CISCO Live ! | Amsterdam, NL
February 9-13,2026

LTRENS-2891

Unified Policy Control with ISE, ACI, and SDA: Leveraging SGTs for Scalable Security

Shivam Kumar, Achintya Murali

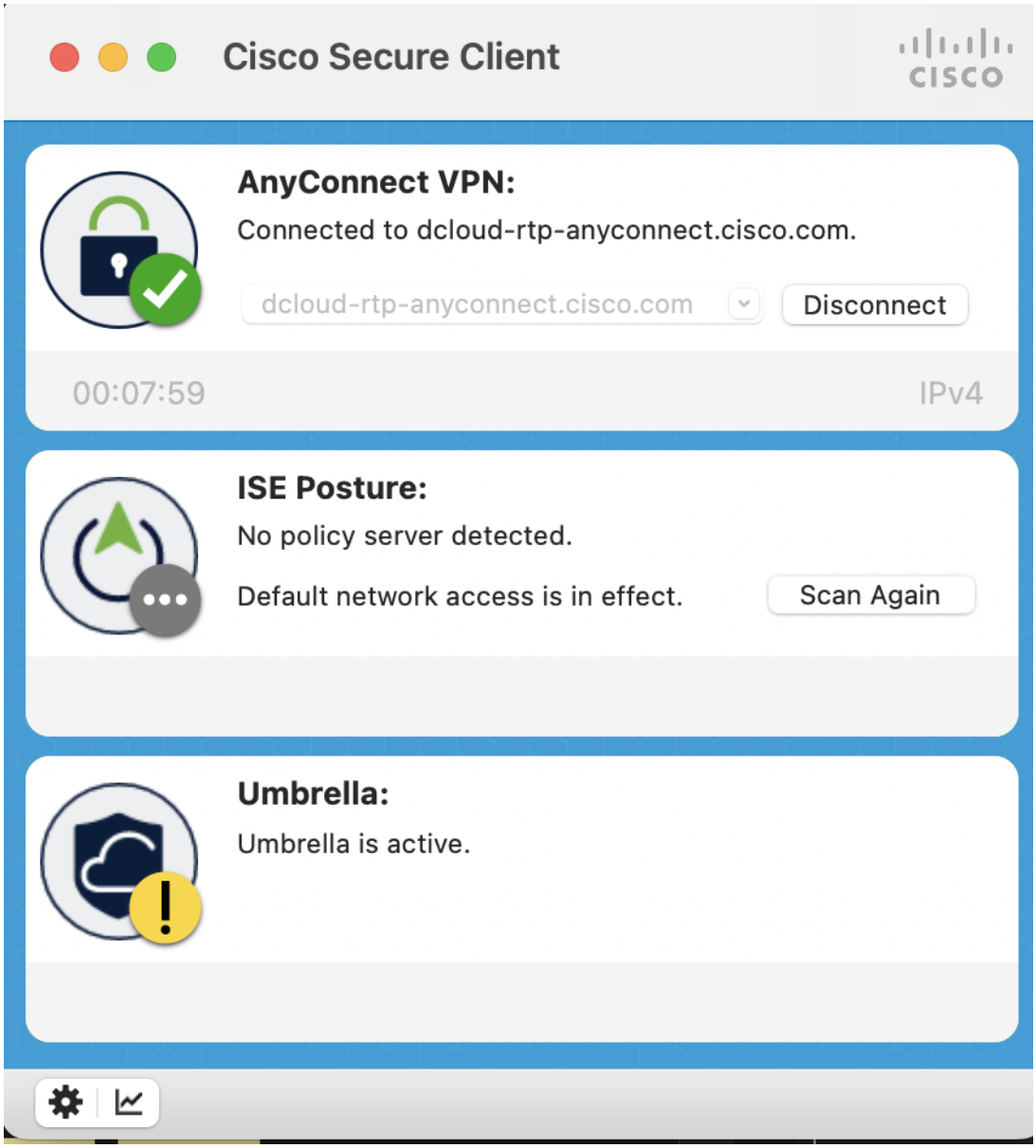
• Introduction

The integration of Cisco Application Centric Infrastructure (ACI) and Cisco Software-Defined Access (SDA) represents a significant step toward achieving an end-to-end intent-based networking architecture. By leveraging Cisco Identity Services Engine (ISE) and Security Group Tags (SGTs), this integration delivers consistent security policies across both data center and campus networks. This lab will provide hands-on experience with the configuration, deployment, and management of this integration, enabling participants to understand how to unify networking and security policies across different domains.

• How to navigate this lab?

Your primary source of accessing our infrastructure will be via your dCloud set up. You've already been logged into the VPN session, however in case you're logged out or have

connectivity issues, reach out to us and we'll do our best to help. Please check if you are connected to the VPN, it should look at follows:



On your desk you can find your user pod number, which you'll use to navigate the infrastructure.

WARNING:
This is a shared environment, and you have been given full admin access. Please be careful and ensure you are only working on your devices/policies. Be doubly careful to check if you are changing in the correct scope/device before you push any changes.

Learning Objectives

In this lab session, you will explore the following key concepts and skills:

1. Understanding the Integration Between Cisco ACI and SDA:
    - Learn how ACI and SDA complement each other to deliver a unified fabric for campus and data center networking.
    - Grasp the benefits of extending segmentation and policies across domains.
2. Cisco Identity Services Engine (ISE) Role:

- Discover how ISE acts as the policy engine for defining and enforcing Security Group Tags (SGTs).
- Learn how to configure ISE for SGT propagation between ACI and SDA environments.

3. Security Group Tags (SGTs) and Their Importance:
    - Understand how SGTs enable identity-based policies and micro-segmentation.
    - Explore the use of SGTs for consistent policy enforcement across the network.
4. Common Policy Framework:
    - Gain insights into creating and managing policies that work seamlessly in both ACI and SDA.
    - Learn how to define policies using Cisco Catalyst Center and APIC controllers.

Lab Objectives

At the end of this lab, you should successfully have understood:

- Hands-on configuration of ACI-SDA integration.
- Synchronization of SGTs across domains using ISE.
- Application of common policy frameworks and verification of policy enforcement.

Lab Scenario

Welcome to our fictional company **ACME Corp**

We want you to walk with us through the stages of expansion of our company and leverage common policy for a consistent and comprehensive security implementation.

We're going to be talking about common policy from the lens of security and segmentation. This lab is a reference to get you started on the journey of envisioning what's possible.
Do talk to us and brainstorm together- we'd love to understand how you plan to use this integration and can help clarify what's coming.

Initially we will then parallelly move to the data center and set up contracts there.. After this implementation, we will start with creation of a campus infrastructure- leveraging Catalyst Center and SDA for dot1x.

Then with the common policy set up- we will look at how we can leverage common policy for a seamless integration and end to end security enforcement.

Access Details

Always keep this open on one screen/Tab!

ACI: https://198.19.219.49

Username: admin

Pass: C1sc0@1234

External Host Machines (simulated by nexus devices)

SW_303 and SW_304:

Username: clemea

Password: cisco!123

Catalyst Center (DNAC):

https://198.18.129.100/

Username: dcloud

Password: C1sco12345

ISE

Session 1 (Users 1-15) 198.18.133.33

Username: admin

Password: C1sco12345

CML Ubuntu Endpoints:

Username: cisco

Password: cisco

CML:

198.18.134.1

Username: admin

Password: C1sco12345

FIABs:

Username: ciscolive

Password: C1sco12345

Dcloud:

Host

dcloud-rtp-anyconnect.cisco.com

Session 1:

User: v292user1

Password: 767279

Scenarios Shown:

1. **Native ACI Segmentation**
2. **Native SDA ISE integration**
3. **ISE ACI integration Workflow**
4. **Outbound and Inbound policy from ISE to ACI using SGTs**
5. **(Theoretical) Understanding Policy Enforcement options**
6. **(Appendix) Use Case Options with Common Policy**

Task 1: ACI Day 0 Configuration

Scenario

ACME Corp is expanding into the data center and has decided to deploy Cisco ACI to provide scalable segmentation and policy-based connectivity.

The environment consists of:

- 1 ACI fabric with 2 leaf switches and 1 spine
- External ‘Switches’ acting as end hosts
- Pre-created tenant objects that will be used throughout this task

---

Learning Objective

This task introduces core Cisco ACI constructs, including:

- Tenant
- VRF
- Bridge Domain (BD)
- Application Profile
- Endpoint Group (EPG)
- Contracts
- Layer 3 Outs (L3Out)

Learning Objective

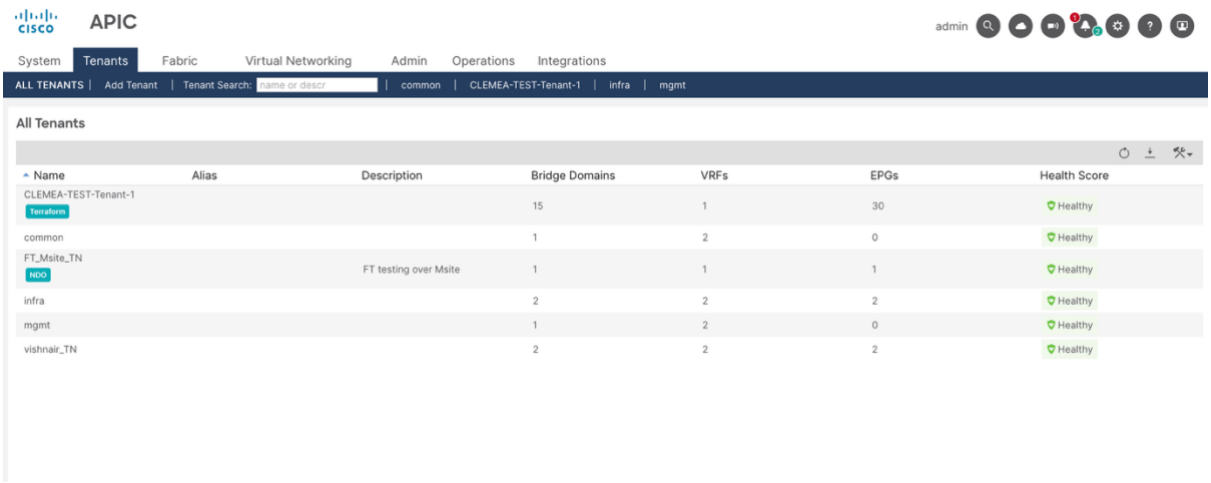
This task introduces core Cisco ACI constructs, including:

- Tenant
- VRF
- Bridge Domain (BD)
- Application Profile
- Endpoint Group (EPG)
- Contracts
- Layer 3 Outs (L3Out)

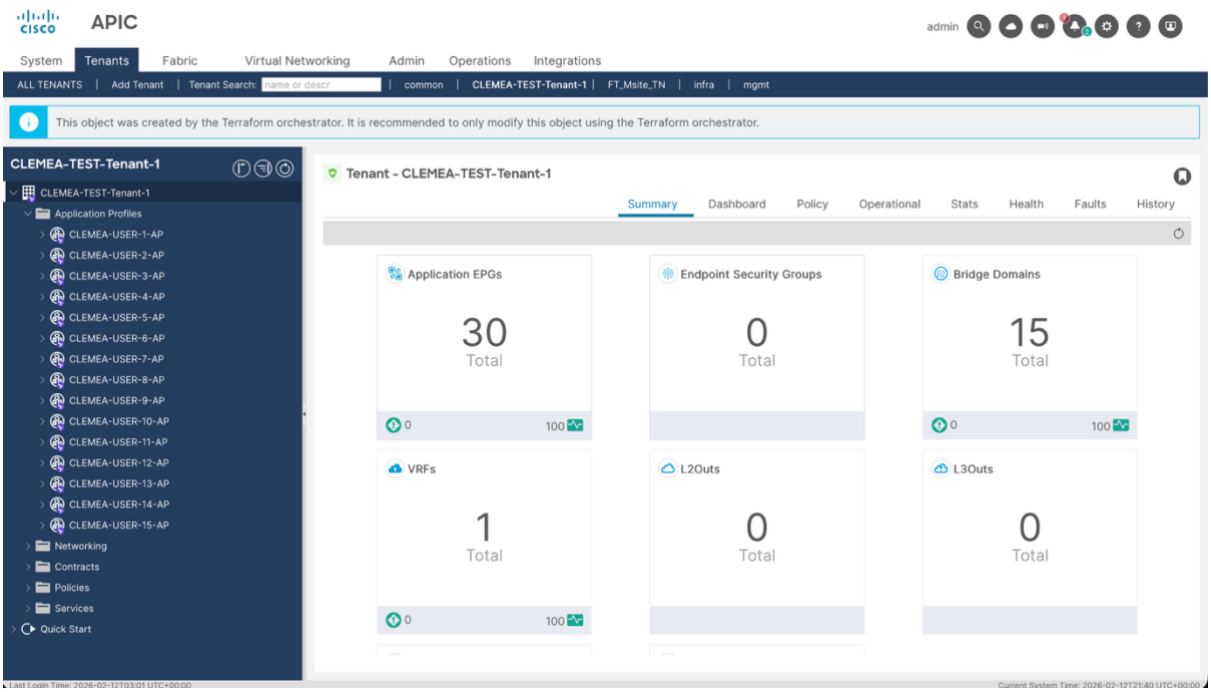
Step 1: Verify Your Assigned Tenant and EPGs

1. Log in to the APIC GUI.

POD 1-15 will have config in CLEMEA-TEST-Tenant-1.

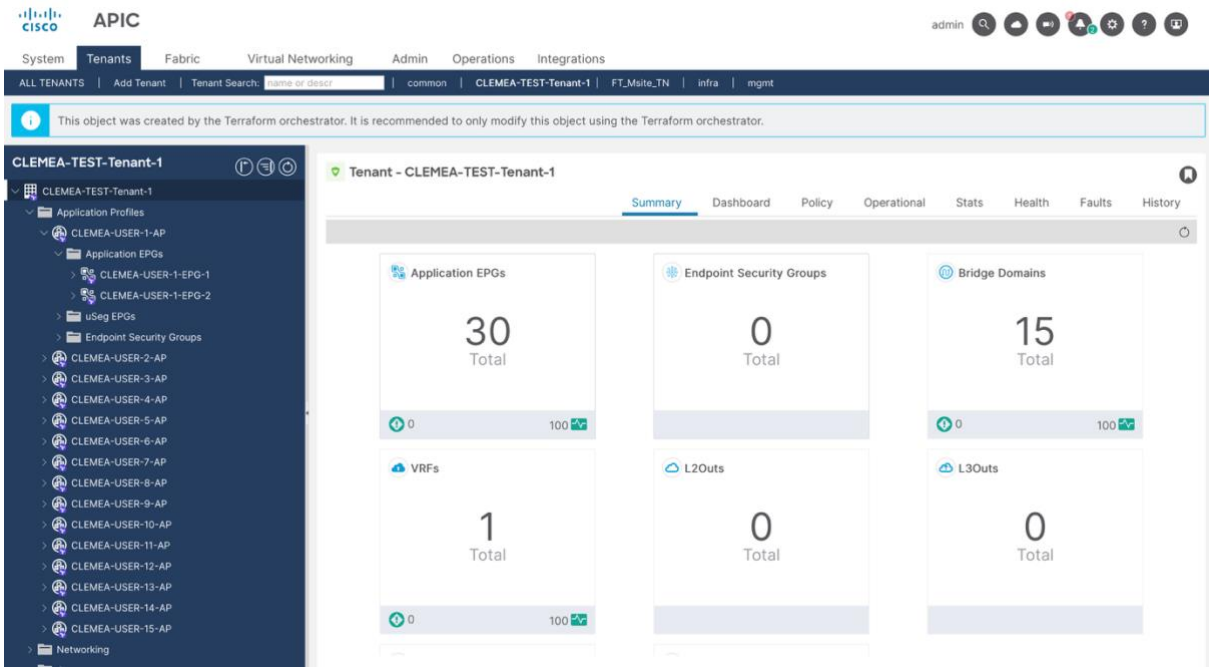


A Tenant in Cisco ACI is a logical container that isolates network resources, like a virtual context or VRF in traditional networking. It allows segmentation of policies, configurations, and users, ensuring multi-tenancy. Think of it to logically group and isolate resources for different teams, applications, or customers within the same fabric.



An Application Profile in Cisco ACI is a logical representation of an application's network and policy requirements. It serves as a container for grouping Endpoint Groups (EPGs) that together define the components of an application (e.g., web, app, database tiers). The application profile organizes the relationships between these EPGs using Contracts to specify how they communicate. Conceptually, it maps to an application-centric view of networking, akin to defining the architecture of a multi-tier application in a traditional network.

Open your application profile and open the available Application EPGs.



You will notice 2 EPGs assigned to your pod

An Endpoint Group(EPG) is a collection of devices (endpoints) with similar policy requirements, grouped based on function or purpose (e.g., web servers, database servers). In classical networking, this is somewhat akin to VLANs, but EPGs operate at a more abstract policy level. They are decoupled from VLANs and subnets, enabling more flexible application-centric network segmentation.

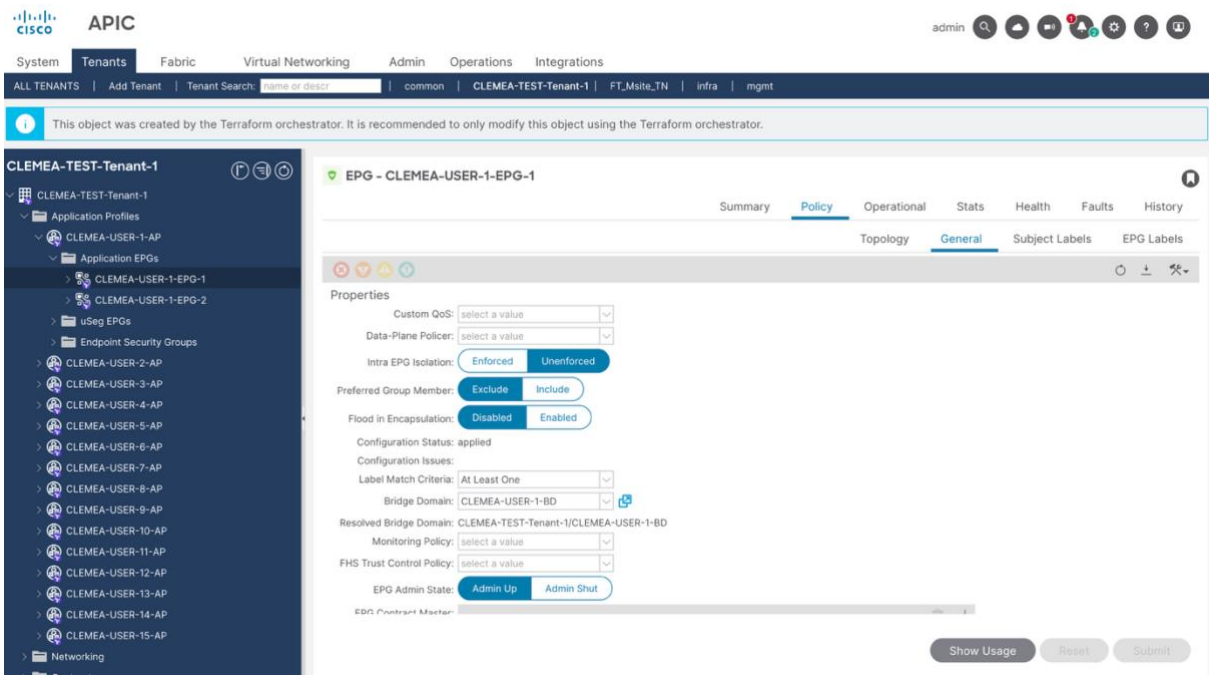
You can have full-free communication inside an EPG. We will Test this shortly.

VRF (Virtual Routing and Forwarding)

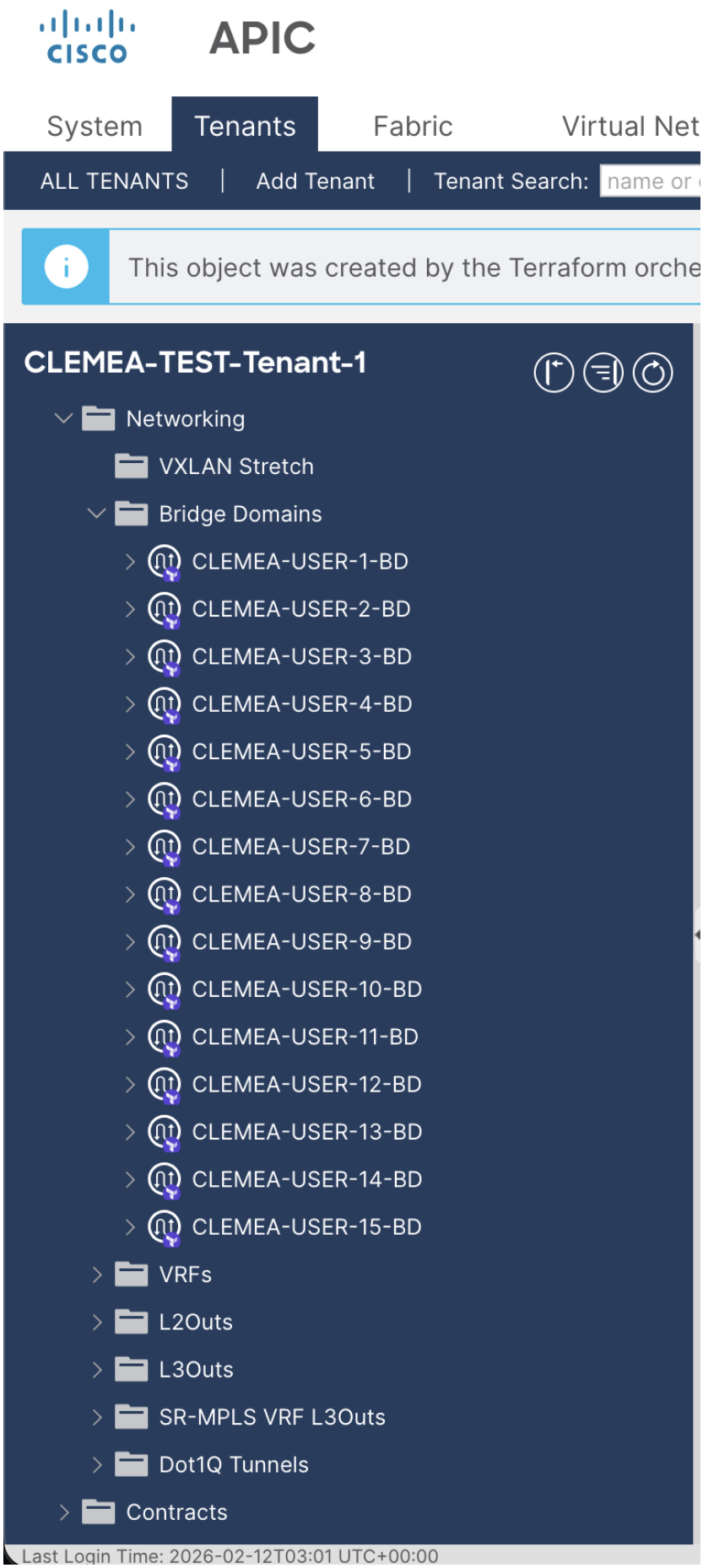
A VRF in ACI is a logical routing instance that provides isolated Layer 3 routing tables within a Tenant. It's like VRFs in traditional networks, allowing for segmentation of routing domains. Each VRF can contain multiple subnets or IP spaces, ensuring no overlap or interference between different applications or tenants.

We have one VRF per tenant configured.

Let us understand which BD these EPGs are a part of.
Click on the EPG. Head over to the Policy Tab and General Tab.
You will see a Bridge Domain for each user mentioned.

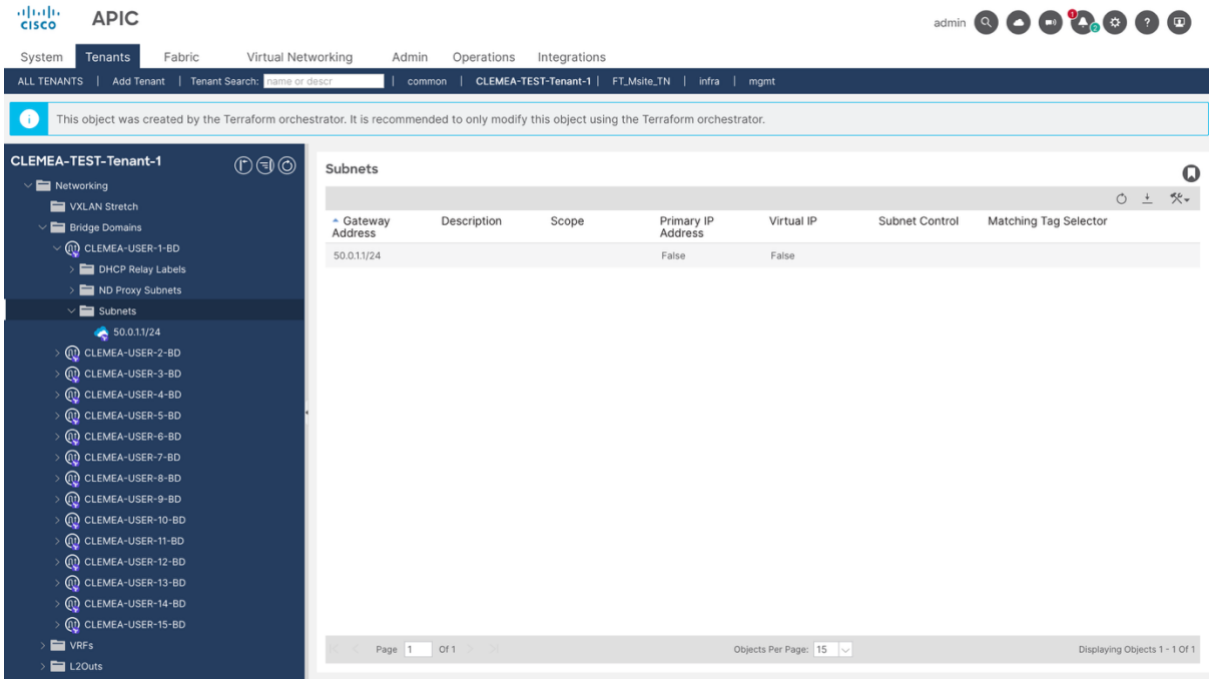


You can pop out with the Blue Pop out arrow to open the BD Settings.
Another way to navigate and see all the BDs is from the Networking tab on the left and go to bridge domains.



Let us open the BD assigned to our user. Understand which Subnet has been assigned to you as well as the gateway IP.

