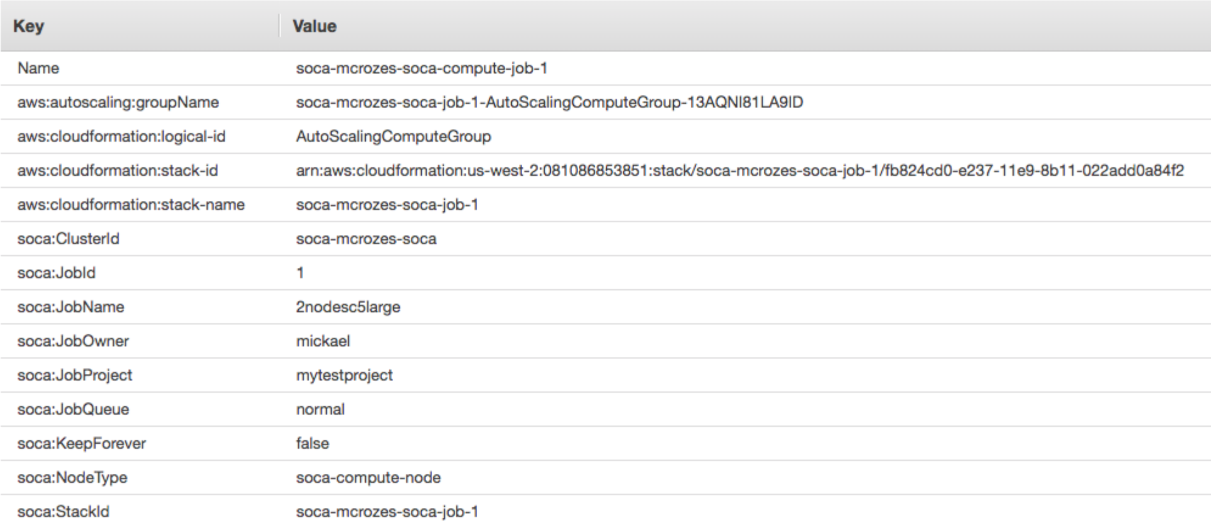
Use the following procedure to create or delete a user through the user interface:

1. In the Admin section, 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 enables 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

Figure 2: 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.

Enable Cost Allocation Tags

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

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

Enable Cost Explorer

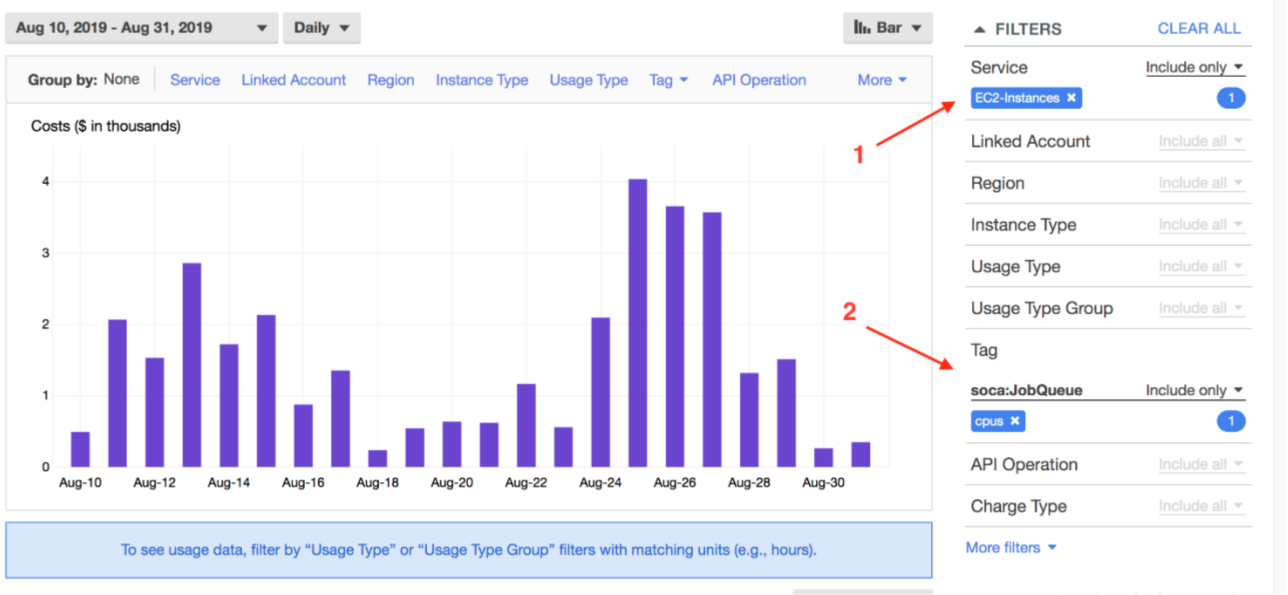
1. In the AWS Cost Management console, select **My Billing Dashboard**
2. Select **Cost Explorer**, then select **Enable Cost Explorer**

Note: Cost Explorer may take up to 24 hours to be enabled.

