



LTRSEC-2000

ISE Deployments in the Cloud - Automate ISE Deployments in AWS and Integrate Them with Entra ID

Jesse Dubois, TAC Security Technical Leader

Patrick Lloyd, Senior Solutions Architect, CX Security Services

Learning Objectives or Table of Contents

Upon completion of this lab, you will be able to:

- Provision an Ubuntu Linux machine with Terraform and Ansible to automate configurations.
- Automatically provision AWS VPC's, Subnets, Routes, and Transit Gateway Attachment via Terraform.
- Provision ISE using a CloudFormation template.
- Provision ISE programmatically into the AWS cloud with your own credentials.
- Configure network device groups, network access devices, users and roles via Ansible configuration scripts.
- Integrate a ROPC external database for authentication to Microsoft Entra ID.

Scenario

In this lab activity, you will learn how to utilize an Ubuntu Linux machine to deploy ISE into an AWS environment. Two separate orchestration models are used, Terraform, which is more suited for deploying workloads and maintaining their state, and Ansible, which is better suited to configure and modify workloads via API. Both Terraform and Ansible scripts, found on github.com for later consumption, will be used to provision the lab, with Terraform used to deploy VPC, subnet, routing table, and internet gateway into an AWS tenant already created. Ansible will then deploy ISE configurations, including users, groups, and network access devices. You will then provision the Entra ID connector, which requires manual intervention to join to Entra ID.

Document Conventions

As part of this document the following highlighting will be used

Note: A time saving note which will indicate aspects of the lab to look out for, or could otherwise cause need for rework.

Warning: Common mistake areas students have made in the past.

A configuration block.

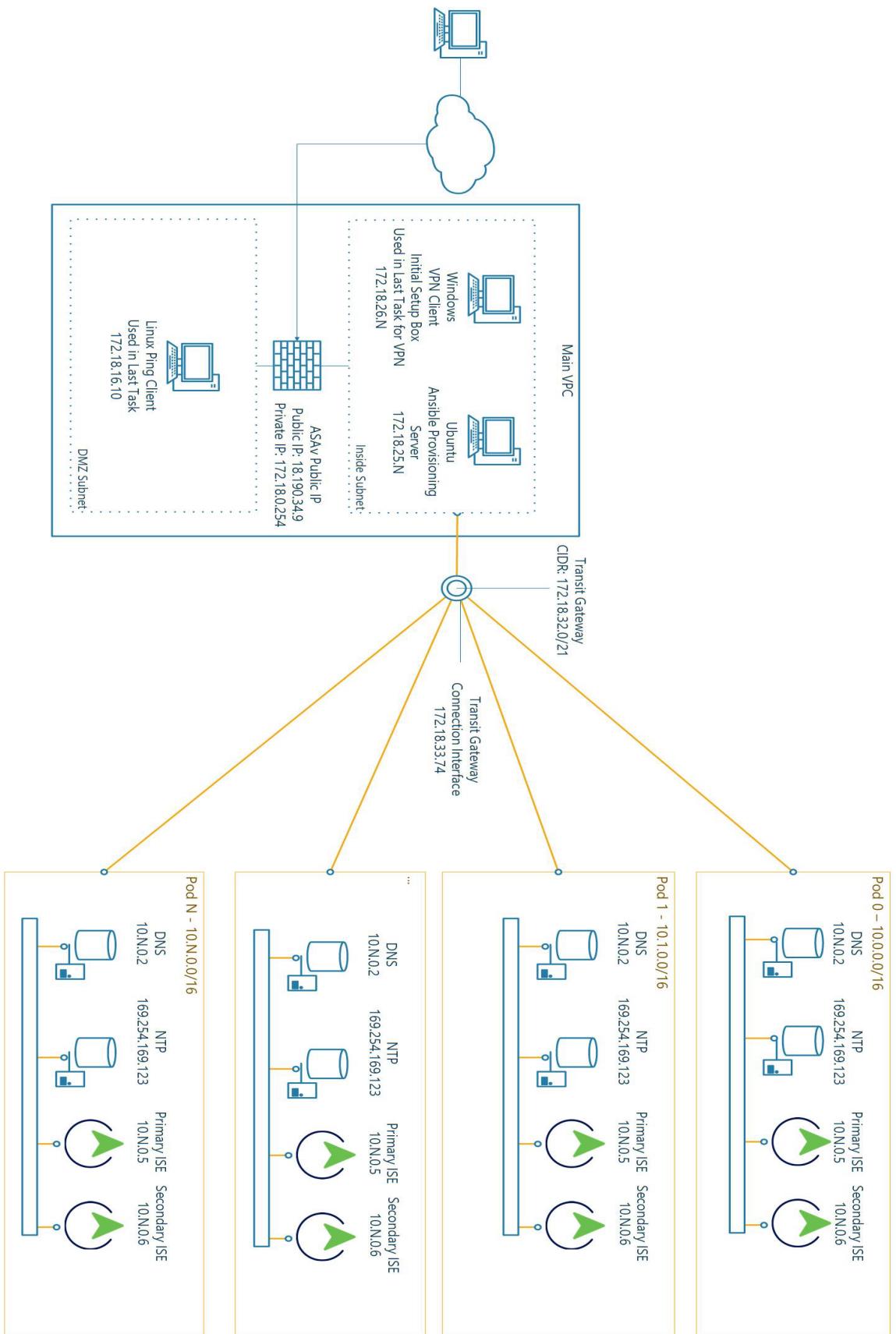
Throughout the lab, there are various places where passwords must be changed. Your proctors should provide you an initial password as well as an updated password before starting.



Recommended Equipment

1. A laptop with VMWare workstation installed.
2. A virtual image you are familiar with.
3. AnyConnect on the virtual image. If you do not have AnyConnect, reach out to your lab proctor.
4. A cell phone with Google Authenticator Installed or your MFA app of choice.
5. Internet Connectivity.

Network Diagram



Task 1: Initial Connectivity

Each pod is accessible via VPN with a provisioning machine on the required subnet. From this provisioning machine, you'll execute tasks allowing you to deploy AWS and ISE components and configurations. Access your pod based on the pod number assigned to you by the instructors:

Table 1-1

Pod Number	Ubuntu Host	AWS Username	RAVPN Username
1	172.18.25.1	pod1-awsadmin	pod1-ravpn
2	172.18.25.2	pod2-awsadmin	pod2-ravpn
3	172.18.25.3	pod3-awsadmin	pod3-ravpn
4	172.18.25.4	pod4-awsadmin	pod4-ravpn
5	172.18.25.5	pod5-awsadmin	pod5-ravpn
6	172.18.25.6	pod6-awsadmin	pod6-ravpn
7	172.18.25.7	pod7-awsadmin	pod7-ravpn
8	172.18.25.8	pod8-awsadmin	pod8-ravpn
9	172.18.25.9	pod9-awsadmin	pod9-ravpn
10	172.18.25.10	pod10-awsadmin	pod10-ravpn
11	172.18.25.11	pod11-awsadmin	pod11-ravpn
12	172.18.25.12	pod12-awsadmin	pod12-ravpn
13	172.18.25.13	pod13-awsadmin	pod13-ravpn
14	172.18.25.14	pod14-awsadmin	pod14-ravpn
15	172.18.25.15	pod15-awsadmin	pod15-ravpn
16	172.18.25.16	pod16-awsadmin	pod16-ravpn
17	172.18.25.17	pod17-awsadmin	pod17-ravpn
18	172.18.25.18	pod18-awsadmin	pod18-ravpn
19	172.18.25.19	pod19-awsadmin	pod19-ravpn
20	172.18.25.20	pod20-awsadmin	pod20-ravpn
21	172.18.25.21	pod21-awsadmin	pod21-ravpn
22	172.18.25.22	pod22-awsadmin	pod22-ravpn
23	172.18.25.23	pod23-awsadmin	pod23-ravpn
24	172.18.25.24	pod24-awsadmin	pod24-ravpn
25	172.18.25.25	pod25-awsadmin	pod25-ravpn
26	172.18.25.26	pod26-awsadmin	pod26-ravpn
27	172.18.25.27	pod27-awsadmin	pod27-ravpn
28	172.18.25.28	pod28-awsadmin	pod28-ravpn
29	172.18.25.29	pod29-awsadmin	pod29-ravpn
30	172.18.25.30	pod30-awsadmin	pod30-ravpn

Step 1: Connect to the Lab VPN

Using AnyConnect VPN, connect to the ASA V to gain access to the inside network. Connect to IP **3.131.72.32**. Choose the group “RA_VPN” and login with the username (**podX-ravpn**) assigned to your pod in table 1-1 above.

At this time, do not log into your “PODXX_VPN” tunnel group.

The password for the account is provided by your proctor. Ignore any certificate warnings presented on connection. There is no public IP access to any of the machines in the pods. When trying to connect to any of the lab machines, ensure VPN is active.

If you do not have the AnyConnect client, browse to <https://3.131.72.32/> and authenticate with the information above. The AnyConnect client will be provisioned during the initial connection process.

Step 2: Connect to AWS

Open a web browser on the lab laptop and navigate to <https://zer0k.signin.aws.amazon.com/console>. Use Account ID zer0k if not already populated. Login with the username assigned to your pod in table 1-1 above (**podX-awsadmin**). The password for the account is **provided by your proctor**.



Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

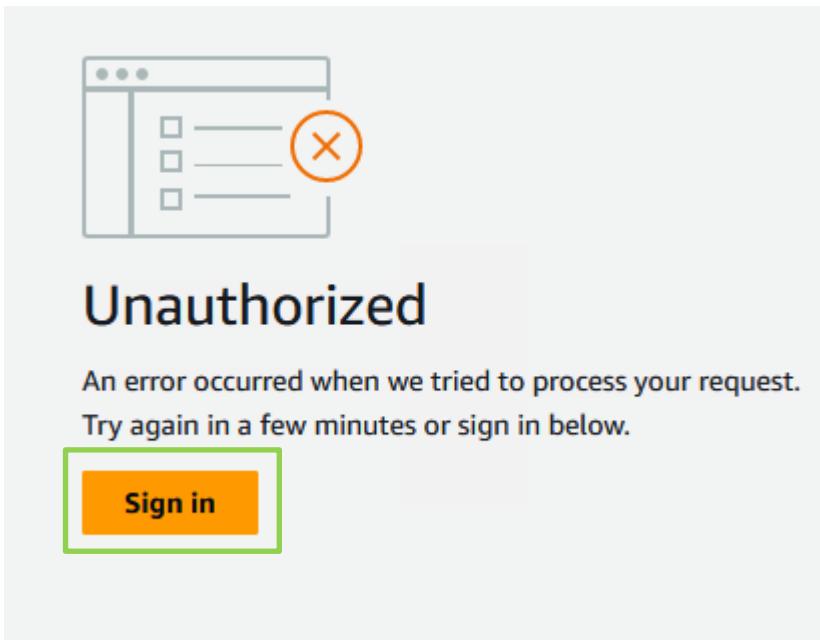
Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

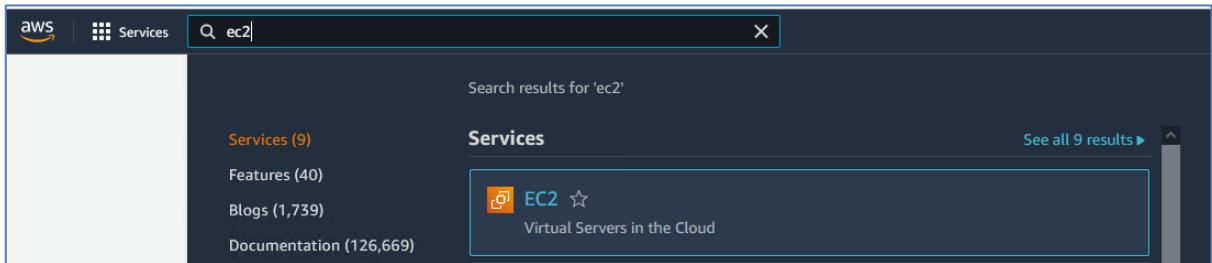
If you are presented with an “Unauthorized” screen, click the Login button and you’ll be redirected to the AWS home page.



Step 3: EC2

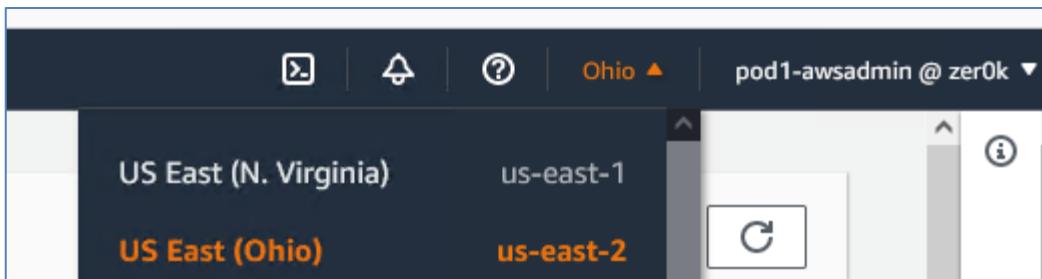
Navigate to the EC2 area of AWS to access available virtual machines and

virtual machine settings. In the search box at the top of the screen, type “EC2” and use the EC2 service link. Please do not access any virtual resources not associated with your pod number. Consult a lab proctor for assistance if you are unsure if a resource is one assigned to you or is one that you created.



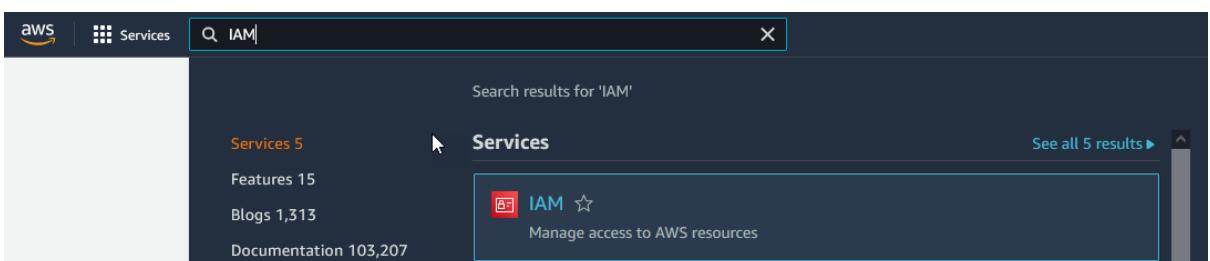
Step 4: Change AWS Regions

The Ansible lab is built within the us-east-2 Ohio region, which is required for connectivity to work between machines. In the EC2 dashboard, ensure that the region, located on the top right of the page, is us-east-2, or “Ohio”.



Step 5: Generate your AWS Access and Secret Key

From the top search bar in AWS, type IAM to navigate to Identity and Access Management. Click IAM.



From the left bar, click “Users” and select your pod username.

Note: You may need to navigate to the second page of credentials or use the search box on this page depending on display settings

A screenshot of the AWS IAM Users page. On the left, a sidebar shows 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings), 'Access reports' (Access analyzer), and 'AWS CloudTrail' (AWS CloudTrail status, Log delivery to CloudWatch Logs). The main pane displays a table of users. The table has columns for 'User name' (with checkboxes) and 'Groups' (dropdown menus). The users listed are 'automation' (None), 'pod0-awsadmin' (LabAdmin), and 'pod1-awsadmin' (LabAdmin). A note at the top says 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar at the top of the main pane says 'Find users by username or access key'.

Warning: Common mistake area!

1. Under the Summary subsection and under Access Key 1, select “Create Access Key”.

Summary

ARN

arn:aws:iam::423620453332:user/pod1-awsadmin

Created

January 03, 2025, 09:33 (UTC-05:00)

Console access

⚠ Enabled without MFA

Last console sign-in

⌚ Today

Access key 1

[Create access key](#)

You will use this access key for your environmental variables used in future steps so that Terraform is able to access your AWS account. When presented with the “Access key best practices & alternatives” select “Local Code” for your use case. Check the box at the bottom of the page for “I understand the above recommendation and want to proceed to create an access key”.

The screenshot shows the AWS IAM Access Key creation process. On the left, a sidebar lists steps: Step 1 (Access key best practices & alternatives), Step 2 (optional: Set description tag), and Step 3 (Retrieve access keys). The main content area is titled "Access key best practices & alternatives". It advises against long-term credentials like access keys to improve security. A list of use cases is provided, with "Local code" selected:

- Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
- Other
Your use case is not listed here.

A warning icon indicates an alternative recommendation: "Alternative recommended: Use an Integrated Development Environment (IDE) which supports the AWS Toolkit enabling authentication through IAM Identity Center. [Learn more](#)". At the bottom, a checked checkbox says "I understand the above recommendation and want to proceed to create an access key."

2. Click Next.
3. For the tag description enter “Pod X local code access key” where pod X is your pod number.
4. Click “Create Access key”
5. Click “Show” to show this secret key.
6. Copy the Access Key and Secret Key to a notepad.

Ensure you copy the secret key, as it will not be available again.

Label your access key and secret access key accordingly, ensuring you paste the right string into the proper variables later in the lab.

In addition, you can download a CSV file with the access and security keys contained within it.

Task 2: Set Up Two Factor Authentication

Each pod used within this lab has administrative access to the test deployment. Therefore, it is best practice that two factor authentication be set up for the administrative user. For this task you'll need an authenticator app such as Google Authenticator

In the IAM -> Users -> <your username> -> Security Credentials area, navigate to **multi-factor authentication (MFA)** section.

Click “Assign MFA Device”.



Enter a name for the device, we recommend **podX-awslab** and select the preferred method of “Authenticator App”.