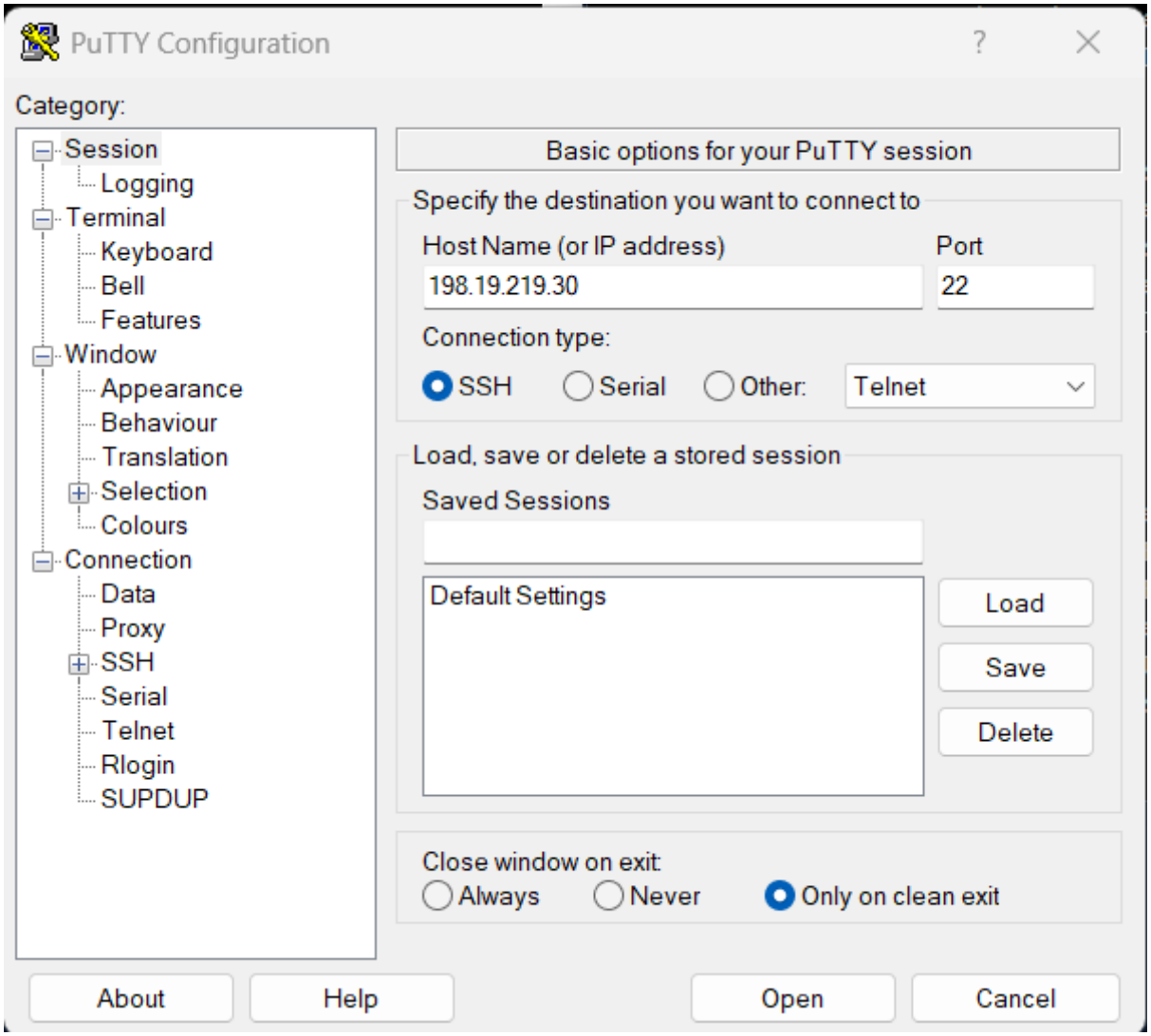
In this example for User-1



BD has a subnet of 50.0.1.0/24 and the gateway IP is 50.0.1.1
Have a look for your user and make a note of your subnet and gateway IP. This is important for the subsequent tasks.

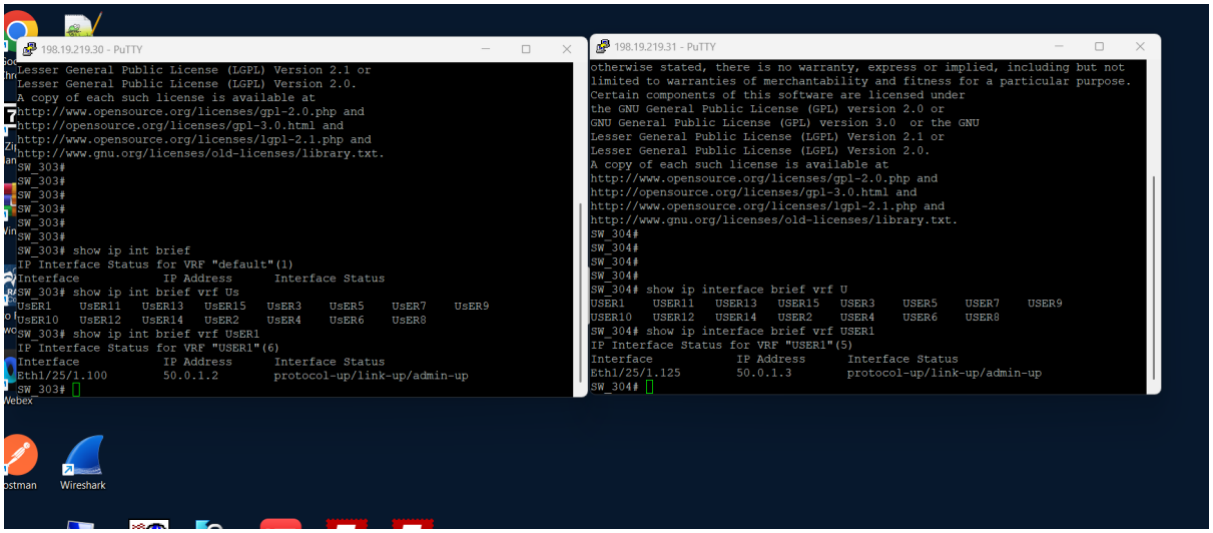
Step 3:
Now we will head over to the external switches SW_303 and SW_304 to verify reachability to ACI

Head over to Putty and login via the credentials (clemea/cisco!123)



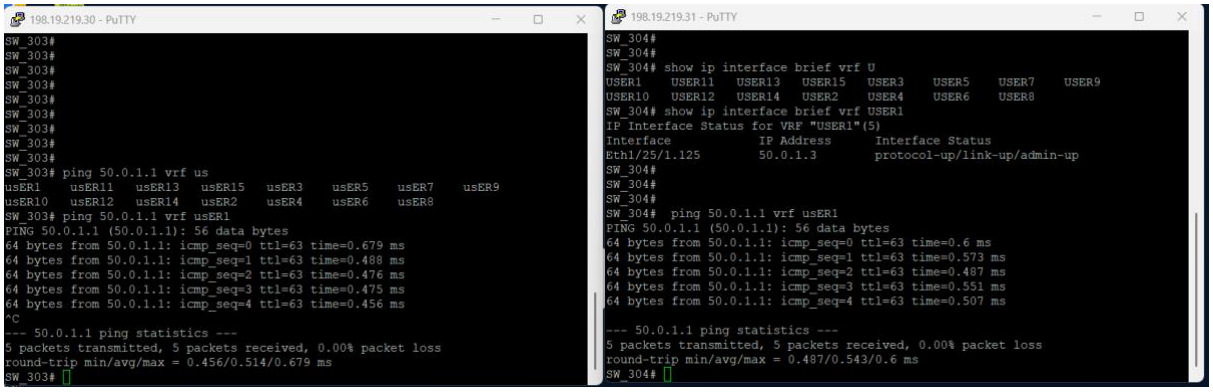
Start by verifying your assigned VRF settings using ‘show ip int brief vrf USER1’

Here we have shown user 1’s vrf, seach for your relevant user

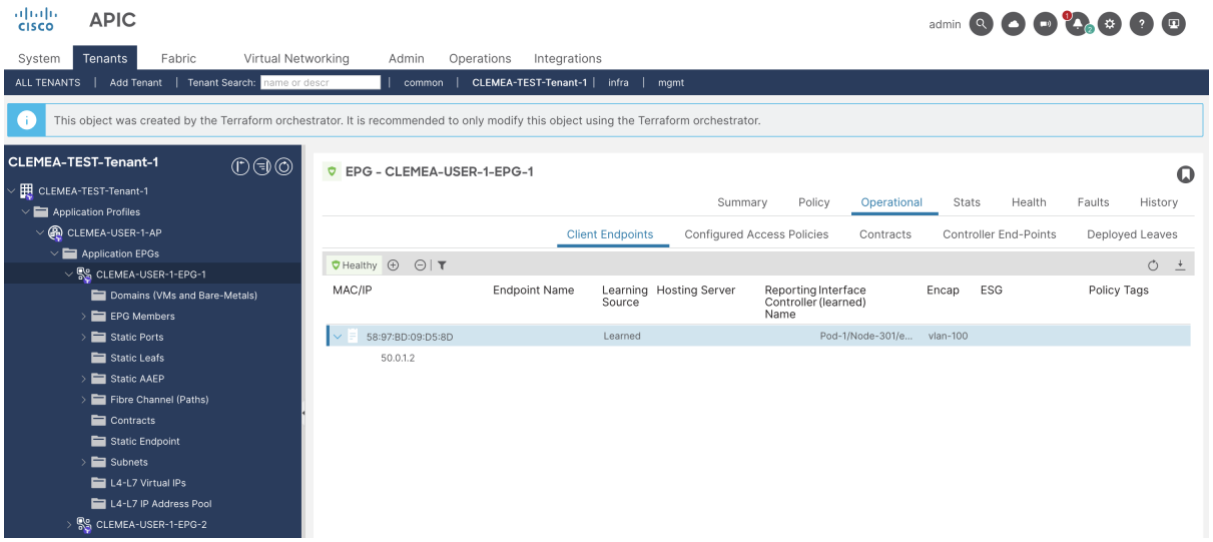


Start by verifying ping to your gateway that you configured on ACI.

For user X: ping 50.0.X.1 from both switches
Below is result for user 1



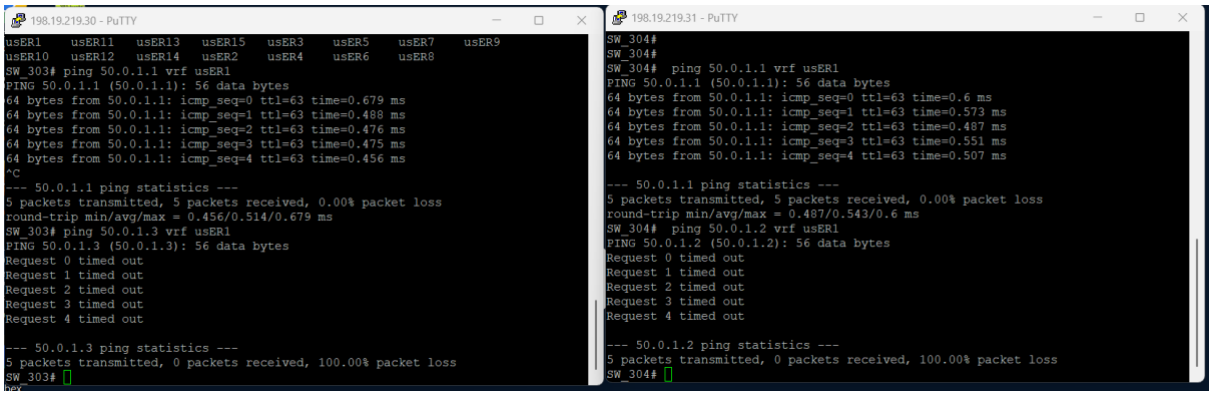
Head over to the APIC and the EPG Operational tab and verify the learning of the IP



Repeat for EPG-2 as well

Step 4:

Now we will attempt to ping the other switch. Perform the ping and see the result



We can see the ping fails. This is because all communication between EPGs is blocked by default.

We therefore need to allow the communication to ping between 303 and 304.

Step 5:

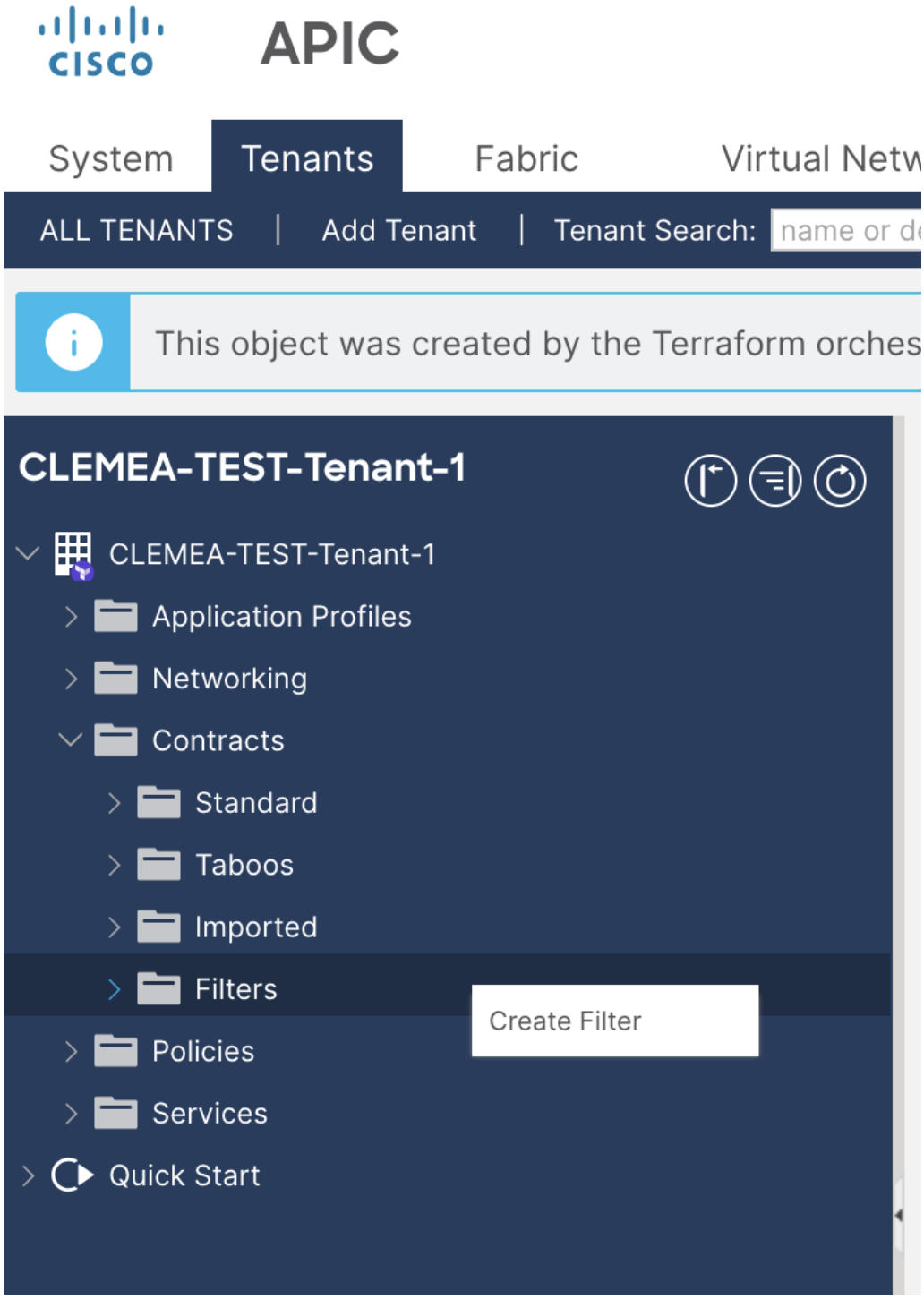
We need to use contracts to allow communication.

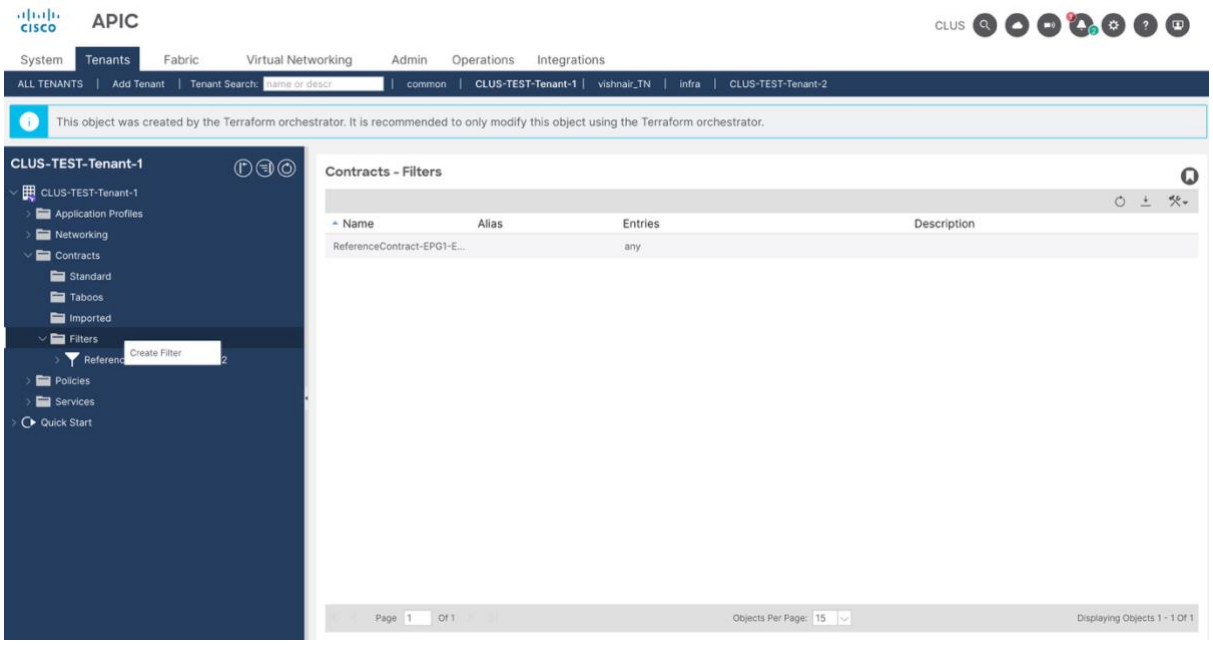
A Contract in ACI defines the communication rules between EPGs, acting as an access control policy. It specifies what type of traffic (e.g., protocols, ports) is allowed between the consumer EPG and the provider EPG. This is analogous to access control lists (ACLs) in traditional networking but is applied at the application level with a centralized policy model.

You must now create your own contract to allow communication between these endpoints.

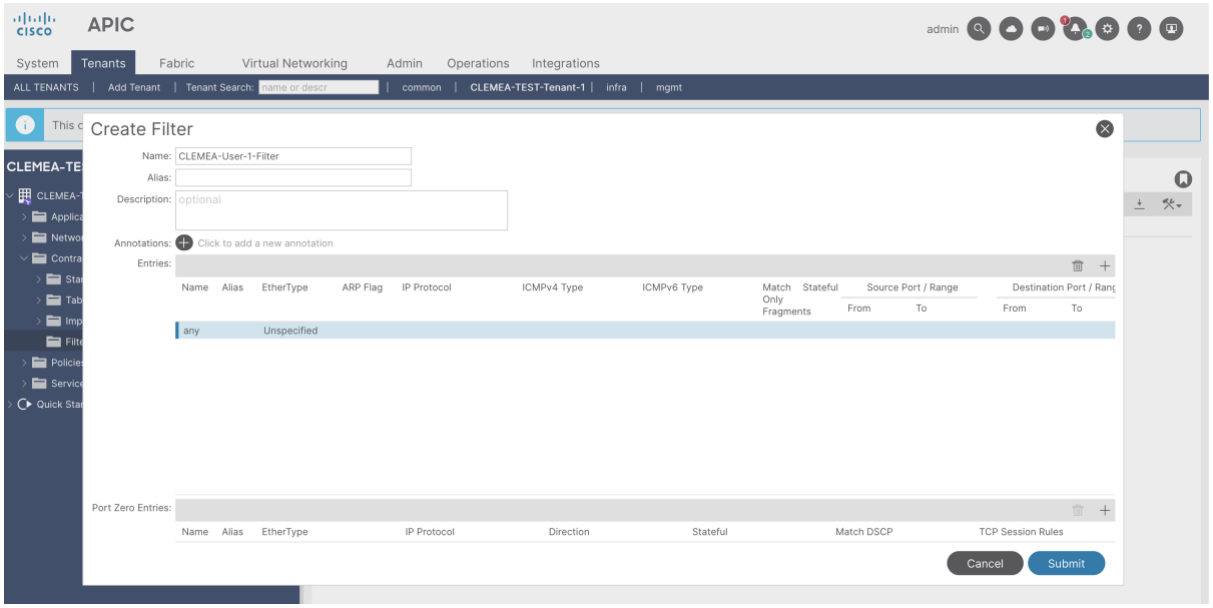
To create a contract, we first need a filter. This defines ‘What’ kind of traffic we are going to be allowing or blocking.
Head over to Contracts>Filters in our tenant

Right click filters to create a new filter

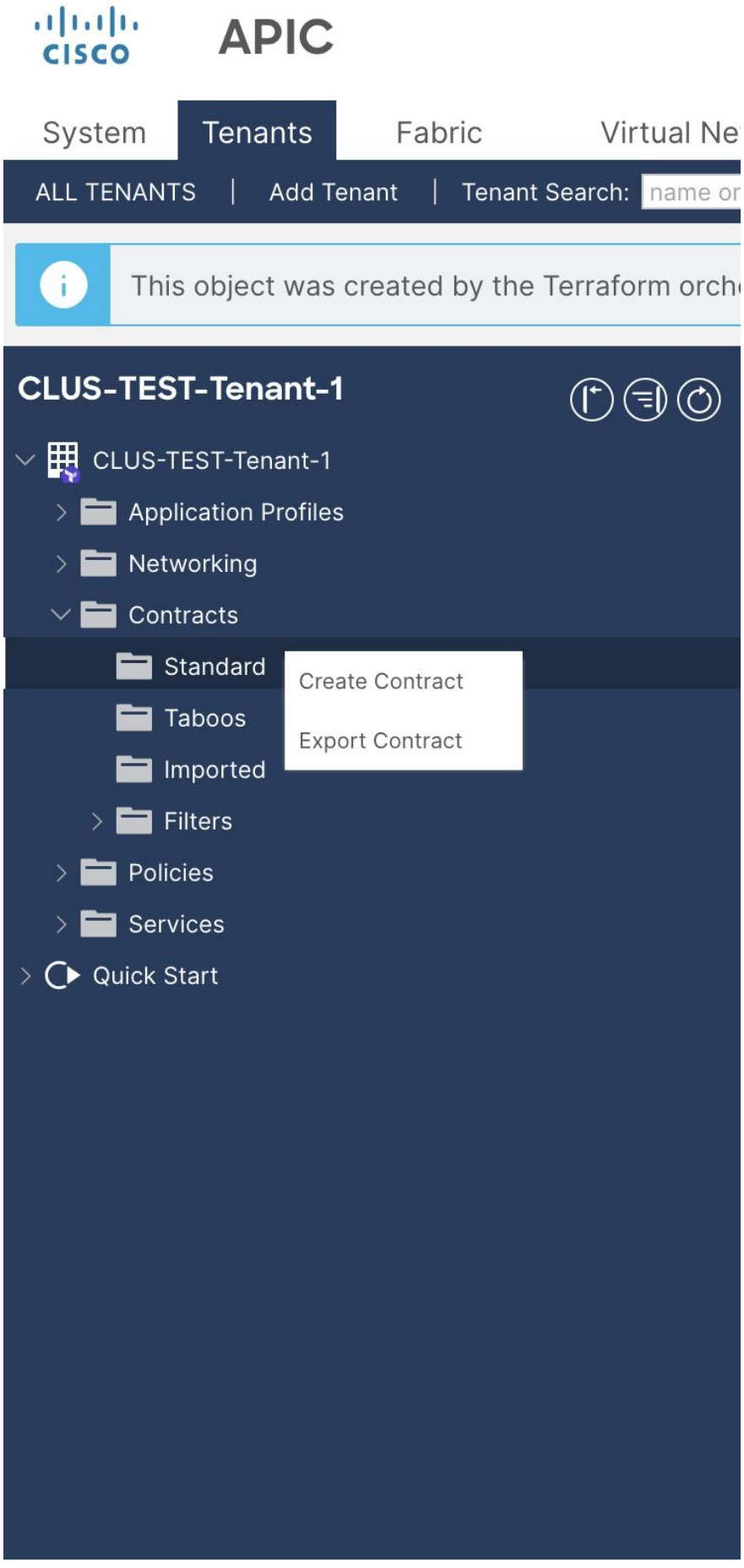


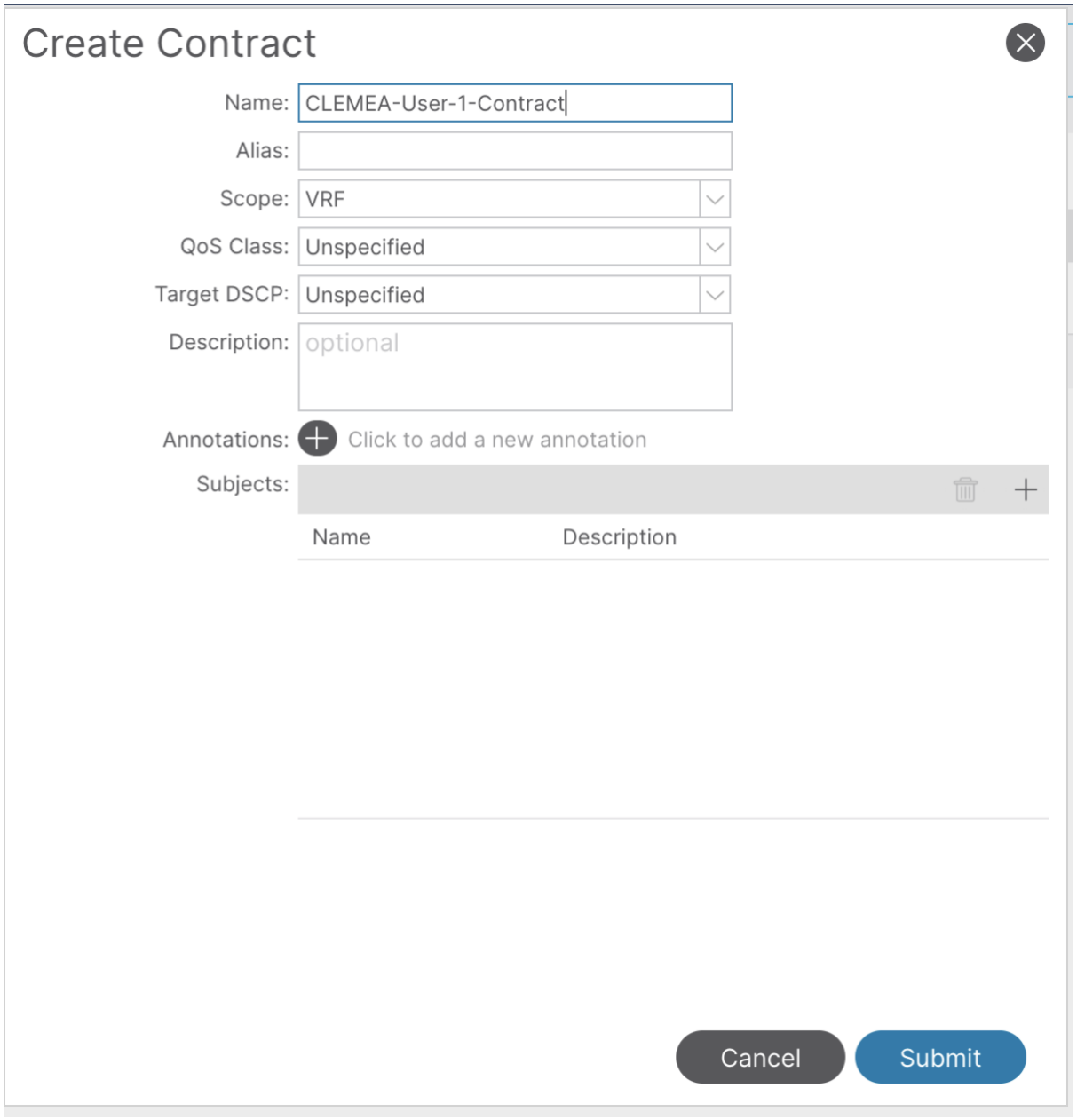


Name the filter CLEMEA-User-X-Filter where X is your Pod number
We will add an entry that matches ‘any’ traffic between our IPs

