Seneca

Lab 1: Setting up the environment

-Pratham Sapra

The goal of this Lab is to create a functional environment that we will use throughout this course.

We will cover the following:

1. Access to AWS Management Console via AWS Academy
2. Explore AWS Credentials and our permissions in AWS Academy environment
3. Creating Amazon Cloud9 IDE
4. Provisioning Amazon EC2 with AWS CLI
5. Securely connect to Amazon EC2 with SSH and AWS Academy provided SSH key pair
6. Optional: Configuring your local IDE to use temporary AWS Academy credentials

Task 1: Accessing AWS Management Console via AWS Academy

AWS Academy Lab environment is provisioned for you for the duration of the Deployment Automaton and Operational Security course and will be available until April 30th.

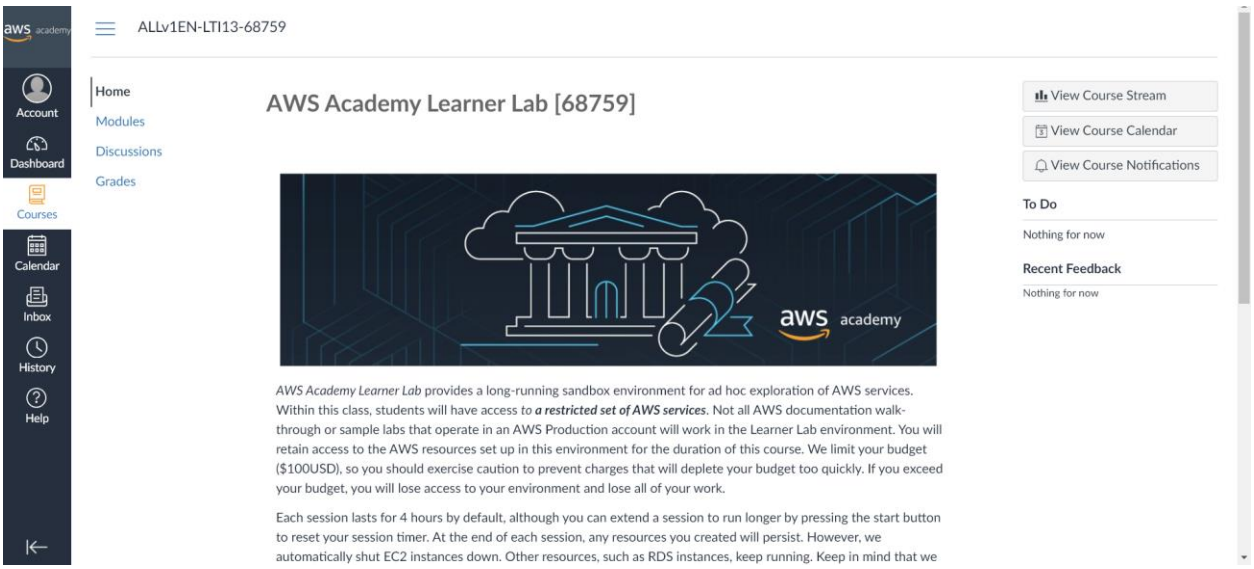
Please note, that the environment is restricted to about 40 out of more than 100 AWS services and your credit limit for the whole term is 100USD.

This means that at the end of your work, you either must clean up all the provisioned infrastructure manually or to press **"Reset"** button which will destroy all the provisioned resources.

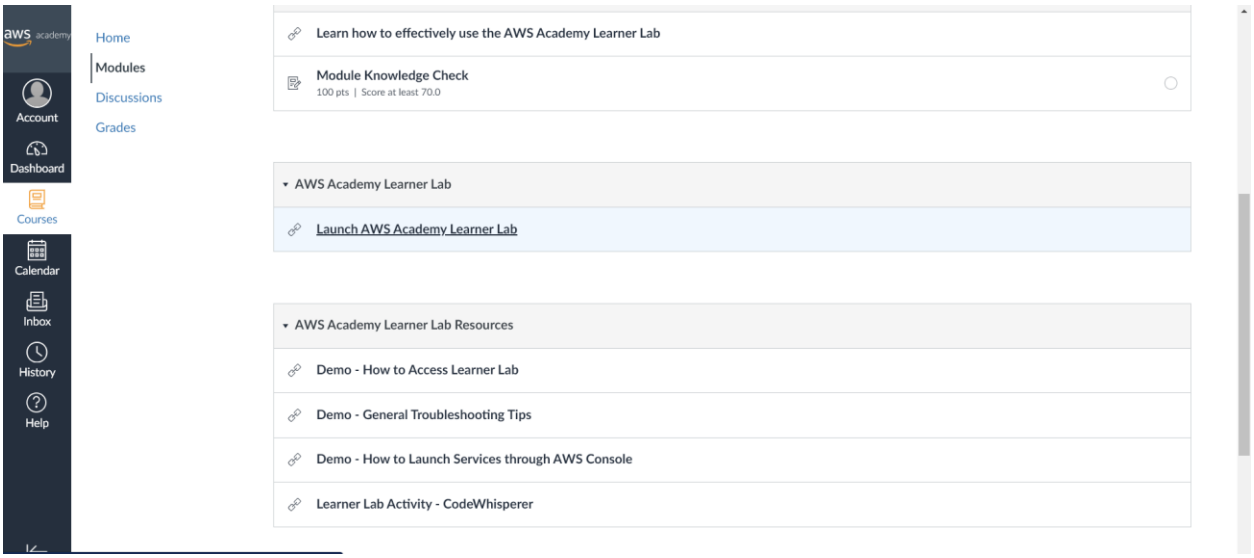
Important: The 100USD credit is about 5 times the amount needed to complete the course, and it cannot be refilled. Please keep a close attention to your remaining credits.