Select MFA device Info

MFA device name

Device name
Enter a meaningful name to identify this device.
 MyCellPhone

Maximum 128 characters. Use alphanumeric and '+ = , . @ - _' characters.

MFA device

Select an MFA device to use, in addition to your username and password, whenever you need to authenticate.

 **Authenticator app**
Authenticate using a code generated by an app installed on your mobile device or computer.

Using Google Authenticator (or your app of choice), scan the QR code and confirm the next two MFA codes.

3

Type two consecutive MFA codes below

Enter a code from your virtual app below

465918

Wait 30 seconds, and enter a second code entry.

603257

[Cancel](#) [Previous](#) [Add MFA](#)

Logout of the console and log in with the user to which you assigned the MFA device. Enter the next two factor authentication code when prompted.

Task 3: Deploy Ubuntu Provisioning Machine / Building Blocks

There are many ways to create an instance in AWS without creating each object ahead of time, but it is helpful to understand each object when CloudFormation templates and scripts are used later in the lab. It is also helpful to know where each of these objects resides when troubleshooting connectivity problems.

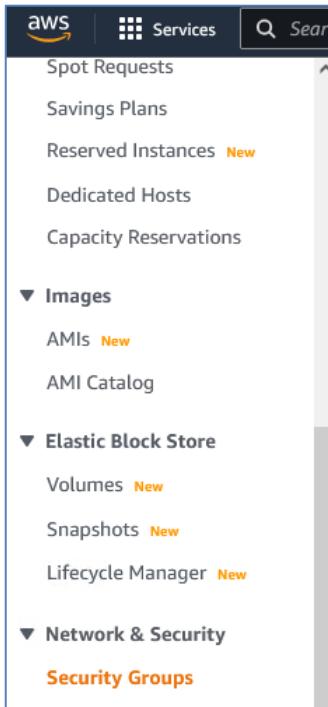
Step 1: Create Security Group

Security Groups are AWS ACLs that control network traffic, by default all outbound network traffic from a security group is permitted as well as the return traffic from that outbound connection (stateful).

1. Navigate back to EC2 using the search bar at the top of the page.

The screenshot shows the AWS search interface. A search bar at the top contains the text 'ec2'. Below the search bar, a sidebar lists categories: Services (12), Features (51), Resources (New), and Blogs (1,889). The main content area displays search results for 'ec2', with a heading 'Search results for 'ec2'' and a sub-heading 'Services'. A card for 'EC2' is shown, featuring the text 'Virtual Servers in the Cloud' and a star icon. To the right of the card, there is a link 'See all 12 results ▶'.

2. In the left navigation panel within the EC2 dashboard, Browse to Network & Security -> Security Groups.



3. Select “Create Security Group” from the upper right-hand side of the page.
4. Name the security group **podX-management-sg** where X is your assigned pod number, add a required description, we recommend “**Management security group for pod X**” where X is your pod number. See the screenshot in step 6 for verification.
5. Make sure the VPC is set to vpc-0a61b7f1d92102ad6 (zer0k-main-vpc) since this is where the provisioning of additional VPCs will be done from.

Note: You should be able to delete the current VPC name and begin typing “zer0k” to show available VPC’s.

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
 Name cannot be edited after creation.

Description Info

VPC Info

6. Add an inbound rule allowing ICMPv4 traffic from 172.16.0.0/12.

The screenshot shows two sections of the AWS CloudFormation interface for defining security rules:

- Inbound rules**: This section has four filter tabs: Type (Custom ICMP - IPv4), Protocol (All), Port range (All), and Source (Custom, 172.16.0.0/12). An "Add rule" button is located below the filters.
- Outbound rules**: This section has four filter tabs: Type (All traffic), Protocol (All), Port range (All), and Destination (Custom, 0.0.0.0/0). An "Add rule" button is located below the filters.

Note: Ensure you added this rule under the inbound rules header.

Note: Cisco monitors all activity from users within the AWS cloud to ensure that limits are placed on reachability. Please ensure you add 172.16.0.0/12 and NOT 0.0.0.0/0 inbound. If this mistake is made, the rule will be removed and proctors will receive an automated notification that a reachability rules has been triggered.

7. Do not add any other rules at this time.
8. Under Tags, add a new tag with a key of “Pod” and a value of your pod number, typically in the range of 1 to 60.
9. Under Tags, add a new tag. Enter “Project” under Key and “LTRSEC-2000” under value.

Tip – AWS does not name anything by default, always add a “Name” tag to easily find the object later.

This screenshot shows the "Tags - optional" section of the AWS CloudFormation interface. It includes a description of what tags are, a table for adding tags, and a note about the maximum number of tags.

Key	Value - optional
Pod	0
Project	LTRSEC-2000
Name	pod0-management-sg

Add new tag
You can add up to 47 more tags

10. Create the security group.

Step 2: Create a Key Pair

Key pairs are used to connect to instances after creation and can be used

for other tasks in AWS such as decrypting Windows passwords.

Note: For the following step, the private key created is NEVER available for download again, if it is lost a new key pair must be created.

1. In the EC2 service, navigate in the left bar to Network & Security -> Key Pairs.



2. Select “Create Key Pair”.
3. Name your keypair podX-keypair where X is your pod number. Leave the default value of RSA and ensure .pem (For use with OpenSSH) is used. In this case the AWS UI populates the Name tag as an option, so no tags are required.

The screenshot shows the "Create key pair" dialog box. At the top, it says "Create key pair Info".
Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.
Name
Name: pod0-keypair
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type | Info
 RSA ED25519
Private key file format
 .pem
For use with OpenSSH
 .ppk
For use with PuTTY
Tags - optional
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

At the bottom right are two buttons: "Cancel" and "Create key pair".

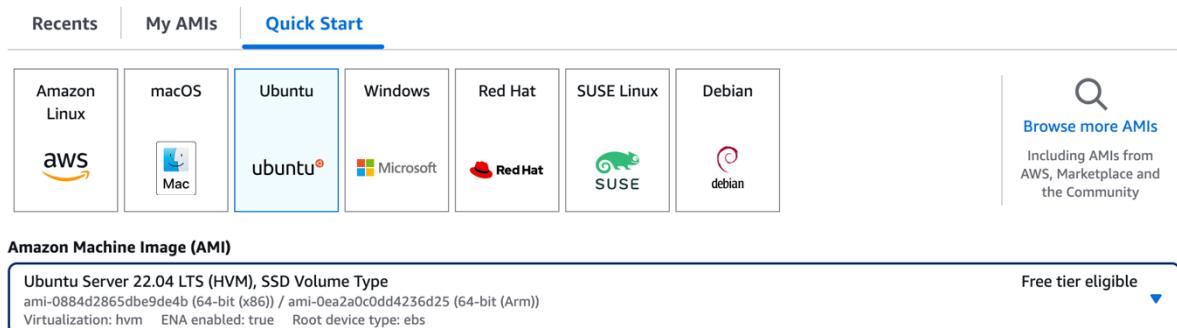
4. Select “Create key pair” and ensure the .pem file is saved to your lab laptop. Downloading the file as .pem will allow it to be used from a Linux or Mac host and can be converted later for use in Putty. This should happen automatically.

This private key is never available for download again, if it is lost a new key pair must be created.

Step 3: Deploy Ubuntu Provisioning Machine

All of the building blocks are in place from a networking standpoint to create an Ubuntu Linux instance.

1. From the Left Navigation bar, navigate to Instances -> Instances.
2. Select “Launch instances” from the upper right corner.
3. Under Name enter “podX-Ubuntu” where X is the number of your assigned pod.
4. Under Application and OS Images, choose Ubuntu under quick start and **Ubuntu Server 22.04** as the AMI. Leave the rest default. There should be a Free Tier Eligible Ubuntu instance chosen.



Note: Some of the settings within this lab are reliant upon Ubuntu 22.04. If a mistake is made to select 24.04 here, you'll need to redeploy from this step onward.

5. If you click another image by mistake in this process, you may be presented with a dialog box warning that some of the current settings will be changed or removed. Confirm these changes.

Some of your current settings will be changed or removed if you proceed

X

Changing your AMI will result in some of your current settings being overridden.

Changes

- Your security group rules will be overridden.
- 1 custom volume(s) will be deleted.

► Details

[Cancel](#)

[Confirm changes](#)

6. Under Instance Type leave it as default t2.micro.
7. Under Key pair, choose the key pair you created under Step 2 of this task. This is the key pair you have downloaded to your local machine. It should have been named “podX-keypair” where X is your assigned pod number.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Pod31-keypair

[Create new key pair](#)

8. Under Network Settings click “Edit”.
9. Set the VPC to vpc-0a61b7f1d92102ad6 (zer0k-main-vpc).
10. Set the Subnet to subnet-0cf86b138b53ba3c9 (zer0k-inside-subnet).
11. Leave Auto Assign Public IP disabled as there is no direct internet access from this subnet.

▼ Network settings

VPC - *required* [Info](#)

vpc-0a61b7f1d92102ad6 (zer0k-main-vpc)
172.18.0.0/16



Subnet [Info](#)

subnet-0cf86b138b53ba3c9
VPC: vpc-0a61b7f1d92102ad6 Owner: 423620453332
Availability Zone: us-east-2a IP addresses available: 2030

zer0k-inside-subnet



[Create new subnet](#)



Auto-assign public IP [Info](#)

Disable

12. Choose Select existing security group and select the security group you created in Step 1 above. It should be podX-management-sg where X is your pod number.

Firewall (security groups) | Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups | Info
Select security groups

pod0-management-sg sg-0e5203d5e1ed7f2e9 X
VPC: vpc-0a61b7f1d92102ad6

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

13. Expand “Advanced network configuration”.

14. Under Primary IP, configure 172.18.25.X where X is your pod number.

Advanced network configuration

Network interface 1

Device index | Info 0 **Network interface | Info** New interface **Description | Info**

Subnet | Info subnet-0cf86b138b53ba3c9 **Security groups | Info** Select security groups **Primary IP | Info** 172.18.25.12

IP addresses available: 2036 Show all selected (1)

Secondary IP | Info Select **IPv6 IPs | Info** Select **IPv4 Prefixes | Info** Select

The selected subnet does not support IPv6 IPs. The selected instance type does not support IPv4 prefixes.

IPv6 Prefixes | Info Select **Assign Primary IPv6 IP | Info** Select **Delete on termination | Info** Select

The selected instance type does not support IPv6 prefixes. A primary IPv6 address is only compatible with subnets that support IPv6.

Interface type | Info Select **Network card index | Info** Select **ENAv Express | Info** Select

The selected instance type does not support multiple network cards. The selected instance type does not support ENA Express.

ENAv Express UDP | Info Select **Idle connection tracking timeout | Info** Enable

The selected instance type does not support ENA Express. Idle connection tracking timeout is only supported on Nitro instances.

Add network interface

15. Leave the options under Configure storage and Advanced details as defaults.

16. Select “Launch instance” on the bottom right of the page.

17. If the creation of the instance is successful, click on “View all instances” to go back to the instance list. Otherwise, notify a proctor.

18. In the EC2 service, browse to Instances -> Instances.

19. Verify that the status check for your podX-Ubuntu machine is showing 2/2 checks passed. If it isn’t, refresh the page until the status is 2/2

checks passed or you see a red status. It will have to be troubleshoot if red, move on once it is 2/2 checks passed.



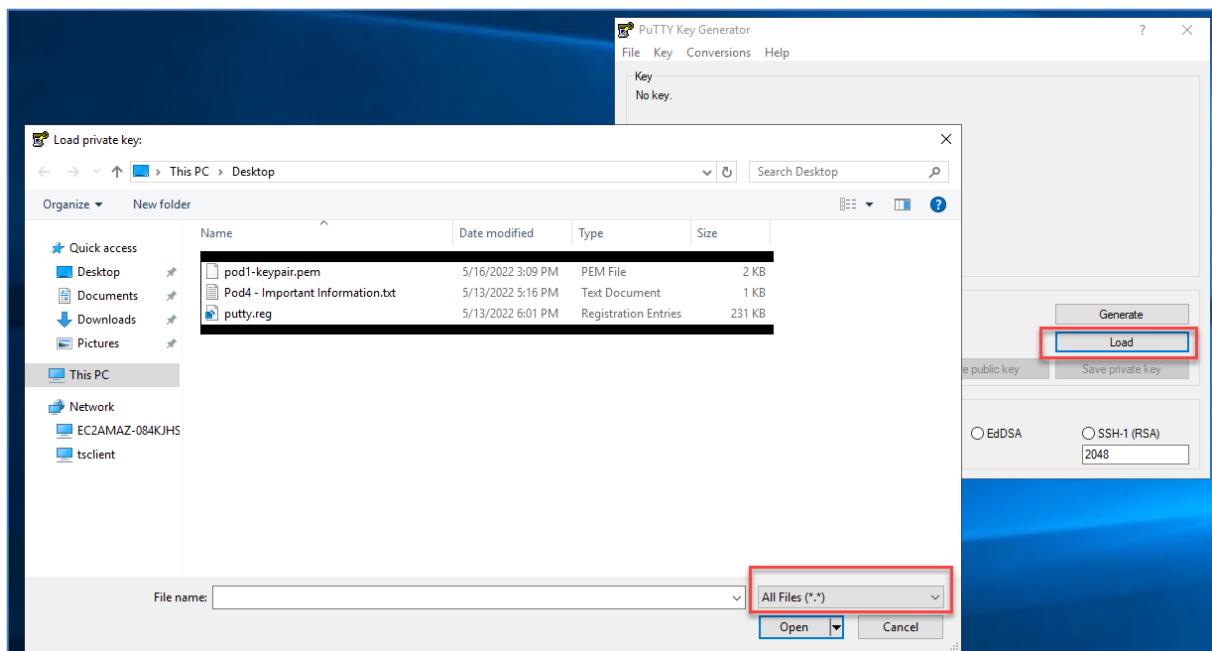
Step 4: Convert PEM to PPK format for dual use

The PEM file downloaded for use as a keypair within AWS will both serve as the keypair within AWS, as well as the connectivity file used for Putty. To use this file within Putty, it must be in PPK format. Locate the PEM file on the lab laptop, it is recommended you move it to the desktop for easy retrieval. By default, Firefox will put it in the Downloads folder.

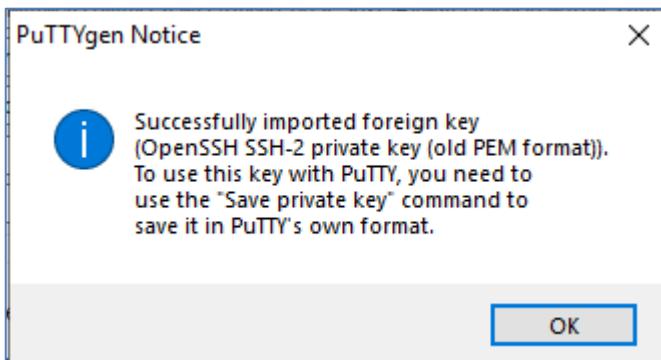
Putty on Windows:

PuTTY is installed on the lab laptop and is recommended to be used, however it can be downloaded if doing this on an alternative machine. If you don't already have PutTY and PuTTYgen, download them from here: <https://www.chiark.greenend.org.uk/~sgtatham/putty/>

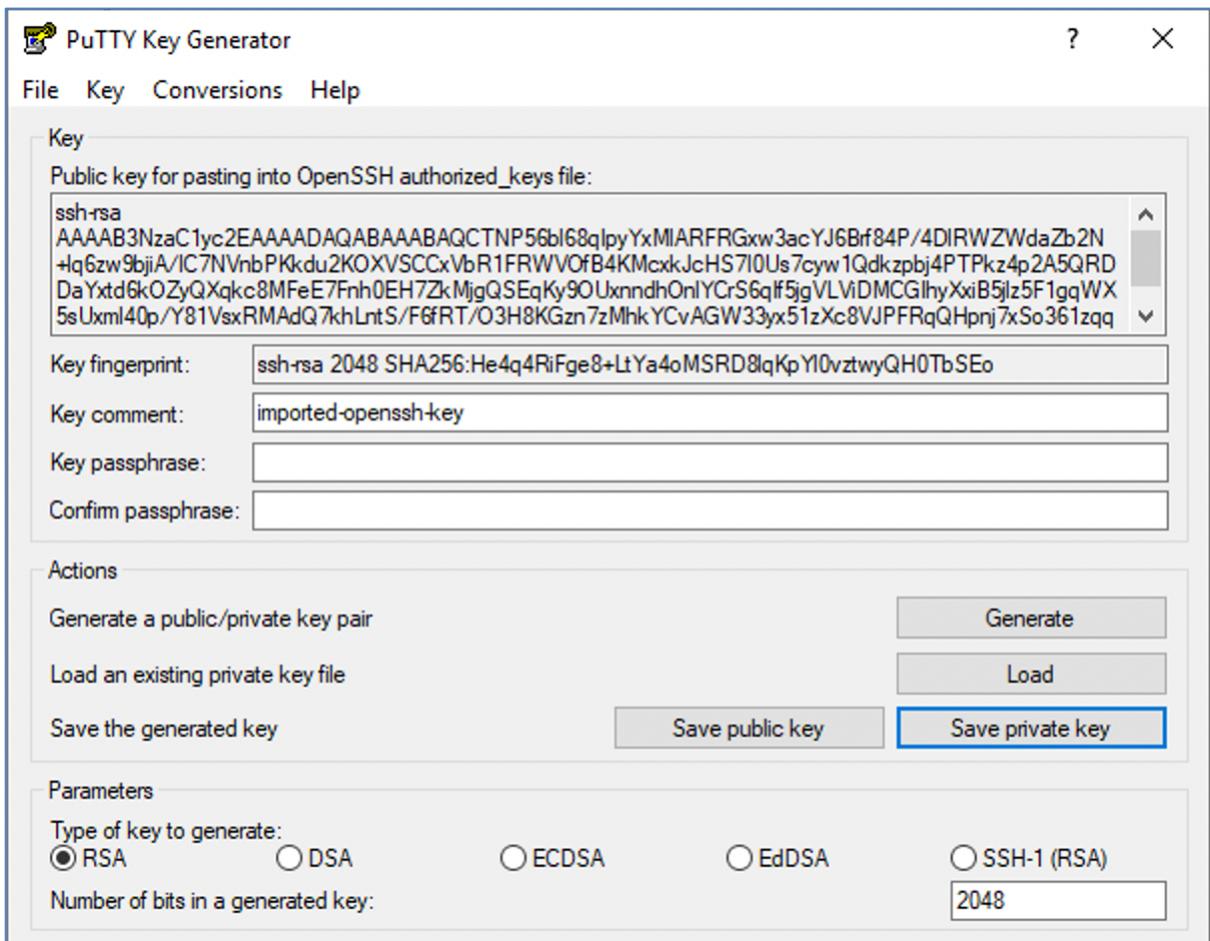
1. In the start menu of the lab laptop, navigate to PuTTY -> PuTTYgen and open PuTTYgen.
2. Click “Load” and change the file type to “All Files (*.*)” to select the PEM keypair you created in the last task, it should have been named podX-keypair.pem where X is your assigned pod number.