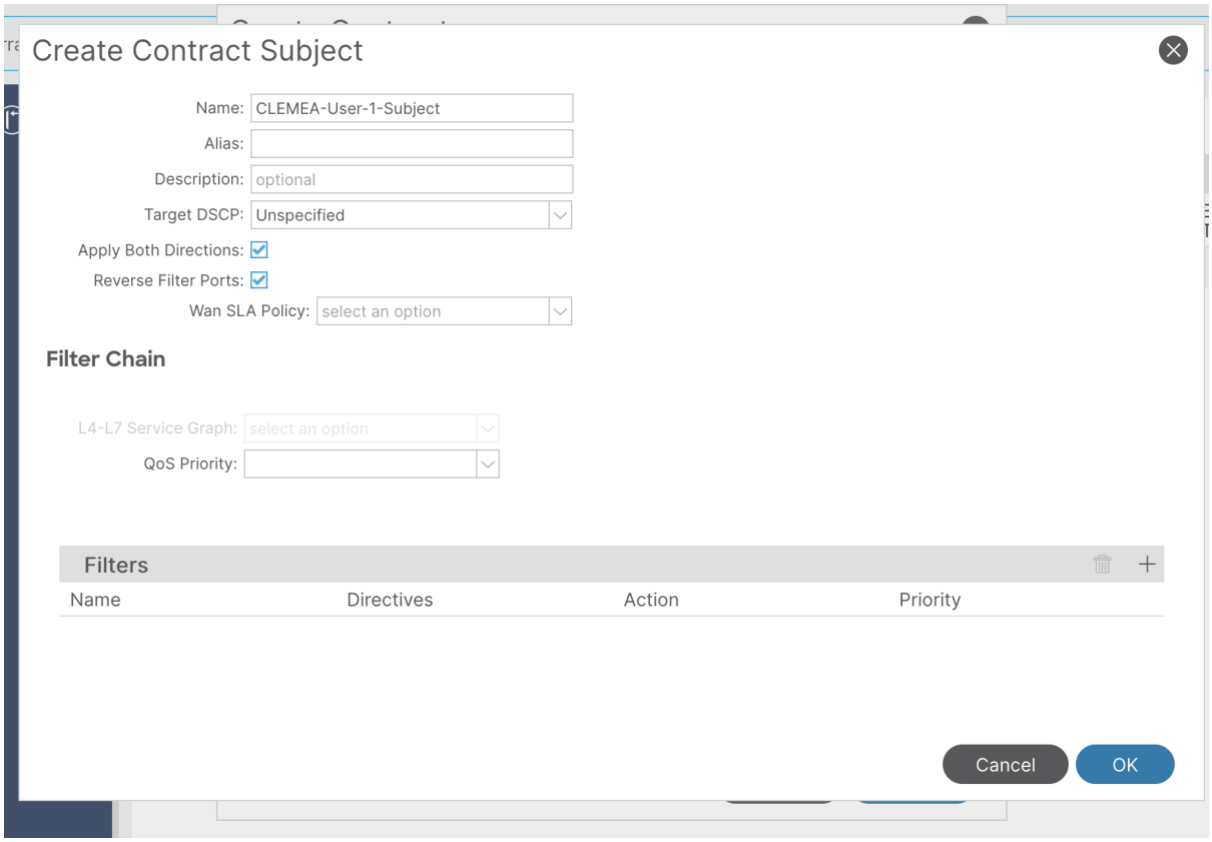


After we hit submit, we can then proceed to creating our contract. You can name it CLEMEA-User-X where X is your pod.

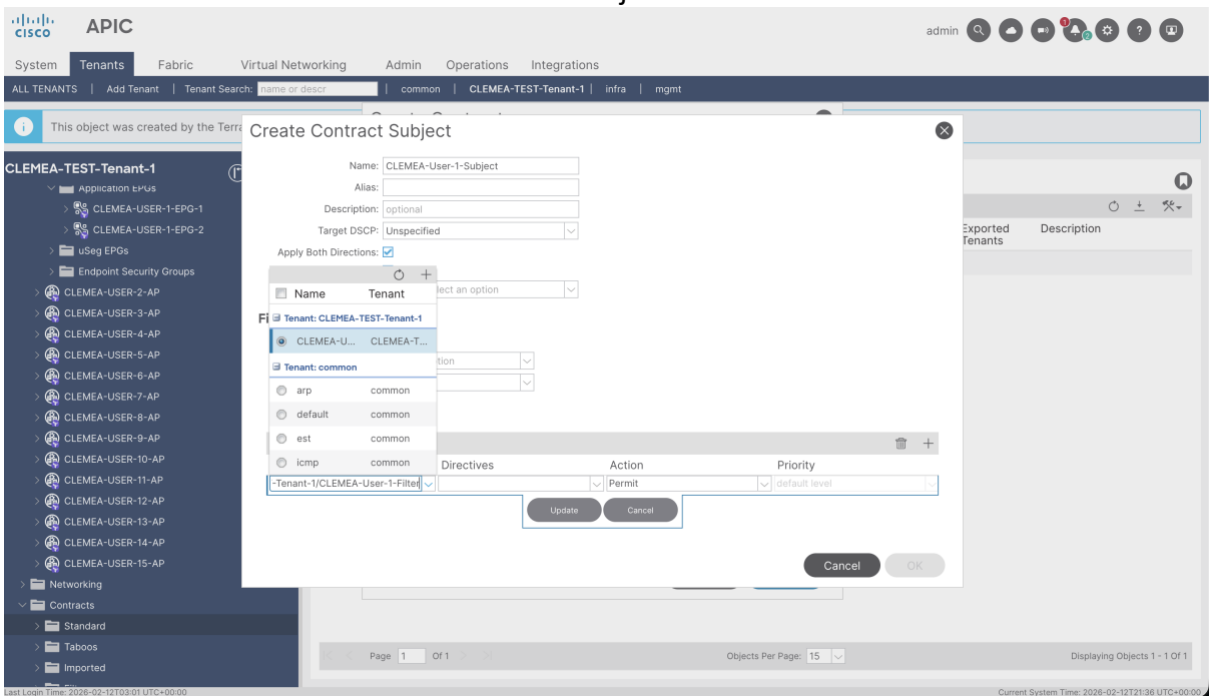




Click + to add a subject
Add subject name CLEMEA-User-X-Subject



Then hit the + near filters and add the filter we just created



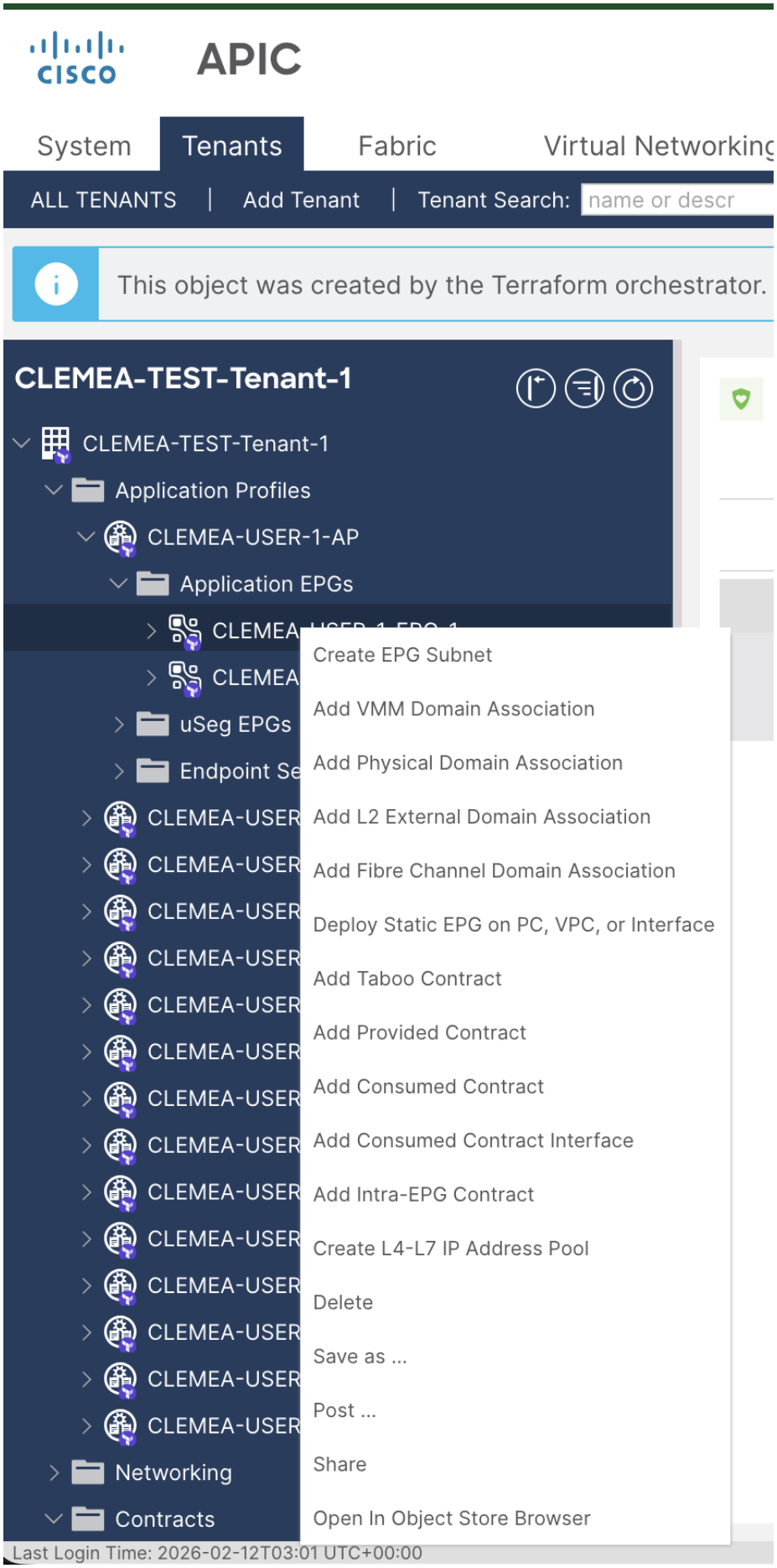
Hit Update, OK and Submit

Now our last step is to apply these to our EPGs

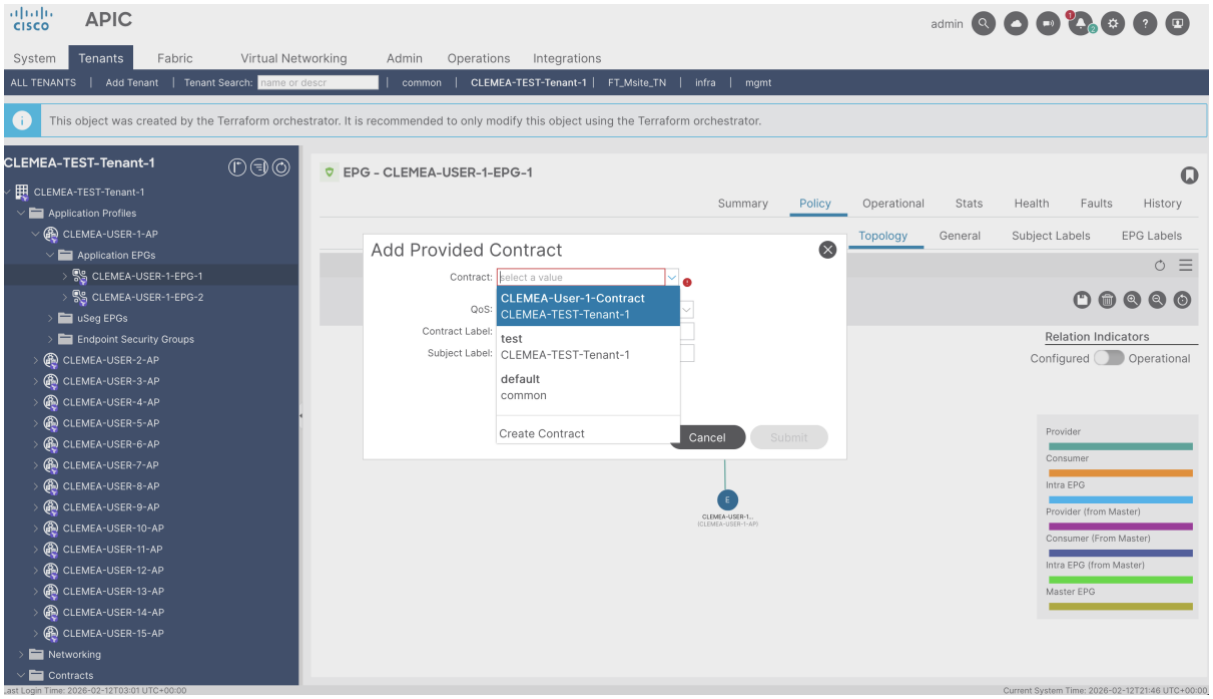
Head over to the EPGs.

With contracts we have the concept of provider and consumer. In this case it does not matter which side is the provider and which the consumer so we can just go ahead and make EPG-1 our provider and EPG-2 our consumer.

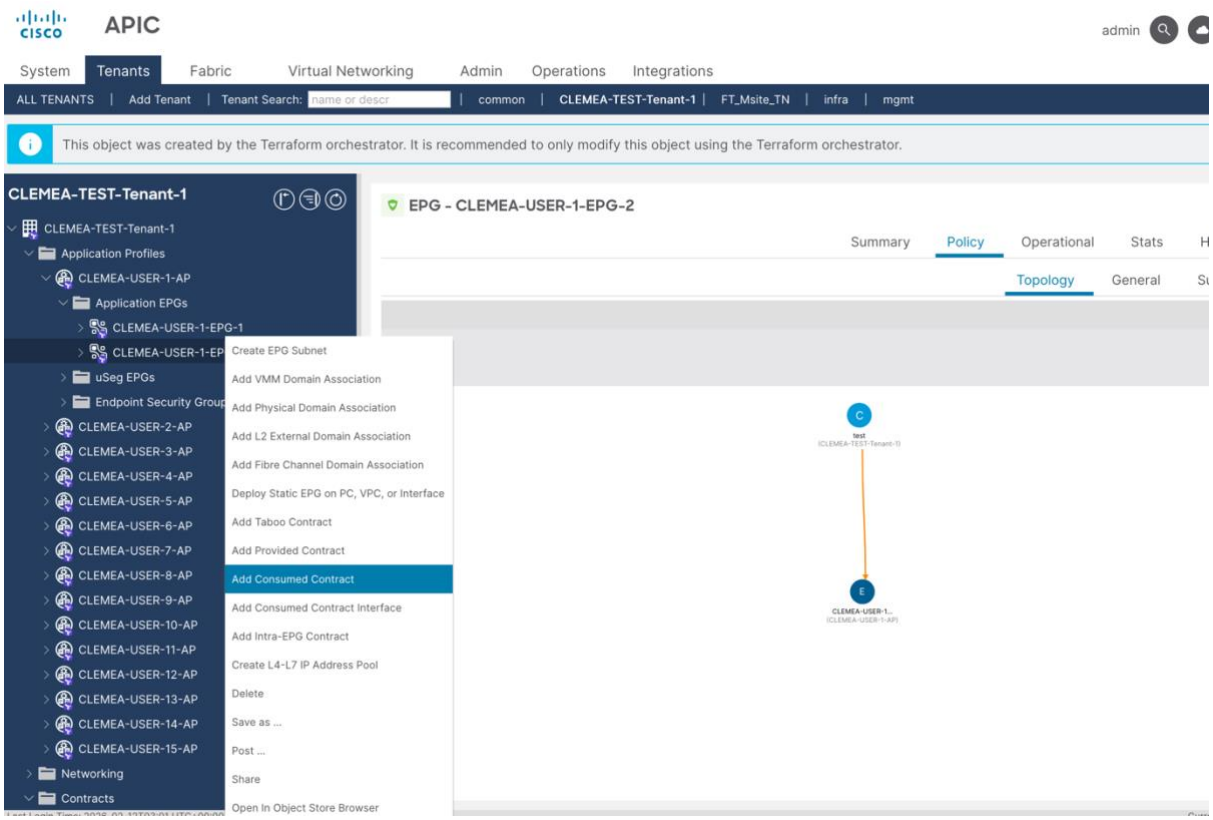
Let us go ahead and right click EPG one and add provider contract;



Select the contract you just created.

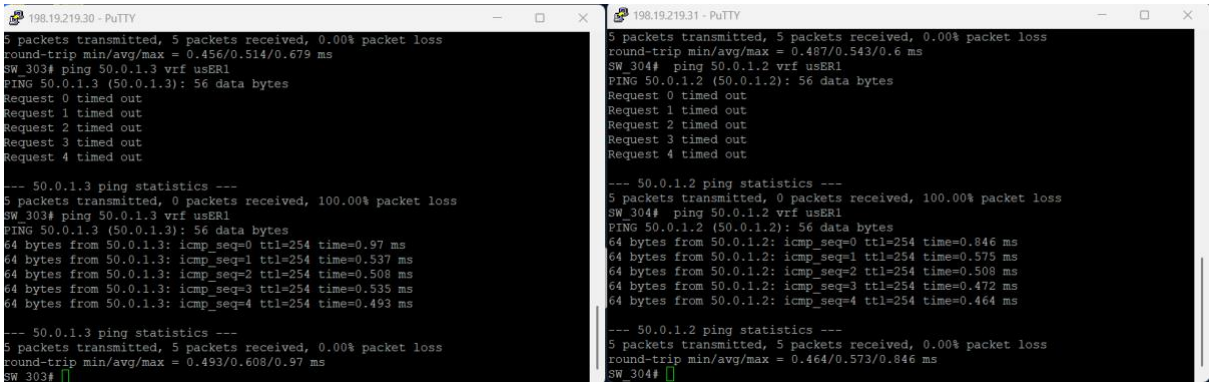


Repeat this to add consumed contract for EPG-2



Select your contract again

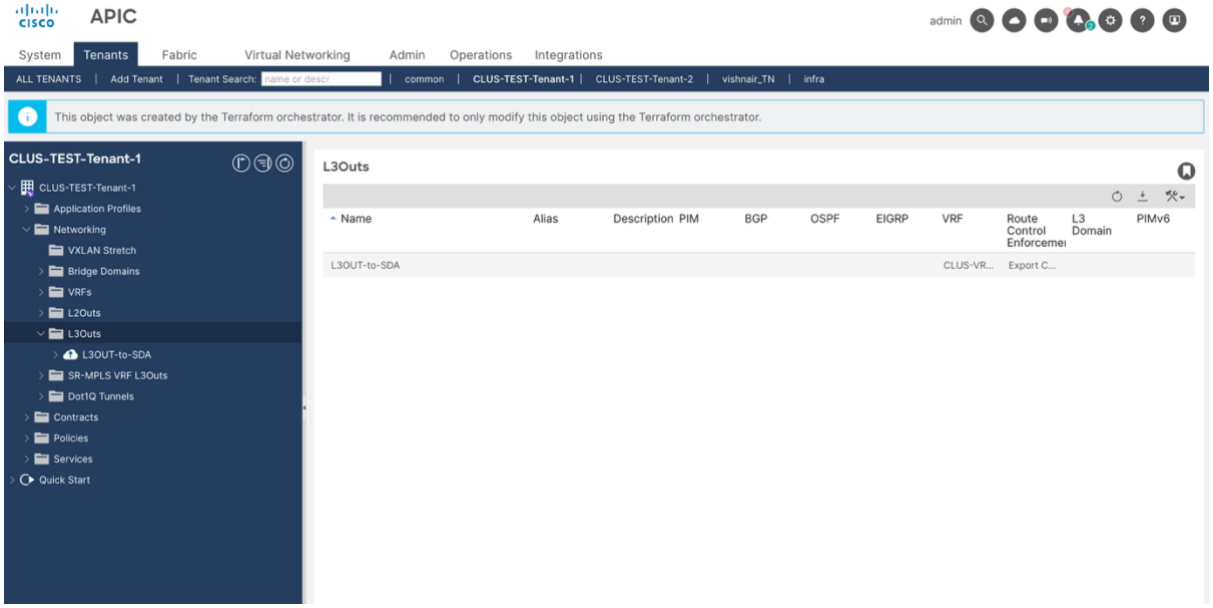
Now test the pings.
You should be able to ping the peer switch now



In case you have any queries here or have issues setting this up- do not worry- let us know and we will investigate it.
Feel free to proceed with the next tasks as there are no dependencies with the previous task!

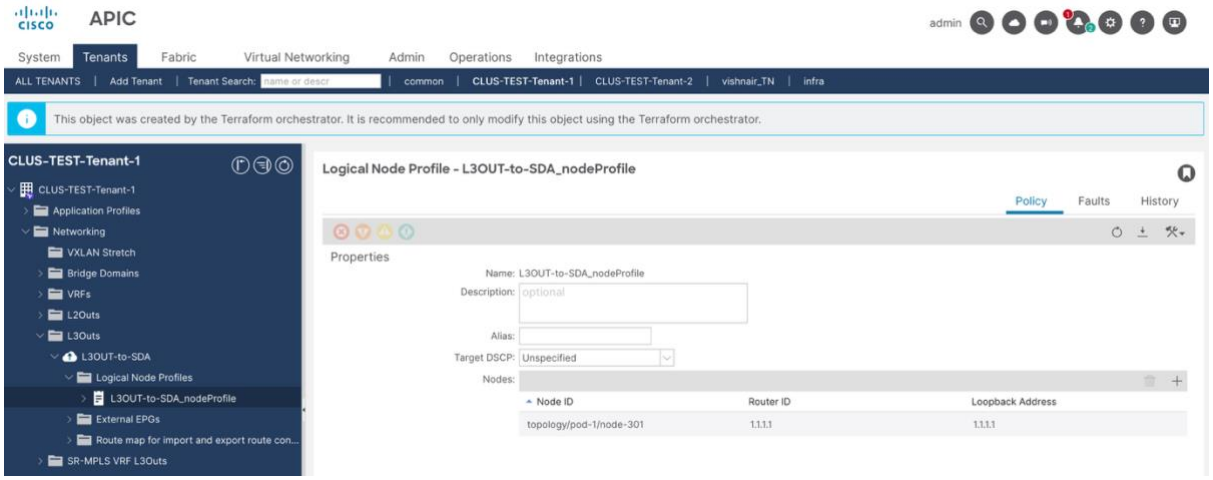
Step 5:

Open L3outs in the networking tab in your Tenant:



Open the L3OUT-to-SDA L3out
Open logical node profile

You can see that this mentions we have a router connected on node 301

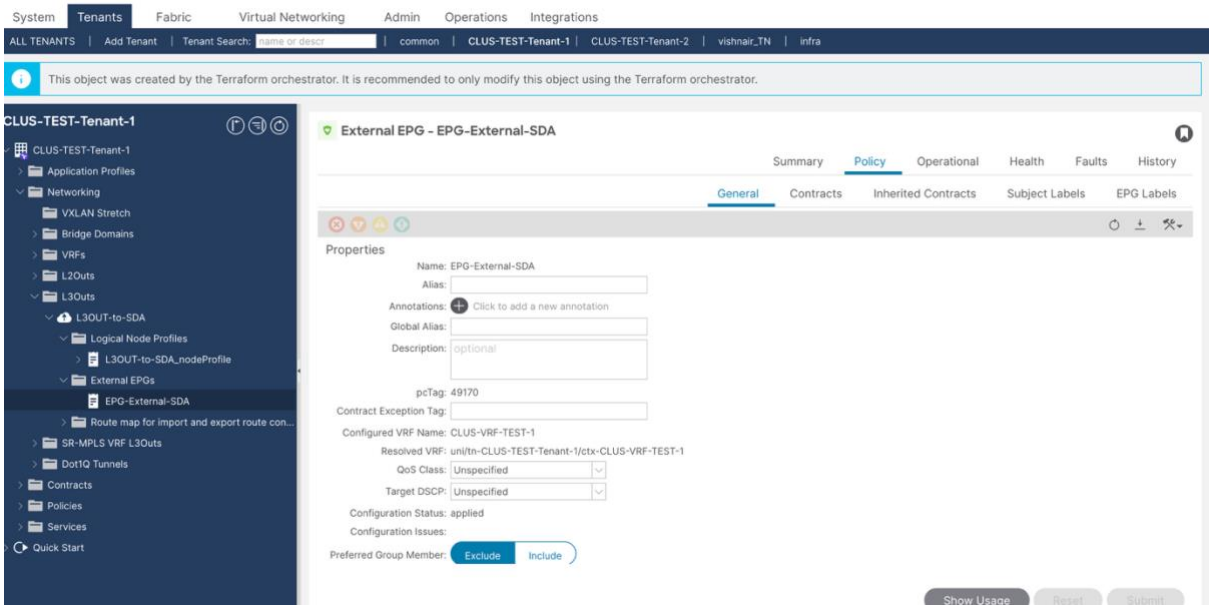


An L3Out is the configuration that enables Layer 3 connectivity between the ACI fabric and external networks (e.g., the internet or a traditional data center network). It is comparable

to configuring a gateway or routing instance in classical networking, providing external routing via protocols like BGP, OSPF, or static routes.

In this case- this is how we will reach our ‘external’ world which is where SDA sits. Later when we perform the integration and set up policies to share EPGs/SGTs- we will also be selecting this L3OUT.

Notice there is also an external EPG here:



This is where all the external traffic is grouped, and we can create relevant contracts as desired to enable communication with the outside world.

We will not be setting up end to end communication between the ACI and SDA in this lab, the objective of this lab is more to understand how segmentation is implemented in ACI and SDA and how you can simplify enforcement via this
integration. We will soon be using these Contracts and EPGs once we set up the Campus side in the next task.

That brings us to the end of the Second Task. We are done setting up our ACI side and understanding how segmentation works in the ACI world.

Now it is time to move to the SDA Side.

