
Stage 1 : Creating EKS Cluster over private network using CloudFormation template

Objective : The objective of this document is to provide step-by-step instructions for users to create EKS cluster, VPC Private network, LAC Writer nodes over K8s deployment. Ensure that you meet requirements. If any issues or errors occur during the process, please consult us or contact the AWS support team for assistance.

Prerequisites

1. Access to the AWS Management Console.
2. CloudFormation template file (e.g., CFT-EKS-LN-1.0.yaml) prepared for EKS stack creation.

Procedure

Step 1: Access the AWS Management Console

1. Open a web browser and navigate to the AWS Management Console (<https://console.aws.amazon.com>).
2. Enter your login credentials to access your AWS account.

Step 2: Navigate to the CloudFormation Service

1. Once logged in, search for “CloudFormation” in the AWS Management Console search bar.
2. Click on the “CloudFormation” service from the search results to open the CloudFormation dashboard.

Step 3: Create a New Stack

1. In the CloudFormation dashboard, click on the “Create stack” button to start the stack creation process.
2. Select “Upload a template file” and click on the “Choose file” button.
3. Browse and select the CloudFormation template file (CFT-EKS-LN-1.0.yaml) from your local system.
4. Click on the “Next” button to proceed.

Step 4: Specify Stack Details

1. Provide a stack name that identifies your EKS stack (e.g., MyEKSSStack).
2. Fill in any required parameters prompted by the CloudFormation template.(KeyPair, Instance Type)

3. Review the other options and settings as per your requirements.
4. Click on the “Next” button to proceed.

Step 5: Configure Stack Options (Optional)

1. If required, configure additional stack options such as tags, permissions, and advanced settings.
2. Review the options and make necessary changes.
3. Click on the “Next” button to proceed.

Step 6: Review and Create the Stack

1. Review the stack details, parameters, and options on the final page.
2. Double-check that all the information is accurate and meets your requirements.
3. Click on the “Create stack” button to start creating the EKS stack.

Step 7: Monitor Stack Creation Progress

1. You will be redirected to the stack detail page, where you can monitor the stack creation progress.
2. Refresh the page periodically to see the status updates.
3. Wait until the stack reaches the “CREATE_COMPLETE” status. This may take a few minutes.(Approx 15 mins)

Step 8: Validation: Access the Created EKS Cluster

1. Once the stack creation is complete, navigate to the AWS Management Console homepage.
2. Search for “EKS” in the search bar and click on the “Amazon EKS” service from the search results.
3. In the Amazon EKS console, you will find your created EKS cluster.

Now proceed to Stage 2: Creating Identity Provider for EKS Cluster.

Stage 2 : Creating Identity Provider for EKS Cluster

Objective: Creating an identity provider for an Amazon Elastic Kubernetes Service (EKS) cluster. The identity provider is necessary for authenticating users and granting them access to the EKS cluster.

Prerequisites

- Access to the AWS Management Console
- A deployed EKS cluster
- Gathered information from the Cloudformation EKS stack outputs:
 - OIDC Url: [Enter the OIDC URL obtained from the EKS stack output]
 - OIDC Audience Name: [Enter the OIDC Audience Name obtained from the EKS stack output]

Procedure

Step 1: Sign in to the AWS Management Console

1. Open a web browser and navigate to the AWS Management Console (<https://console.aws.amazon.com>).
2. Enter your login credentials to access your AWS account.

Step 2: Navigate to the IAM (Identity and Access Management) service.

1. Once logged in, search for “IAM” in the AWS Management Console search bar.
2. Click on the “IAM” service from the search results to open the IAM dashboard.

Step 3: Creating an Identity Provider

1. In the IAM dashboard, click on “Identity providers” in the left-hand navigation pane under Access management

Step 4: Add an Identity provider

1. Click on the “Add provider” button.
2. On the “Add provider” page, select “OpenID Connect” as the provider type.

Step 5: Enter the following details:

- Provider URL: (e.g., "<https://oidc.eks.us-west-1.amazonaws.com/id/292194D91A482781SDED2235S>").
- Audience: [sts.amazonaws.com]

Step 6: Collect the Fingerprint

1. Click on the “Get Thumbprint”.
2. Now Click on the “Add Provider” button.

Step 7: Validation

1. If everything appears correct, You will see your Identity provider in the home page.

You are now ready to proceed to the next step, which involves running a script in the AWS Cloud Shell. The `.env` file will be used in the script to configure the deployment.

