Deploying in Amazon Web Services

Prerequisites and Guidelines

You must have reviewed and completed the general prerequisites described in the Deployment Overview.

In addition, the following apply when deploying in the Amazon Web Services (AWS):

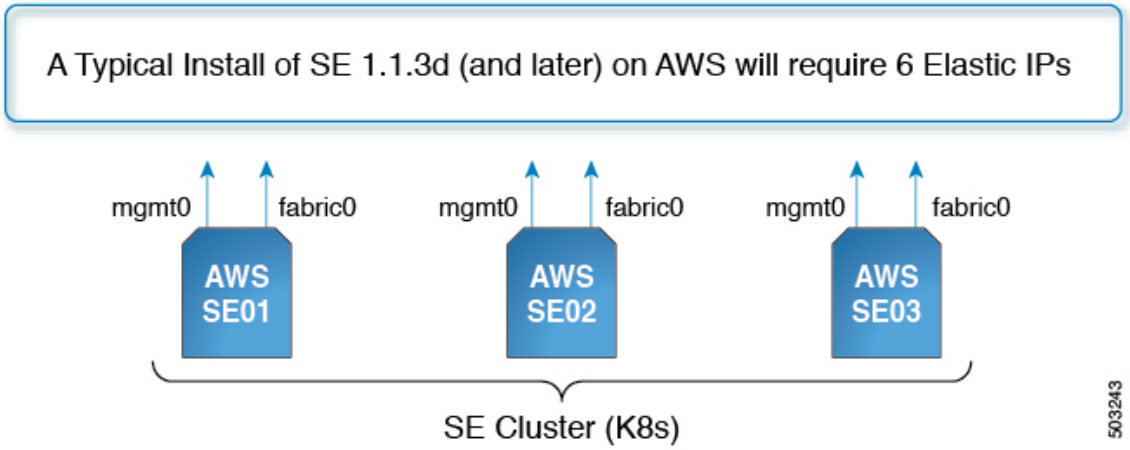
- You must have appropriate access privileges for your AWS account.

  You must be able to launch multiple instances of Elastic Compute Cloud (`m5.2xlarge`) to host the Application Services Engine cluster.

- At least 6 AWS Elastic IP addresses.

  A typical Application Services Engine deployment in AWS requires 6 AWS Elastic IP addresses as shown in the following figure:

  Figure 1: Elastic IPs



- Detailed information about AWS configuration is outside the scope of this document, but in short, to create a VPC:

- In your AWS console, navigate to **Computer** > **EC2**.
- In the EC2 Dashboard, click **Network & Security** > **Elastic IPs** and note how many Elastic IPs are already being used.
- In the EC2 Dashboard, click **Limits** and note the maximum number of **EC2-VPC Elastic IPs** allowed.

  Subtract the number of IPs already being used from the limit to get. Then if necessary, click **Request limit increase** to request additional Elastic IPs.

- You must create a VPC (Virtual Private Cloud).

  A VPC is an isolated portion of the AWS cloud for AWS objects, such as Amazon EC2 instances. Detailed information about AWS configuration is outside the scope of this document, but in short, to create a VPC:

  - In your AWS console, navigate to **Networking & Content Delivery Tools** > **VPC**.
  - In the VPC Dashboard, click **Your VPCs** and choose **Create VPC**. Then provide the **Name Tag** and **IPv4 CIDR block**.

    The CIDR block is a range of IPv4 addresses for your VPC and must be in the `/16` to `/28` range. For example, `10.9.0.0/16`.

- You must create an Internet Gateway and attach it to the VPC.

  Internet Gateway is a virtual router that allows a VPC to connect to the Internet. Detailed information about AWS configuration is outside the scope of this document, but in short, to create an Internet Gateway:

  - In the VPC Dashboard, click **Internet Gateways** and choose **Create internet gateway**. Then provide the **Name Tag**.
  - In the **Internet Gateways** screen, select the Internet Gateway you created, then choose **Actions** > **Attach to VPC**. Finally, from the **Available VPCs** dropdown, select the VPC you created and click **Attach internet gateway**.

- You must create a routes table.