Task 2: SDA and ISE Bringup

Step 1:
Let us first examine our set up. Head over to the CML login after connecting to vpn of your session

CML: 198.18.134.1

Username: admin
Password: C1sco12345

Based on the number/pod assigned, you will be one of the 15 users creating your own SDA network.
We will simulate the SDA network by creating a fabric using a single Cat 9k switch as a FIAB (Fabric in a box), i.e. having the role of border, control plane and edge all assigned to one box.

Your endpoint is an Ubuntu device that is pre-configured to do Dot1x, it is connected on interface Gi1/0/2 of your switch.
This authentication traffic will then leave the fabric via the Fusion device and reach to your ISE and DNAC.
Login to your respective ISE & DNAC based on the session assigned to you.

For reference, here is the table explaining the interfaces connected within CML:

Table: CML Interface Connections

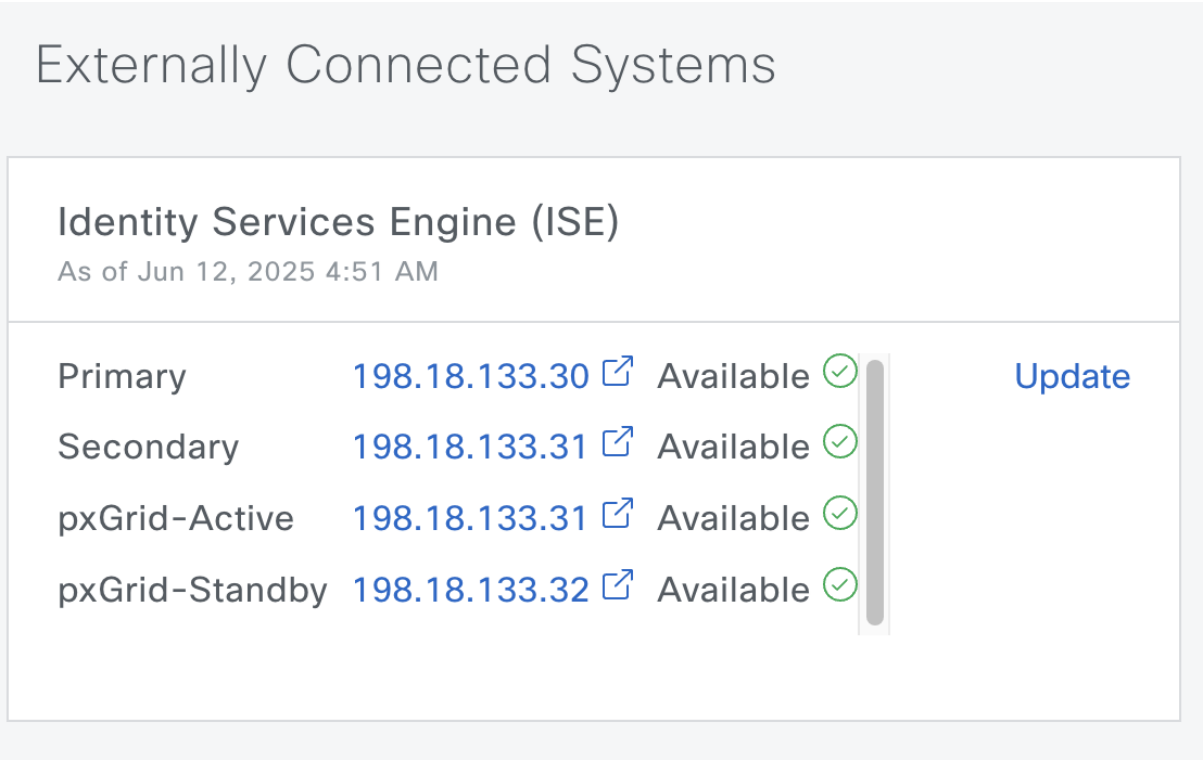
Fib-1	Gi1/0/1	Fusion_1 : Gi1/0/1
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/2
Fib-2	Gi1/0/1	Fusion_1 : Gi1/0/3
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/4
Fib-3	Gi1/0/1	Fusion_1 : Gi1/0/5
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/6
Fib-4	Gi1/0/1	Fusion_1 : Gi1/0/7
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/8
Fib-5	Gi1/0/1	Fusion_1 : Gi1/0/9
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/10
Fib-6	Gi1/0/1	Fusion_1 : Gi1/0/11
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/12
Fib-7	Gi1/0/1	Fusion_1 : Gi1/0/13
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/14
Fib-8	Gi1/0/1	Fusion_1 : Gi1/0/15
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/16
Fib-9	Gi1/0/1	Fusion_1 : Gi1/0/17
	Gi1/0/2	Ubuntu

	Gi1/0/3	Fusion_1 : Gi1/0/18
Fib-10	Gi1/0/1	Fusion_1 : Gi1/0/19
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_1 : Gi1/0/20

Fib-11	Gi1/0/1	Fusion_2 : Gi1/0/1
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_2 : Gi1/0/2
Fib-12	Gi1/0/1	Fusion_2 : Gi1/0/3
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_2 : Gi1/0/4
Fib-13	Gi1/0/1	Fusion_2 : Gi1/0/5
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_2 : Gi1/0/6
Fib-14	Gi1/0/1	Fusion_2 : Gi1/0/7
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_2 : Gi1/0/8
Fib-15	Gi1/0/1	Fusion_2 : Gi1/0/9
	Gi1/0/2	Ubuntu
	Gi1/0/3	Fusion_2 : Gi1/0/10

For users of session 2, starting with fib-16, also uses the same logic as above.
However, fib 1-15 are in session 1 of CML while fib 16-30 are part of session 2 CML.

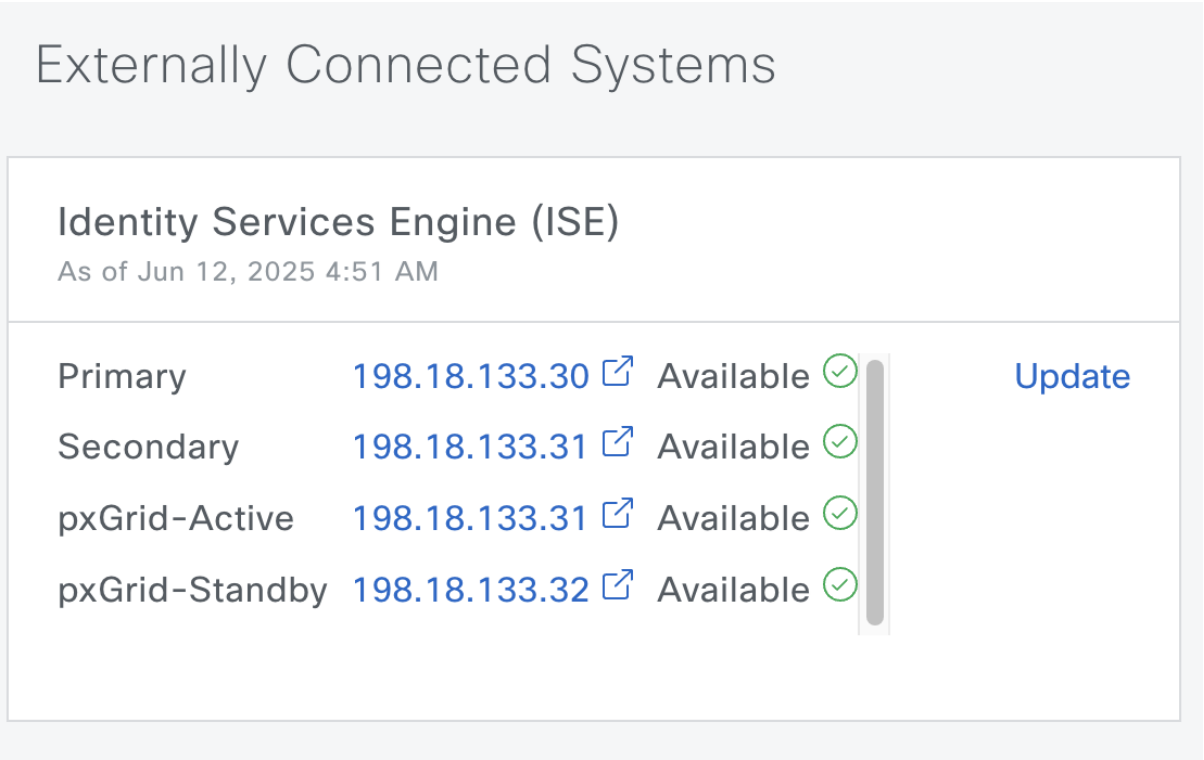
Step 2: (Theoretical)
We will start with building our SDA network from DNAC (or CatC) first, login to the UI and head to System > System 360 > Externally Connected Systems



Click on Update to see the details of ISE integration which was already done for the convenience of all users.

The server is added as “ISE” and integrated with pxGrid enabled. We are also using both RADIUS and TACACS capabilities of ISE.
TACACS will be used for the network device login while RADIUS is for endpoint, user authentication.

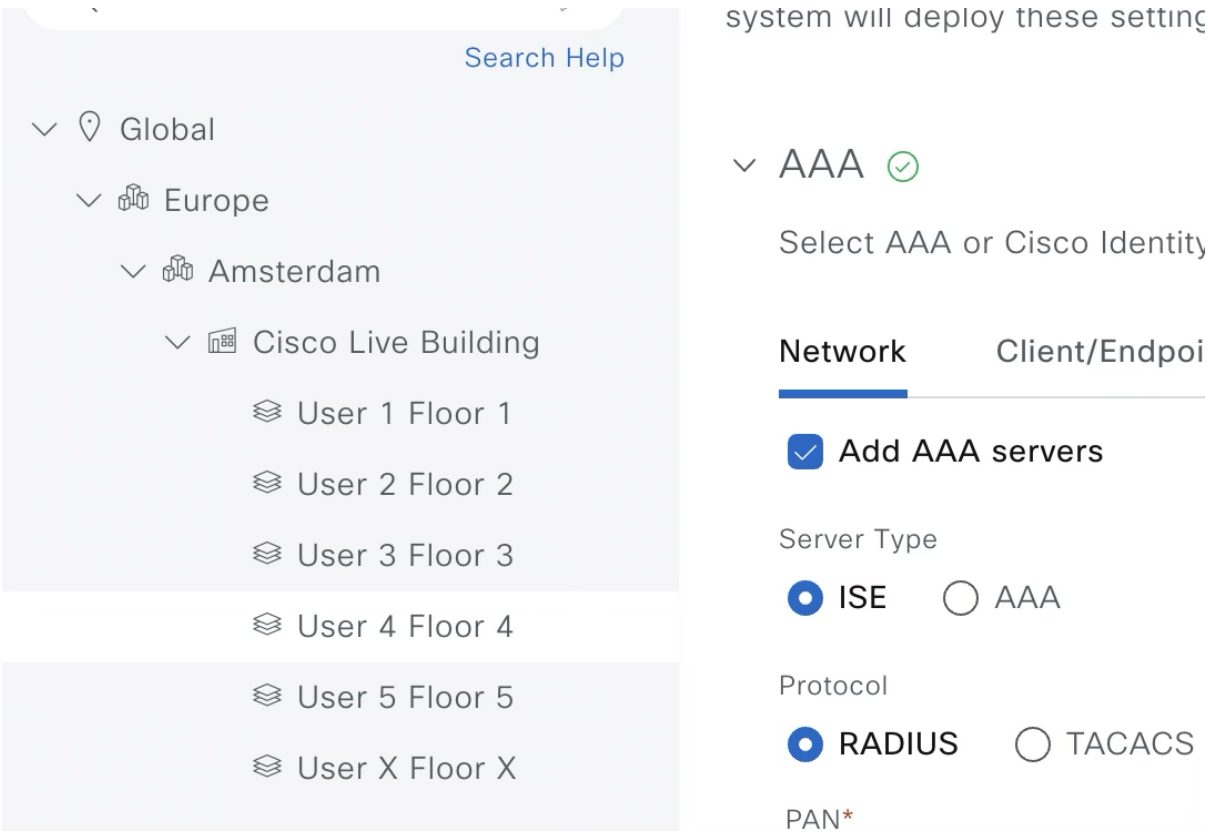
Step 2: (Theoretical)
We will start with building our SDA network from DNAC (or CatC) first, login to the UI and head to System > System 360 > Externally Connected Systems



Click on Update to see the details of ISE integration which were already done for the convenience of all users.

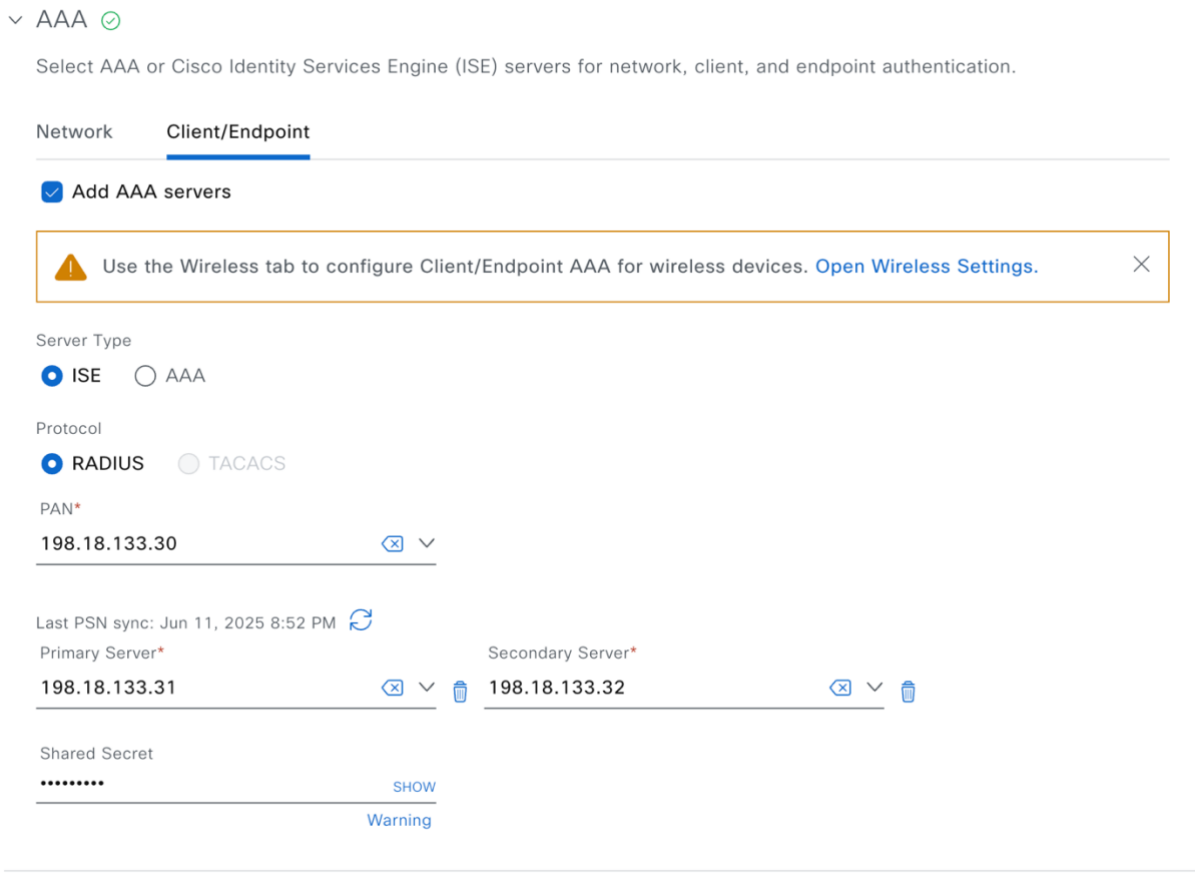
The server is added as “ISE” and integrated with pxGrid enabled. We are also using both RADIUS and TACACS capabilities of ISE.
TACACS will be used for the network device login while RADIUS is for endpoint, user authentication.

Step 3:
Go to Design > Network Hierarchy > You will see the area and the floor is created here for all users.
Based on your user number, verify a matching “User x Floor x” area under ‘Cisco Live’ section.



Click on Amsterdam and three dots to go to Settings:

The ISE configuration is already added here which should then be inherited on your floor automatically.
Verify settings under your floor to ensure this data is present there.

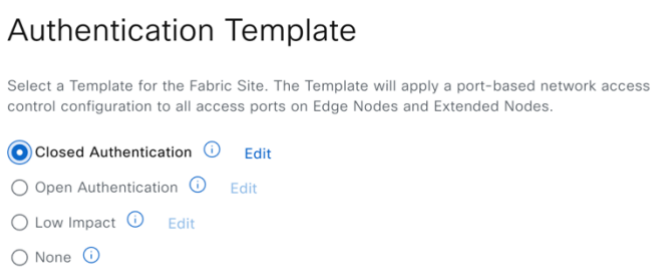


Similarly, using the tabs on the top menu, also verify data under Global Credentials and IP Address Pools.
'CLEMEA' and 'clemeav3' are the credentials to be used for network devices and snmp.

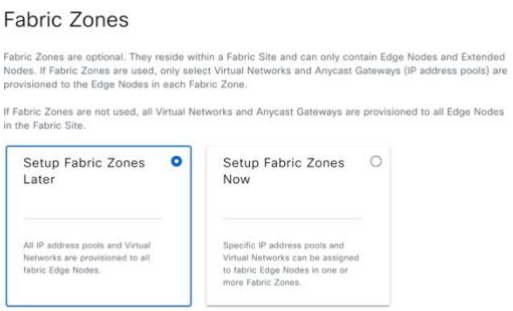
Step 4:
Create a Fabric Site
From main menu, go to Provision> SD Access > Fabric Sites > Create a Fabric Site

Fabric site will be created per user and this will be your own SDA network, separate from other users, with an individual IP pool, layer 3 VN and a single FIAB box connecting your user to the network.

Click through the mandatory options with 'Next' until you reach Authentication template, where Closed authentication will be selected.
This template would configure the port to do 802.1x authentication in closed mode.

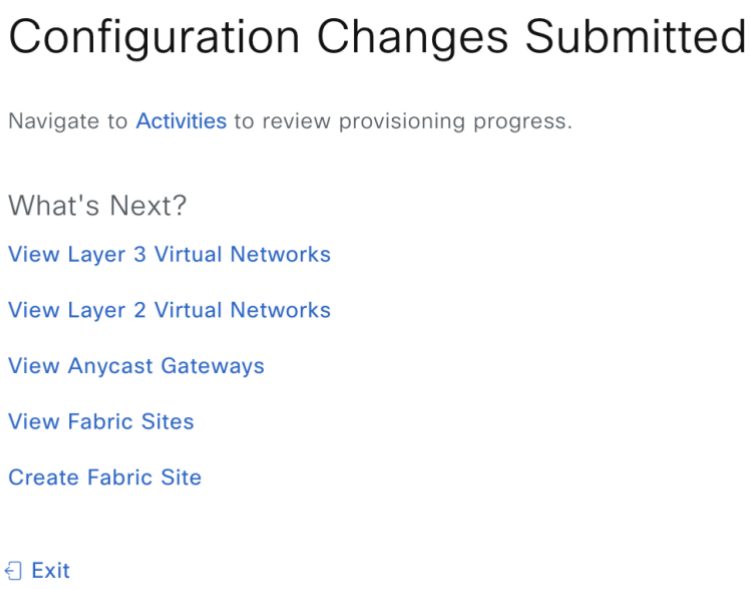


Select “Setup Fabric Zone Later” as this is not in the scope of this lab activity.

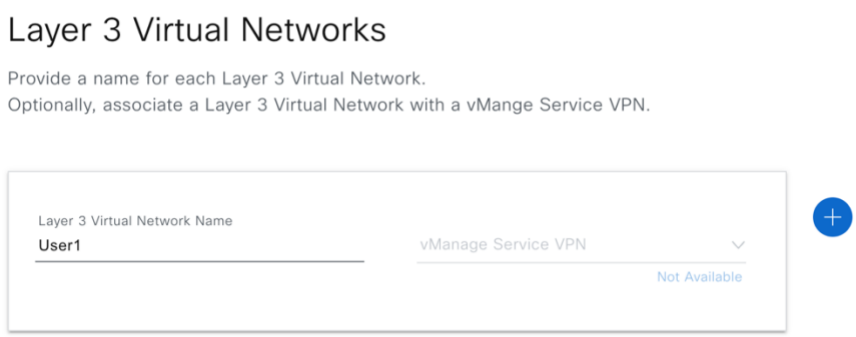


Click through the next steps till you ‘Save Intent’ and Deploy. This is saving intent as no network devices are present in your Fabric Site yet.

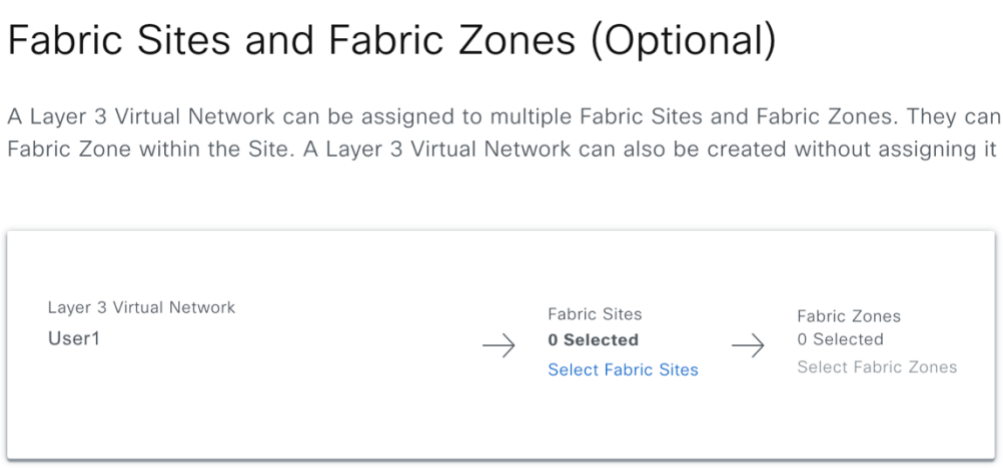
Continue navigating through to next part by selecting “Layer 3 Virtual Networks”

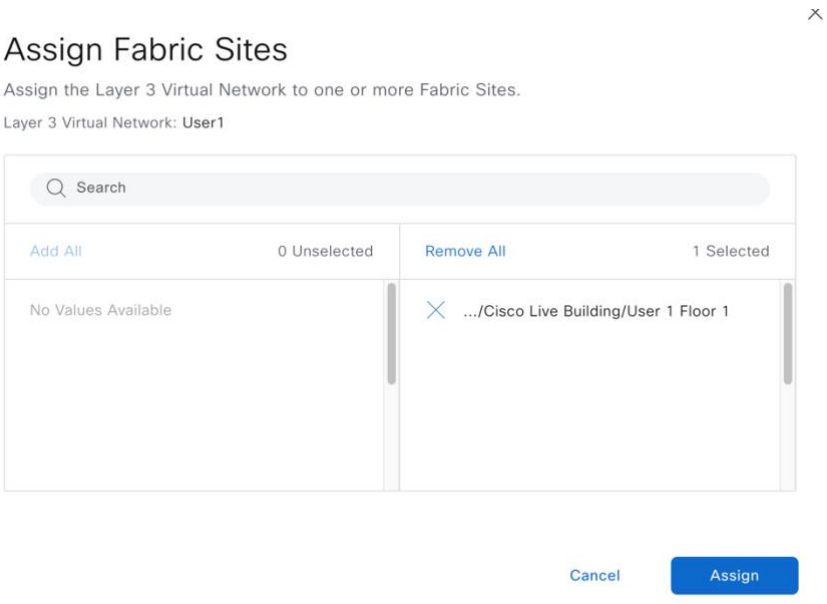


Create VN based on your user number.



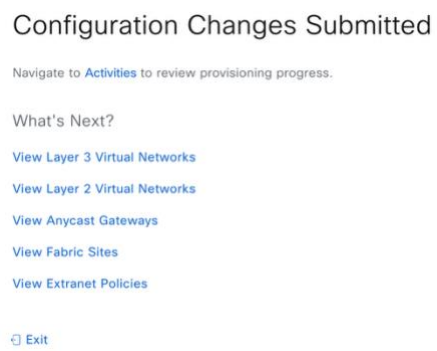
In the next steps, select the Fabric Site that you earlier created for yourself, this can easily be searched with the user number

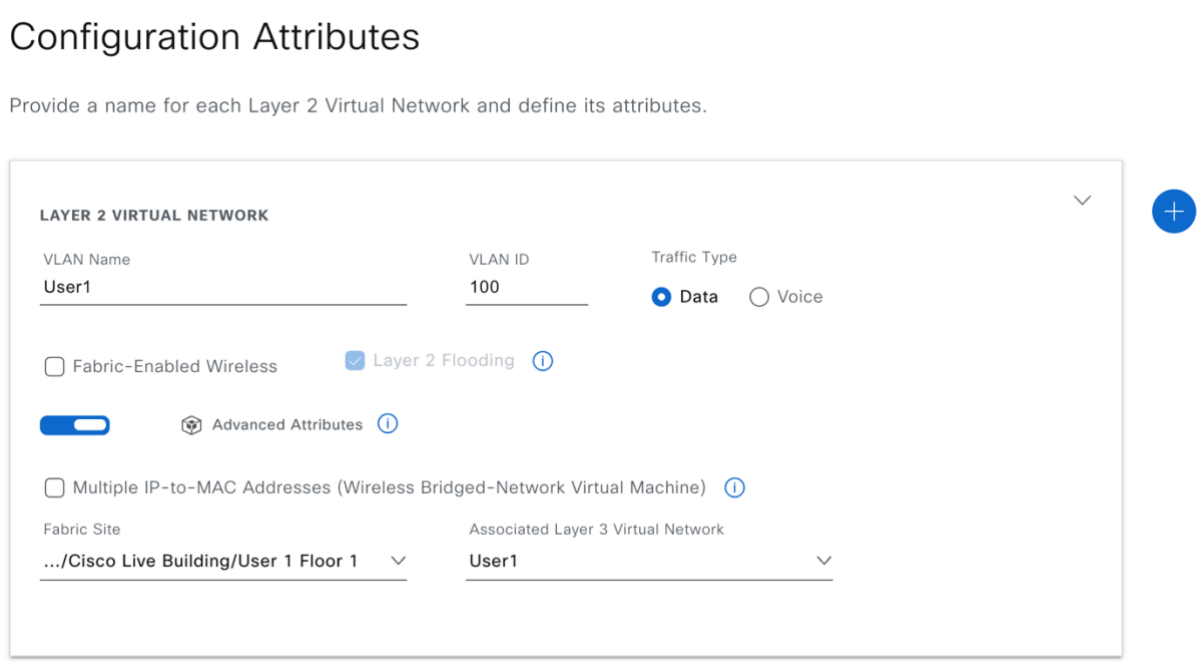




Just as before, Save intent and Deploy.
Always wait for these tasks to complete.