3. The selection of the keypair should result in a successful imported foreign key dialog.



- Leave RSA chosen as the default key type under parameters. Save the **private key** without passphrase to the same location as the PEM file. Name this keypair “podX-keypair.ppk” where X is your assigned pod number.



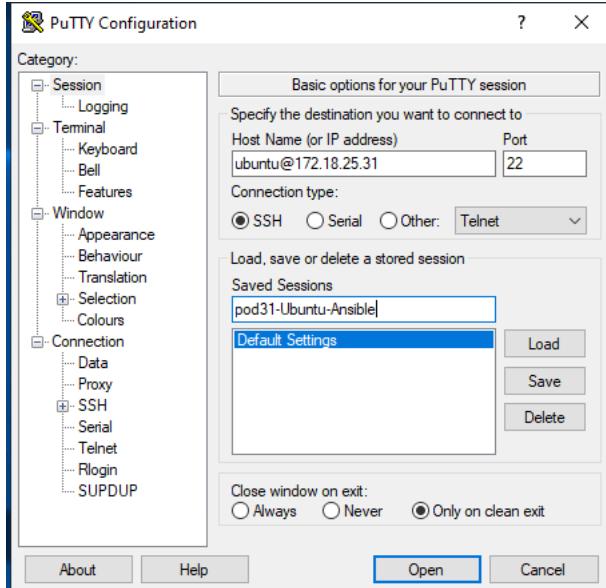
- Close PuTTYgen.
- This converted key will be used later in the next step.

Step 5: Configure Putty for SSH on Local Machine

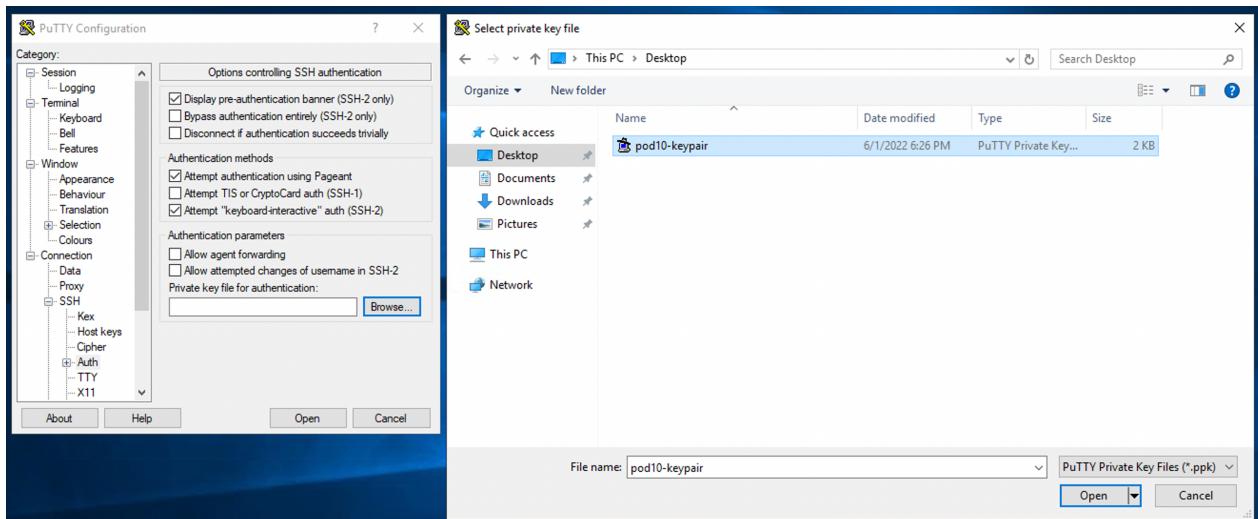
In this section the SSH client is just being setup,

Note: SSH will not work until Step 6 is completed.

1. On the lab laptop, navigate back to the desktop and open Putty from the shortcut.
2. The default screen is Session, set the Hostname to “ubuntu@172.18.25.X” and Saved Sessions as “podX-Ubuntu” where X is your assigned pod number. Click the save button.



3. Navigate to Connection -> SSH -> Auth -> Credentials. In the “Private key file for authentication” box at the bottom of the screen, select the private key created from the PEM file in step 3. It should have been named podX-keypair.ppk located on the desktop from the previous preparation step.



4. Optional: Navigate to Session -> Logging, select “All Session Output”, browse, and place the log file in the desktop. This is strictly for future troubleshooting if desired.
5. Do not change any additional settings.
6. Go back to Session and save again.
7. Attempt to SSH to your podX-Ubuntu machine from your Windows host using the previously set up profile. **It timed out!**

Optional: If using Linux/Mac Command Line

The permissions of the key file must be changed to 400 for the command line ssh client to use the key:

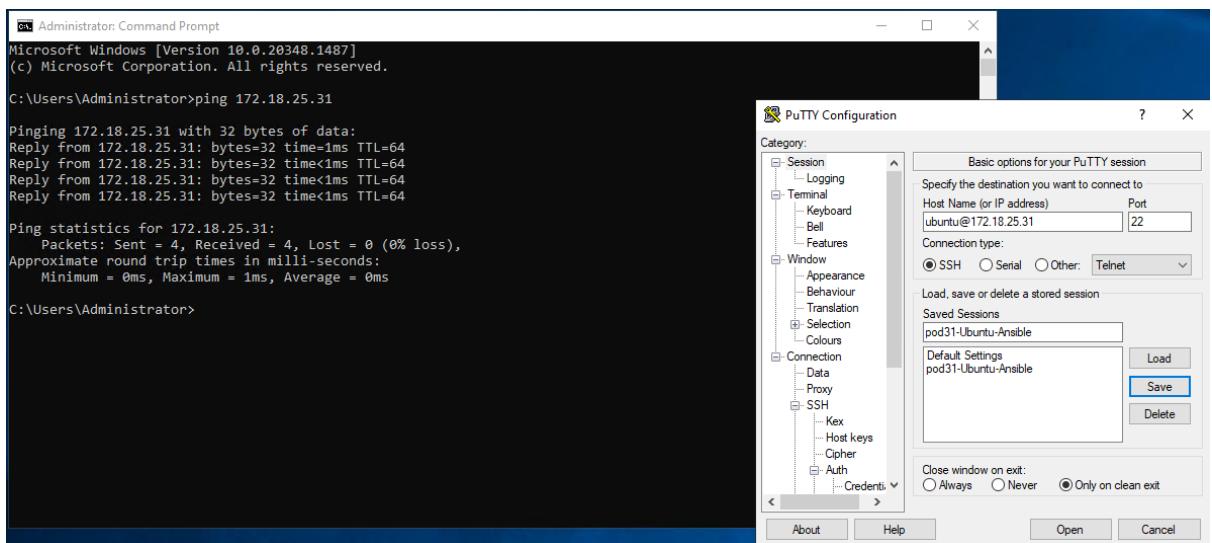
```
> chmod 400 <keyfile>
```

To use the private key file use ssh -i

```
> ssh -i <keyfile> <username>@<host>
```

Step 6: Verify Connectivity

1. From your lab laptop, ping your podX-Ubuntu server, remember the IP is 172.18.25.X. This should be successful. If it isn't, recheck your security group. The following steps will also help to troubleshoot.



2. Back in the AWS console, use the search bar to search for “VPC Reachability Analyzer”

Search results for 'vpc reachability'

Features

Services

Resources New

Documentation

Features

VPC Reachability Analyzer

VPC feature

Show more ▶

3. Click Create and analyze path.
4. Name it “SSHtoPodX-Ubuntu” where X is the pod number you were assigned.
5. Set the source type to “Instances” and locate the instance named “zer0k-lab-automation”

Path Source

Source type

Instances

Source | [Info](#)

Source



i-03d476072bcb82919

zer0k-edge-ASAv

i-0af9bd77074c86869

zer0k-private-linux

i-0342c576b2d6f556d

zer0k-lab-automation

6. Set the destination type to “IP Address” and enter the IP of your Ubuntu machine which should be 172.18.25.X where X is your pod number.
7. Expand “Additional packet header configurations at source – *optional*.”
8. Set the destination port to “22” for SSH and the protocol as TCP

Path destination

Destination type

IP Address

Destination address

Enter IP address

172.18.25.1

▼ Additional packet header configurations at destination - *optional*

You can optionally specify the packet header details of your traffic to evaluate its network reachability. Reachability Analyzer determines whether traffic matching these packet headers would be permitted or dropped based on your network configurations such as Security Groups.

Source address

Enter IP address

192.0.2.1

Destination address

Enter IP address

192.0.2.1

Source port

Enter port or port range. Example 80 or 0-65535.

Enter number or range

Destination port

Enter port or port range. Example 80 or 0-65535.

22

Protocol

Protocol

Use the appropriate protocol

TCP

9. Select Create and analyze path in the bottom right.
10. You will see the Analyses as pending, refresh the page until you see success. The reachability status should be Not Reachable.

The screenshot shows the 'Analyses' tab selected. A single analysis entry is listed:

Analysis ID	Analysis run date	Reachability status	Intermediate comp...	State
nia-0013048bf55708970	January 30, 2025, 12:1...	☒ Not reachable	-	☑ Succeeded

11. Check under Explanations to see why AWS thinks your instance isn't reachable on port 22. It should be because of a missing security group rule.

Explanations

None of the ingress rules in the following security groups apply: sg-0865ec82ad05fbb4f. See [sg-0865ec82ad05fbb4f](#).

► Details

12. Click on the link to the security group. Click "Edit inbound rules" under the "Inbound rules" tab.
13. Add another inbound entry for SSH from 172.16.0.0/12, all hosts are behind the ASA firewall already so allowing from the private subnet is fine for this lab.

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID

Type Info

Protocol Info

Port range Info

Source Info

sgr-0fde0796315dcfb06

All ICMP - IPv4

ICMP

All

Custom

Q

172.16.0.0/12 X

-

SSH

TCP

22

Custom

Q 172.16.0.0/12 X

172.16.0.0/12 X

Add rule

14. Save the rule.

15. Go back to the VPC Reachability Analyzer using the search bar.

16. Check the box next to your previously saved analysis. Click “Analyze path”. Click confirm.

Summary <small>Info</small>		Actions ▾		Analyze path
Path ID nip-03b069f01e71e8a9f	Last analysis date May 30, 2024, 8:55 (UTC-04:00)	Reachability status ⌚ Pending	Last analysis status ⌚ Pending	
Source i-003cd5752fcc87c9a	Source account ID 423620453332	Destination eni-0bc70ab4d252a427e	Destination account ID 423620453332	
Destination port -	Protocol TCP			

Analyze path X

Path ID nip-03b069f01e71e8a9f	Source i-003cd5752fcc87c9a	Destination eni-0bc70ab4d252a427e
----------------------------------	-------------------------------	--------------------------------------

To confirm that you want to analyze this path, press the *Confirm* button below.

Include an intermediate component filter - *optional*

Enter Amazon Resource Name

Cancel Confirm

17. You may need to use the refresh button in the pane to see a new entry in the analyses table.

Analyses (1/2) <small>Info</small>					
<input type="text"/> Filter path analyses					
Analysis ID	Analysis run date	Reachability status	Intermediate comp...	State	
<input checked="" type="checkbox"/> nia-0fcfd18991dfc6e3a9	January 30, 2025, 12:2...	⌚ Pending	-	⌚ Pending	
<input type="checkbox"/> nia-0013048bf55708970	January 30, 2025, 12:1...	☒ Not reachable	-	☒ Succeeded	

18. There will be a new entry pending under Analysis, click refresh until it succeeds. The Reachability status should now be reachable. If it is not, view the Explanations.

Analyses (1/2) Info					
<input type="text"/> Filter path analyses					
	Analysis ID	Analysis run date	Reachability status	Intermediate comp...	State
<input checked="" type="checkbox"/>	nia-0fcd18991dfc6e3a9	January 30, 2025, 12:2...	Reachable	-	Succeeded
<input type="checkbox"/>	nia-0013048bf55708970	January 30, 2025, 12:1...	Not reachable	-	Succeeded

19. From your lab laptop, attempt to SSH to your provisioning machine, accepting any key acceptance messages. It should succeed this time and you should be at an ubuntu command prompt.

```

ubuntu@ip-172-18-25-31: ~
Usage of /: 19.9% of 7.57GB  Users logged in: 0
Memory usage: 21%          IPv4 address for eth0: 172.18.25.31
Swap usage: 0%
0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-18-25-31:~$ 
ubuntu@ip-172-18-25-31:~$ 
ubuntu@ip-172-18-25-31:~$ 

```

Task 4: Configure DNS in Route 53

ISE requires DNS for many operations including initial database priming and any internode operations.

Table 4-1

Pod	Primary Hostname	Primary IP Address	Secondary Hostname	Secondary IP Address
Pod 1	pod1-ise1	10.1.0.5	pod1-ise2	10.1.0.6
Pod 2	pod2-ise1	10.2.0.5	pod2-ise2	10.2.0.6
Pod 3	pod3-ise1	10.3.0.5	pod3-ise2	10.3.0.6
Pod 4	pod4-ise1	10.4.0.5	pod4-ise2	10.4.0.6
Pod 5	pod5-ise1	10.5.0.5	pod5-ise2	10.5.0.6
Pod 6	pod6-ise1	10.6.0.5	pod6-ise2	10.6.0.6
Pod 7	pod7-ise1	10.7.0.5	pod7-ise2	10.7.0.6
Pod 8	pod8-ise1	10.8.0.5	pod8-ise2	10.8.0.6
Pod 9	pod9-ise1	10.9.0.5	pod9-ise2	10.9.0.6
Pod 10	pod10-ise1	10.10.0.5	pod10-ise2	10.10.0.6
Pod 11	pod11-ise1	10.11.0.5	pod11-ise2	10.11.0.6
Pod 12	pod12-ise1	10.12.0.5	pod12-ise2	10.12.0.6
Pod 13	pod13-ise1	10.13.0.5	pod13-ise2	10.13.0.6
Pod 14	pod14-ise1	10.14.0.5	pod14-ise2	10.14.0.6
Pod 15	pod15-ise1	10.15.0.5	pod15-ise2	10.15.0.6
Pod 16	pod16-ise1	10.16.0.5	pod16-ise2	10.16.0.6
Pod 17	pod17-ise1	10.17.0.5	pod17-ise2	10.17.0.6
Pod 18	pod18-ise1	10.18.0.5	pod18-ise2	10.18.0.6
Pod 19	pod19-ise1	10.19.0.5	pod19-ise2	10.19.0.6
Pod 20	pod20-ise1	10.20.0.5	pod20-ise2	10.20.0.6
Pod 21	pod21-ise1	10.21.0.5	pod21-ise2	10.21.0.6
Pod 22	pod22-ise1	10.22.0.5	pod22-ise2	10.22.0.6
Pod 23	pod23-ise1	10.23.0.5	pod23-ise2	10.23.0.6
Pod 24	pod24-ise1	10.24.0.5	pod24-ise2	10.24.0.6
Pod 25	pod25-ise1	10.25.0.5	pod25-ise2	10.25.0.6
Pod 26	pod26-ise1	10.26.0.5	pod26-ise2	10.26.0.6
Pod 27	pod27-ise1	10.27.0.5	pod27-ise2	10.27.0.6
Pod 28	pod28-ise1	10.28.0.5	pod28-ise2	10.28.0.6
Pod 29	pod29-ise1	10.29.0.5	pod29-ise2	10.29.0.6
Pod 30	pod30-ise1	10.30.0.5	pod30-ise2	10.30.0.6

1. Search for “Route 53” in the AWS Console.
2. Under DNS Management, choose “Hosted zones”
3. Zer0k has already been registered so click on zer0k.org.

The screenshot shows the AWS Route 53 Hosted zones interface. At the top, there's a breadcrumb navigation: Route 53 > Hosted zones. Below that, a header says "Hosted zones (1)". A note below the header states: "Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings." There are four buttons: a grey "C" button, "View details", "Edit", and "Delete". A prominent orange "Create hosted zone" button is also visible. Below these buttons is a search bar with the placeholder "Filter hosted zones by property or value". The main table lists the single hosted zone: "zer0k.org" (Hosted zone name), "Private" (Type), "Route 53" (Created by), and "2" (Record count). The table has columns for Hosted zone name, Type, Created by, and Record count.