Note: Ensure that you keep the `.env` file secure and do not share it with unauthorized individuals. It contains sensitive information related to your EKS cluster.

Now proceed to Stage 3: Running the Script in AWS Cloud Shell.

Stage 3 : Running Script in AWS Cloud Shell

Objective: Run the deployment script in AWS Cloud Shell using the uploaded `.env` file. Ensure that you have obtained the correct `.env` file and filled up respective information from the CloudFormation EKS stack outputs and review the `.env` file for node customization requirements. The script is responsible for deploying resources related to the EKS cluster and requires the `.env` file containing the necessary information.

Prerequisites

1. Access to the AWS Management Console.
2. CloudFormation stack for EKS cluster successfully created.
3. Identity provider Created.
4. The `.env` file with required information obtained from the CloudFormation stack outputs.

Procedure

Step 1: Access the AWS Management Console

1. Open a web browser and navigate to the AWS Management Console (<https://console.aws.amazon.com>).
2. Enter your login credentials to access your AWS account.

Step 2: Open AWS Cloud Shell

1. Once logged in, search for “CloudShell” in the AWS Management Console search bar.
2. Click on the “AWS CloudShell” service from the search results to open the Cloud Shell.

Step 3: Upload the `.env` File

1. In the AWS Cloud Shell terminal, click on the “Upload/Download files” button in the toolbar.
2. Browse and select the `.env` file containing the required information (e.g., IP address, EKS cluster name, region, Node Name, Email address).
3. Wait for the file to upload successfully.

Step 4: Run the Deployment Script

1. In the AWS Cloud Shell terminal, execute the following command to download and run the deployment script desired network: select between (mainnet / testnet)

```
source <(curl -s -L https://raw.githubusercontent.com/ravinayag/lacchain-eks/master/deploy.sh) testnet
```

2. The script will start running and deploying the necessary resources for the EKS cluster.
3. Monitor the script execution for any errors or prompts for user input, if applicable.
4. Wait for the script to complete the deployment process.

Step 5: Verify the Deployment

1. Once the script execution is complete, verify that all the required resources for the EKS cluster have been successfully deployed.
2. Check the AWS Management Console or run appropriate commands to validate the deployment.

Bonus Stage : Operational Commands

To perform operational commands and retrieve specific information about your EKS cluster, you can utilize the `ops.sh` script. This script provides various commands that can be executed in the AWS Cloud Shell or a terminal environment. Here are the available commands:

- `getPodRestart` : Use this command to restart the writer pods in the cluster.
- `getNodeId` : Use this command to obtain the Node ID of the Writer Node.
- `getNodeAddress` : Use this command to fetch the address of the Writer node.
- `getConnectionStatus` : Use this command to check the status of the nodes's connection.

- `getbesuLogs` : Use this command to view the besu container logs

To execute any of these commands, follow these steps:

Step 1: Make sure you have the `ops.sh` script available. If not follow stage 3 again.

Step 2: Open the AWS Cloud Shell or a terminal environment.

Step 3: Navigate to the location where the `ops.sh` script is located.

Step 4: Run the following command to source the script:

```
source ops.sh
```

Once the script is sourced, you can run any of the commands mentioned above by simply typing the command in the AWS Cloud Shell and pressing Enter. The output will be displayed accordingly.

Note : Ensure that you have appropriate permissions and access to the EKS cluster before running these commands.

Repo file Contents

This repository contains the following files:

- `.env` : An environment variable file for LAC-Chain.
- `deploy.sh` : A script to deploy the Kubernetes resources for the mainnet or testnet.
- `ops.sh` : A script to operate the Kubernetes resources and LACChain.
- `CFT-EKS-LN-1.0.yaml` : CloudFormation template for creating an EKS Cluster.
- `LAC-K8s-pro-testnet-1.0.yaml` : Kubernetes resources for the LAC Pro testnet.
- `LAC-K8s-mainnet-1.0.yaml` : Kubernetes resources for the LAC mainnet.
- `Readme.md` : A SOP document that provides instructions for creating and operating the files in this repository.

Please refer to the `Readme.md` file for detailed information and step-by-step instructions on how to use these files.