Step 5: Using the next navigation option on the page, create Layer 2 Virtual Networks



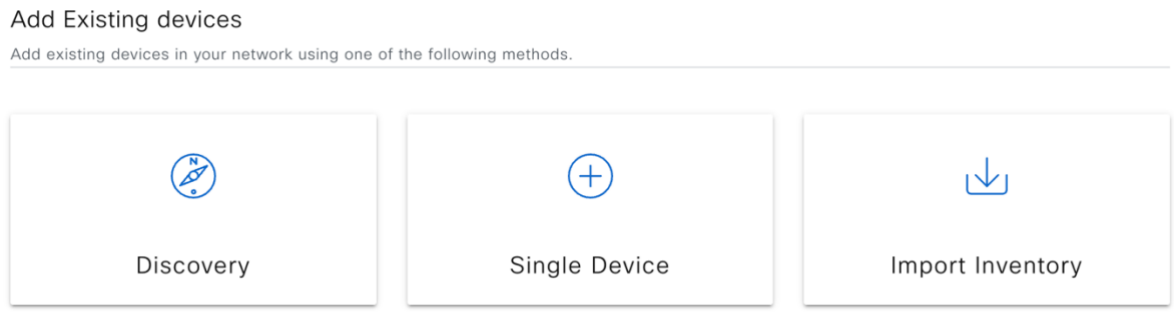


Enable the advance attributes and also map your fabric site with the layer 3 VN.

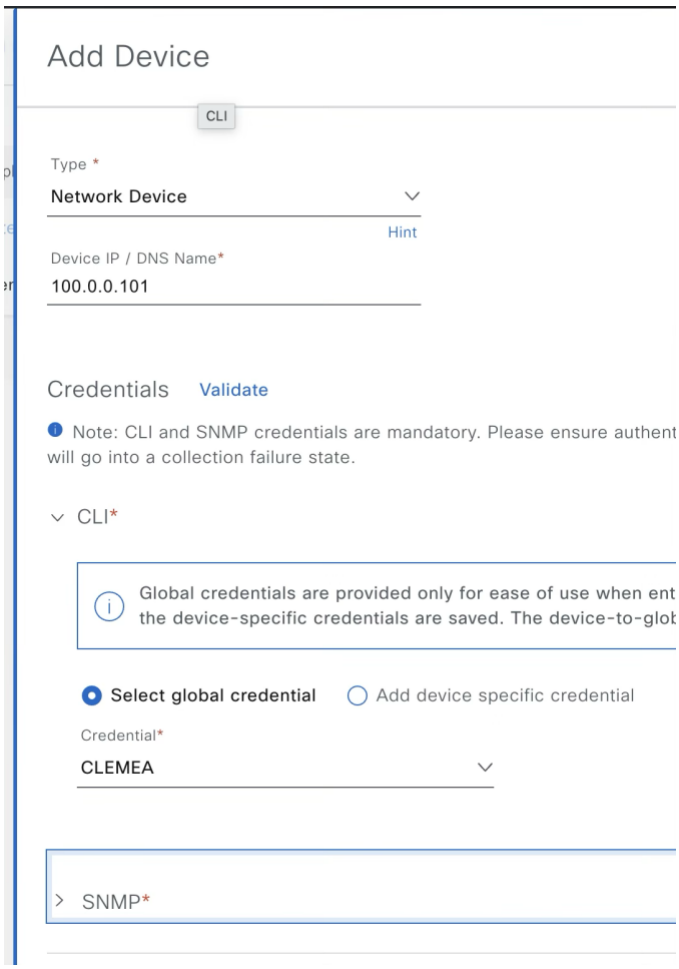
Every user can create a vlan id based on their number, for example User 16 would have vlan 160 and user 17 vlan 170, and so on

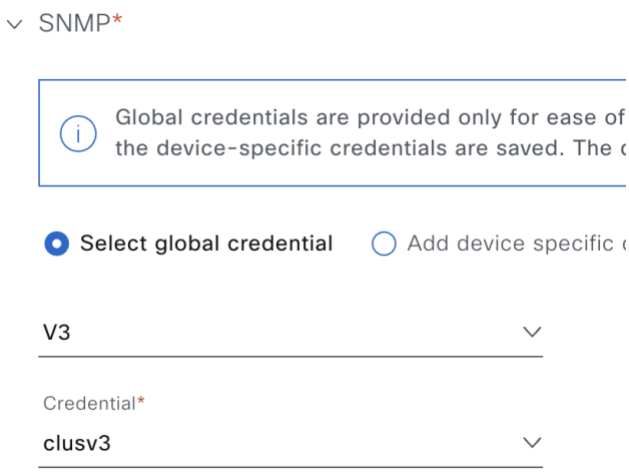
Save the intent and deploy.

Step 6:
We will now start with the discovery of your network device so that it can be provisioned by CatC with the correct role.
Go to provision> Network devices > Inventory > click on Add Device to add a single device to your network.



Using the Loopback IPs mentioned in the table below, enter the loopback IP of your device and then select credentials as per the image:





Then protocol must be set to SSH2, click Add.

Reference: This is the table with the Loopback IPs of all 15 switches in each session:

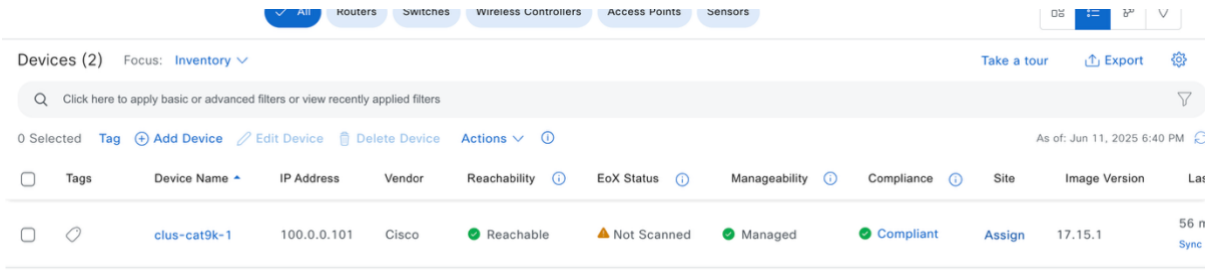
Table: FIAB Loopback IPs for claiming a device

Fiab 1	100.0.0.101
Fiab 2	100.0.0.102
Fiab 3	100.0.0.103
Fiab 4	100.0.0.104
Fiab 5	100.0.0.105
Fiab 6	100.0.0.106
Fiab 7	100.0.0.107
Fiab 8	100.0.0.108
Fiab 9	100.0.0.109
Fiab 10	100.0.0.110
Fiab 11	100.0.0.111
Fiab 12	100.0.0.112
Fiab 13	100.0.0.113
Fiab 14	100.0.0.114
Fiab 15	100.0.0.115

The session 1 has these names for the first 15 users.
With **Session 2,** the second CML session also uses the same set of loopback IPs on the devices.
So user 16 can use the same loopback as the user 1.

After adding the device, it will show up in inventory while its reachability and status will be updated in a few minutes.

The CatC will use the provided credentials to login to the device and configure/read it via SNMPv3.

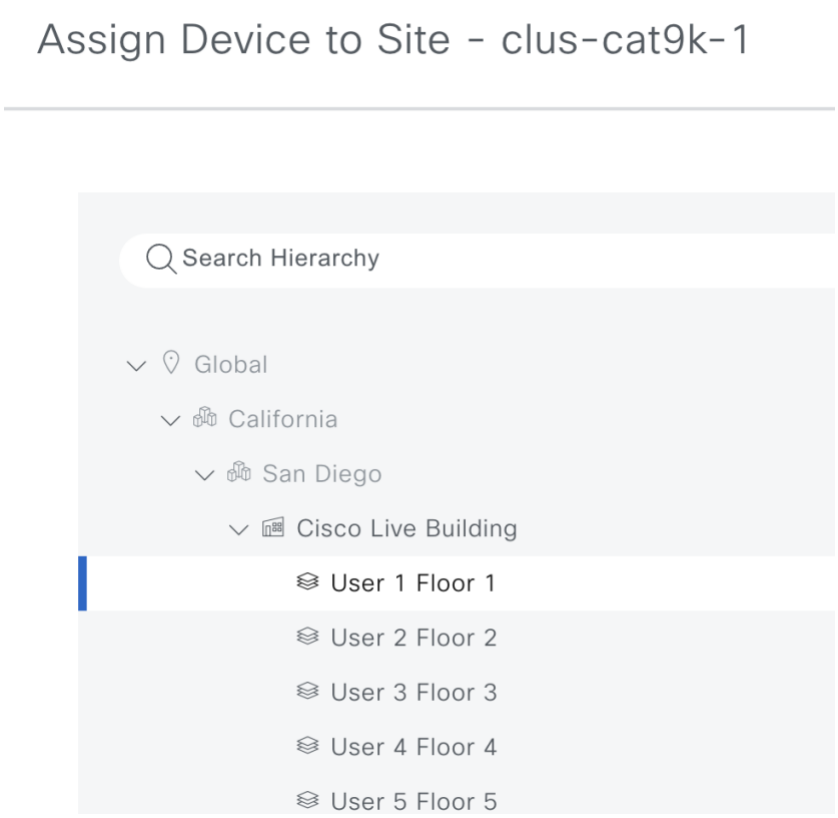


After your device shows as reachable and Managed, proceed to next step.

Step 7:

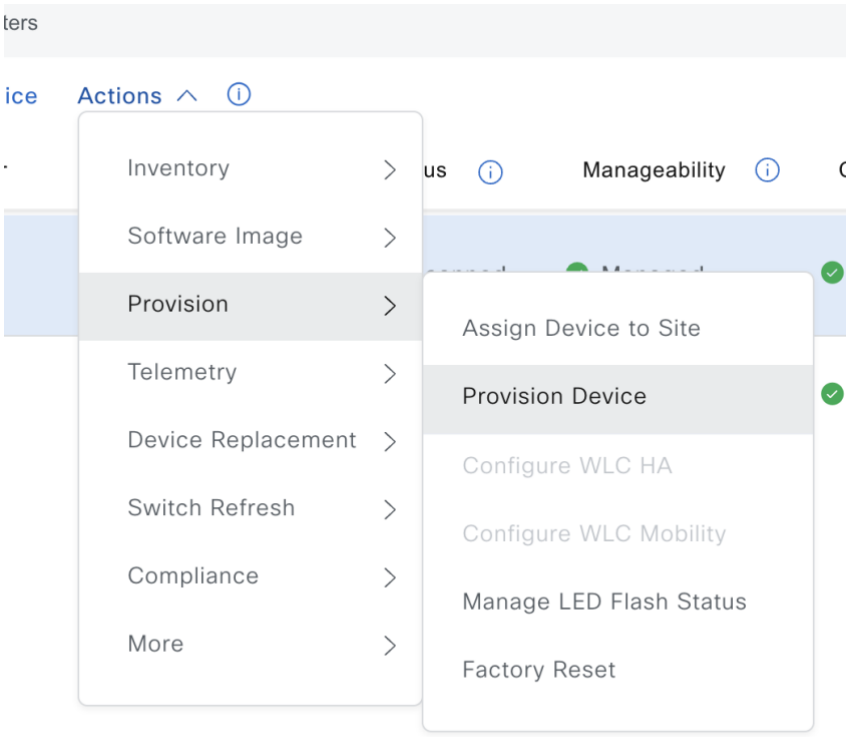
We will now assign the device to the fabric site that was created.

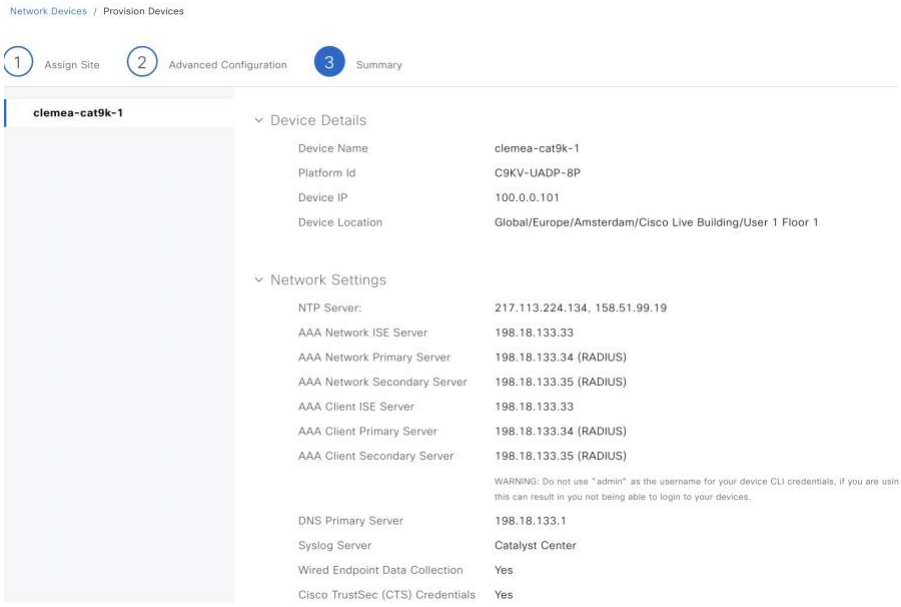
On the same page as step 6, click on “Assign” next to your device and proceed to select the fabric site that you created earlier.



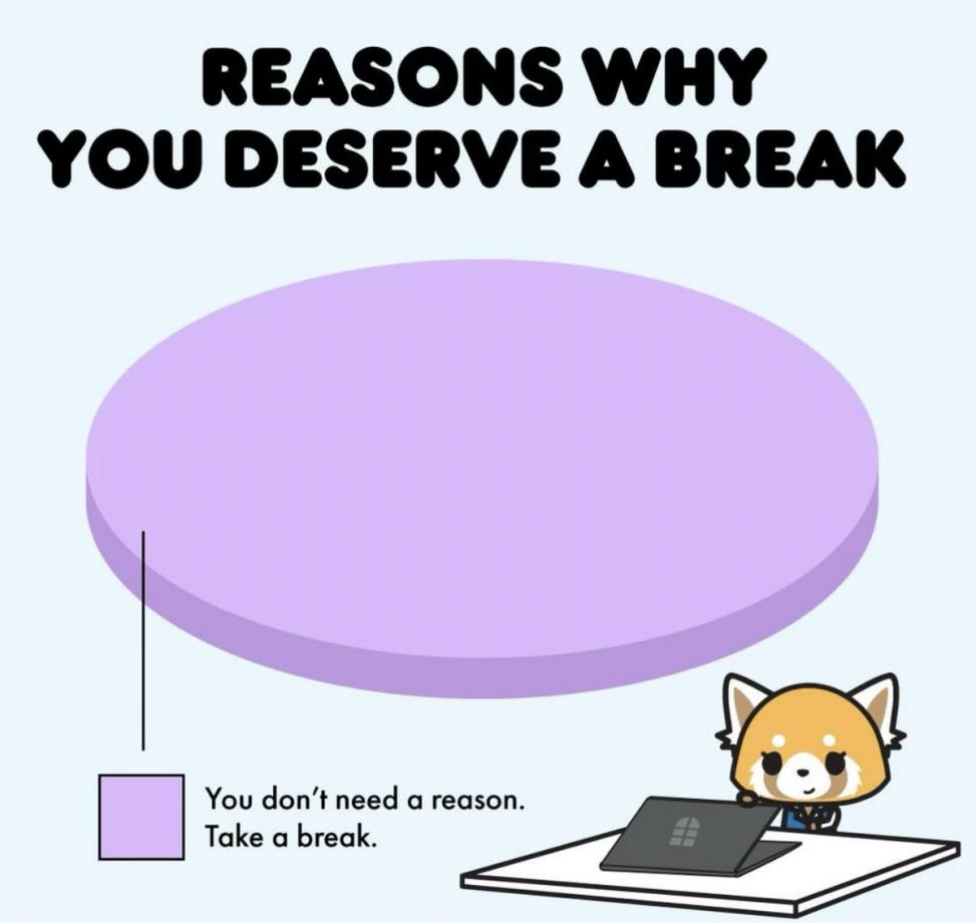
Step 8:
As the next step of building the SDA network, we must now provision our device with the configuration data present on CatC.
Provisioning a device adds all the relevant AAA servers, settings, etc. so that the device can now be used and managed via CatC.

Go back to provision> Network devices > Inventory and select your device.
Under Actions, provision the device.



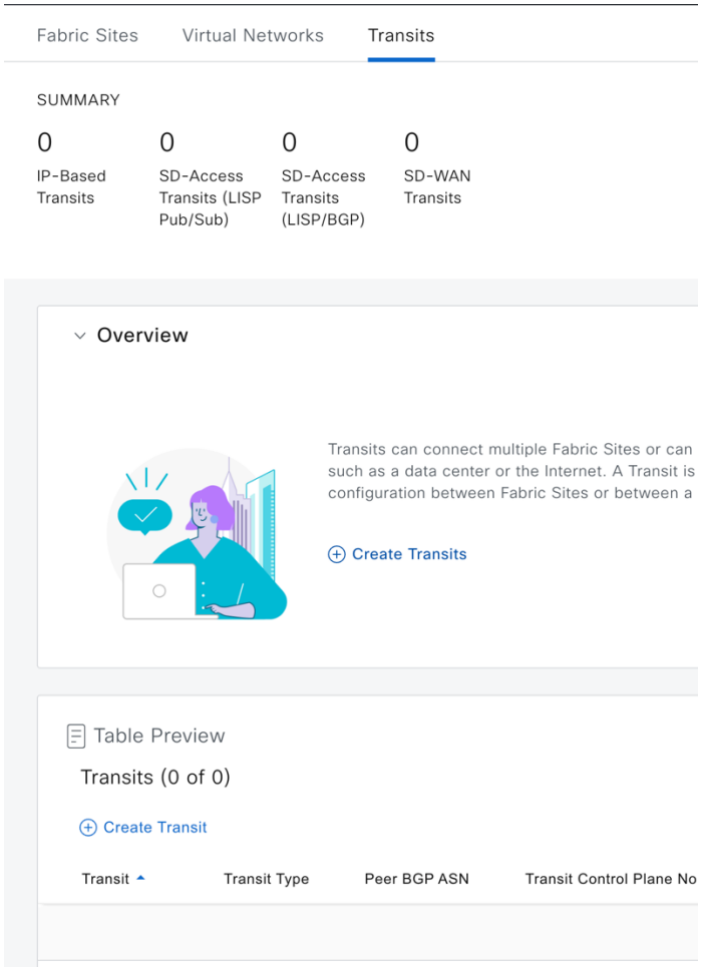


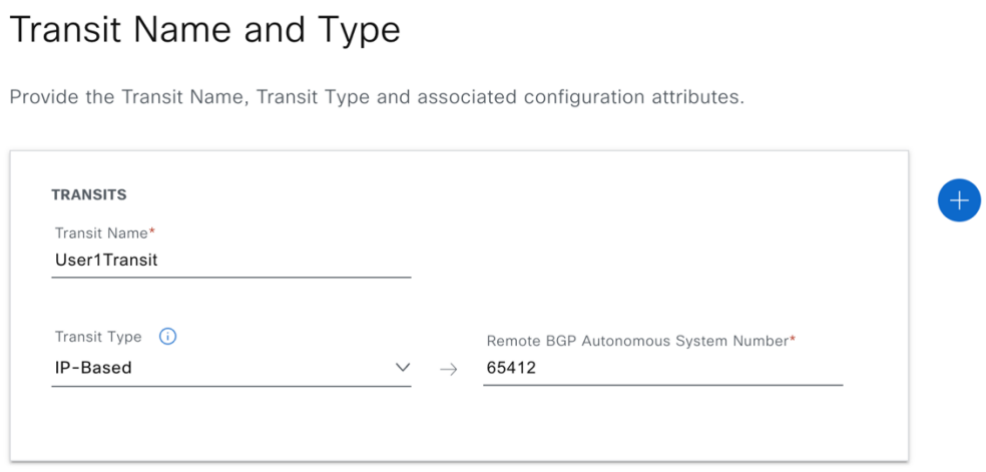
Provisioning a device can take up to 5 mins or more. After starting this, please review the task status from Activities > Tasks.
Wait for this task to complete.



Step 9:
The next components of an SDA network to be added are Transits and assigning device roles.
We will make use of traditional IP based transits.

Go to Provision > SD Access > Transit > Create



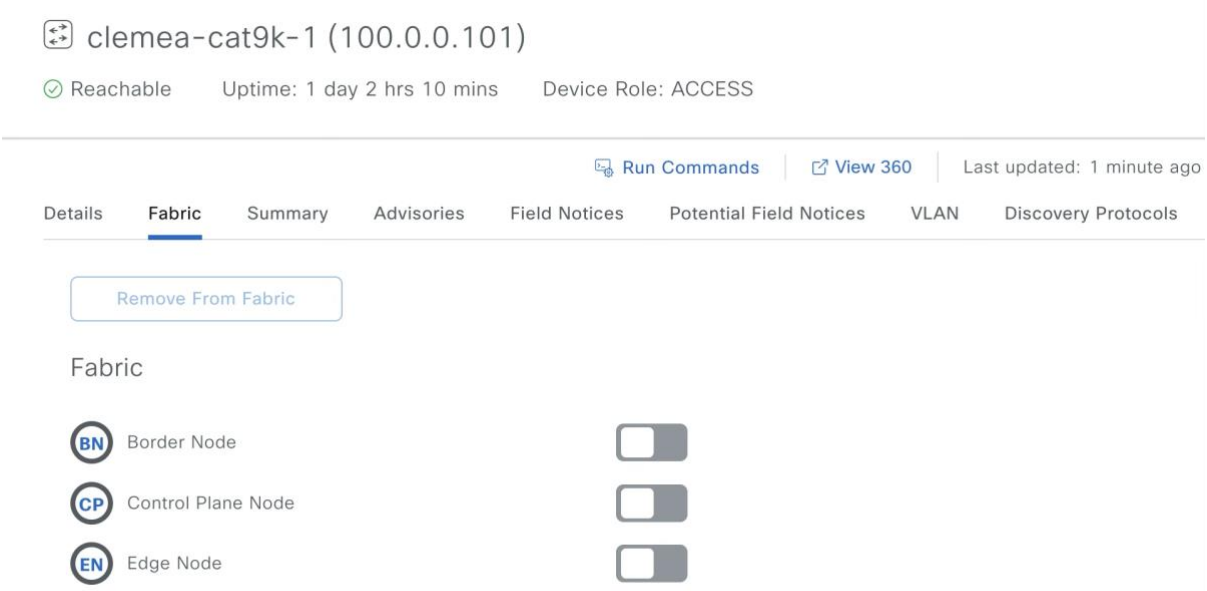


The BGP number can be any value within a valid range.

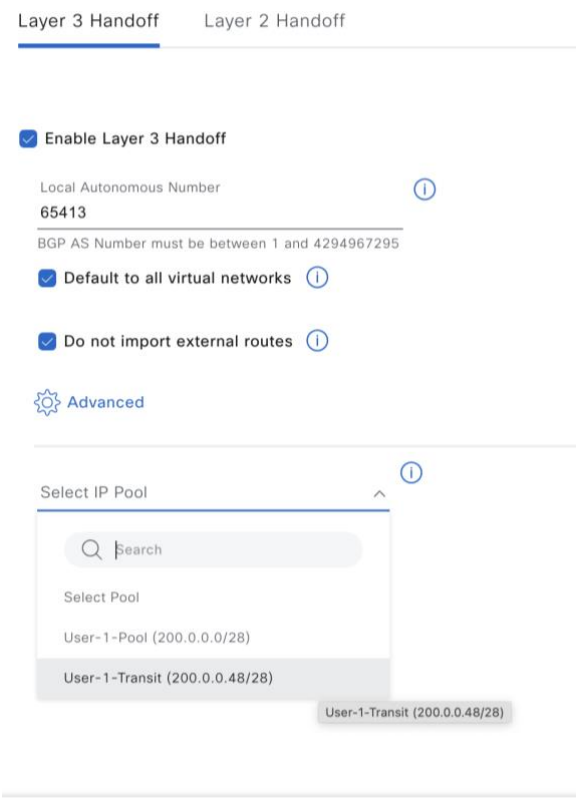
Save intent & Deploy.

Step 10:
By now, your device provisioning must have been completed. Proceed to configure roles on this device.

Go to Fabric Sites and open your assigned fabric.
The device your provisioned and assigned must be visible here, click on it to reach this page:

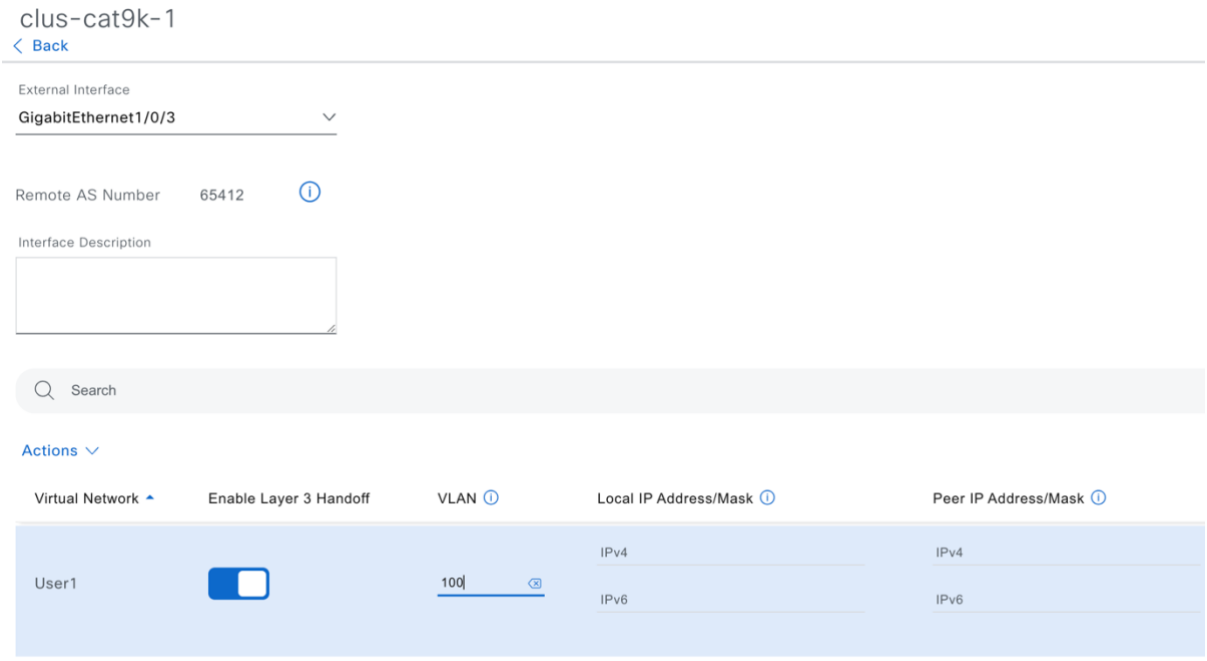


Let’s start by assigning the role of Border first,



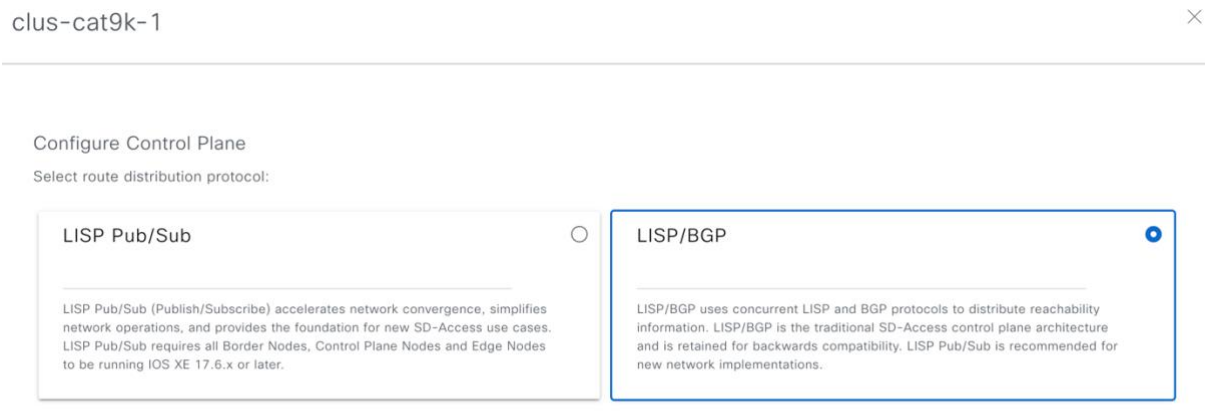
Enable the setting as shown in the image above, also selecting the IP Transit pool reserved for your user.