4. As part of the ISE provisioning, the Primary ISE node in your pod (10.x.0.5) will be provisioned into Route53. To help understand the task done via script, you will register the second node manually.
5. Click “Create record” and enter the following information (replace X with your pod number):
 - a. Record name – podX-ise2
 - b. Value – 10.X.0.6
6. Leave the rest of the values as default then click “Create records”.

The screenshot shows the AWS Route 53 'Records' page with the following details:

- Records (3)**: The current search behavior is set to 'Automatic'.
- Actions**: Buttons for 'Delete record' and 'Import zone file'.
- Create record**: A prominent orange button.
- Filter**: A search bar with placeholder 'Filter records by property or value'.
- Navigation**: Page 1 of 1.
- Columns**: Headers for 'Record name', 'Type', and 'Routing policy'.
- Data**:

Record name	Type	Routing policy
zer0k.org	NS	Simple
zer0k.org	SOA	Simple
pod31-ise2.zer0k.org	A	Simple

Task 5: VPC Automation

Terraform will be used to provision ISE into AWS, and the required components which allows ISE to connect to the internet. It will maintain the state of the AWS resources and ISE instance, as well as allow for an easy tear down post-lab or in a real world scenario, should you use these instructions after Cisco Live!.

Step 1: Install Terraform

1. In the preceding step, you should have established an SSH session to the Ubuntu provisioning machine for your pod. The Ubuntu machine should be sitting at the default prompt after login.

```
ubuntu@ip-172-18-25-1: ~
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Jan 24 18:29:21 UTC 2025

System load: 0.07           Processes:          107
Usage of /: 21.7% of 7.57GB Users logged in:      0
Memory usage: 20%           IPv4 address for eth0: 172.18.25.1
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

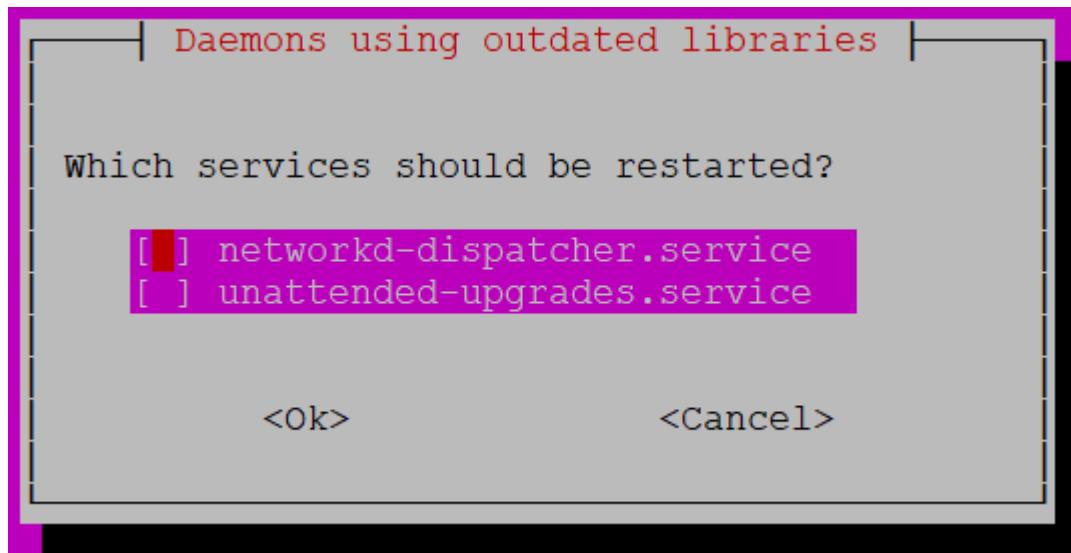
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-18-25-1:~$
```

2. Ubuntu uses apt for its package manager, first apt needs to be updated to get the most current package and version list from the default package repository: ***sudo apt-get update***

Note: If this step fails due to connectivity, check your security groups to ensure you didn't modify the outbound security group rule.

3. Upgrade the packages on the system to the most current versions:
sudo apt-get upgrade. Answer “Y” when shown how much space the upgrades will take.
4. You will be prompted to restart services that are using some of the packages that were updated. Leave the defaults selected and select (tab, enter) OK.



5. This lab will use a number of open-source libraries for Terraform.

Install the GNU-PG Common Software Properties library by running:

```
sudo apt-get install -y gnupg software-properties-common
```

The package may already be up to date, at which point you would see the following screenshot. Otherwise, updates will be performed.

```
ubuntu@ip-172-18-25-1:~$ sudo apt-get install -y gnupg software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

6. With Terraform being an open-source package, it is important to verify that the terraform package we're installing has not been tampered with.

Download the Terraform package fingerprint with the following command (all one command):

```
wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
```

```
--2025-01-24 18:39:53-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 108.156.184.65, 108.156.184.19, 108.156.184.5, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|108.156.184.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

          100%[=====] 3.89K --.-KB/s   in 0s

2025-01-24 18:39:53 (1.75 GB/s) - written to stdout [3980/3980]
```

7. Verify the fingerprint with the following command:

```
gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
```

--fingerprint

The following output should be presented indicating the fingerprint is valid, the signature in **bold** should match the screenshot until 2028:

```
gpg: directory '/home/ubuntu/.gnupg' created
gpg: /home/ubuntu/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
-----
pub rsa4096 2023-01-10 [SC] [expires: 2028-01-09]
798A EC65 4E5C 1542 8C8E 42EE AA16 FCBC A621 E701
uid      [unknown] HashiCorp Security (HashiCorp Package
Signing) <security+packaging@hashicorp.com>
sub rsa4096 2023-01-10 [S] [expires: 2028-01-09]
```

8. Terraform is a custom repository, which can be thought of as a custom “app store”. We’ll need to add the repository to the set which Ubuntu uses to ensure Terraform can be downloaded. Run the following command:
`echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list`
9. We’ll once again need to update our repositories to ensure Ubuntu pulls down the index of available packages, specifically so we get the latest version of Terraform. Run the following command: **sudo apt-get update**
10. Install Terraform from the repository: **sudo apt-get install terraform**
11. If asked which services should be restarted, repeat the same process as in the preceding steps and hit **<Tab>**, **Enter** to keep the defaults.
12. Once complete, run **terraform -help** to verify installation succeeded and Terraform is now an accessible package on your system.

```
Usage: terraform [global options] <subcommand> [args]
```

The available commands for execution are listed below.
The primary workflow commands are given first, followed by less common or more advanced commands.

Main commands:

init	Prepare your working directory for other commands
------	---

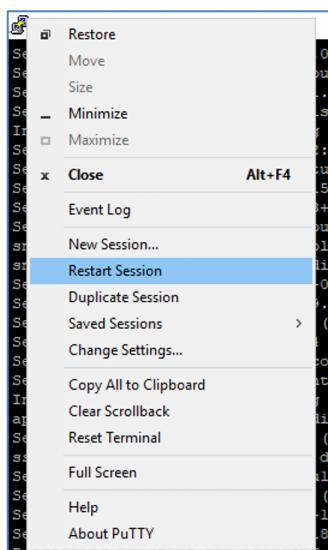
```
validate    Check whether the configuration is valid
plan       Show changes required by the current configuration
apply      Create or update infrastructure
destroy    Destroy previously-created infrastructure
<snip>
```

Congratulations, you now have your first orchestration package installed!

13. Terraform provides for tab completion as part of the package, which assists with ease of operation. Run the following commands to enable tab completion: `touch ~/.bashrc && terraform -install-autocomplete`
14. There are some services that were unable to be restarted individually, including the tab completion for terraform. Restart the system to get those services using the latest package versions: `sudo shutdown -r now`. This will prevent further restart messages in the future.

Note: Now is the perfect time to stretch and get some water. It'll take up to 2 minutes for the virtual machine to start. In the meantime, you'll receive a "Connection Refused" error.

15. Restart the putty session by right clicking the putty icon on the window and selecting Restart Session. If it times out, try again until it connects, it could take a couple of minutes. If prompted, use username **ubuntu** once again.



Step 2: Clone the GIT Repository for Script Execution

Clone the scripts to be used for execution from GIT into the production provisioning machine. These will be used to execute all tasks for the remainder of the lab. There isn't enough time in the lab for you to write the scripts but as you go through and execute them, take some time to read over the scripts first.

1. Verify you are in the home directory by executing ***pwd***, you should be in the /home/ubuntu directory. If you are not, change directory with:

```
cd /home/ubuntu
```

2. Download (Clone) the GIT repository to the local machine. Execute the command:

```
git clone https://github.com/minimavus/LTRSEC-2000
```

The output should look similar to the following:

```
Cloning into 'LTRSEC-2000'...
...
Receiving objects: 100% (101/101), 165.20 KiB | 2.01 MiB/s,
done.
Resolving deltas: 100% (32/32), done.
```

3. There should now be an 'LTRSEC-2000' directory in your home directory at this point, run ***ls -al*** to verify.

```
ubuntu@ip-172-18-25-2:~$ ls -al
total 44
drwxr-x-- 6 ubuntu ubuntu 4096 Jan 30 21:35 .
drwxr-xr-x 3 root   root   4096 Jan 30 21:05 ..
-rw----- 1 ubuntu ubuntu  655 Jan 30 21:34 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3813 Jan 30 21:34 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jan 30 21:30 .cache
drwx----- 2 ubuntu ubuntu 4096 Jan 30 21:33 .gnupg
-rw-r--r-- 1 ubuntu ubuntu  807 Jan  6 2022 .profile
drwx----- 2 ubuntu ubuntu 4096 Jan 30 21:05 .ssh
-rw-r--r-- 1 ubuntu ubuntu    0 Jan 30 21:31 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu ubuntu 181 Jan 30 21:32 .wget-hsts
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 30 21:35 LTRSEC-2000
```

4. Change directories into the LTRSEC-2000 directory and verify there is both an **Ansible** and **Terraform** directory:

```
cd LTRSEC-2000
```

```
ubuntu@ip-172-18-25-2:~/LTRSEC-2000$ ls -al
total 44
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 30 21:35 .
drwxr-x--- 6 ubuntu ubuntu 4096 Jan 30 21:35 ..
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan 30 21:35 .git
-rw-rw-r-- 1 ubuntu ubuntu 110 Jan 30 21:35 .gitignore
drwxrwxr-x 6 ubuntu ubuntu 4096 Jan 30 21:35 Ansible
-rw-rw-r-- 1 ubuntu ubuntu 18456 Jan 30 21:35 README.md
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan 30 21:35 Terraform
```

Step 3: Configure the Terraform Environmental Variables to be Used

As part of the deployment process, multiple variables are used to ensure configurations are populated and access keys are available to Terraform and Ansible. This step configures variables for Terraform. Additional steps are found later in the lab to do the same for Ansible.

1. Change directory into your Terraform directory to modify your pod number and key pair. Run the following commands:

cd Terraform

ls -al

2. Verify the “variables.tf” file exists.

```
ubuntu@ip-172-18-25-1:~/LTRSEC-2000/Terraform$ ls -al
total 24
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan 24 19:01 .
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 24 19:01 ..
-rw-rw-r-- 1 ubuntu ubuntu 4995 Jan 24 19:01 main.tf
-rw-rw-r-- 1 ubuntu ubuntu 263 Jan 24 19:01 userdata.tftpl
-rw-rw-r-- 1 ubuntu ubuntu 2155 Jan 24 19:01 variables.tf
```

3. Access the variables.tf file by running: ***nano variables.tf***
4. Change the pod number in the variables file from the default of “<TBD>” to your pod number. Replace the 0 and keep the quotation marks.

```
variable "pod_number" {
    description = "Assigned Pod Number"
    type = string
    default = "X"
}
```

5. Change the keypair that will be used to access ISE to “podX-keypair” where X is your pod number.

```
variable "pod_keypair" {
    description = "Key Pair Created By the Student"
    type = string
    default = "podX-keypair"
}
```

6. Save the file by using the key combination Ctrl+X, answer “Y” to save the configuration, and press **Enter** to confirm we’ll be saving to the current file.
7. To prevent malicious actors from accessing the lab with what we publish to git, there is one file that needs to be created by the user with sensitive information such as the access key and secret key. While in the Terraform working directory, run the command **nano terraform.tfvars** This command will create a new file call `terraform.tfvars`.
8. Paste the following into `terraform.tfvars`, populating your access key, secret key, and password accordingly:

```
#AWS keys (not shared with github)
aws_access_key = "<YOUR ACCESS KEY>"
aws_secret_key = "<YOUR SECRET KEY>"
ise_password = "<INITIAL ISE PASSWORD ENDING WITH !>"
```

Maintain all quotation marks and ensure they are straight quotation marks, not curly. Replace the access key and secret key with the values you saved from the Ubuntu machine creation. Enter the password provided by the proctors.

9. Save the file by using the key combination Ctrl+X, answer “Y” to save the configuration, and press Enter to confirm we’ll be saving to the current file.

Step 4: Deploy the AWS components, and Identity Services Engine

1. Explore what the Terraform script will do by using the **more main.tf** command. Some areas to focus on within the script are the following:
 - a. Provider “aws”
 - b. #Create Pod VPC
 - c. #Create Private Subnet
 - d. #Create Transit Gateway Attachment
 - e. #Create Private Route Table
 - f. #Attach the Route Table to the Subnet

- g. #Create Security Group for ISE
 - h. #Get AMI from Market Place for which ever region is being used
 - i. #Create a DNS Entry for the Node
 - j. #Create the ISE Instance
2. We'll now run the terraform script, but need to first initialize terraform to pull in the files and tasks it'll need to perform. Run the command: **terraform init** to initialize terraform.

```
Initializing the backend...
Initializing provider plugins...
  - Finding latest version of hashicorp/aws...
  - Installing hashicorp/aws v5.84.0...
  - Installed hashicorp/aws v5.84.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!
```

3. We'll now want to verify our “plan”, or what Terraform will do before finally executing on it. Run the command: **terraform plan** to see what changes will be made. Verify these plans in accordance with step 2:
- a. # aws_vpc.pod-vpc will be created
 - b. # aws_subnet.pod-private-subnet will be created
 - c. # aws_security_group_rule.allow_tls_ipv4 will be created
 - d. # aws_security_group_rule.allow_ssh_ipv4 will be created
 - e. # aws_security_group.ise-security-group will be created
 - f. # aws_route_table_association.a will be created
 - g. # aws_route_table.pod-private-rt will be created
 - h. # aws_route53_record.www will be created
 - i. # aws_instance.ise-instance will be created
 - j. # aws_ec2_transit_gateway_vpc_attachment.zer0k-transit-gateway will be created

The plan should end with the following verbiage:

```
Plan: 14 to add, 0 to change, 0 to destroy.
```

Note: If you forgot to add in the access key, secret key, or ISE password, you will be prompted at this point to add them in. This dialog will look like the following:

```
ubuntu@ip-172-18-25-1:~/LTRSEC-2000/Terraform$ terraform plan
var.aws_access_key
Enter a value: AKIAWFIN7EPKC5YH4OLP

var.aws_secret_key
```

```
Enter a value: 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
```

```
var.ise_password
```

```
Enter a value: CLUS2023Party!
```

Note: You'll find that many of the configuration attributes associated with ISE and AWS are human readable within the output that Terraform returns. We encourage you to read through and understand what is being applied and compare it to the network diagram at the beginning of the lab. ISE will take up to 30 minutes to install and initialize, during which is a great time to develop your understanding.

4. You may now apply the terraform configurations with the apply command, this will create the resources into AWS. Run the following command: **terraform apply**. Answer **yes** when prompted whether you want to perform these actions.
5. When successfully applied, the script should end with:

```
Apply complete! Resources: 11 added, 0 changed, 0 destroyed.
```

Outputs:

```
instance_id = "i-053350f3ff9514710"
instance_private_ip = "10.2.0.5"
ise_password = "<Provided by your proctor>"
pod-key-name = "Creating Instance with Key Name: podX-keypair"
pod_number = "X"
```

Copy these values to your notepad for use in the next task.

Step 5: Explore the AWS Environment to Ensure the Network Environment Was Provisioned

With the scripts being successfully run it is time to verify that they created and configured the environment as expected.

1. Verify the VPC was deployed. Navigate to the search bar in AWS, type VPC. Click VPC in the results and navigate to “Your VPC’s”. The VPC should be named ISEinAWS-podX where X is your assigned pod number.

<input type="checkbox"/>	zer0k-pod0	vpc-0105fb3ba3f6cbec3	Available	10.0.0.0/16
<input type="checkbox"/>	ISEinAWS-pod9	vpc-0dbba522e2e80f740	Available	10.9.0.0/16
<input checked="" type="checkbox"/>	ISEinAWS-pod10	vpc-06c27a7f1c1c05e39	Available	10.10.0.0/16
<input type="checkbox"/>	zer0k-main-vpc	vpc-0a61b7f1d92102ad6	Available	172.18.0.0/16

2. Navigate to Subnets in the left menu. Verify the Private subnet is deployed in accordance with your pod. It should be named ISEinAWS_podX_Private_Subnet where X is your assigned pod number.

Subnets (1/8) [Info](#)

Find resources by attribute or tag						
	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
<input type="checkbox"/>	zer0k-public-subnet	subnet-0653589a8ba2fa824	Available	vpc-0a61b7f1d92102ad6 zer0...	<input type="checkbox"/> Off	172.18.0.0/21
<input type="checkbox"/>	zer0k-private-subnet	subnet-059dc621173c4d170	Available	vpc-0a61b7f1d92102ad6 zer0...	<input type="checkbox"/> Off	172.18.16.0/21
<input type="checkbox"/>	zer0k-inside-subnet	subnet-0cf86b138b53ba3c9	Available	vpc-0a61b7f1d92102ad6 zer0...	<input type="checkbox"/> Off	172.18.24.0/21
<input checked="" type="checkbox"/>	ISEinAWS-pod1_Private_Subnet	subnet-0d52cd201c2539dcc	Available	vpc-060aecfa62678565a ISEin...	<input type="checkbox"/> Off	10.1.0.0/24

3. Verify the subnet allocated to your pod is in the form of 10.X.0.0/24 where X is your pod number.
4. Navigate to Route Tables in the left menu. Verify a routing table was provisioned for your pod. Under the column “VPC” a route table should be named ISEinAWS-podX where X is your assigned pod number.

Route tables (1/10) [Info](#)

Find resources by attribute or tag						
	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input type="checkbox"/>	ISEinAWS-pod0_Transit_Attach	rtb-02f5e679d9d136aa3	subnet-0256780f7c9e8a...	-	No	vpc-0a564561747453d46 ISEinAWS-pod0
<input type="checkbox"/>	zer0k-main-public-routes	rtb-0761a46ae6aad8b1b	2 subnets	-	No	vpc-0a61b7f1d92102ad6 zer0k-main-vpc
<input type="checkbox"/>	zer0k_pod4_RT_Privat	rtb-0062199a6c3fb8acc	-	-	No	vpc-01ab3e8c88c6a633a ISEinAWS-pod4
<input type="checkbox"/>	zer0k-main-routes	rtb-02c985d44a05c0051	-	-	Yes	vpc-0a61b7f1d92102ad6 zer0k-main-vpc
<input type="checkbox"/>	zer0k_pod4_RT_Privat	rtb-03bcc71afac143b3d	-	-	No	vpc-01ab3e8c88c6a633a ISEinAWS-pod4
<input checked="" type="checkbox"/>	-	rtb-0ef1f5556b5c4fb03e	-	-	Yes	vpc-060aecfa62678565a ISEinAWS-pod1

5. Verify a Transit Gateway Attachment exists for your pod. Under the Column “Name” the transit attachment should be named “ISEinAWS-podX_Transit_Attach” where X is your pod number.

Step 6: Install Ansible and Modify Variables for Your Pod

While ISE initializes now is a good time to get started on the next task, while Terraform is great at deploying resources, Ansible is better at configuration, let’s get Ansible installed and prepared.

1. Change your directory back to the home directory with `cd /home/ubuntu`.
2. This lab will use a python virtual environment. Venv is not installed by default so install it: **`sudo apt-get install python3.10-venv`**. A virtual python environment allows for different projects on the same machine to use different versions of shared python libraries. Answer **Y** when prompted to install.
3. Create a virtual environment for this lab, this will create a new folder in the `/home/ubuntu` directory: **`python3 -m venv ISEonAWS`**.
4. Activate the newly created ISEonAWS virtual environment by running the activation script: **`source ISEonAWS/bin/activate`**. This script sets up environment variables on the system to use the activated virtual environment. You should see the linux prompt prepended with `(ISEonAWS)` at this time.

Note: If you log out of Ubuntu, you'll need to repeat the command **source ISEonAWS/bin/activate** to get back into your virtual environment.

5. Install the python packages required to run ansible for ISE: **python3 -m pip install ansible boto boto3 botocore ciscoisesdk jmespath**.
This step will take a few minutes.

Step 7: Modify your ansible Variables

1. Your current working directory, or directory that you are within should be /home/ubuntu/. Verify this by running the command **pwd** meaning “present working directory”.

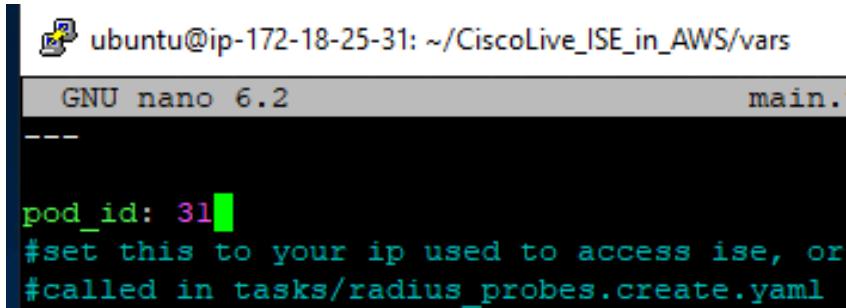
```
ubuntu@ip-172-18-25-1:~/LTRSEC-2000/Terraform$ pwd  
/home/ubuntu/
```

2. Change directories into Ansible using command **cd LTRSEC-2000/Ansible**.
3. Ansible's variables exist in the “vars” directory. Edit the vars/main.yaml file to update your pod number, which will be utilized when we configure ISE in later steps. Refer to the output from the terraform apply command in your notepad if you forget any of these values.
Execute the commands:

```
cd vars  
nano main.yaml
```

4. Edit the line in main.yaml **pod_id:** to match your pod number.

Note: Update the pod_id: to only have the pod number in it, as seen in the following screenshot. This is a variable used throughout the lab, and only takes in a numerical value.



```
ubuntu@ip-172-18-25-31: ~/CiscoLive_ISE_in_AWS/vars  
GNU nano 6.2          main.yaml  
---  
pod_id: 31  
#set this to your ip used to access ise, or  
#called in tasks/radius_probes.create.yaml
```

5. Scroll down in main.yaml to ensure the “ise_username” variable is set to **iseadmin**. If it is not, edit it. Do not change any other variables at this time.

6. Set the ise_password to the *updated* ISE password, which ends with a \$. This password is provided by your proctor.

Note: Because of the order of operations within the lab, Terraform will deploy ISE with an initial password ending in !. This is expected. ISE will require a password reset before Ansible is used to configure users, groups, and network access devices. This password will end with \$. Both passwords are printed on your information card for your pod.

7. Exit and save from nano utilizing the key sequence “Ctrl-X, Y, <enter>”
8. Once back at the Ubuntu command line, replace the access key and secret key with the one you created in Task 1, Step 5 in the file “PodX Important Information.txt” on the Windows desktop. It should look like the following:

```
export AWS_REGION=us-east-2
export AWS_ACCESS_KEY=XXXXXXXXXXXXXXXXXXXX <----  

Your Access Key Here
export AWS_SECRET_KEY=YYYYYYYYYYYYYYYYYYYYYYYY <----Your  

Secret Key Here
export ise_verify=false
```

Note: Ensure there are no spaces in any of the exported variables.

Note: If you get logged out of Ubuntu, you'll need to repeat the export of variables, as they are local to your session.

9. Paste the export commands block of text into the Ubuntu CLI, line by line.

Note:

- Right click in PuTTY will paste commands, do not use ctrl+v.
- Do not paste the entire block, paste line by line to ensure no spaces are added.
- Ensure there are no spaces before or after any of the variable names.

10. Verify by running the **env** command:

```
ubuntu@ip-172-18-25-1:~/LTRSEC-2000/Ansible/vars$ env
```

```

SHELL=/bin/bash
AWS_ACCESS_KEY=<YOUR ACCESS KEY>
AWS_SECRET_KEY=<YOUR SECRET KEY>
AWS_REGION=us-east-2
PWD=/home/ubuntu/LTRSEC-2000/Ansible/vars

```

Step 8: Monitor the Environment and ISE Deployment

- ISE will take up to 30 minutes to initiate. Depending on how long it took to install Ansible, you may have time before ISE is fully deployed, during which time you can monitor it from within the AWS GUI. Within your web browser, navigate back to EC2 by using the search bar on top of the AWS page.
- Create a filter in EC2 for your pod instances, enter “podX” where x is your pod number in the search bar to filter down to just your pod instances.

Instances (8) [Info](#)

The screenshot shows the AWS EC2 Instances page. A search bar at the top contains the text "pod1". Below the search bar, a list of instances is displayed, each with a checkbox, name, instance ID, state, type, and status check results. The first instance, "pod0-ise1", has its checkbox checked and is highlighted with a blue border. Its details are expanded:

- Use:** "pod1"
- Tags values**
- Created_by** = **pod1-awsadmin**
- Name** = **pod1-ubuntu-ansible**
- Name** = **pod1-ise1**
- Pod** = **pod1**

- Check the status of podX-ise1, wait until it has completed 3/3 checks before proceeding to the next step. The node will need to cycle through initializing, ½ checks passed, and 3/3 checks passed before being pingable.

<input type="checkbox"/>	Name 🔗	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/>	pod0-ise1	i-0171a64348aee64dc	Running 🔗 🔗	c5.9xlarge	3/3 checks passed
<input type="checkbox"/>	pod0-Ubuntu	i-0b070ab4c1d373ffb	Running 🔗 🔗	t2.micro	2/2 checks passed

- You can also monitor the state of ISE as it comes up through the System log. Check the box next to your ISE instance then in the top right go to Actions -> Monitor and troubleshoot -> Get system log. This is helpful to ensure that the user_data provided to the instance is correct.

System log

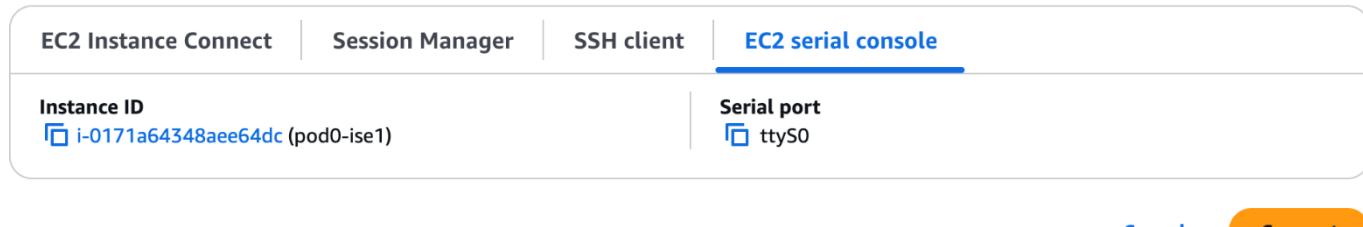
Review system log for instance i-0171a64348aee64dc as of Fri Jan 31 2025 14:16:34 GMT-0500 (Eastern Standard Time)

```
[[0;32m OK [0m] Started Logout off all iSCSI sessions on shutdown.  
[[0;32m OK [0m] Started Enable periodic update of entitlement certificates..  
[[0;32m OK [0m] Started GSSAPI Proxy Daemon.  
[[0;32m OK [0m] Reached target NFS client services.  
[[0;32m OK [0m] Reached target Remote File Systems (Pre).  
[[0;32m OK [0m] Reached target Remote File Systems.  
Starting Crash recovery kernel arming...  
Starting Permit User Sessions...  
[[0;32m OK [0m] Started Permit User Sessions.  
Starting Hold until boot process finishes up...  
Starting Terminate Plymouth Boot Screen...  
[[0;32m OK [0m] Started Command Scheduler.  
[[0;32m OK [0m] Started Job spooling tools.  
  
Press <Enter> to continue  
DEV-ISE-125 login:  
  
ISE: Validating user provided data:  
  
ISE: User provided data validation Passed
```

10. The serial console can also be used by checking the box next to the instance then select “Connect” on the top right of the EC2 console. On the Connect to instance screen, select the EC2 Serial console then click “Connect” in the bottom right.

Connect to instance Info

Connect to your instance i-0171a64348aee64dc (pod0-ise1) using any of these options



Step 9: Establish an SSH session with ISE

Note: ISE will need to initialize and can take up to 30 minutes to fully deploy. You can verify the status of your ISE instance in EC2 -> Instances. In the Status check header in the EC2 dashboard, the status for ISE must have all checks passed, as opposed to “initializing”

pod10-ise1	i-0e38dab8285aeaaa7	Running	c5.4xlarge	2/2 checks passed
------------	---------------------	---------	------------	-------------------

1. From the Windows PC open a command prompt and attempt to ping ISE. ISE’s IP address should be in the form of 10.X.0.5 where X is your pod number.

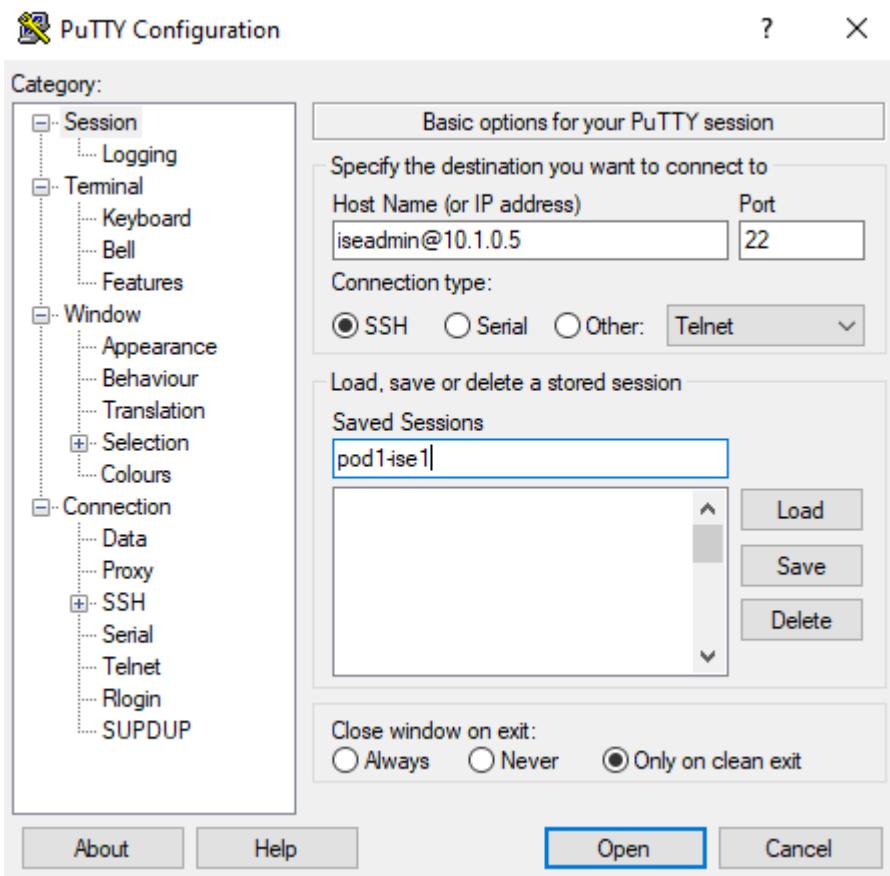
```
Windows Command Prompt  
Microsoft Windows [Version 10.0.19045.5371]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\palloyd-test-pc>ping 10.1.0.5  
  
Pinging 10.1.0.5 with 32 bytes of data:  
Reply from 10.1.0.5: bytes=32 time=57ms TTL=63  
Reply from 10.1.0.5: bytes=32 time=57ms TTL=63  
Reply from 10.1.0.5: bytes=32 time=56ms TTL=63  
Reply from 10.1.0.5: bytes=32 time=56ms TTL=63  
  
Ping statistics for 10.1.0.5:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 56ms, Maximum = 57ms, Average = 56ms
```

2. In the following steps, we'll establish an SSH session with ISE to verify services transition through a Not Running -> Initializing -> Running cycle. This should be done from the lab.

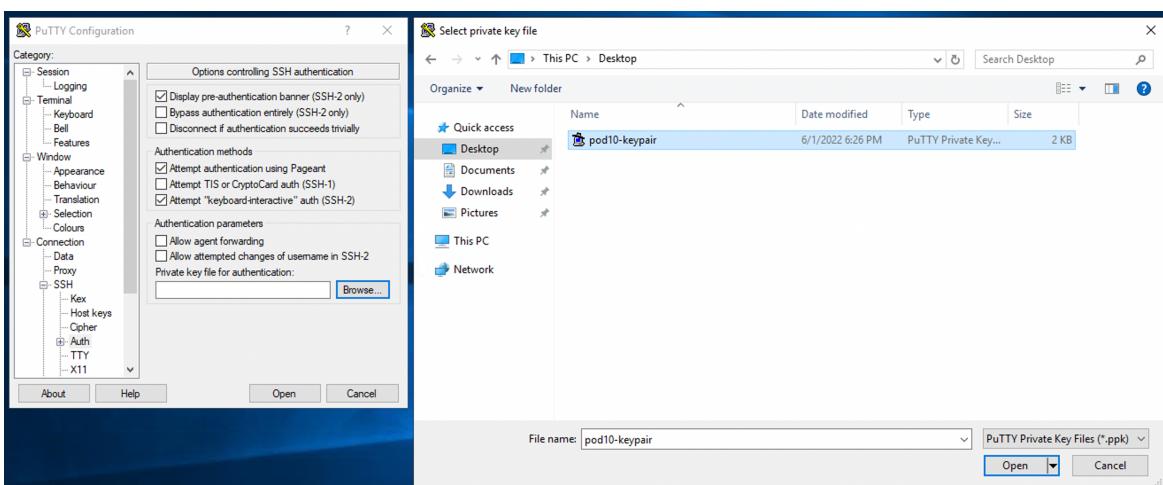
NOTE: SSH will fail until ISE is fully provisioned and running. Within this process the ISE server is provisioned, the database primed, the ISE node restarted, and only then is the application server and SSH process available. This can take up to **30 minutes**. Progress can be monitored in the AWS console within the EC2 Instances area. This will manifest itself as:

```
“(ISEonAWS) ubuntu@ip-172-18-25-4:~ $ ssh -i ~/.ssh/ISEinAWS-  
pod4.pem iseadmin@10.4.0.5  
admin@10.4.0.5: Permission denied (publickey).”
```

3. On the Windows PC, navigate back to the desktop and open Putty from the shortcut.
4. The default screen is under the Session category. Set the Hostname to “iseadmin@10.X.0.5” and Save the Session as “podX-ise1” where X is your assigned pod number. Click the save button.



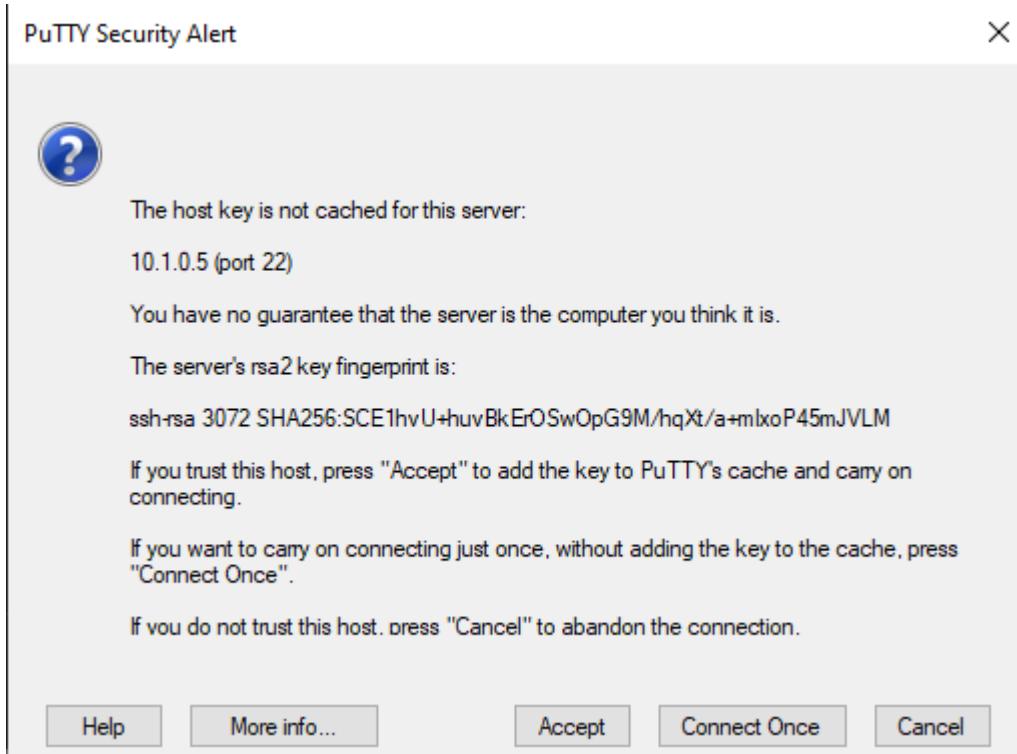
5. Navigate to Connection -> SSH -> Auth -> Credentials. In the “Private key file for authentication” box at the bottom of the screen, select the private key created from the PEM file earlier in the lab. It should have been named podX-keypair.ppk located on the desktop from the previous preparation step.



6. Optional: Navigate to Session -> Logging, select “All Session Output”, browse, and place the log file in the desktop. This is strictly for future troubleshooting if desired.
7. Do not change any additional settings.
8. Go back to Session and save again.

9. Click Open.

10. If presented with a Putty Security Alert, accept the fingerprint.



11. You should now have access to the ISE console.

A screenshot of a terminal window titled "10.1.0.5 - PuTTY". The window shows the following text:

```
Failed to log in 0 time(s)

iseadmin connected from 172.16.1.12 using ssh on podl-isel
podl-isel/iseadmin#
```

The cursor is located at the end of the command line.

12. **Optional:** If you are unable to SSH to ISE and receive an error of **Permission denied (publickey)**, ISE may still be provisioning. A procedure to connect via the Amazon EC2 Serial Console is provided in Appendix 1 to observe the provisioning process.

13. ISE will need to be in a running state to access the GUI. The Application server will need to be in a running state before this is the

case.

Run the following command:

```
podXX-ise/admin# show application status ise
ISE PROCESS NAME      STATE      PROCESS ID
-----
-----
Database Listener      running    32895
Database Server        running    85 PROCESSES
Application Server     not running
...
Application Server     initializing 49728
...
Application Server     running
```

The application server will transition through each of the states of not running, initializing, and running as seen in the snippet above.

14. Once the Application Server shows as “running” move on to the next step.

Step 10: Verify the running configuration of ISE to align with your Pod

Verify the running configuration, including IP address for the private facing interface, aligns with expected IP’s for your pod. Also verify DNS is provisioned to the correct server.

Execute the command: ise/admin# **show run**

Expected Output:

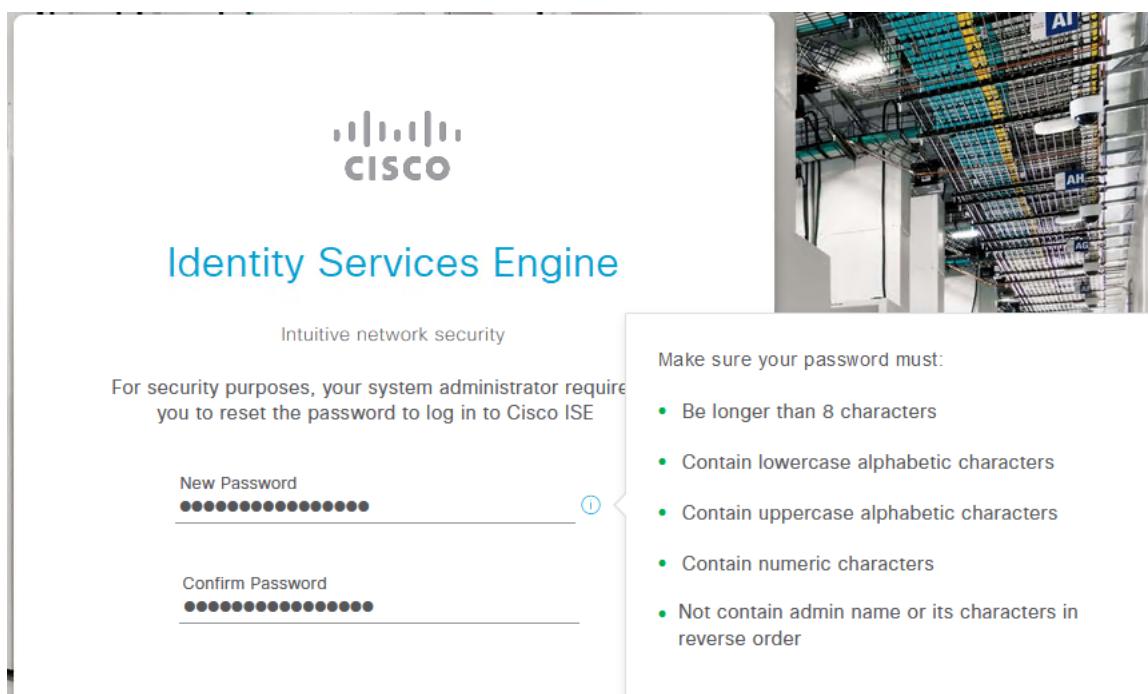
```
Generating configuration...
!
hostname ise
ip domain-name zer0k.org
interface GigabitEthernet 0
  ip address 10.X.0.5 255.255.255.0
!
ip name-server 169.254.169.253
ip default-gateway 10.X.0.1
clock timezone UTC
!
```

Step 11: Reset the ISE Password

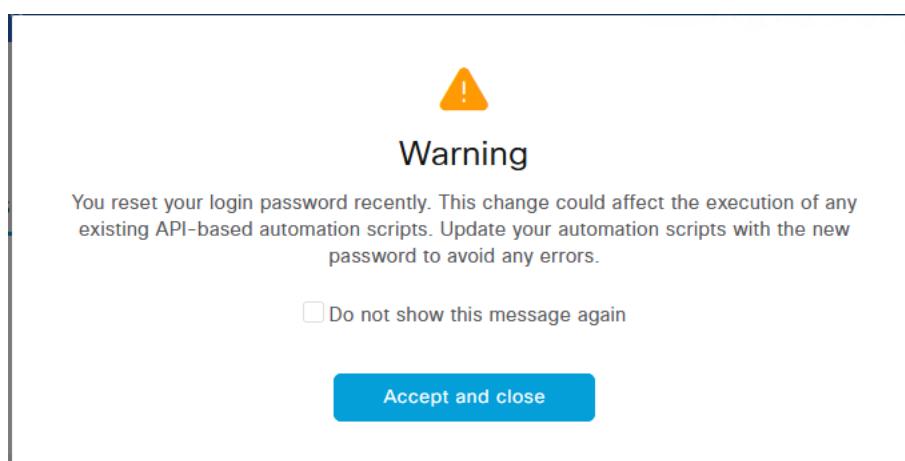
ISE forces the user to change their password on first login. Default settings of requiring complex passwords prevent the user from resetting

their password back to the same password used. The password therefore needs to change. In a previous step, a note was presented indicating a different password should be used in the Ansible variables file. This password will be what you change the GUI password to as part of this step.

1. Launch a web browser to ISE (you will need to populate the full URL of <https://10.X.0.5>) from your Windows PC, accepting any certificate errors. You will be prompted to reset the ISE GUI password. Reset the password as prompted to the updated password ending in \$, or the **Updated password provided by your proctor**.



2. You'll be logged out of ISE. Log back into ISE using the administrative username **iseadmin**, using the updated password.
3. If presented, accept the warning about resetting the password, noting this would be a concern for this lab.



4. Using the hamburger menu on the top left of the screen, navigate to Administration -> Identity Management -> Groups -> User Identity Groups
5. Verify only default ISE groups are populated, including OWN_ACCOUNTS, ALL_ACCOUNTS

<input type="checkbox"/>	 ALL_ACCOUNTS (default)	Default ALL_ACCOUNTS (default) User Group
<input type="checkbox"/>	 Employee	Default Employee User Group
<input type="checkbox"/>	 GROUP_ACCOUNTS (default)	Default GROUP_ACCOUNTS (default) User Group
<input type="checkbox"/>	 GuestType_Contractor (default)	Identity group mirroring the guest type
<input type="checkbox"/>	 GuestType_Daily (default)	Identity group mirroring the guest type
<input type="checkbox"/>	 GuestType_SocialLogin (default)	Identity group mirroring the guest type
<input type="checkbox"/>	 GuestType_Weekly (default)	Identity group mirroring the guest type
<input type="checkbox"/>	 OWN_ACCOUNTS (default)	Default OWN_ACCOUNTS (default) User Group

6. Navigate to Administration -> Identity -> Identities
7. Verify there are no users present in the identity store
8. Navigate to Administration -> Network Device Groups
9. Verify only default ISE NDG's are present.

<input type="checkbox"/>	All Device Types	All Device Types
<input type="checkbox"/>	All Locations	All Locations
<input type="checkbox"/>	> Is IPSEC Device	Is this a RADIUS over IPSEC Device

10. Navigate to Administration -> Network Devices
11. Verify no Network Devices are currently present.

Step 12: Configure ISE Programmatically with Ansible

Note: Should the following steps be performed outside of the lab or across regions, ensure that the correct IP is used for the ise.configuration.yaml script. If not, the script will perform 1000 pings before timing out.

If this happens it can be interrupted with **ctrl+c**.

Warning: If you did not complete step 5.9.6, the following will fail with

an error indicating authentication is required.

1. Open the SSH session to the Ubuntu Ansible machine. You may need to reset the session using the icon in the top left of the window.

```
y_vpc_attachment.zer0k-transit-gateway: Still creating...
y_vpc_attachment.zer0k-transit-gateway: Creation complete
tach-08fbff72ecfc6ffe9]
ces: 11 added, 0 changed, 0 destroyed.

pod-key-name - Creating Instance with Key Name: pod1-keypair"
ubuntu@ip-172-18-25-1:~/LTRSEC-2000/Terraform$
```

2. Navigate back to base Ansible directory with:

`cd /home/ubuntu/LTRSEC-2000/Ansible`

3. Ensure you are running within the Python virtual environment for Ansible, as indicated by (ISEonAWS) preceding your machine's hostname. If not, run the command

`source /home/ubuntu/ISEonAWS/bin/activate`

4. Verify the ISE password used for Ansible is the updated password ending in \$. Run the command **`more vars/main.yaml | grep password`**

```
ise_password: Cis12345!
ise_rest_password: "{{ ise_password }}"
repository_password: Cis12345!
```

5. Configure ISE from the Ansible virtual environment, to populate the Network Device Groups, Network Devices, User Roles, and Users. Run these commands from the Ubuntu-Ansible machine.

Execute the command: **`ansible-playbook ise.configuration.yaml`**

Expected Output:

PLAY [ISE Configuration Playbook]

