Enterprise Cluster

Set up and configure an Enterprise cluster

# Objectives

Enterprise clusters are designed for production-ready environments that require private endpoint networking but without manual scalability and other advanced features of a dedicated cluster. In this lab, you will create an Enterprise cluster and connect it via a private link to an AWS VPC.

# Labs

## Create an AWS VPC

* Confluent Cloud Enterprise Clusters currently support a subset of AWS regions. Pick your AWS region out of the following:
* af-south-1 (Cape Town)
* ap-east-1 (Hong Kong)
* ap-south-1 (Mumbai)
* ap-southeast-1 (Singapore)
* ap-southeast-2 (Sydney)
* ca-central-1 (Canada Central)
* eu-central-1 (Frankfurt)
* eu-west-1 (Ireland)
* eu-west-2 (London)
* me-south-1 (Bahrain)
* me-central-1 (UAE)
* us-east-1 (N. Virginia)
* us-east-2 (Ohio)
* us-west-2 (Oregon)
* Create a VPC in the chosen region with an Internet Gateway, a route to the Internet, and a small instance (Jumphost) to which you can connect.  
  You will also need 3 subnets across the first three availability zones (AZ).  
    
  You should use the Terraform scripts at <https://github.com/sknop/simple-vpc> if you are still getting familiar with AWS. Here are the instructions:
  + git clone <https://github.com/sknop/simple-vpc>
  + cd simple-vpc
  + cd aws
  + Copy terraform.tfvars.template to terraform.tfvars
  + Edit terraform.tfvars and pick the AWS region of your choice
  + Adjust other parameters as required
  + Run terraform init -upgrade
  + Verify your settings with terraform plan
  + Run terraform apply
  + Check the output for clues on the details of your setup

## Create a dedicated cluster first

The next lab ([Lab 7: Dedicated Cluster and Private Links](https://docs.google.com/document/u/1/d/1R4nhvzaQEWnnj7IFF2fBFYrTUXl6m-KoZWjQMHfcByg/edit)) requires a dedicated cluster that will take a couple of hours to complete, therefore we will set this cluster up beforehand.

* Create a dedicated cluster in your Confluent Cloud Console (UI)
  + Choose 2 CKUs (for a multi-zone cluster)
  + Pick AWS and the region of your VPC, and Multi-Zone availability
  + Choose PrivateLink for your networking
    - Create a new network, enabling private DNS resolution. Give it a name.
    - Ensure you pick the same 3 network zones you chose for your VPC. Most regions only have 3 AZs, but some have more.
  + Choose Automatic for your encryption key - unless you know exactly what you are doing with AWS key management.
  + Skip payment
  + Name the cluster, then launch it

This will take 1-2 hours to provision. In the meantime …

## Create the Enterprise Cluster

* Create an Enterprise cluster in your Confluent Cloud Console (UI).
  + Choose the same AWS region where you created your VPC instance
  + Choose 99.99% SLA (this invokes deployment across 3 AZ)
  + Skip the networking configuration for now
  + Skip the payment
  + Pick a name for your cluster and launch it

## Connect the cluster to AWS

* In your new cluster, navigate to the Network Management panel (it is flagged with an exclamation mark and a yellow triangle)
* Click on “Create network configuration”
  + Choose “From your VPC or VNet to Confluent Cloud” (left panel)
  + Give the Gateway a name
  + Choose the correct cloud provider and region (usually AWS)
  + Submit the changes
* Follow the instructions for configuring the Gateway via the AWS Console
  + Click on AWS Console instructions for detailed documentation
  + Your DNS hostnames and resolution are already enabled
  + Copy the PrivateLink Service ID (for AWS)
* In AWS, create a VPC endpoint with referencing your Private Link Attachment
  + In the VPC Service, navigate to Endpoints
  + Create endpoint
  + Choose a name
  + Choose the “Endpoint services that use NLBs and GWLBs” type
  + Enter the service name copied from the “Configure Gateway” Panel
  + Select your VPC
  + Disable DNS name (if enabled)
  + Choose all three subnets, and pick the private subnet ids (if available)
  + Pick a suitable security group or create a security group following the instructions in the documentation.
* Complete the link by entering the newly created attachment ID in the Network Management panel (in step 4).
* Give it a name
* Finally, complete the connection by updating your DNS entry for your cluster as per [documentation](https://docs.confluent.io/cloud/current/networking/aws-platt.html?ajs_aid=f43c3624-9cd6-423d-ba05-b381750f0427&ajs_uid=198337#set-up-dns-resolution).