Add external interface, which is always Gig 1/0/3, as described in the table earlier.



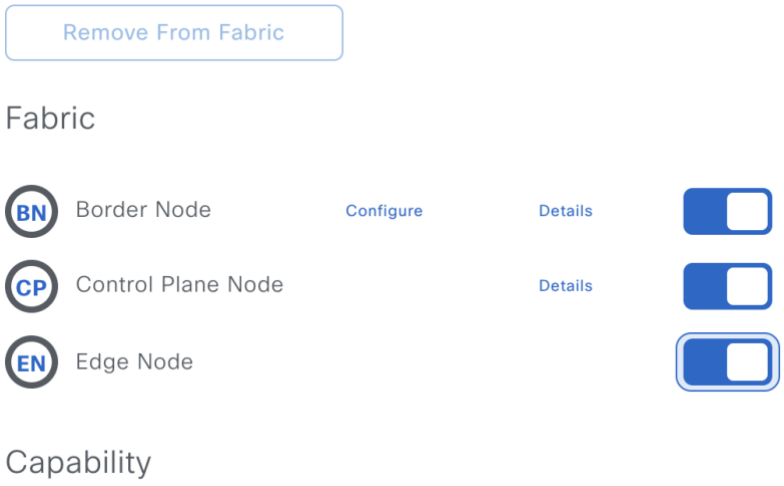
If you had selected the IP pool in last step, then there is no need to add IP addresses/mask here.

As next step, enable Control Plane with LISP/BGP



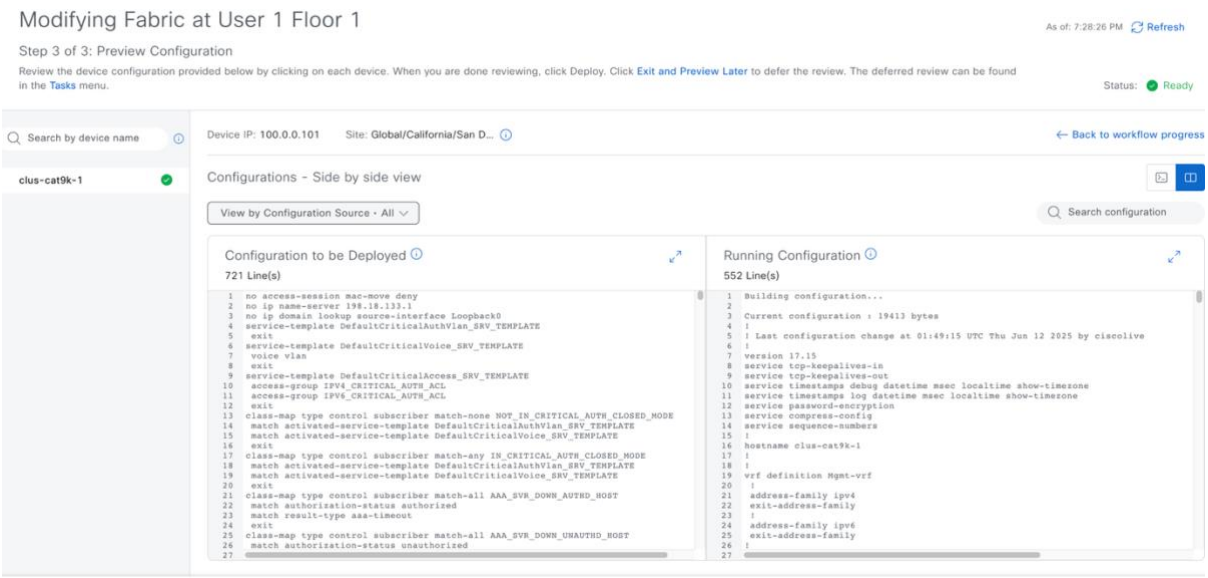
Also enable Edge node.

In the end, it should look like this:



Save and deploy.

The CatC will show you the configuration before pushing:

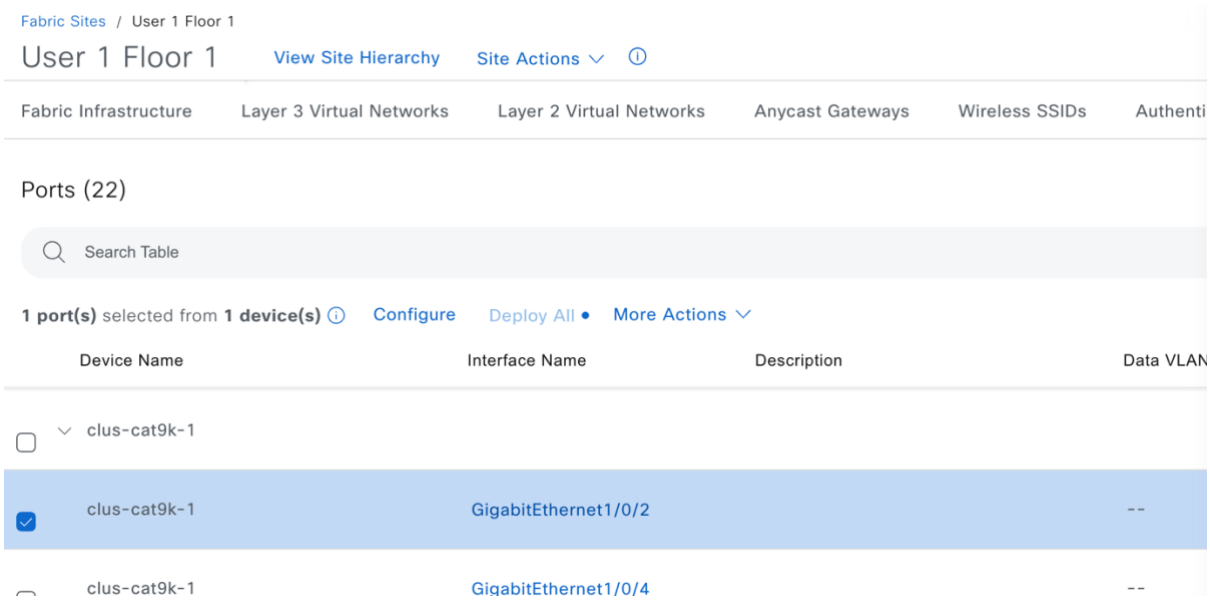


This task can take more than a few minutes, please monitor the status from Activities > Tasks.

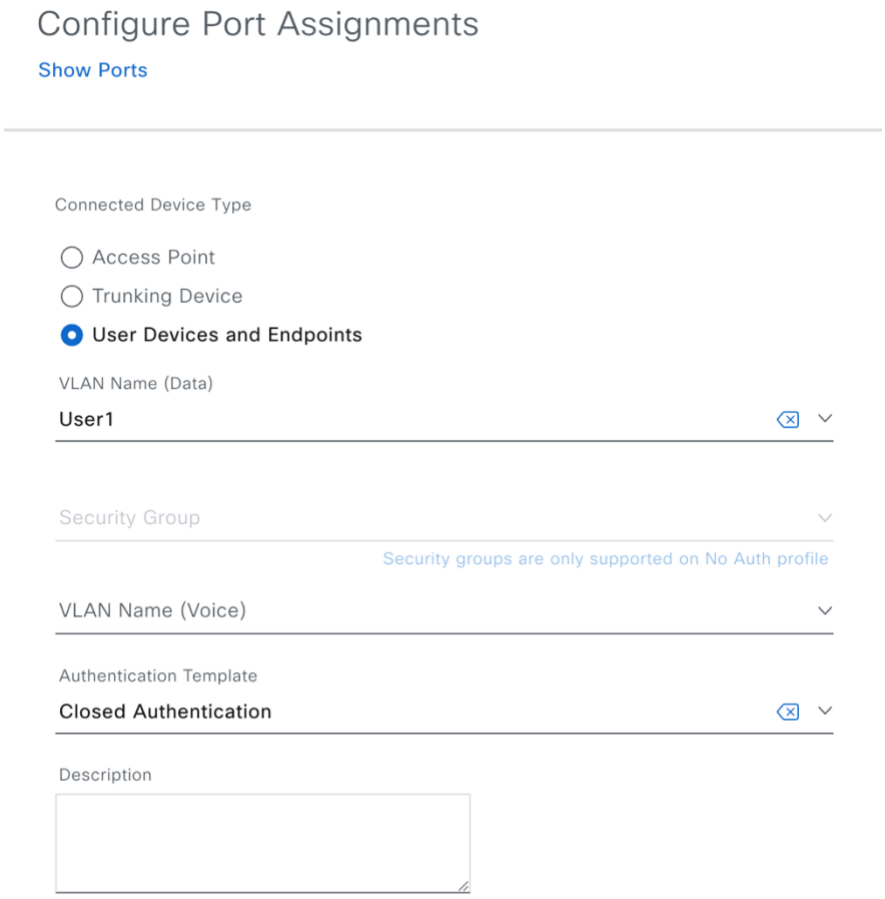
Step 11:

After the device has the assigned roles from the last step. Come back to the Fabric site page and go to “Port Assignment”.

Every device has the Ubuntu connected to Gi 1/0/2, so this is the port to configure with Dot1x.



Select the role as “User device and endpoints” and assign the appropriate vlan and template.



Click on Deploy All > Apply.

Wait for this task to complete.
We have now successfully completed the SDA part of the configuration.

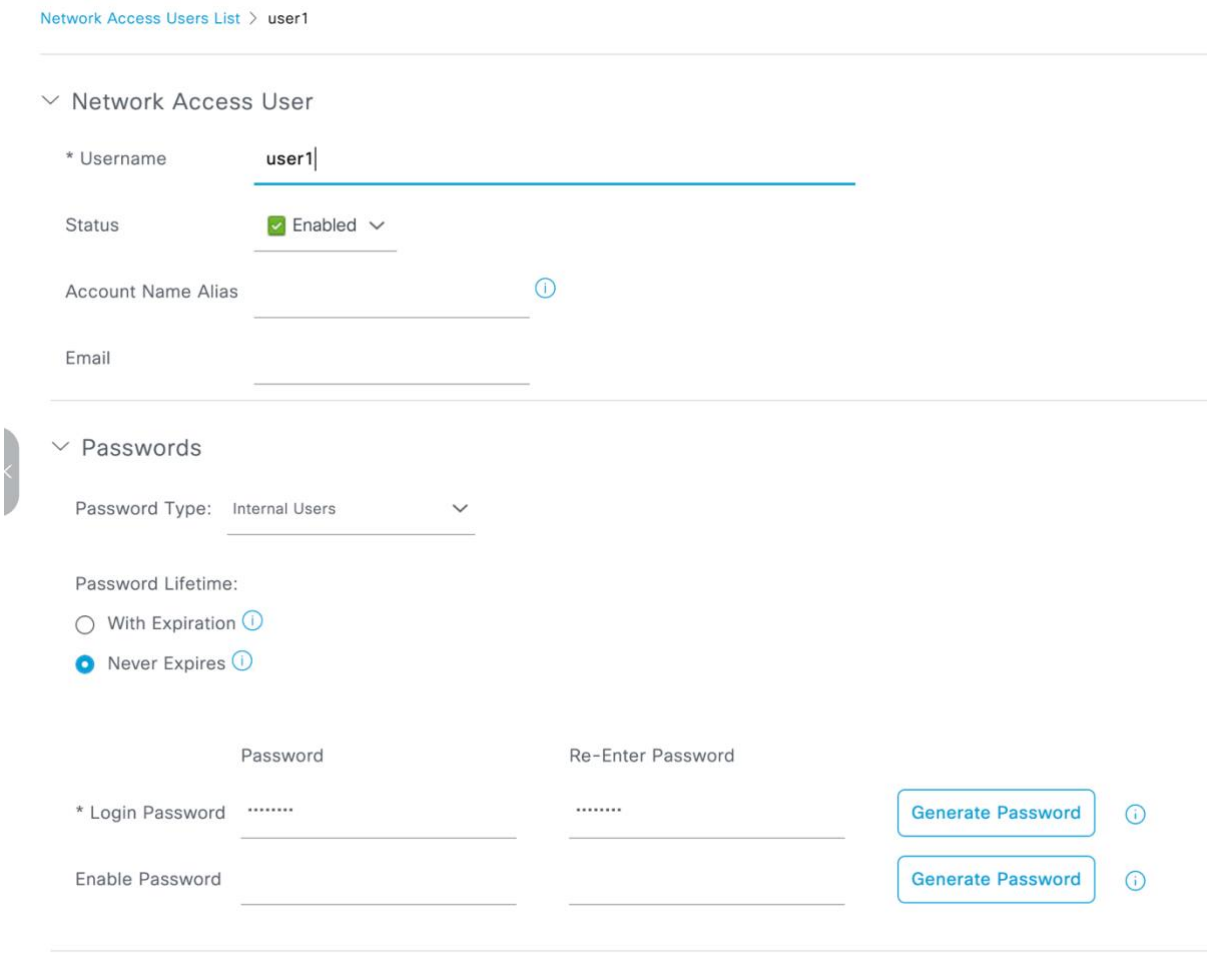
Step 12:
We will now proceed with creating a user and policy on ISE.
For 802.1x to work, we need a policy corresponding to your username and an internal user account that can authenticate on ISE.

Login to the ISE server corresponding to your session.
Go to Administration > Identity Management > Identities > Users

Add a user based on your account

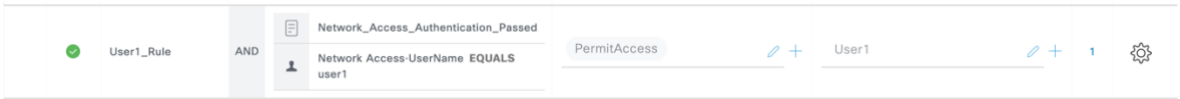
Then dot1x users must have these credentials:

Username: user1 (with the number changing based on your pod)
Password: L0veCiscoLive



Now Navigate to Policy > Policy sets > under the default policy

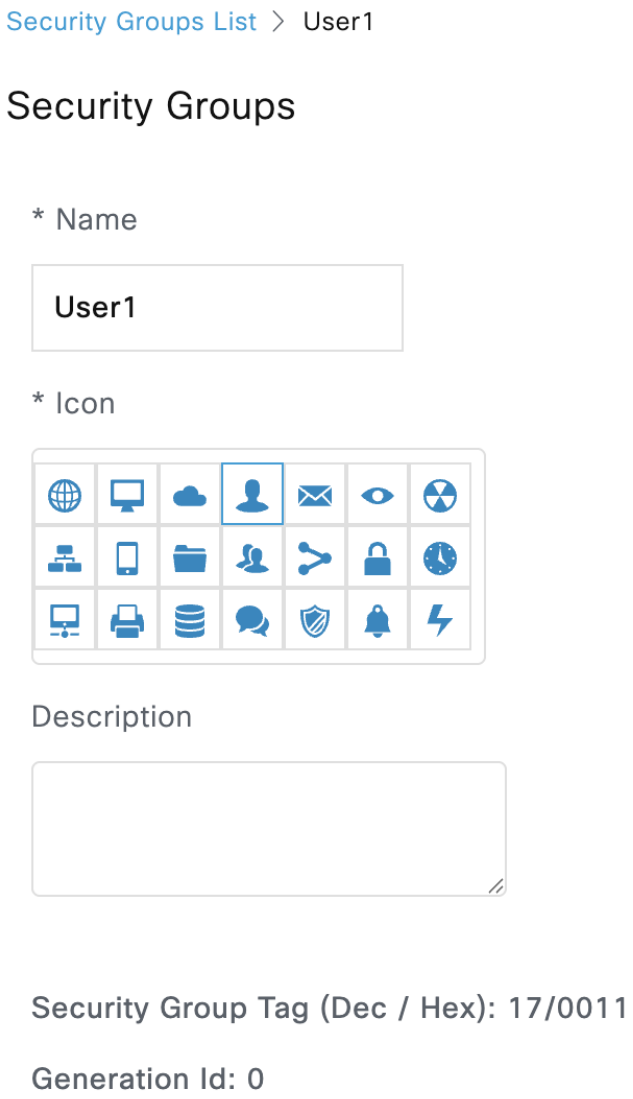
Under Authorization , you may see a sample policy for user1 is already created, which you can then use to duplicate and create your own specific policy.



The SGT to be assigned here can be created as described below:

Go to Work centers > Trustsec > Trustsec components and add an SGT.

If an SGT for your user does not exist yet, create it here.



The SDA network is set up and ISE is configured to dynamically assign SGTs to users.

Step 12:
The next step would be to configure Ubuntu device with correct Dot1x credentials.

These devices have been pre-configured to do Dot1x on ens3 port, connected to the fabric.

Next, manually assign an IP address (from your reserved pool) and trigger a Dot1x authentication.

Login to Ubuntu console from CML

Hint: Go to CatC and look at IP address pool reservations of your Floor, to find an appropriate IP and default gateway for your endpoint.

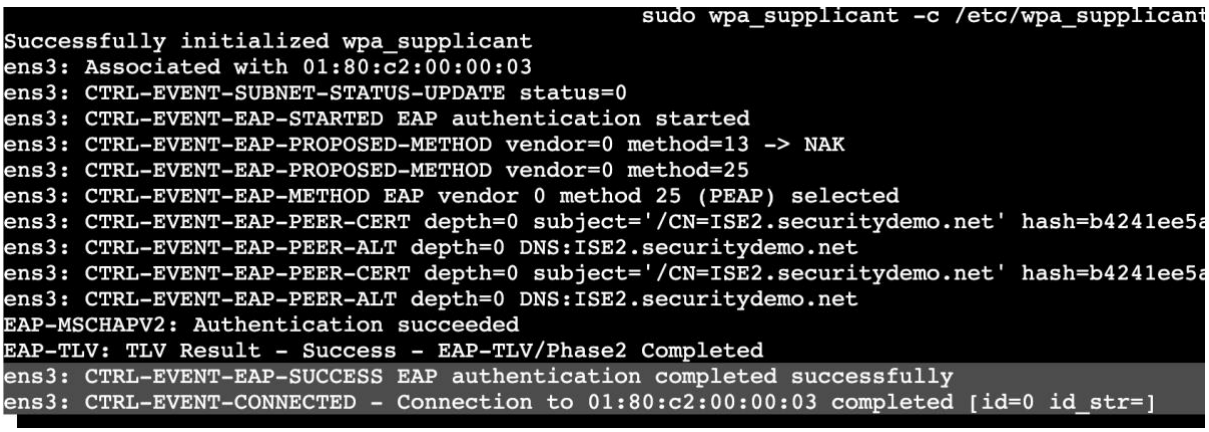
Example of adding IP adddress for User1 from pool assigned to it:

```
sudo ip address add 200.0.0.18/28 dev ens3

sudo ip route add 0.0.0.0/0 via 200.0.0.1
```

Trigger a dot1x authentication using this command:

```
sudo wpa_supplicant -c /etc/wpa_supplicant/wpa_supplicant.conf
-D wired -i ens3
```



A message will appear on screen saying Authentication Succeeded if it worked.

You can also verify the assigned SGT and auth status from the console of your FIAB and from ISE live logs.

Note:
In case your ubuntu machine does not have the EAP configuration, you would need to connect ens2 to ext-connector and then run the following commands

sudo apt-get update
sudo apt install wpasupplicant

cd /etc/wpa_supplicant/

sudo vi wpa_supplicant.conf

```
country=US
ctrl_interface=/var/run/wpa_supplicant
update_config=1
ap_scan=0
network={
        key_mgmt=IEEE8021X
        eap=PEAP
        identity="user29"
        password="L0veCiscoLive"
        eapol_flags=0
}
```

sudo ip link set ens2 down

sudo ip link set ens3 up

Task 3: ACI and ISE integration

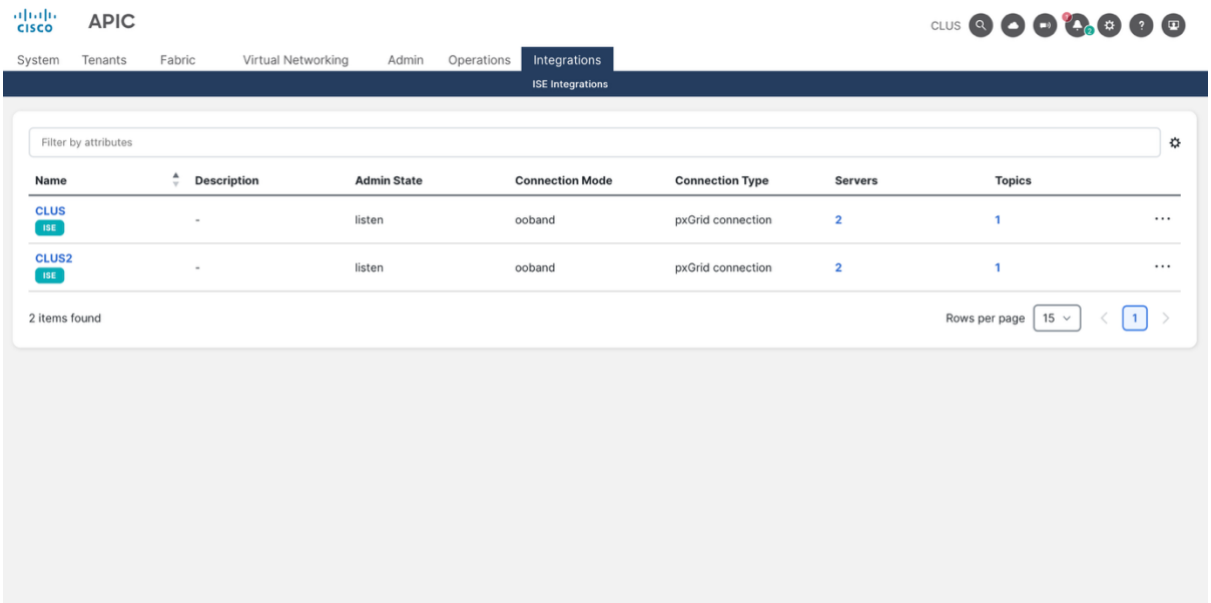
In this Task we will see the workflow to integrate ISE and APIC. ACME Corp is looking to start embracing common policy and wants to see how this is possible.

The basic integration has already been done for our whole infra so we will only be walking through the integration parameters in the first half of this task. In the second part, we will work with our EPGs/SGTs for several use cases.

Step 1:

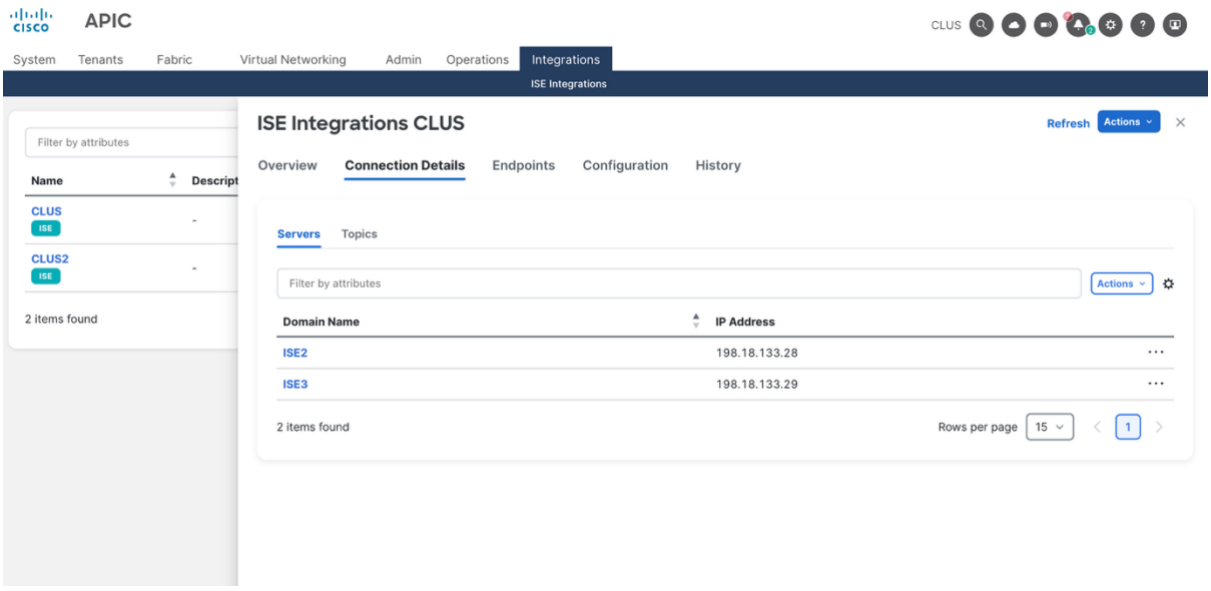
Head over to the ISE and APIC on different Tabs.