```

TASK [Query for ISE instances in project "'ISEinAWS-palloyd'"]...
TASK [Show instances]...
TASK [Test for ISE Application Server Initialization]...
TASK [Ping 10.X.0.5]...
TASK [Wait for <private_IP> App Server (GUI)]...
TASK [Show <private_IP> Initialized]...
TASK [Enable ISE ERS & OpenAPIs]...
TASK [Enable ISE OpenAPIs (ISE 3.1+)]...
TASK [Show ISE OpenAPIs Enabled Status]...
TASK [Show ISE OpenAPIs Disabled Status]...
TASK [Get ISE ERS APIs Status]...
TASK [Enable ise.zer0k.org ERS APIs]...
TASK [Show ise.zer0k.org ERS Enabled Status]...
TASK [Show ise.zer0k.org ERS Disabled Status]...
TASK [Create RADIUS Probes - identity_group and internal_users]...
TASK [Create `RADIUS_Probes` identity group]...
TASK [Create Internal Users]...
TASK [Create Internal User Accounts]...
TASK [Create Network Device Groups]...
TASK [Create demo network_devices]...
TASK [Include Endpoints]...
TASK [Create Endpoints]...
PLAY RECAP
localhost      : ok=39  changed=6  unreachable=0  failed=0
skipped=4  rescued=0  ignored=0

```

Step 13: Verify Newly Configured Users, Groups, and Network Access Devices

1. On your Ubuntu provisioning machine, navigate to ~/LTRSEC-2000/Ansible/vars
2. Execute a more on each file to explore what is expected to be configured, including users, endpoints, groups, network device groups, and network devices.
3. Using the hamburger menu on the top left of the screen, navigate to Administration -> Identity Management -> Groups -> User Identity Groups
4. Verify the RADIUS_Probes group has been created as a group to assign users to.

User Identity Groups

	Edit	Add	Delete	Import	Export
	Name	Description			
<input type="checkbox"/>	ALL_ACCOUNTS (default)	Default ALL_ACCOUNTS (default) User Group			
<input type="checkbox"/>	Employee	Default Employee User Group			
<input type="checkbox"/>	GROUP_ACCOUNTS (default)	Default GROUP_ACCOUNTS (default) User Group			
<input type="checkbox"/>	GuestType_Contractor (default)	Identity group mirroring the guest type			
<input type="checkbox"/>	GuestType_Daily (default)	Identity group mirroring the guest type			
<input type="checkbox"/>	GuestType_SocialLogin (default)	Identity group mirroring the guest type			
<input type="checkbox"/>	GuestType_Weekly (default)	Identity group mirroring the guest type			
<input type="checkbox"/>	OWN_ACCOUNTS (default)	Default OWN_ACCOUNTS (default) User Group			
<input type="checkbox"/>	RADIUS_Probes	Group for RADIUS probe internal users			

5. Navigate to Administration -> Identity -> Identities
6. Verify the users found in the “internal_users.thomas.yaml” are configured as expected.

Identities	Groups	External Identity Sources	Identity Source Sequences	Setting																														
Users Latest Manual Network Scan Res...	<h2>Network Access Users</h2> Edit Add Change Status Import Export <table border="1"> <thead> <tr> <th>Status</th> <th>Username</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled chmula</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled cocarson</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled elparis</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled jedubois</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled meraki_8021x_test</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled palloyd</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled radius-probe</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled thomas</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/> Enabled vibobrov</td> <td></td> </tr> </tbody> </table>				Status	Username	Description	<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled chmula		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled cocarson		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled elparis		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled jedubois		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled meraki_8021x_test		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled palloyd		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled radius-probe		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled thomas		<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled vibobrov	
Status	Username	Description																																
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled chmula																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled cocarson																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled elparis																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled jedubois																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled meraki_8021x_test																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled palloyd																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled radius-probe																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled thomas																																	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Enabled vibobrov																																	

7. Navigate to Administration -> Network Devices
8. Verify only one network device, the ASA Headend, was configured.

Network Devices



<input type="checkbox"/>	Name	IP/Mask	Profile Name	Location	Type	Description
<input type="checkbox"/>	ASA_Headend	172.18.24.254/32	Cisco <small>(i)</small>	AMER	VPN_Headend#ASA	ASA_Headend

Task 6: Entra ID integration via ROPC for Remote Access VPN

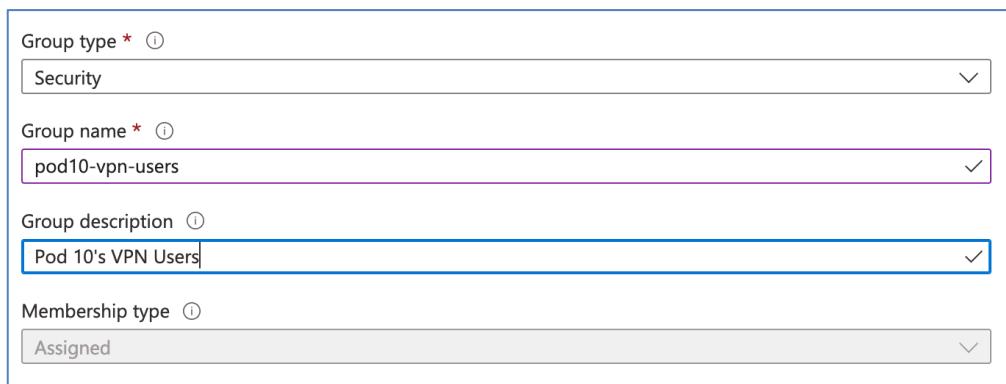
This task will have you create the basics needed for ISE integration with Entra ID for VPN authentication.

Note: Midway through the following task, you are asked to create a VPN user and set an initial password. Use the Initial password **provided by your proctor** and change this in the subsequent step to change the password to **Updated password provided by your proctor**

Step 1: Log into Entra ID and Configure Users/Groups

Note: You may be asked to change your password during the first login.

1. Log into <https://entra.microsoft.com/> with the pod administrator: **podX-admin@zer0k.onmicrosoft.com** where X is the number of your pod and the password **is provided by your proctor**
2. On the left navigation bar browse to Identity -> Groups -> All Groups.
3. Select New Group. This will be used on ISE to determine which users are allowed to log into the VPN.
4. Leave the Group Type as Security and Name the group **podX-vpn-users** where X is your pod number, give it a description for users who will be allowed to log into VPN later.



Group type * (i)
Security

Group name * (i)
pod10-vpn-users

Group description (i)
Pod 10's VPN Users

Membership type (i)
Assigned

Note: It may take a minute before the group shows up in the All Groups

area of Entra ID. Please be patient and hit refresh.

5. Using the left navigation bar, choose Identity -> Users -> All users.
6. Select New user at the top and select Create new user.
7. Name it **podX-vpn-user** where X is your pod number and give it a display name.

The screenshot shows the Microsoft Entra ID interface. The left sidebar has sections like Home, What's new, Diagnose & solve problems, Favorites, Identity, Overview, Users, All users, Deleted users, and User settings. The main area is titled 'Users' and shows a list of users under 'All users'. At the top right, there are buttons for '+ New user', 'Delete', and 'Download'. The user list includes:

User	Display Name	Last Name
AK	Aditya K R (adikr)	adi
AW	Alex Wight	aw
AG	Anvesh Gattu (agattu)	agi
BS	Blaine Schmidt (blschmid)	bli
BE	Brandon Enright (SVIC)	bre

8. Complete the basic information for a user, utilizing the @zer0k.onmicrosoft.com domain, and the initial password **provided by your proctor**.

The screenshot shows the 'Create new user' form. The left sidebar is identical to the previous screenshot. The main form has tabs for Basics, Properties, Assignments, and Review + create. The Basics tab is selected. It contains fields for User principal name (set to 'pod31-vpn-user'), Mail nickname (set to 'pod31-vpn-user'), Display name (set to 'Pod31 VPN User'), Password (set to 'CLEMEA2025Party!'), and Account enabled (checkbox checked). There are also checkboxes for 'Derive from user principal name' and 'Auto-generate password'.

9. Click Next to advance to Properties. No information is required in this tab. Click Next to advance to Assignments.
10. Click "Add Group" at the top of the page to be presented with the

“Select Group” pop out.

Home > Users >

Create new user

Create a new internal user in your organization

Basics Properties **Assignments** Review + create

Make up to 20 group or role assignments. You can only add a user to a maximum of 1 administrative unit.

+ Add administrative unit + Add group + Add role

No assignments to display.

Review + create < Previous Next: Review + create >

11. Under “Select Group”, check the box next to the group you created in the prior step, click “Select”.

Select group

Try changing or adding filters if you don't see what you're looking for.

Search

1 result found

All Groups

Name	Type	Details
<input checked="" type="checkbox"/> pod31-vpn-users	Group	

Selected (1)

Reset

 pod31-vpn-users 

Select

12. Click the Next button to advance to “Review + Create”. Create the user.

Create new user ...

Create a new internal user in your organization

Basics Properties Assignments Review + create

Basics

User principal name	Pod99-vpn-user@zer0k.onmicrosoft.com	
Display name	Pod99-vpn-user	
Mail nickname	Pod99-vpn-user	
Password	CLUS2023Party!	
Account enabled	Yes	

Properties

User type	Member
-----------	--------

Assignments

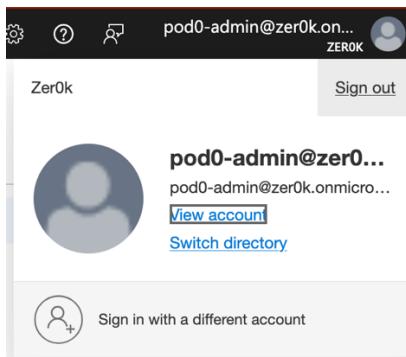
Administrative units	
Groups	pod31-vpn-users
Roles	

Create

< Previous

Next >

13. Entra ID requires the password to be reset before authentication can be done to that user. This cannot be done through the VPN auth so click your username at the top right of the screen and “sign out”



14. Navigate back to <https://entra.microsoft.com>, choose “User another account”, and sign in with the user you just created (the @zer0k.onmicrosoft.com suffix is required).



Sign in

to continue to Microsoft Entra

pod0-vpn-user@zer0k.onmicrosoft.com



No account? [Create one!](#)

[Can't access your account?](#)

Back

Next

-
15. You will be prompted to reset your password. Use the **Updated password provided by your proctor** Password as your new password.



pod99-vpn-user@zer0k.onmicrosoft.com

Update your password

You need to update your password because this is the first time you are signing in, or because your password has expired.

••••••••••••••

••••••••••••••

••••••••••|

Sign in

16. When asked to configure the Microsoft Authenticator app, click “Ask Later”



pod99-vpn-user@zer0k.onmicrosoft.com

Action Required

Your organization requires additional security information. Follow the prompts to download and set up the Microsoft Authenticator app.

[Use a different account](#)

[Learn more about the Microsoft Authenticator app](#)

You have 14 days until this is required.

Ask later

Next

17. If Entra ID does not allow you to delay the MFA configuration, follow the prompts with the authenticator app used when logging into AWS to set up an additional account for Entra ID. Use the “I want to use a different

authenticator app” if you used Google Authenticator or another app of your choice.

Microsoft Authenticator



Start by getting the app

On your phone, install the Microsoft Authenticator app. [Download now](#)

After you install the Microsoft Authenticator app on your device, choose "Next".

[I want to use a different authenticator app](#)

[Next](#)

[I want to set up a different method](#)

Keep your account secure

Authenticator app

Scan the QR code

Use the authenticator app to scan the QR code. This will connect your authenticator app with your account.

After you scan the QR code, choose "Next".



[Can't scan image?](#)

[Back](#)

[Next](#)

18. Once you successfully log in, switch back to the pod admin user.



Sign in with a different account

Step 2: Configure application integration in Entra ID

1. Using the left navigation bar brows to Identity -> Applications -> App registrations and choose “New registration”

2. Name the app **podX-ise-integration** where X is your assigned pod number, and leave the Support account types as “Accounts in this organizational directory only”. This application will not need a redirect URI.

*** Name**
The user-facing display name for this application (this can be changed later).

Supported account types
Who can use this application or access this API?
 Accounts in this organizational directory only (ZeroK only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
 Personal Microsoft accounts only
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

3. Click Register.
4. On the overview screen now presented, copy out the “Essentials” displayed at the top of the page. You will need the client ID and tenant ID later.

5. On the left sub-menu bar, select “Manage -> Certificates & secrets”. This is the OAuth secret key that will be used to integrate ISE with Entra ID. Click “New Client Secret” and enter a description of “podX-ISE” where X is your pod number. Click Add at the bottom of the dialog box.

Add a client secret

Description	pod31-ise
Expires	Recommended: 180 days (6 months)

6. Copy the secret value to a notepad before leaving the page!

Warning: Common mistake area!

This value cannot be viewed again after leaving the page. If you forget this step, delete the client secret and create a new one. It is **highly recommended** that you label the ID and value separately in the notepad document.

Description	Expires	Value	Set ID
Secret for ISE Integration	12/3/2022	Stm8Q~b8ft_lul29HAdBEVZbtnJXrLDRo4 ...	a4911819-5d55-4bad-9812-1d537f2d19de

7. Configure the application for ROPC by going to the Authentication tab on the left sub-menu. ISE does not use a URI based redirect flow, therefore scroll down to Advanced Settings and set the “Allow public client flows” setting to Yes. Click Save.

Advanced settings

Allow public client flows ⓘ

Enable the following mobile and desktop flows:

Yes

No

- App collects plaintext password (Resource Owner Password Credential Flow) [Learn more ↗](#)
- No keyboard (Device Code Flow) [Learn more ↗](#)
- SSO for domain-joined Windows (Windows Integrated Auth Flow) [Learn more ↗](#)

8. On the left sub-menu, click Token Configuration
9. Under Token Configuration on the top menu, add a groups claim so that you can later create ISE policies that refer to groups in Entra ID. The groups claim should allow access to the following group types:

- Security Groups
- Directory Roles
- All Groups

Click Add.

Select group types to include in Access, ID, and SAML tokens.

- Security groups
- Directory roles
- All groups (includes distribution lists but not groups assigned to the application)
- Groups assigned to the application

10. Add API permission for the groups graph API. In the left menu select “API permissions” then click Add a permission. In the window that pops up choose “Microsoft Graph” then “Application Permissions”.
11. Search for “group” then under the group drawer, select Group.Read.All, notice that Admin consent required is shown as Yes.
12. Select Add Permissions at the bottom of the screen.

Select permissions

group

Permission

Admin consent required

➤ Calls

▽ Group (1)

Group.Create ⓘ
Create groups

Yes

Group.Read.All ⓘ
Read all groups

Yes

Group.ReadWrite.All ⓘ
Read and write all groups

Yes

13. Before moving on ask your lab proctor to provide consent for your new

API Permission.

14. Open the overview page for the application, that information will be needed in the next steps.

Step 3: Enable the ROPC feature in ISE

1. Go to Administration -> Identity Management -> Settings -> External Identity Sources Settings Store Settings and ensure the REST ID Store is enabled. In previous versions of ISE this was disabled by default, so this is a verification step.

The screenshot shows the 'External Identity Sources Settings' page in Cisco ISE. The 'Settings' tab is selected. On the left, there's a sidebar with options like 'User Custom Attributes', 'User Authentication Settings', 'Endpoint Purge', 'Endpoint Custom Attributes', and 'External Identity Sources Settings'. The 'External Identity Sources Settings' option is highlighted with a blue border. The main content area has a heading 'External Identity Sources Settings' followed by 'REST ID Store'. It contains a note: 'To allow integration of REST identity stores with Cisco ISE, click the radio button below. It takes a few minutes to enable the REST ID Store settings. After the settings are enabled, you can add REST ID stores to Cisco ISE in the [External Identity Source](#) page.' Below this is a blue toggle switch labeled 'REST ID Store' which is turned on. Further down is a section for 'Multi-Factor Authentication' with a grey toggle switch labeled 'MFA' which is turned off.

Step 4: Register ISE with Entra ID

1. In ISE, browse to Administration -> Identity Management -> External Identity Sources from the hamburger menu icon at the top left of the ISE UI.
2. From the REST option, add a new entry.
3. Name it **zer0k_entraid** and fill in the information obtained from the overview page under the application created in Step 2 from Entra ID. The client secret (Secret Value) should have been saved to a notepad. The username suffix for this lab is **@zer0k.onmicrosoft.com**.

^ Essentials

Display name	: pod10-ise-integration
Application (client) ID	: 6846a86b-36b5-4e7c-bc96-7e98b5dda0a8
Object ID	: 89de3286-75f2-46f9-9d3c-318164d42ad0
Directory (tenant) ID	: 6d92df37-cb73-4152-bfd5-54c7828c09c4
Supported account types	: My organization only

Note: You'll need BOTH the secret value and the Client ID from two separate screens in this step. Both should be contained in a notepad document. In the Entra ID overview, the Client ID is the "Application

(client) ID” and the tenant ID is the “Directory (tenant) ID”.

The screenshot shows the 'General' tab selected in a configuration interface. The form fields include:

- Name***: zer0k_entraid
- Description**: (empty text area)
- REST Identity Provider***: Microsoft Entra ID (with a dropdown arrow)
- Client ID***: 398b-e949-4b11-aaaa-7473ebfc9e5b
- Client Secret***: (redacted)
- Tenant ID***: df37-cb73-4152-bfd5-54c7828c09c4
- Username Suffix**: @zer0k.onmicrosoft.com (with a copy icon)

At the bottom is a blue button labeled "Test connection".

4. Test the connection, if it is successful go to the groups tab and select the **podX-vpn-users** group you created earlier.
5. In order to use groups in policy they need to be selected here in the ROPC connection. Click Add. In the “Select Groups” screen, click “Retrieve Groups”

Select Groups

Please click on "Retrieve Groups" button to load groups. And select needed groups from Directory. 5000 is the maximum that can be retrieved in one go. Set filters to see other groups. Groups are retrieved when name "starts with" user input.

The screenshot shows a search interface for selecting groups. At the top, there is a 'Name Filter *' input field and a 'Retrieve Groups' button. Below that is a pagination control with 'Rows/Page' set to 8, and buttons for navigating between pages (1, 2, 3, etc.). A 'Go' button and a message indicating '8 Total Rows' are also present. On the right, there is a 'Filter' dropdown. The main area displays a list of groups, each with a checkbox and a name. The names listed are: LTRSEC-2000-Admins, Administrators, LTRSEC-2012-Admins, LTRSEC2012_IT, LTRSEC2012_MKT, pod4-vpn-users, LTRSEC2012_SALES, and Pod Admins. At the bottom right are 'Close' and 'Save' buttons.

6. Select the group created in the previous step, with the format “**podX-vpn-users**” where X is your pod number.
7. Save the connection.

Step 5: Configure ISE Policy for VPN Authentication

1. Go to the ISE menu -> Policy -> Policy Sets and select the “+” in the top left to add a new policy set.
2. Give the policy set a name of “PodX-VPN-Policy” where X is your pod number, and select “Default Network Access’ under Allowed Protocols / Server Sequence.

The screenshot shows a table titled 'Policy Sets'. It has columns for Status, Policy Set Name, Description, Conditions, and Allowed Protocols / Server Sequence. A search bar is at the top. One row is visible, showing a green checkmark icon, the name 'Pod4-VPN-Policy', and a '+' sign under 'Conditions'. Under 'Allowed Protocols / Server Sequence', it says 'Default Network Access' with a delete ('x') and edit ('+') icon.

Status	Policy Set Name	Description	Conditions	Allowed Protocols / Server Sequence
<input type="text"/> Search				
	Pod4-VPN-Policy		+	Default Network Access (x) (e) +

3. Click the “+” under conditions which will launch the conditions studio.
4. The network device and network device group were created by the initial configuration ansible scripts. They will be used here to match authentication from the VPN headend.
- Use the “All Dictionaries” drop down to filter for “DEVICE”.

Editor

DEVICE-Device Type|

Equal | Set to

Select attribute for condition

Dictionary	Attribute	ID	Info
DEVICE	Device Type	(i)	
DEVICE	IPSEC	(i)	
DEVICE	Location	(i)	

- Choose Device: Device Type and choose the “All Device Types” device type with the equals operator. The save button will allow you to save the condition for later, in this case choose “Use” at the bottom of the page to use it now.

Editor

DEVICE-Device Type

Equal | All Device Types

Set to 'Is not'

NEW | AND | OR

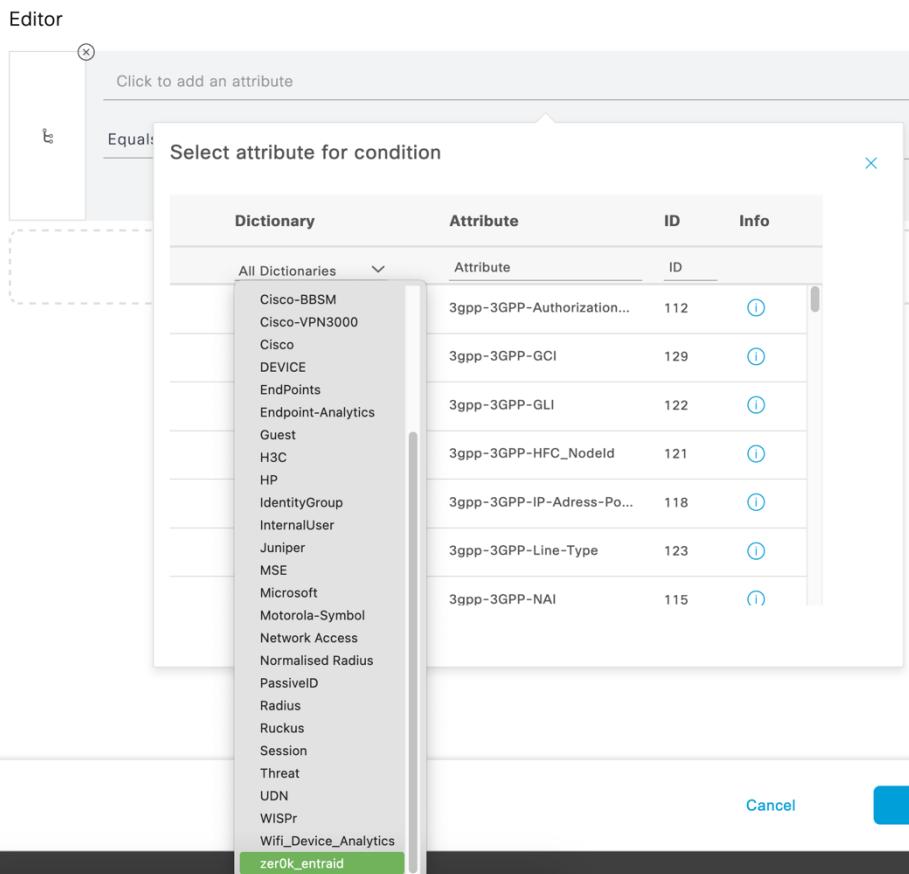
- The Conditions studio should close after hitting “Use”.
- Save the policy sets.
- Click the “View” arrow on the right side of the policy set line to enter the policy set configuration.
- Expand “Authentication Policy” and set the default entry to what you named your ROPC connection (zer0k_entraid).

zer0k_azure

> Options

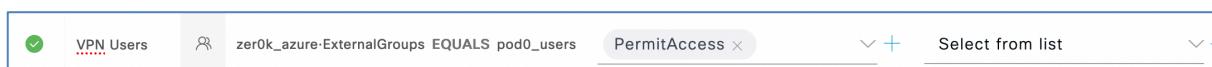
- Expand “Authorization Policy” and click the “+” on the top of the authorization policy list to add a new entry, this entry will be used to match the group created on Entra ID and permit access to the VPN.
- Name the rule “VPN Authorization” then click the + under conditions to launch the condition studio and create a new condition.
- Click “click to add an attribute” and use the “All Dictionaries” drop down to filter for your “zer0k_entraid” external dictionary.

- Choose your ROPC connection name: External Groups and then select the group you have added your user to in Entra ID (podX-vpn-users).



- Leave the default as equals and choose use at the bottom of the screen to use the newly created condition.

10. Back on the Policy Set page set the Result to “Permit Access” then save.

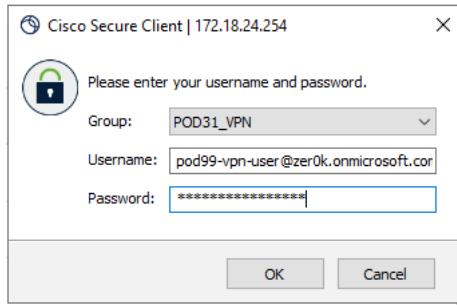


Task 7: Test VPN Authentication

Step 1: Test VPN Authentication

Note: In this section of the lab you will be logging out of the lab VPN, while you are logged out any lab SSH or GUI connections will be unavailable. When reconnecting you may need to re-add environment variables to your SSH connections.

- Log out of the current VPN connection from Secure Client on the lab laptop.
- Reconnect to the same VPN address 3.131.72.23 but this time use your PODX_VPN group and podX-vpn-user entraid user. Use the password ending in \$



3. After successfully connecting you should be able to ping the super secret linux server at 172.18.16.10.
4. On ISE, browse to the ISE Menu -> Operations -> Radius -> Live Logs. Here you can see the authentication attempts and successful authentications to the VPN.

Status	Details	Repea...	Identity	Endpoint ID	Endpoint...	Authenti...	Authoriz...	Authoriz...
			Identity	Endpoint ID	Endpoint Pr	Authenticat	Authorizatio	Authorizatio

5. Click the details icon on one of the successful authentications where you can see the attributes available on the authentication and the steps ISE took to authenticate the user.

Overview		Steps
Event	5200 Authentication succeeded	Step ID Description Latency (ms)
Username	pod99-vpn-user@zer0k.onmicrosoft.com	11001 Received RADIUS Access-Request 0
Endpoint Id	02:FE:3C:44:20:2F ⊕	11017 RADIUS created a new session 0
Endpoint Profile		15049 Evaluating Policy Group 0
Authentication Policy	Pod31-vpn-policy >> Default	15008 Evaluating Service Selection Policy 0
Authorization Policy	Pod31-vpn-policy >> VPN Authorization	15048 Queried PIP ~ DEVICE.Device Type 4
Authorization Result	PermitAccess	15041 Evaluating Identity Policy 7
		15013 Selected Identity Source ~ zer0k_Azure 2
		25103 Perform plain text password authentication in external REST ID store server ~ zer0k_Azure 0
		25100 Connecting to external REST ID store server ~ zer0k_Azure 36
		25101 Successfully connected to external REST ID store server ~ zer0k_Azure 240
		25104 Plain text password authentication in external REST ID store server succeeded ~ zer0k_Azure 0
		25107 REST ID store server respond with groups ~ zer0k_Azure 0
		25110 User groups inserted to session cache ~ zer0k_Azure 1
		22037 Authentication Passed 0
		24715 ISE has not confirmed locally previous successful machine authentication for user in Active Directory 0
		15036 Evaluating Authorization Policy 0
		24209 Looking up Endpoint in Internal Endpoints IDStore ~ pod99-vpn-user@zer0k.onmicrosoft.com 1
		24217 The host is not found in the internal endpoints identity store 16
		15016 Selected Authorization Profile ~ PermitAccess 9

Task 8: (Bonus) Deploy Secondary ISE through AWS Marketplace from CloudFormation Template

Step 1: Subscribe to ISE in AWS Marketplace

1. Navigate to <https://zer0k.signin.aws.amazon.com/console> Use Account ID zer0k if not already populated. Login with the username assigned to your pod in table 1-1 above (**podX-awsadmin**). The password for the

account is provided by your proctor.



Sign in as IAM user

Account ID (12 digits) or account alias

zer0k

IAM user name

pod13-awsadmin

Password

••••••••••••••|

Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

2. In the AWS console search for AWS Marketplace Subscriptions.
3. Go to Discover Products on the left menu.
4. Search for ISE and click on Cisco Identity Services Engine (ISE).

Search AWS Marketplace products

cisco identity services engine

cisco identity services engine (5 results) showing 1 - 5



[Cisco Identity Services Engine \(ISE\)](#) 

By [Cisco Systems, Inc.](#)  | Ver 3.3.0
[10 external reviews](#) 

Cisco Identity Services Engine (ISE) on AWS enables Network Access Control (NAC) service workloads on AWS, you can unify the policy management of your...

5. Click on “Continue to Subscribe” on the top right

Cisco Identity Services Engine (ISE)

By: [Cisco Systems, Inc.](#) Latest Version: 3.3.0

Cisco ISE on AWS provides secure network access control for IoT, BYOD, and corporate owned endpoints. Cisco ISE enables you to easily segment network access for employees, contractors, [+ Show more](#)

Linux/Unix ☆☆☆☆☆ 0 AWS reviews | 10 external reviews [i](#)

BYOL

Continue to Subscribe

Request a private offer

Save to List

Typical Total Price
\$0.68/hr

Total pricing per instance for services hosted on c5.4xlarge in US East (N. Virginia). [View Details](#)

6. Click on Continue to Configuration.
7. Under fulfilment option, select “CloudFormation Template”.

Cisco Identity Services Engine (ISE)

< Product Detail [Subscribe](#) [Configure](#)

Configure this software

Choose a fulfillment option and software version to launch this software.

Fulfillment option

Select a fulfillment option

- Select a fulfillment option
- Amazon Machine Image
- CloudFormation Template**

Amazon Machine Image
Deploy a vendor-provided Amazon Machine Image (AMI) on Amazon EC2

CloudFormation Template
Deploy a complete solution configuration using a CloudFormation template

8. A new drop down will be displayed, select Cisco Identity Services Engine (ISE).
9. 2 new drop downs will be presented, select 3.3.0 (July 09, 2023) and select the Region as US East (Ohio). If you don't use 3.3.0 you won't be able to join the node created by ansible earlier.
10. Click Continue to Launch on the top right.
11. Under Choose Action, select “Launch CloudFormation”, then select the Launch button.



Cisco Identity Services Engine (ISE)

< Product Detail Subscribe Configure [Launch](#)

Launch this software

Review the launch configuration details and follow the instructions to launch this software.

Configuration details

Fulfillment option	Cisco Identity Services Engine (ISE) Cisco Identity Services Engine (ISE) <i>running on c5.4xlarge</i>
Software version	3.3.0
Region	US East (Ohio)

[Usage instructions](#)

Choose Action

[Launch CloudFormation](#)

Choose this action to launch your configuration through the AWS CloudFormation console.

[Launch](#)

Step 2: CloudFormation Stack Creation

Marketplace will launch a new tab for CloudFormation with a link to the publicly available CloudFormation Template uploaded to S3.

1. Copy the Amazon S3 URL and open it in a new tab/browser and save it to your local machine. This can be used later directly in CloudFormation rather than going through the subscription as another option. This lab will not use this method unless initial Cloud Formation fails.

CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready

Use a sample template

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL
Provide an Amazon S3 URL to your template.

Upload a template file
Upload your template directly to the console.

Amazon S3 URL
`https://s3.amazonaws.com/awsmp-fulfillment-cf-templates-prod/bedef662-aba4-427e-b523-7c93cd50111c.fdc8e997-b533-4691-866e-58856e94aa8e.template`

Amazon S3 template URL

S3 URL: `https://s3.amazonaws.com/awsmp-fulfillment-cf-templates-prod/bedef662-aba4-427e-b523-7c93cd50111c.fdc8e997-b533-4691-866e-58856e94aa8e.template`

2. With the “Template is ready” option selected, and the Amazon S3 URL

still populated, click Next.

3. The template includes all the provisioning information needed to launch an ISE instance. Fill out all the fields:
 - a. Stack Name - Since this is a single instance set this to the hostname podX-ise2 where X is your pod number.
 - b. Hostname - podX-ise2 where X is your pod number.
 - c. Instance Key Pair - Select the key pair you created in Task 1 and have already saved to your local machine.
 - d. Management Security Group - ISEinAWS-podX where X is your assigned pod number. This was created in the Ansible ISE deployment steps.
 - e. Management Network - Select the subnet in the new VPC created in Task 3, it should be zer0k_podX_Private_Subnet where X is your assigned pod number.
 - f. Management Private IP - Given in the Table above as configured in Route53. It will be 10.X.0.6 where X is your assigned pod number.
 - g. Time Zone - Etc/UTC
 - h. Instance Type - C5.4xlarge
 - i. EBS Encryption - false
 - j. Volume Size - 600
 - k. DNS Domain - zer0k.org
 - l. Name Server - 10.X.0.2 where X is the number of your assigned pod.
 - m. NTP Server - 169.254.169.123
 - n. ERS - yes
 - o. OpenAPI - yes
 - p. pxGrid - no
 - q. pxGrid Cloud - no
 - r. Password - **Provided by your proctor**

Step 1
[Create stack](#)

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review pod31-ise2

Specify stack details

Provide a stack name

Stack name

pod31-ise2

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you

Instance Details

Hostname

Enter the hostname. This field only supports alphanumeric characters and hyphen (-). Th

pod31-ise2

Instance Key Pair

To access the Cisco ISE instance via SSH, choose the PEM file that you created in AWS for

pod31-keypair-20240126

Management Security Group

Choose the Security Group to attach to the Cisco ISE interface. Create a Security Group in

Select List<AWS::EC2::SecurityGroup::Id>

sg-076adef5f3e88a94b X

4. Click Next.
5. Under Tags add a Key “Name” with value “podX-ise2” where X is the number of your pod. CloudFormation is going to add an instance, EBS volume, and network interface that will all get named via this tag.
6. Under Tags, add a tag. Enter “project” under Key and “ISEinAWS-podX” under value, where X is your pod number.
7. Leave the rest of the options default though you are welcome to browse through them to see the options AWS provides. Click Next when you are done.
8. Review the information to make sure you entered everything correctly, then click “Submit” at the bottom of the page.
9. Under events you should see “CREATE_IN_PROGRESS”, refresh the table until you see podX-ise2 with “CREATE_COMPLETE”.

Timestamp	Logical ID	Status	Status reason
2022-05-09 16:44:06 UTC-0400	pod0-ise	CREATE_COMPLETE	-
2022-05-09 16:44:04 UTC-0400	IseEc2Instance	CREATE_COMPLETE	-
2022-05-09 16:43:57 UTC-0400	IseEc2Instance	CREATE_IN_PROGRESS	Resource creation Initiated
2022-05-09 16:43:55 UTC-0400	IseEc2Instance	CREATE_IN_PROGRESS	-
2022-05-09 16:43:49 UTC-0400	pod0-ise	CREATE_IN_PROGRESS	User Initiated

Task 9: (Bonus) Clean Up

Step 1: Terraform Destroy

1. If you are finished with your lab and have no further questions, Terraform tracks the status of the ISE node and configurations that were run. Navigate back to the Ubuntu machine.
2. Use the command **pwd** to verify you are currently under the Ansible directory:

```
(ISEonAWS) ubuntu@ip-172-18-25-2:~/LTRSEC-2000/Ansible$ pwd  
/home/ubuntu/LTRSEC-2000/Ansible
```

3. Change directory back to the Terraform directory. Use the command

```
cd ..  
cd Terraform
```

4. Enter command `terraform destroy`. Enter “yes” when prompted:

```
Do you really want to destroy all resources?  
Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.
```

5. Verify the last line of the output is:

```
Destroy complete! Resources: 11 destroyed.
```

6. Within your web browser, navigate back to your AWS instance.

Navigate to EC2 -> Instances.

7. Use the filter at the top of the screen to filter to your pod instances, entering “podX” where X is your pod number.
8. Verify your ISE node has been terminated.

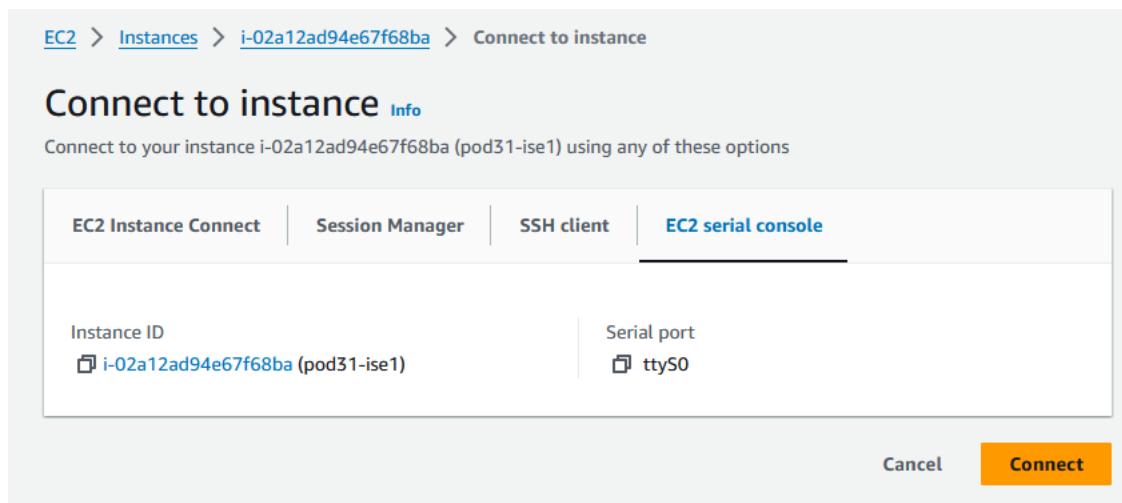
Instances (2) [Info](#)

<input type="text"/> Find Instance by attribute or tag (case-sensitive)		
<input type="button" value="pod2"/> X	Clear filters	
<input type="checkbox"/>	Name 🔗	Instance ID
<input type="checkbox"/>	pod2-ise1	i-053350f3ff9514710

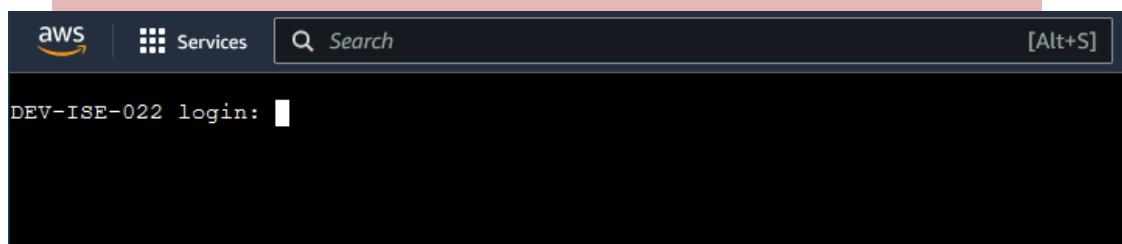
9. You Subnet, routing table, and transit gateway attachment all rely on your VPC being present. Navigate to VPC using the search bar, and “Your VPCs”. Verify your VPC, named “ISEinAWS-podX” where X is your pod number, is no longer present.

Appendix 1: Using the EC2 Serial Connect to Monitor ISE Deployment Status

1. If you would like to see the ongoing status of the ISE deployment you can connect through the virtual serial. In your AWS Console, click the instance name and click the “Connect” button on the top left.
2. Select EC2 Serial Console, and connect



Note: If your ISE host name is not in the format of podXX-ise upon initial connection, **do not issue any commands**. The ISE node is still programmatically deploying. Wait 5 minutes before hitting enter again to get the proper login prompt.



3. **Optional:** Run the following command:

```
podXX-ise/admin# show application status ise
```

ISE PROCESS NAME	STATE	PROCESS ID

Database Listener	running	32895
Database Server	running	85 PROCESSES
Application Server	not running	
...		
Application Server	initializing	49728
...		
Application Server	running	

4. On the CLI where the ssh command was run, you will be prompted to accept the ssh fingerprint of the ISE server. Enter yes.

5. Run the following command:

```
podXX-ise/admin# show application status ise
```

ISE PROCESS NAME	STATE	PROCESS ID
<hr/>		
<hr/>		
Database Listener	running	32895
Database Server	running	85 PROCESSES
Application Server	not running	
...		
Application Server	initializing	49728
...		
Application Server	running	

6. Once the Application Server shows as “running” move on to the next step.

**Thank you for your interest in the lab and joining us at Cisco Live!
Please remember to do your session survey, it helps us continue to offer sessions like this!**