Query Cost Explorer

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

   The following example shows the **EC2-Instance** cost group by day for the queue named **cpus**:



Service: Include only — EC2-Instances

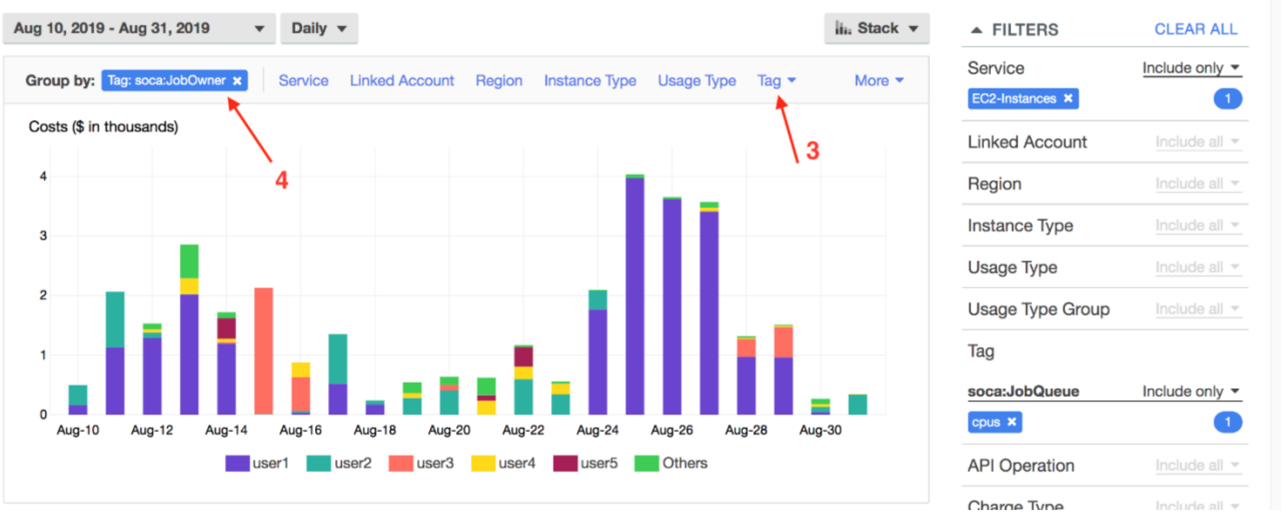
soca:JobQueue: Include only — cpus

Figure 3: 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; Tag: soca:JobQueue: Include only — cpus

Figure 4: 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.

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.

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. Use the following procedure to update or install certificates:

1. In the AWS Management console, navigate to AWS Certificate Manager.
2. In the certificate drop-down, select the applicable **certificate**. Then, select **Actions**, and select **Reimport Certificate**.
3. In the applicable certificate, enter the **Body**, **Private Key**, and **Chain Certificate**. Then, select **Review** to verify the entry is valid.
4. Copy the **Certificate ID**. Note that we recommend binding the Certificate ID to the ALB. For more information, see Upload your SSL certificate.
5. After five minutes, verify your endpoint by using your new SSL certificate.

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 enable SSO authentication to the web interface, see What is Scale-Out Computing on AWS? in the *Scale-Out Computing on AWS Knowledge Base*.

> **Note:** This solution uses an OpenLDAP service to manage user accounts. We recommend enabling a connection to an external LDAP directory with encryption (LDAPS).

Additional Resources

AWS services

- AWS CloudFormation
- AWS Lambda
- AWS Auto Scaling
- AWS Cost Explorer
- AWS Budgets
- Amazon Elastic File System
- AWS Secrets Manger
- AWS Identity and Access Management
- Amazon EC2
- Application Load Balancer
- AWS Backup

Other documentation

- Scale-Out Computing on AWS Knowledge Base

Appendix A: Launch the Stack (Custom Installer)

For customers who want to maintain customization and extensibility, this solution allows you to deploy the AWS CloudFormation template using a custom installer in your hosted repository for production environments.

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

What We'll Cover

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

Step 1. Download the Solution Template

- Download the AWS CloudFormation template

Step 2. Create and Upload the Build

- Create and upload the build

Step 3. Launch the Stack

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

Prerequisites

Before deploying this solution, you must create a new Amazon Simple Storage Service (Amazon S3) bucket in your AWS account. Or, you can use an existing Amazon S3 bucket.

Step 1: 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 2: Create and Upload the Build

Use the following procedure to create and upload the build:

1. Using Python2 or Python3, in a terminal window run the following command:

```
$SOCA_DIR/source/manual_build.py
```

This command will create a compressed installer folder on your local machine. For example, `soca-installer-`*`<xxxx.tar>`*.

2. Navigate to the Amazon S3 console, and upload the newly created installer folder to your Amazon S3 bucket.

> **Note:** Copy the Amazon S3 folder name and the root folder name for the installer folder you uploaded. You will use these names when deploying the AWS CloudFormation template.
>
> You can create multiple versions of this solution in the same Amazon S3 bucket. However, the name for each installer folder must be unique.

Step 3. Launch the Stack

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

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

1. Sign in to the AWS Management console, select **Choose a file**
2. Navigate to the installer folder you downloaded in Step 3, select the `scale-out-computing-on-aws.template`
3. Launch the template, by following the deployment procedure from step 3.

Appendix B: 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 Zone. 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 project documentation.

Appendix C: Collection of Operational Metrics

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 enabled, 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:

```
Mappings:
  Send:
    AnonymousUsage:
      Data: Yes
```

to

```
Mappings:
  Send:
    AnonymousUsage:
      Data: No
```

Source Code

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

Document Revisions

Date	Change
November 2019	Initial release
April 2020	Integrated AWS Backup for solution durability and AWS Cognito customization option to enable Single-Sign On (SSO) for end users. See the list of v2 updates here: https://github.com/awslabs/scale-out-computing-on-aws/blob/master/CHANGELOG.md
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. See the full list of v2.5 updates here: https://github.com/awslabs/scale-out-computing-on-aws/blob/master/CHANGELOG.md

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.