On the APIC let us login and go over to the ‘Integrations’ tab



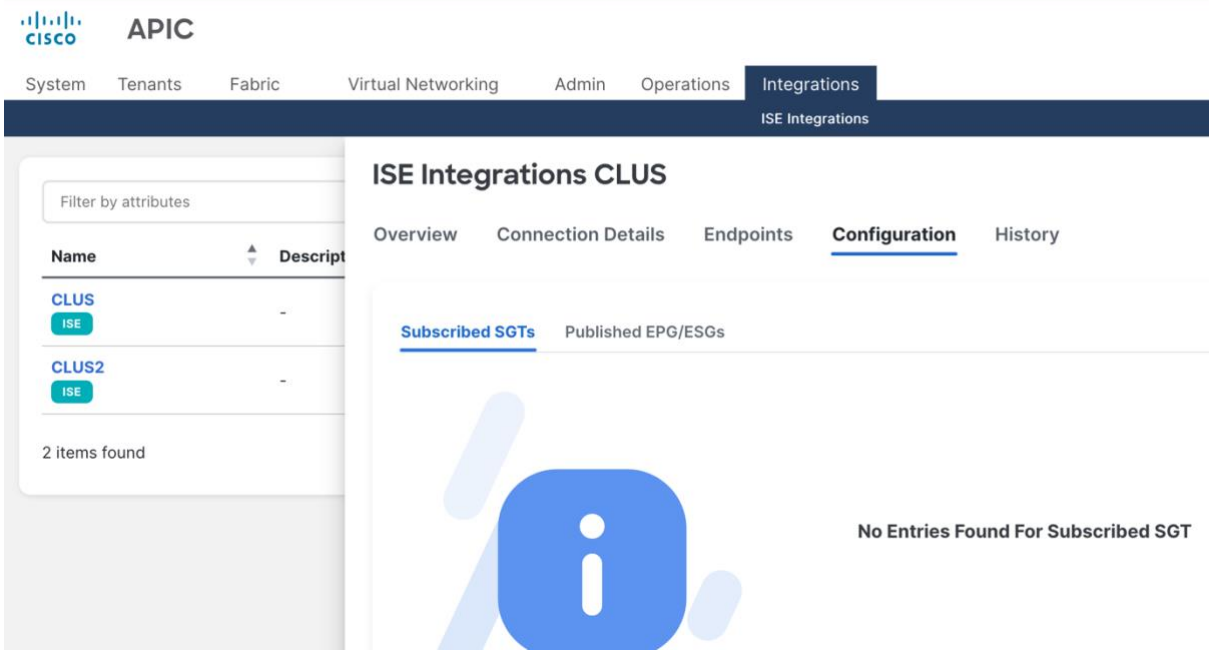
You will see 2 integrations. We will use CLEMEA1

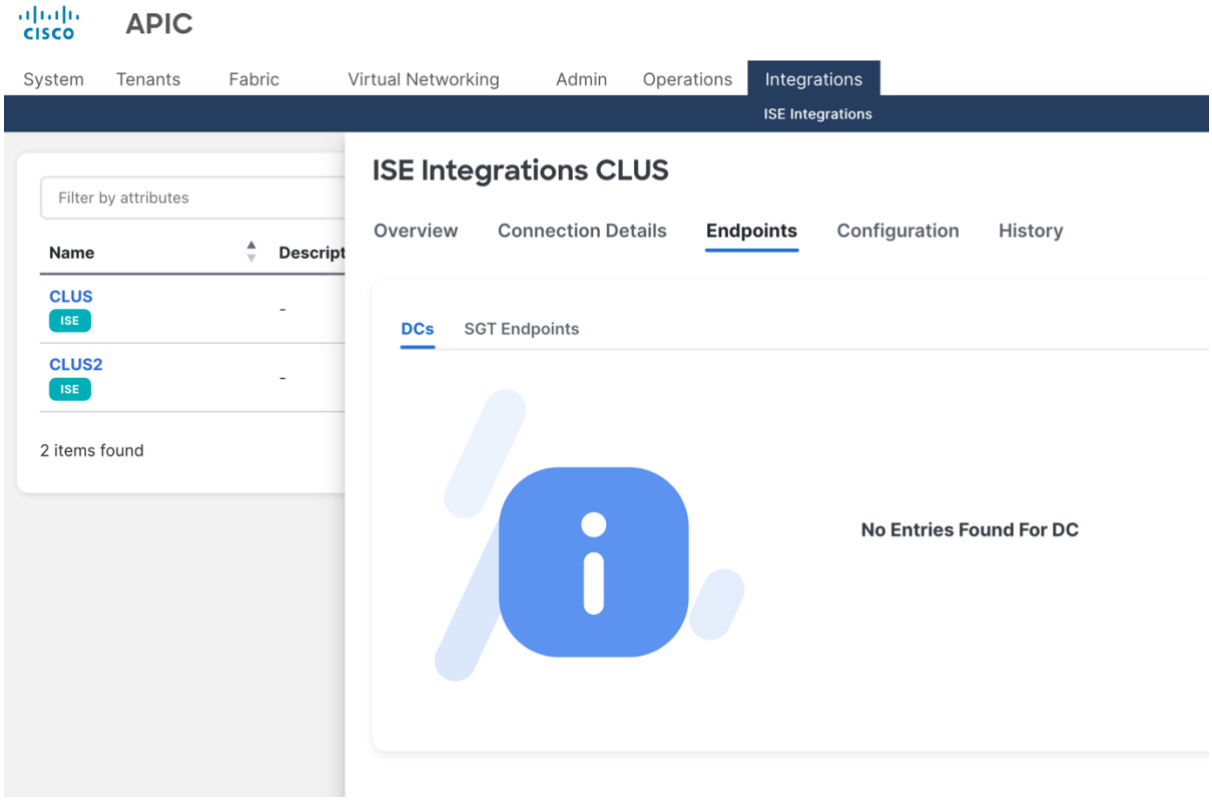
Click on your respective connection and Connection Details



You will see the IP of the integrated ISE

Initially the Endpoints and Configuration tabs are empty. They may have some details from other users but wont have your EPGs/SGT.





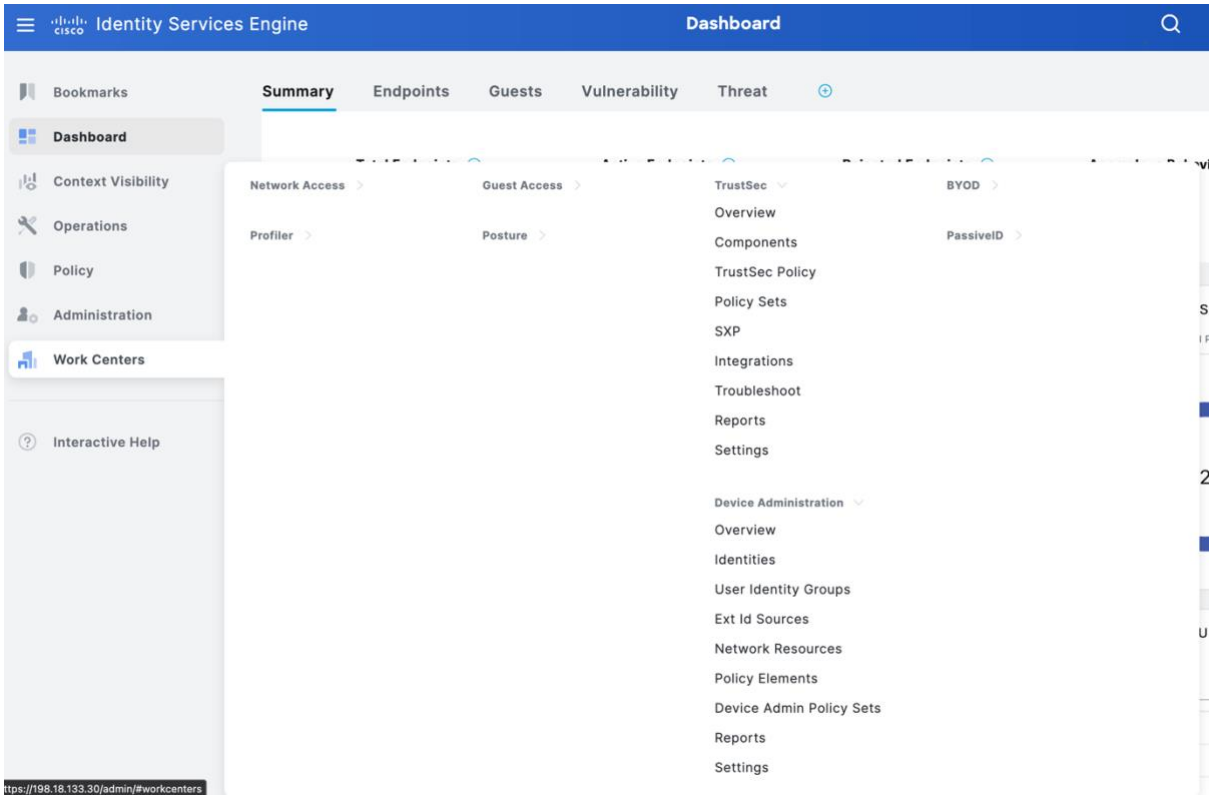
Step 2:

All of this is auto imported and pushed when we go through the integration workflow on ISE.

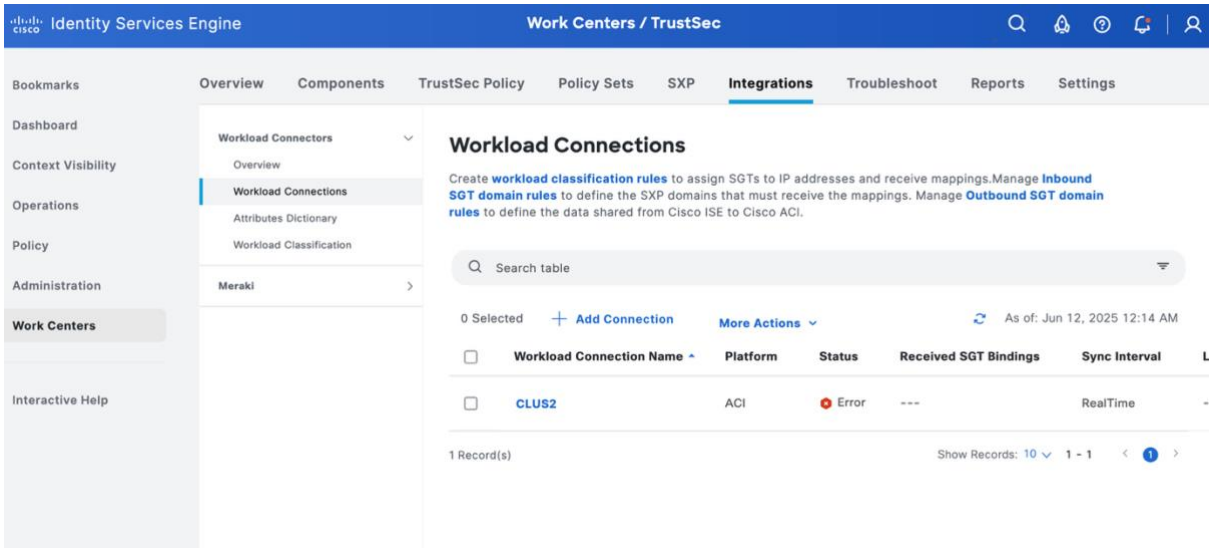
Let us head over to ISE and examine the integration workflow from ISE's Perspective

Once again to remind you we will use ISE 198.18.133.33.

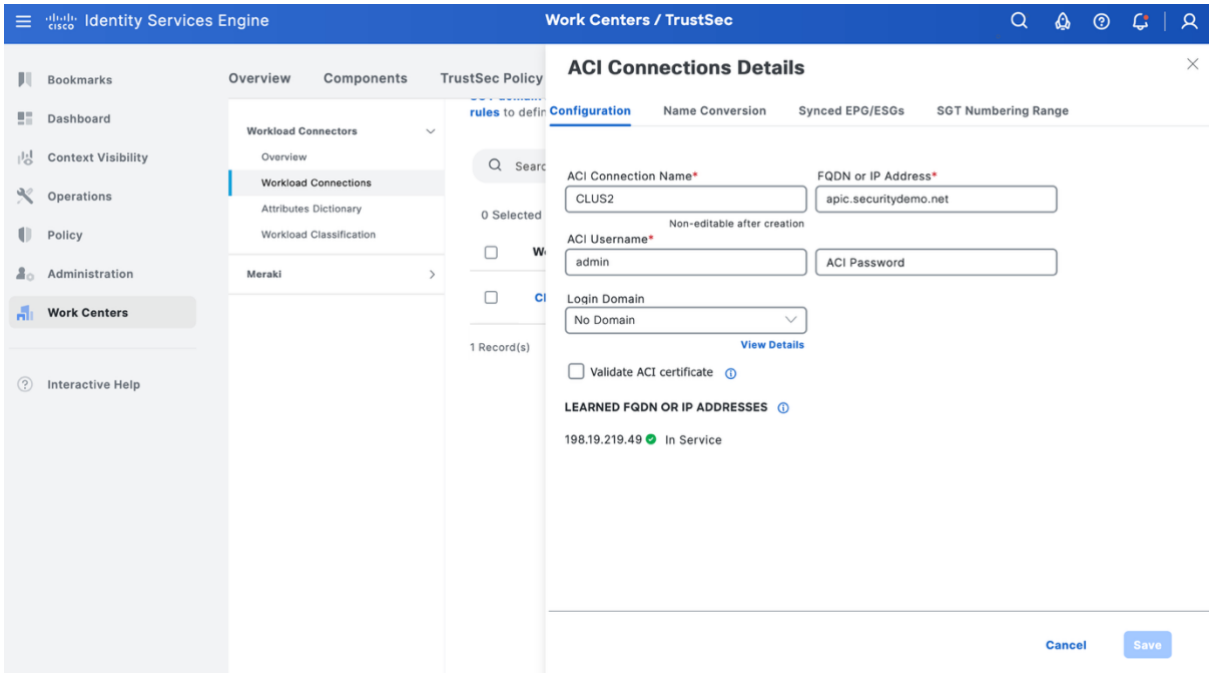
Once you login to your respective ISE and go to Work Centres> Integrations

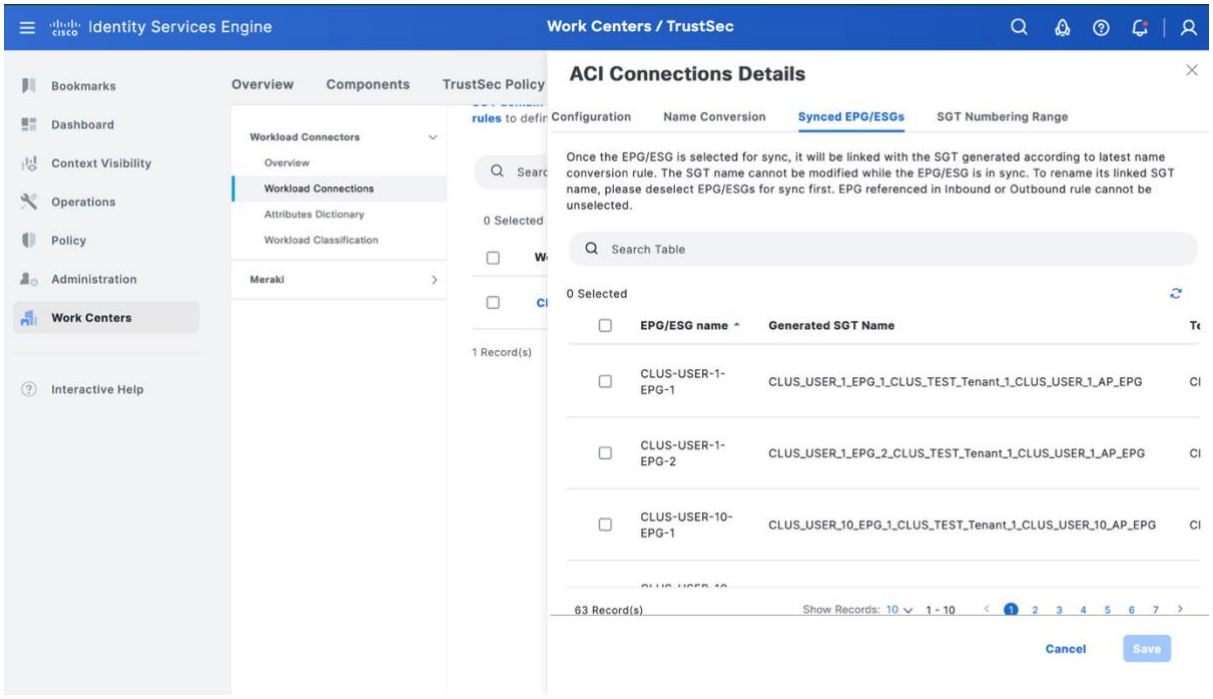


Click on Workload Connections to see our connection



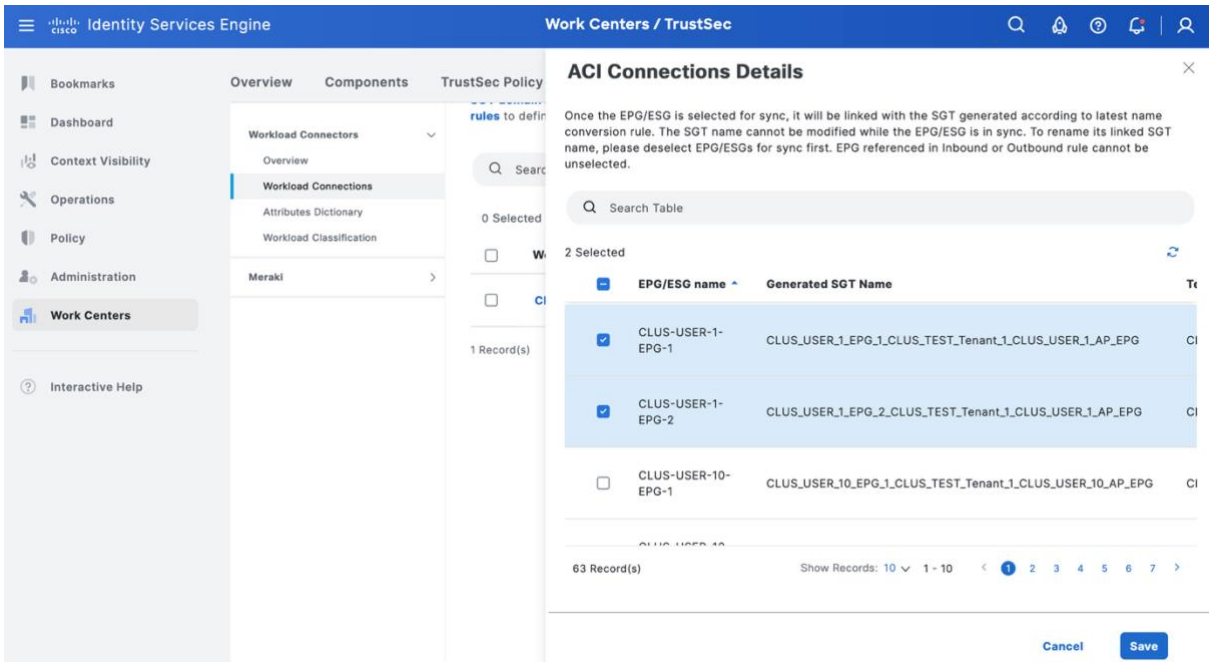
Click on the connection to examine how it integrates with ACI





Step 3:

Select the 2 EPGs you have been assigned and select Save.



Be careful in case someone else is saving as well at the same time- it may need you to resave.

This concludes the workflow of integration between ACI and ISE.

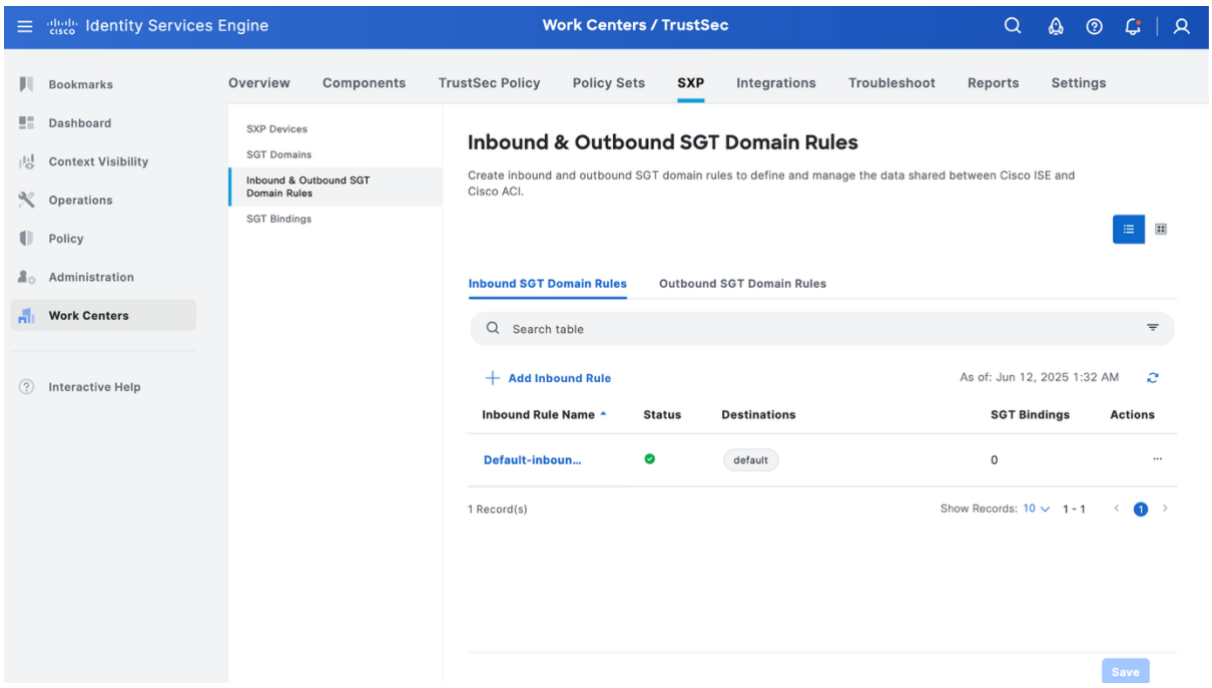
In the Next Task we will use Inbound and Outbound policies for communication

Please do reach out if you have any questions so far as the next task will require clear understanding of SGTs and EPGs/Contracts.

Task 4: Inbound and Outbound Policies

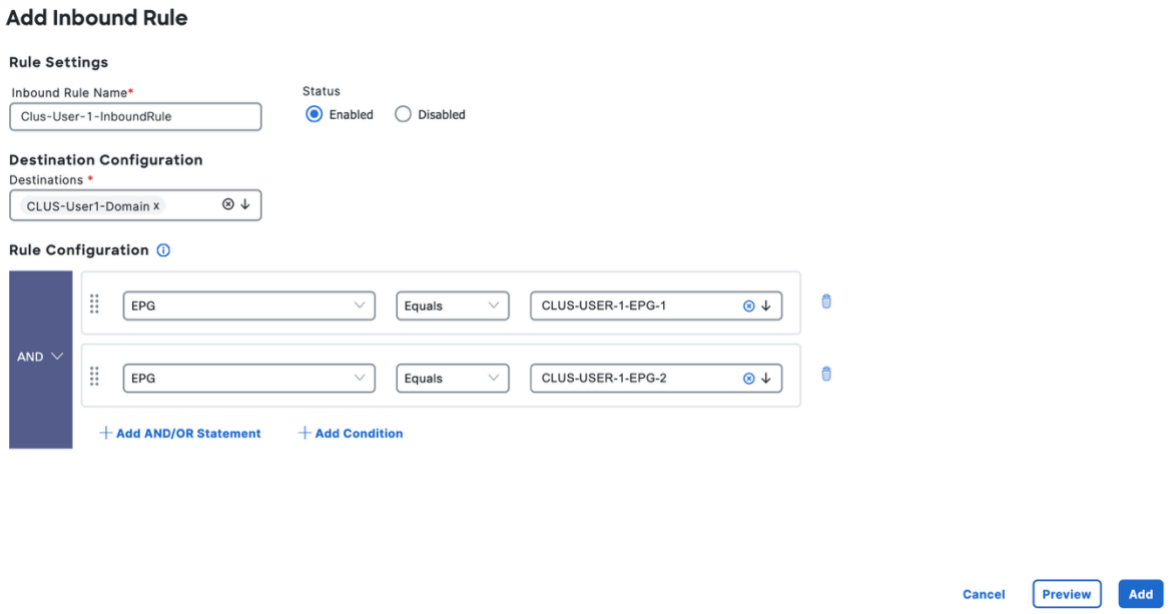
Inbound and Outbound Policies
We will now go over the use cases and options to integrate.

Go to Work Centers and SXP Menu and head to the Inbound and outbound SGT Rules Tab.



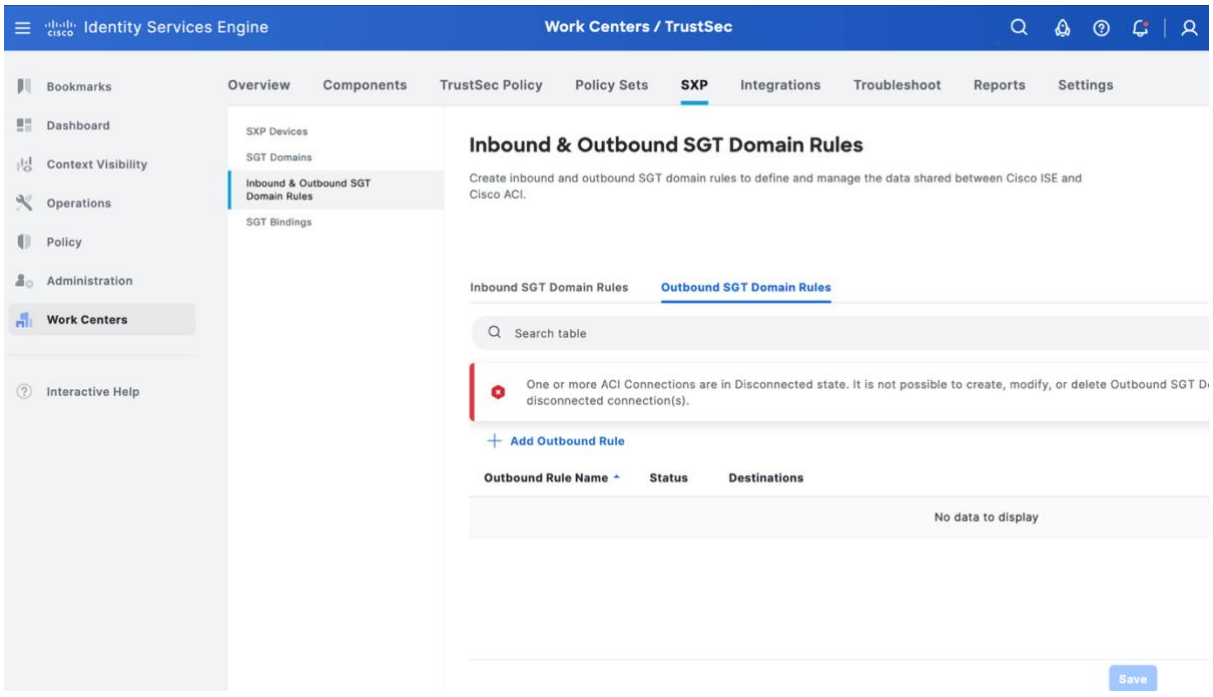
Create a rule by clicking on Add Imbound Rule

You may create a new destination with your user name. Make sure to add the 2 EPGs assigned to your user name

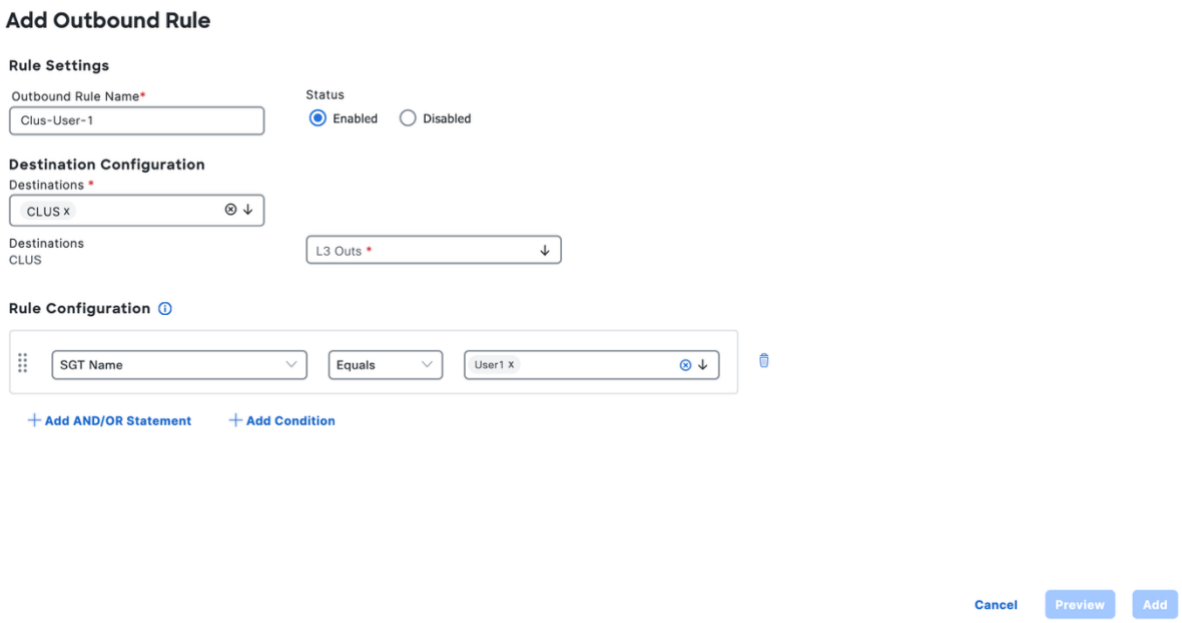


This way you can use the EPG related classifier on the SDA side.