  Routes table is used for connecting the subnets within your VPC and Internet Gateway to your Application Services Engine cluster. Detailed information about AWS configuration is outside the scope of this document, but in short, to create a routes table:

  - In the VPC Dashboard, click **Route Tables**, choose the **Routes** tab, and click **Edit routes**.
  - In the **Edit routes** screen, click **Add route** and create a `0.0.0.0/0` destination. From the **Target** dropdown, select `Internet Gateway` and choose the gateway you created. Finally, click **Save routes**.

- You must also create a key pair.

  A key pair consists of a private key and a public key, which are used as security credentials to verify your identity when connecting to an EC2 instance. To create a key pair:

  - Navigate to **All services** > **Compute** > **EC2**.
  - In the EC2 Dashboard, click **Network & Security** > **Key Pairs**. Then click **Create Key Pairs**.
  - Provide a name for your key pair, select the **pem** file format, and click **Create key pair**.

This will download the `.pem` private key file to your system. Move the file to a safe location, you will need to use it the first time you log in to an EC2 instance's console.

By default only PEM-based login is enabled for each node. If you'd like to be able to SSH into the nodes using a password, you will need to explicitly enable password-based logins. You can do that by logging into each node separately using the PEM file the first time, then executing the following command:

```
# acidiag login prompt enable
```

Deploying the Cisco Application Services Engine in AWS

This section describes how to deploy Cisco Application Services Engine cluster in Amazon Web Services (AWS).

Before you begin

- You have familiarized yourself with deployment options, recommendations, Application Services Engine connectivity described in Deployment Overview and Requirements.
- Ensure that you meet the requirements and guidelines described in Prerequisites and Guidelines, on page 1.

Step 1 Subscribe to Cisco Application Services Engine product in AWS Marketplace.

a) Log into your AWS account and navigate to the AWS Management Console

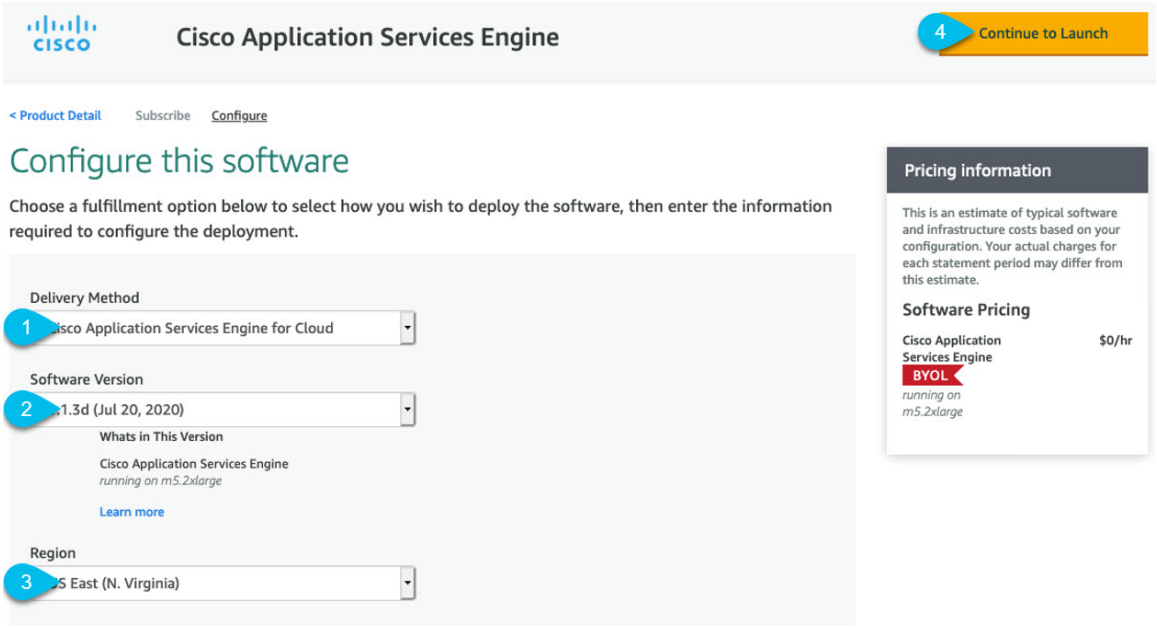
The Management Console is available at https://console.aws.amazon.com/.

b) Navigate to **Services** > **AWS Marketplace Subscriptions**.
c) Click **Manage Subscriptions**.
d) Click **Discover products**.
e) Search for **Cisco Application Services Engine** and click the result.
f) In the product page, click **Continue to Subscribe**.
g) Click **Accept Terms**.