## Test your cluster

* SSH into your jumphost in the VPC you created and to which you set up the private link endpoint.
* Verify that you can connect to your new cluster from your jumphost.
  + Use openssl s\_client -connect <your-kafka-endpoint> for a first test
  + Create a client configuration for a C++ client including an API Key
  + Install kafkacat on your jumphost. If you used Ubuntu, you can use  
    sudo apt install kafkacat.
  + Test the connection with kafkacat -F <your-property-file> -L.
  + Install Java (11 or 17) on your jumphost.
  + Install the Kafka Java Clients (kafka-topics, kafka-console-producer, etc.) on your jumphost
  + Create a client configuration for Java
  + Create a topic, the produce and consume from it on your Enterprise cluster.

## Proxy configuration

If you navigate in the UI to the topic, you can see that the topics are not accessible via the console. The problem is that the console needs to communicate with the cluster in a different network, and this network is only accessible from your VPC via the private endpoint.

In a production environment, a network administrator could set up a VPN to your private VPC and set up a DNS entry to reroute the traffic. For this bootcamp, we can use a workaround via Proxy.

Either:

* Install a proxy service on your jumphost, for example, haproxy. Configure this proxy to point to your Kafka cluster. The documentation below will help you figure out the required settings.

**Or:**

* Alternatively, you can use a dynamic (SOCKS v5) proxy.
  + For this, you need to set up a dynamic proxy by adding the option “-D port” to your SSH command when connecting to your jumphost, for example:  
      
    ssh -D 8081 jumphost
  + Install a dynamic proxy in your browser, for example, Standard FoxyProxy (<https://getfoxyproxy.org/>).
  + Configure the Proxy to point to localhost:port (here, port 8081) with a URL pattern that matches your cluster and a regular expression at the beginning and end, for example like this  
      
    [https://.\*lkc-m2yg51.dom1w4k12gl.eu-west-1.aws.confluent.cloud\*](about:blank)  
    (Do not forget to enable your Proxy for your Patterns)
* Test the proxy solution with your cluster
* After completing these labs, you can delete the Enterprise cluster again. It consumes resources and hence credits.

# References

<https://docs.confluent.io/cloud/current/clusters/cluster-types.html#enterprise-clusters>

<https://docs.confluent.io/cloud/current/networking/aws-platt.html?ajs_aid=f43c3624-9cd6-423d-ba05-b381750f0427&ajs_uid=198337#aws-privatelink-for-enterprise-clusters>

<https://docs.confluent.io/platform/current/installation/installing_cp/zip-tar.html#get-the-software>

<https://docs.confluent.io/cloud/current/networking/ccloud-console-access.html#configure-a-proxy>

# Expected Outcomes

Create an Enterprise Cluster and successfully connect it via Private Link to your AWS VPC.

Verify access via OpenSSL and Kafka tools.

Successfully create a Proxy solution to enable adding and modifying topics in the Confluent Cloud UI.

# Check your understanding

This colour marks advanced questions.

* How does an Enterprise cluster differ from a Basic or Standard cluster?
  + Are there any similarities?
* Which networking options are available for an Enterprise cluster?
* Why would you pick an Enterprise cluster over a dedicated cluster?
* Why are the topics not visible in the Console (UI) unless you create a Proxy?

# 