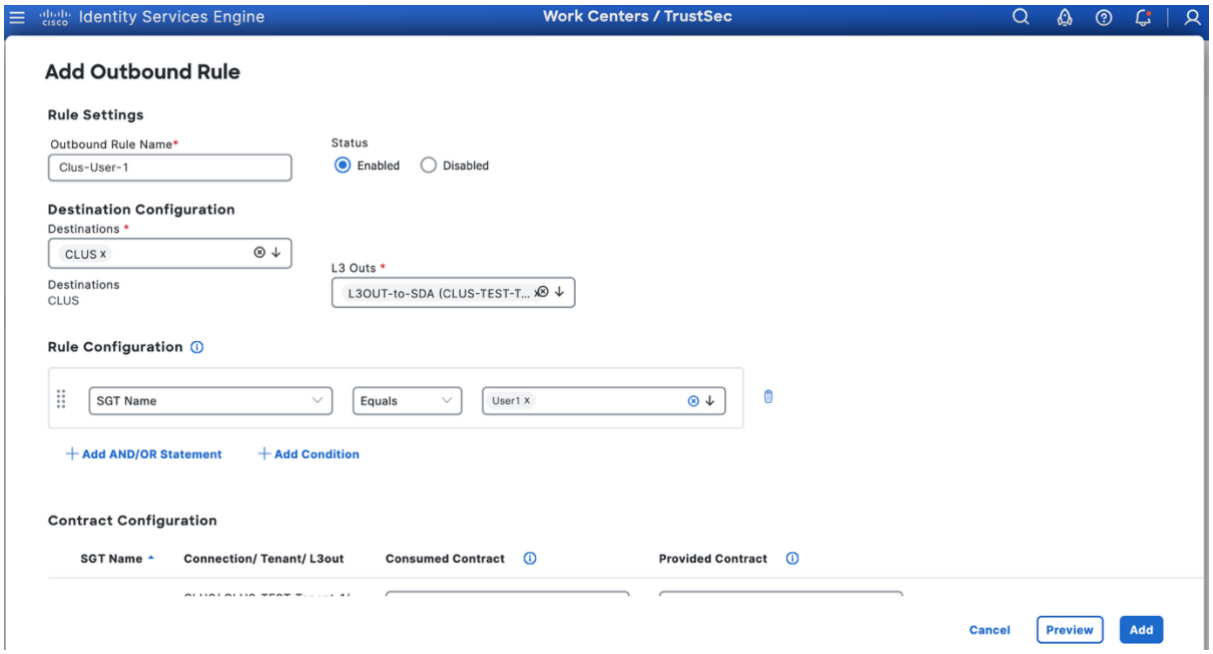
Now let us check outbound policy. Make sure to save before.



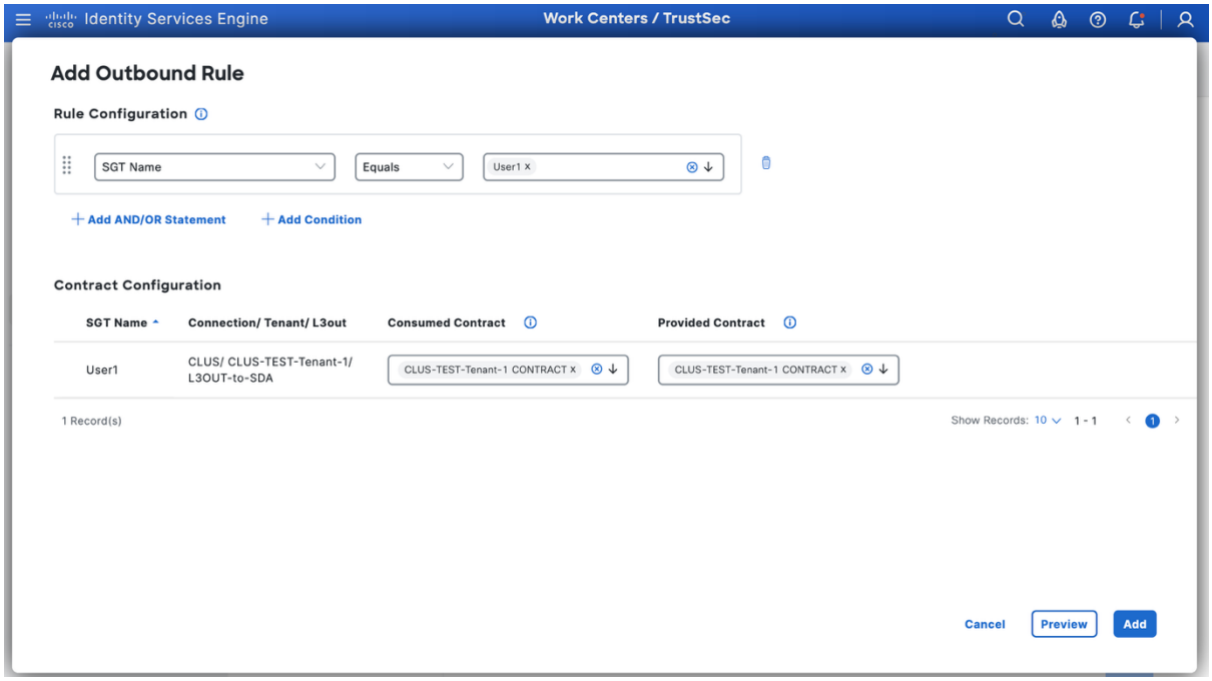
Go over to the outbound policy tab and click on Add Outbound Rule



You can make it match the previously created SGT that you used.



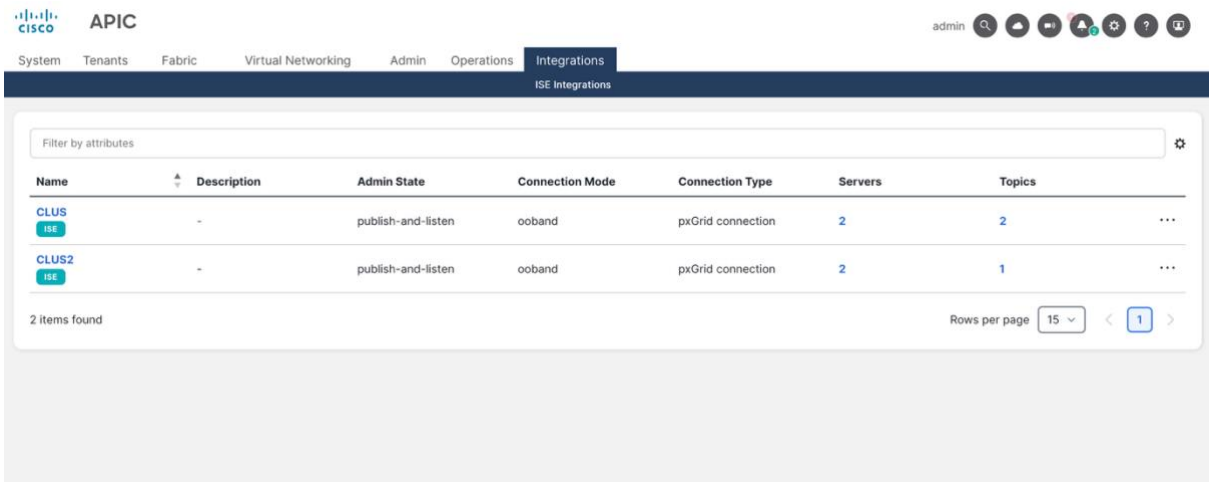
We then also need to show which contract we use for which communication. Select the Contract we used in Task 1 as both consumed and provided.



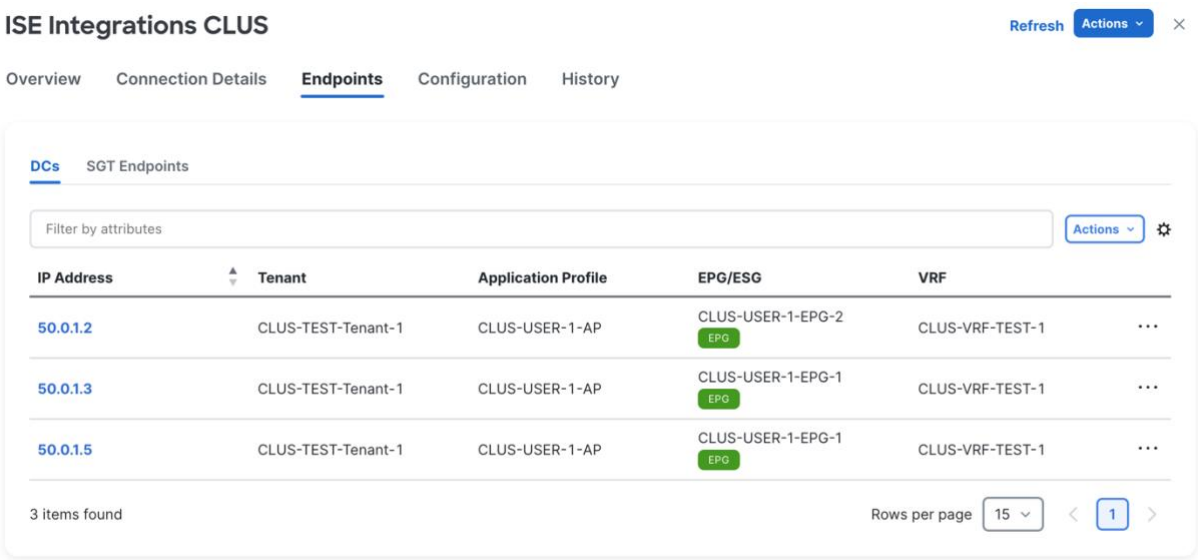
Add and then proceed to save.

Now let us head over to ACI and look at what this config caused.

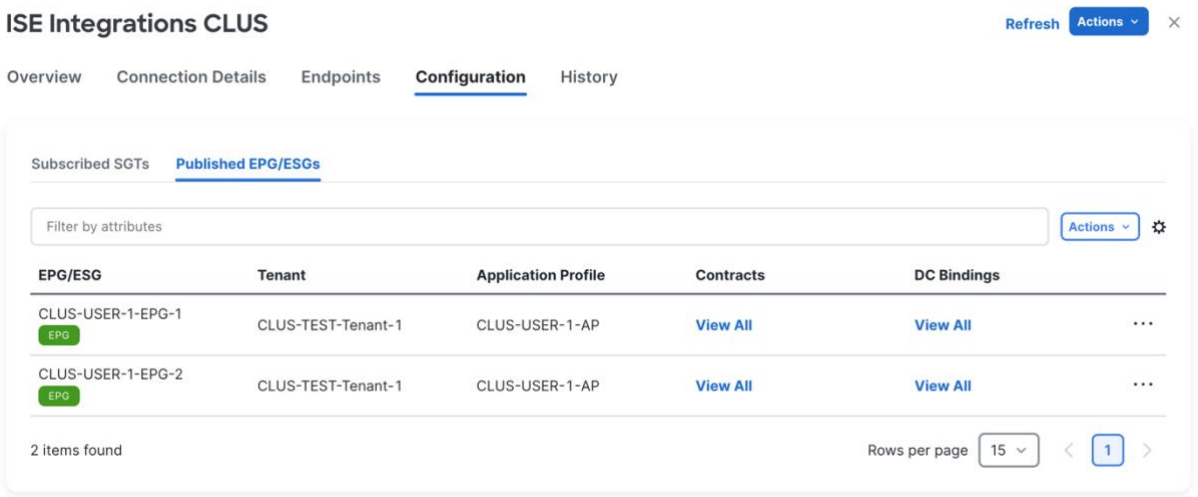
Select the correct Connection Instance



When you check the endpoints Tab, you see all the learnt endpoints from ACI side

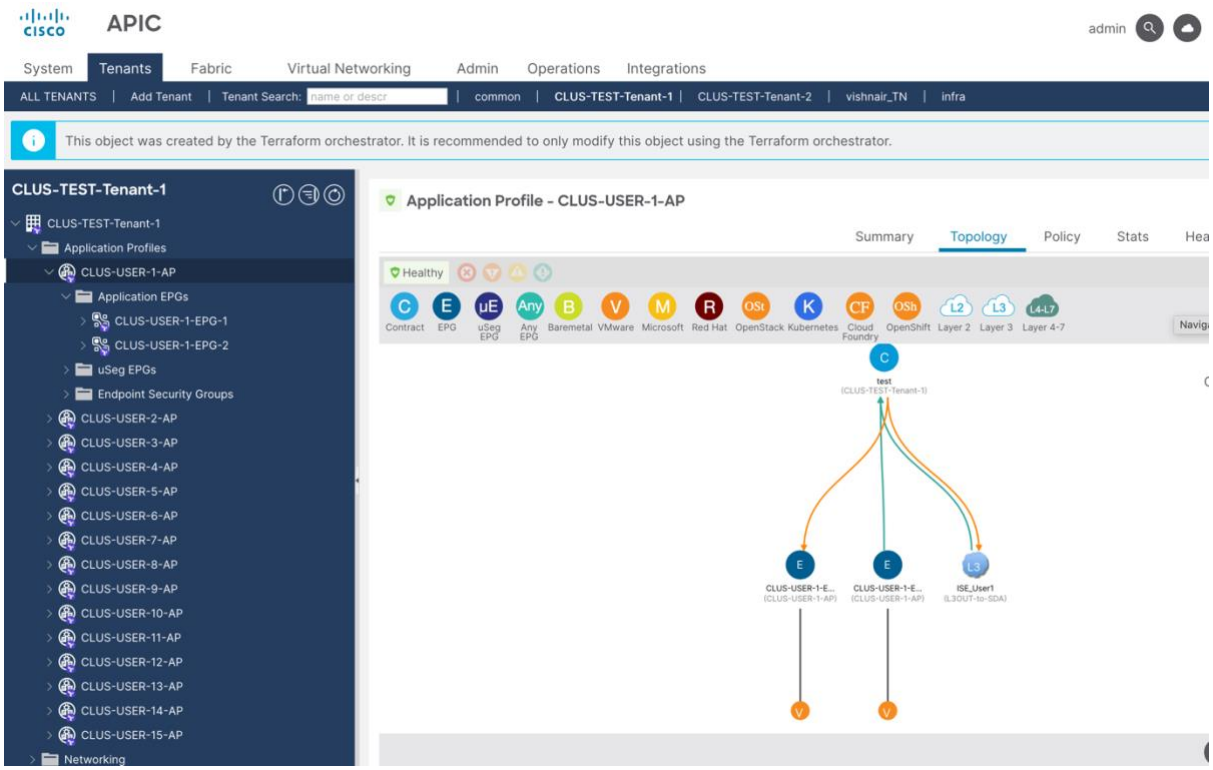


Additionally when we move to configuration tab, we see that we are Subscribed to ISE_User1 and Publish EPG 1 and 2 for out user



You can then go to verify how the contracts have been set up

Head over to Tenants and your tenant and AP and click on topology:



You can see the communication established to and from the external EPG via the same contract we created.

This shows how we can leverage and set up common policy either via an APIC or ISE/SGT enforcement.

Task 5 (Theoretical/Discussion): Models- how and where to control policies

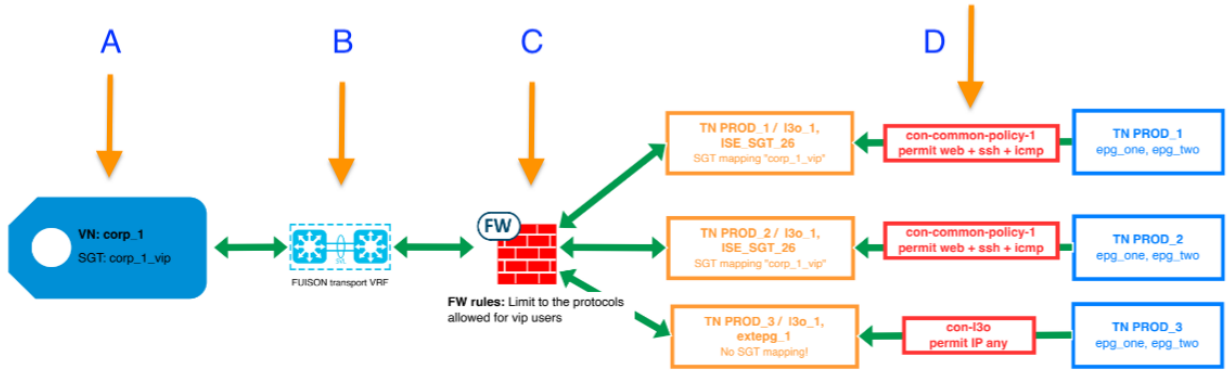
Overview

- Control points to enforce the policies
- Considerations where and how to enforce policies
- Start point, what to enforce where in the path.

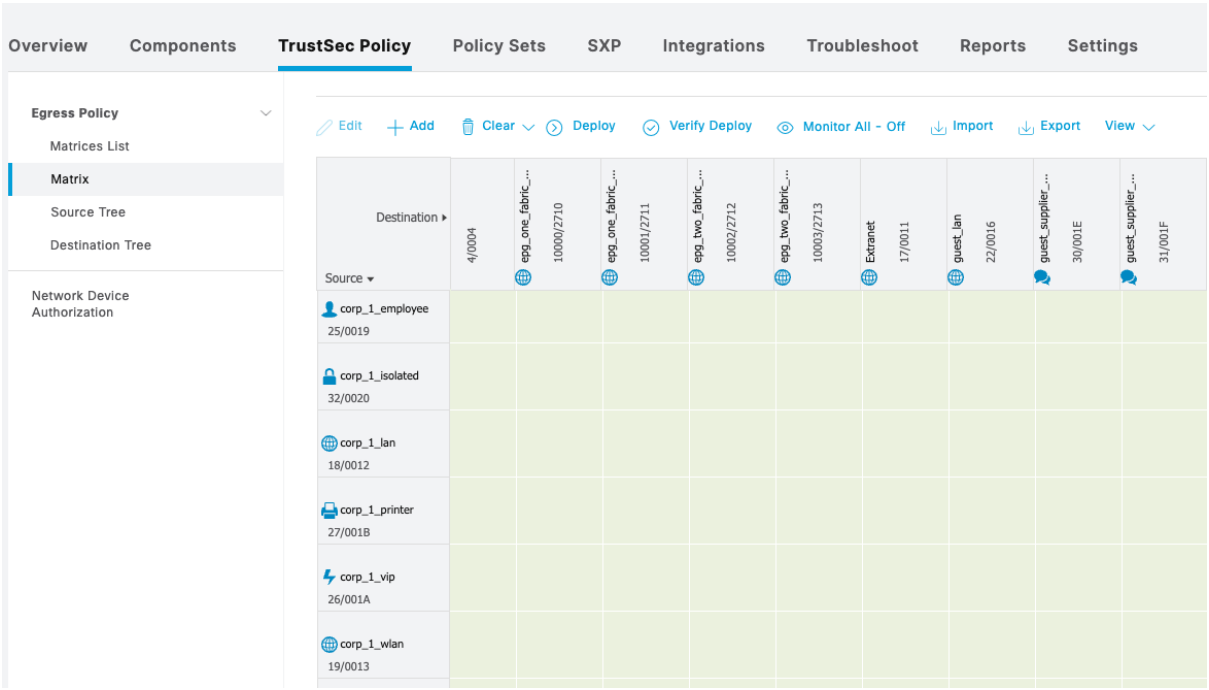
No lab, will be purely a discussion/theoretical section in the below fictional scenario.

Control Points on the Path

As shown in diagram below, there are 4 points where policy could be enforced via ISE exchange, information of SDA SGTs and ACI EPGs.



(A) ISE policies Via the ISE and TrustSec Policy Egress Policy Matrix, the SGTs as well the EPGs will be visible. Via the matrix in ISE below, policies could be added from simple "deny IP" or "allow IP" towards permit or deny protocols similar to ACL



(B) WAN Traffic Engineering In the lab we are using a Fusion router, which represents the WAN. The next step for common policy, is that the SGT and EPG information is distributed to SDWAN. What could be done is, that inside of SDWAN Traffic Engineering is used to provide certain SGT to EPG traffic more bandwidth or preferred path.

(C) Firewall Rules: In current setup there is no firewall in the path from SDA fabrics towards ACI. Today the firewall can join pxGrid and receive the SGT as well the EPG information. The result would be, that firewall rules can be applied based on SGT and EPG information used in the other domains.

Because in SDA it is quite common that one IP pool is shared by multiple SGTs, to define a firewall rule set based on IPs is very hard. To be precise either host IPs must be added dynamically to ISE, or IP pool per SGT must be added.

In ACI it is not common to use one Bridge Domain for different EPGs, but the usage of ESGs is increasing. With the use of ESGs, similar situation will occur, one network / one Bridge Domain could be used for many ESGs.

(D) ACI contracts attached to SGT external EP: In the current lab setup, this is the only method applied. The ISE controls which SGTs in SDA will be transported to which ACI tenant. For each SGT transported, one external EPG will be setup by ISE. The contracts in ACI, in the lab setup for example "con-common-policy-1", will be defined in AAC and rolled out towards ACI. As well in AAC the association which EPGs will be provider for the contract, is done.

But the control which ACI external EPG reprents a SGT, will consume which contract, is done via ISE!

Results are:

- The granularity of control, like only "permit IP" to certain protocols, is done in AAC / ACI.
- Which path is allowed or simply blackholed, like in tenant PROD_2 for "corp_1_isolated" is done ISE.

Which Control Points to use?

Control at every point to maximum granularity is probably no good idea. One suggestion, even maybe easy start point:

- Permit IP / deny IP in SDA from SGTs towards outside “ACI SGTs”
- Permit IP / deny IP in ACI from EPGs to “external EPG SGTs” via contracts
- Granular filters for protocols on the firewalls

Matrix for pro and cons for different models

- Suggestion for models:
    - (1) Simple model above
    - (2) All control on ISE via policies, no control in ACI, some rules on Firewall
    - (3) All control in ACI via contract, no control on ISE, some rules on Firewall
    - (4) No rules on ISE and ACI, all rules on Firewall

For WAN the policy enforcement is not to allow or drop certian traffic. For WAN the enforcment would be to provide certain traffic e.g. higher or lesser bandwidth.

Enforcement Point	Pro	Con
TrustSec Policy SDA Fabric	• Less Unknown Tagged traffic inside the Fabric • Dropping traffic as early as possible for traffic originated in Campus	• Stateless • Needs SXP to each Bordernode -> Scaling Issues

Enforcement Point	Pro	Con
		• Needs to learn every IP-SGT binding of every Fabric -> Scaling Issues
WAN Traffic Engineering	Traffic Engineering not possible on other enforcement points	If no Traffic Engineering based on policies is required, don't use it
Firewall	• Stateful • Uses pxGrid as transportation protocol, no scalling concerns of SXP • SGT based Ruleset instead of IP Based Ruleset	• Throughput -> Scaling Issues
ACI	• Datacenter Admin keeps Control of Datacenter • Uses pxGrid as transportation protocol, no SXP is required to Border Nodes • Dropping traffic as early as possible for data center originated traffic	• Stateless • Needs to learn every IP-SGT binding of every Fabric -> Scaling Issues
Combination of enforcement points above	• A Mix of Enforcement Points can fix the Issues of Others like scale and/or granularity of policies • Possible to use enforcement points with only "drop/permit" and one other with more granular policies	• Not a clear vision where traffic got dropped -> Troubleshooting Issues

Appendix: Use Cases on Common Policy

1. The obvious and underestimated, no enforcement via SGTs / EPGs
    - Considerations:
        - There is no difference between hosts from SDA or endpoints from ACI.
        - Could be useful for infrastructure services required by all users in all SGTs and all VRFs.
        - Could also help for scale considerations, no need to learn the ACI endpoints or SDA host mappings.
        - Might be the most difficult one, because it is hard to generalize.
2. Keep the doors open for the not-categorized
    - Considerations:
        - Special rules might apply for some SGTs.
        - Other hosts of SGTs also need to access the resources, but can be treated as "default"
        - There might also be hosts, which are not tagged anyhow with SGTs
3. Different eligibility between identified hosts

    - Consideration:
        - The identified hosts are grouped into different SGTs and get different eligibility.
4. Lower eligibility for "bad-hosts" SGTs
    - Considerations:
        - Due to security reasons, some groups might get lower eligibility than groups of identified hosts via SGTs and even "default groups".
        - This could be the use case for malware infected host which are moved into quarantine or isolation, versus more rights for known hosts.
5. Block "bad-hosts" SGTs from resources
    - Considerations:
        - The "bad-hosts" shouldn't have access to defined resources at all.
        - The "bad-hosts" will be grouped via SGTs
        - Goal would be to black-hole the access for "bad-hosts".
6. Some SGTs require same eligibility
    - Considerations:
        - The eligibility for certain resources are the same for different group of hosts identified by SGTS.
        - To minimize the maintenance effort and to keep the solution scaleable, the same rules should be used.
7. Not-categorized hosts shouldn't have access
    - Considerations:
        - End devices which are attached to a SGT, are allowed to access resources.
        - The eligibility of identified hosts, might be different
        - Other end devices which are not-categorized and learned via SGT, shouldn't have any access to resources.