It may take a couple of minutes for the subscription to be processed.

h) Finally click **Continue to Configuration**.

Step 2 Select software options and region.



a) From the **Delivery Method** dropdown, select `Cisco Application Services Engine for Cloud`.

b) From the **Software Version** dropdown, select the version you want to deploy.

We recommend version 1.1.3d or later.

c) From the **Region** dropdown, select the regions where the template will be deployed.

This must be the same region where you created your VPC.

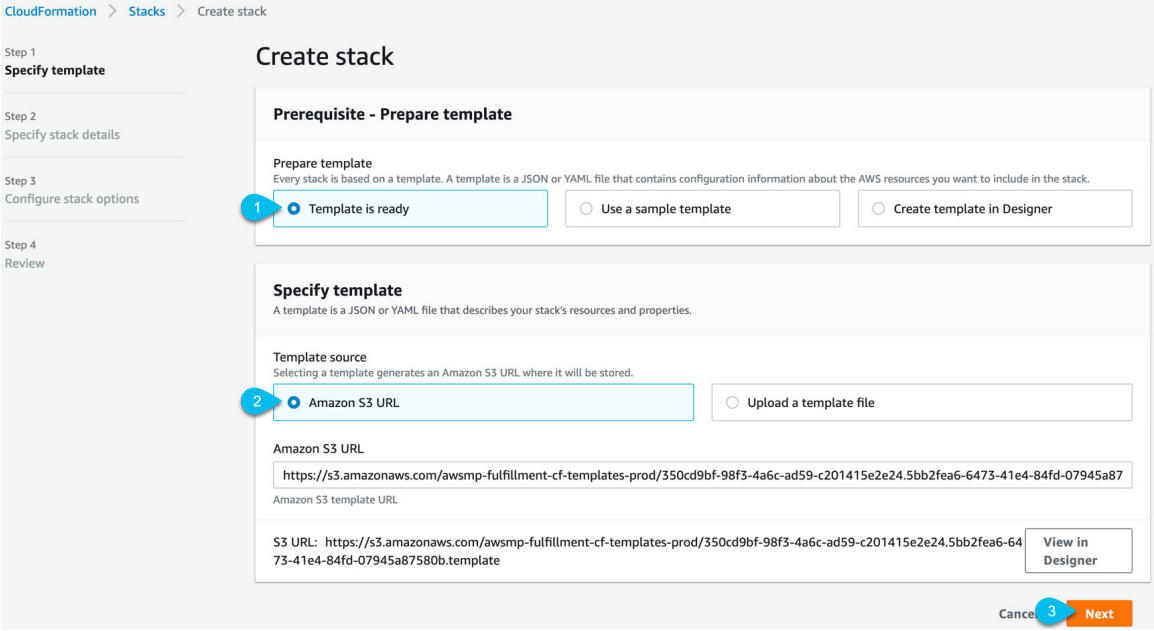
d) Click the **Continue to Launch**.

The product page appears, which shows a summary of your configuration and enables you to launch the cloud formation template.

Step 3 From the **Choose Action**, select `Launch CloudFormation` and click **Launch**.

The **Create stack** page appears.

Step 4 Create stack.



a) In the **Prerequisite - Prepare template** area, select `Template is ready`.

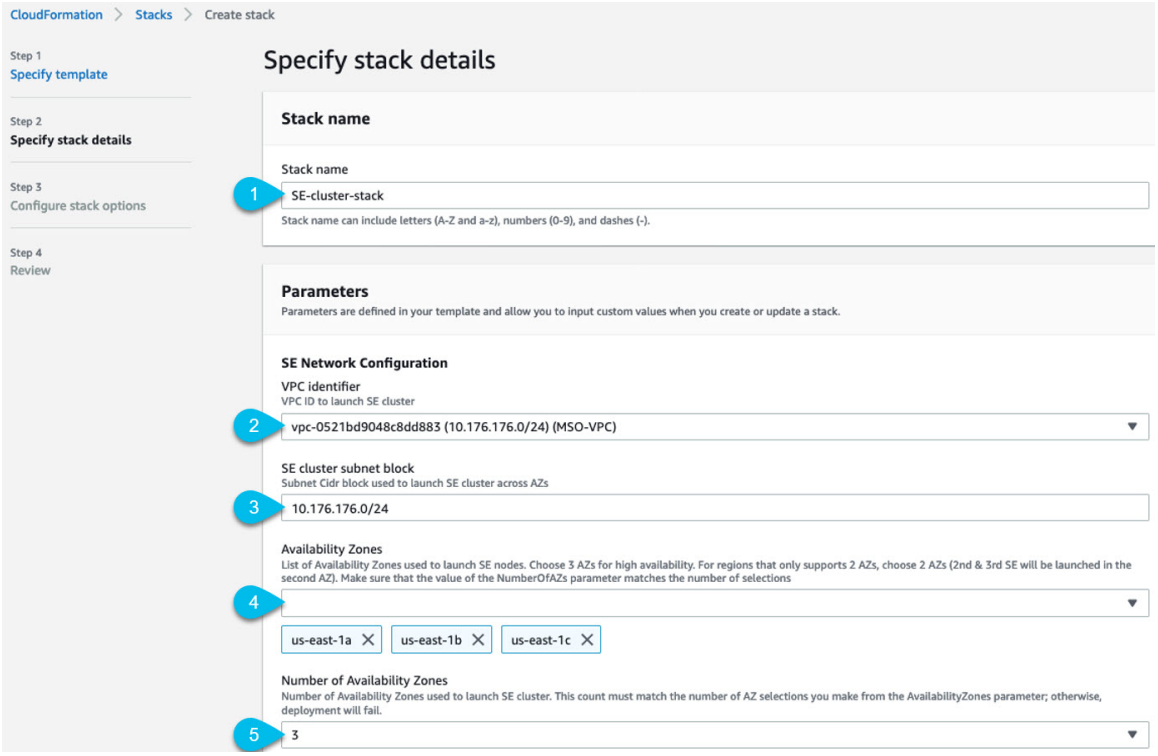
b) In the **Specify Template** area, select `Amazon S3 URL` for the template source.

The template will be populated automatically.

c) Click **Next** to continue.

The **Specify stack details** page appears.

Step 5 Specify stack details.



a) Provide the **Stack name**.

b) From the **VPC identifier** dropdown, select the VPC you created.

For example, `vpc-038f83026b6a48e98(10.176.176.0/24).`

c) In the **SE cluster Subnet block**, provide the VPC subnet CIDR block.

Choose the subnet from the VPC CIDR that you defined. You can provide a smaller subnet or use the whole CIDR.

For example, `10.176.176.0/24`.

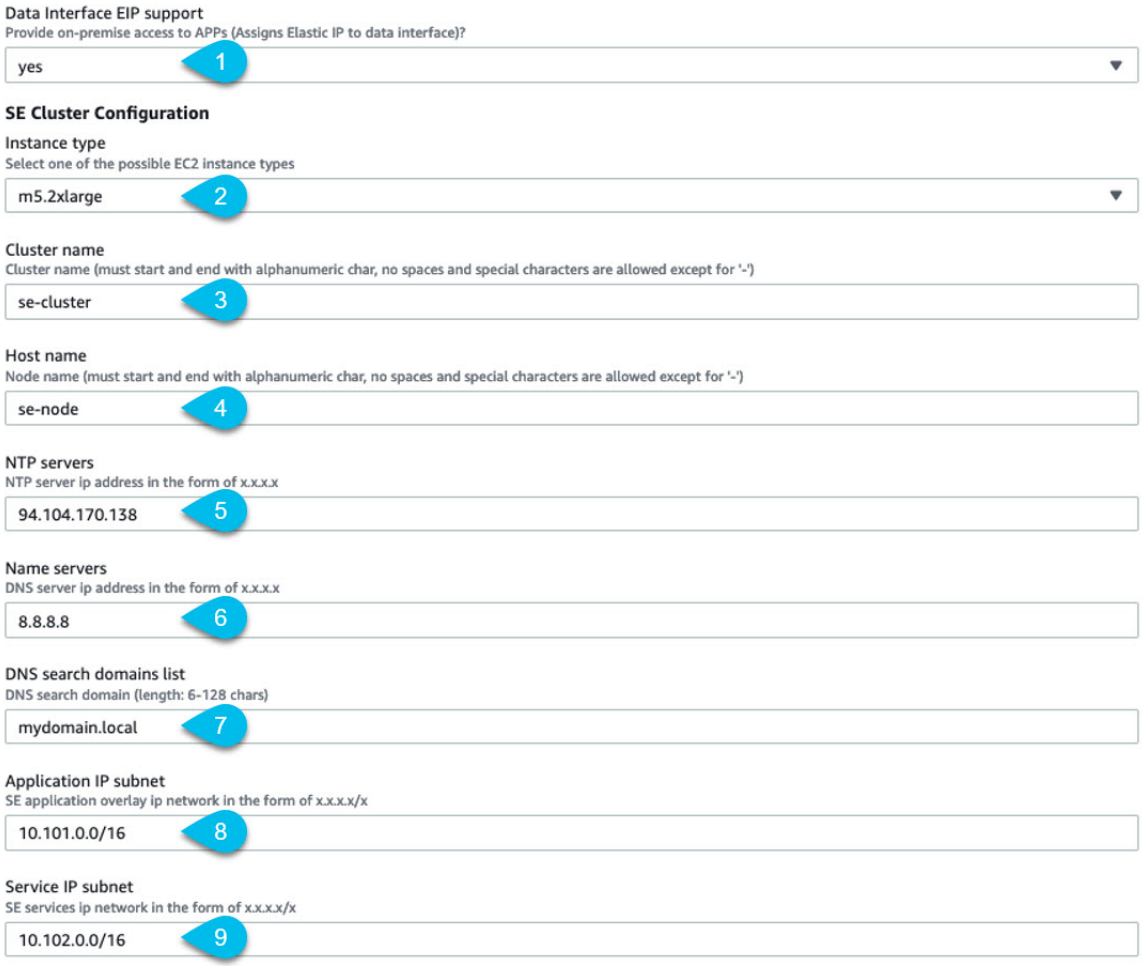
d) From the **Availability Zones** dropdown, select one or more available zones.

We recommend you choose 3 availability zones. For regions that support only 2 availability zones, 2nd and 3rd nodes of the cluster will launch in the second availability zone.

e) From the **Number of Availability Zones** dropdown, select the number of zones you added in the previous substep.

Ensure that the number matches the number of availability zones you selected in the previous substep.

Provide the rest of the node information.



a) Enable **Data Interface EIP support**.

   This field enables external connectivity for the node. External connectivity is required for communication with Cisco ACI fabrics outside AWS.

b) From the **Instance type**, select `m5.2xlarge`
c) Provide the **Cluster name**.

   The cluster name must be the same across all nodes you deploy.

d) Provide the **Host name**.

   The host name must be unique for each node.

e) Provide the **NTP servers** information.
f) Provide the **Name servers** information.
g) (Optional) Provide the **DNS search domains list**.
h) Provide the **Application IP subnet**.

   The application overlay network defines the address space used by the application's services running in the Application Services Engine. This must be a `/16` subnet.

   For example, `10.101.0.0/16`.

Note Communications between containers deployed in different Application Services Engine nodes is VXLAN-encapsulated and uses the data interfaces IP addresses as source and destination. This means that the Application Overlay and Service Overlay addresses are never exposed outside the data network and any traffic on these subnets is routed internally and does not leave the cluster nodes. As such, when configuring these networks, ensure that they are unique and do not overlap with any existing networks or services you may need to access from the Application Services Engine cluster nodes

i) Provide the **Service IP subnet**.

The services network is an internal network used by the Application Services Engine and its processes. This must be a `/16` subnet.

For example, `10.102.0.0/16`.

Finally, provide the login and access information.

a) In the **Password** fields, provide the password.

This password will be used for the Application Services Engine's `rescue-user` login, as well as the initial password for the GUI's `admin` user.

b) From the **SSH key pair** dropdown, select the key pair you created.

c) In the **Access control** field, provide the external network allowed to access the cluster.

For example, `0.0.0.0/0` to be able to access the cluster from anywhere.

d) Click **Next** to continue.

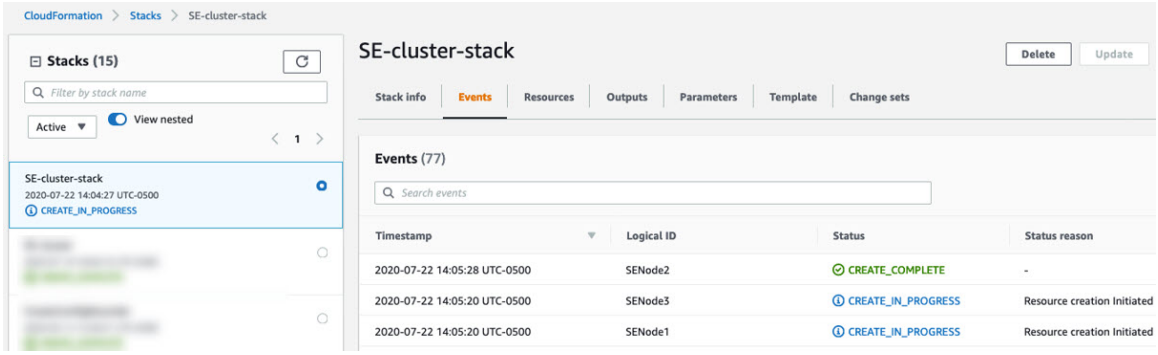
Step 6 In the **Advanced options** screen, simply click **Next**.

Step 7 In the **Review** screen, verify template configuration and click **Create stack**.

Step 8 Wait for the instance deployment to complete, then start the instance.

You can view the status of the instance deployment in the **CloudFormation** page, for example `CREATE_IN_PROGRESS`. You can click the refresh button in the top right corner of the page to update the status.

When the status changes to `CREATE_COMPLETE`, you can proceed to the next step.



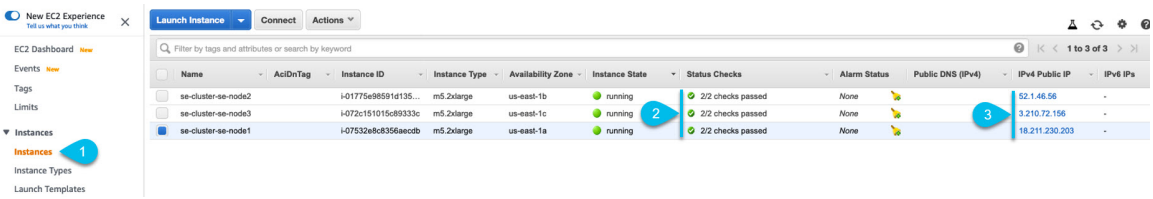
Step 9 Verify that the cluster is healthy.

It may take up to 30 minutes for the cluster to form and all the services to start.

After all three nodes' status is `CREATE_COMPLETE`, proceed with the following substeps to verify cluster health.

a) Verify that the AWS EC2 instances are up and running.

Navigate to **Services** > **EC2**. Then confirm that the **Status Checks** tab displays `2/2` checks.



b) Login in to one of the nodes.

You will need to use the private key `.pem` file you downloaded when creating a key pair in the following command:

```
$ ssh -i <pem-file-name>.pem rescue-user@<node-ip-address>
```

c) Verify that the cluster is up and running.

You can check the current status of cluster deployment by logging in to any of the nodes and running the `acidiag health` command.

While the cluster is converging, you may see the following outputs:

```
$ acidiag health
k8s install is in-progress
```

```
$ acidiag health
k8s services not in desired state - [...]
```

```
$ acidiag health
k8s: Etcd cluster is not ready
```

When the cluster is up and running, the following output will be displayed:

```
$ acidiag health
All components are healthy
```

d) Log in to the Application Services Engine GUI.

After the cluster becomes available, you can access it by browsing to any one of your nodes' management IP addresses. The default password for the `admin` user is the same as the `rescue-user` password you chose for the first node of the Application Services Engine cluster.

When you first log in, you will be prompted to change the password.

Step 10 (Optional) Enable password-based login.

By default only PEM-based login is enabled for each node. If you'd like to be able to SSH into the nodes using a password, you will need to explicitly enable password-based logins. You can do that by logging into each node separately using the PEM file the first time, then executing the following command:

```
# acidiag login prompt enable
```