To work with the AWS environment provisioned by AWS Academy, please follow the instructions below:

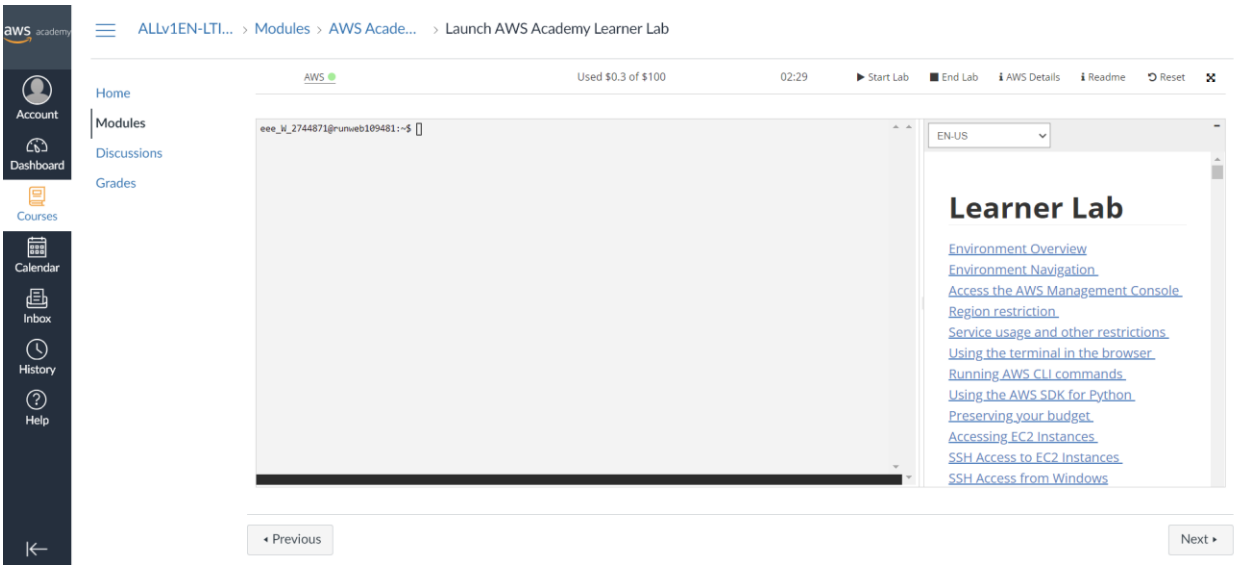
1. Select Modules from the course menu.



2. Click through the Learner Lab link

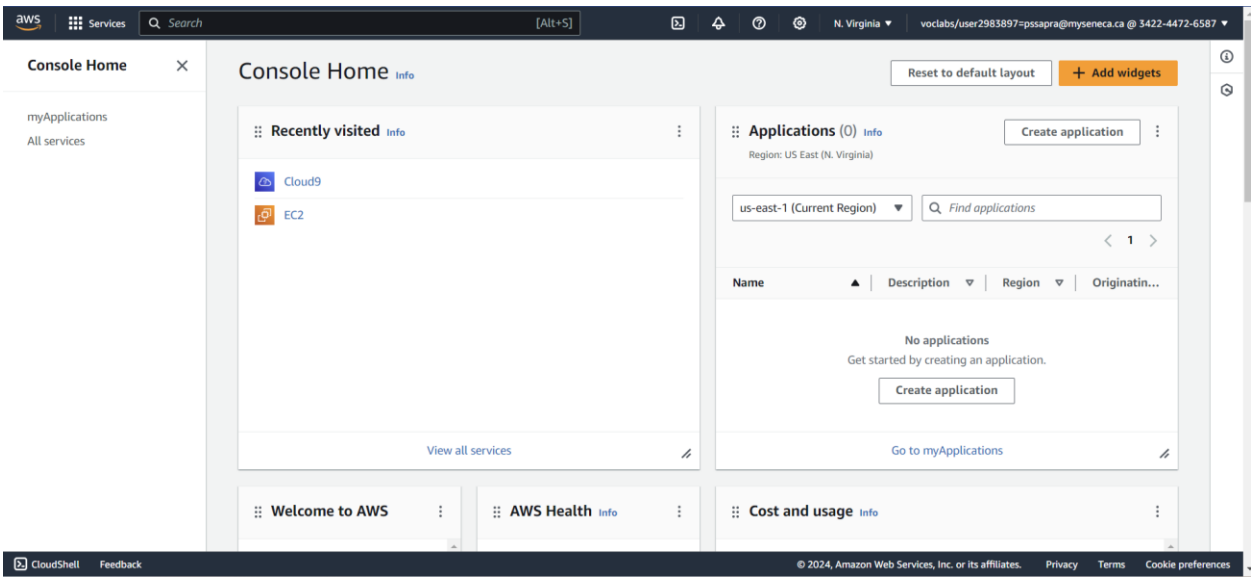


3. Note: If this is your first time you may have to refresh the web page to see and accept the terms and conditions. Start the lab by pressing the “Start Lab” button.

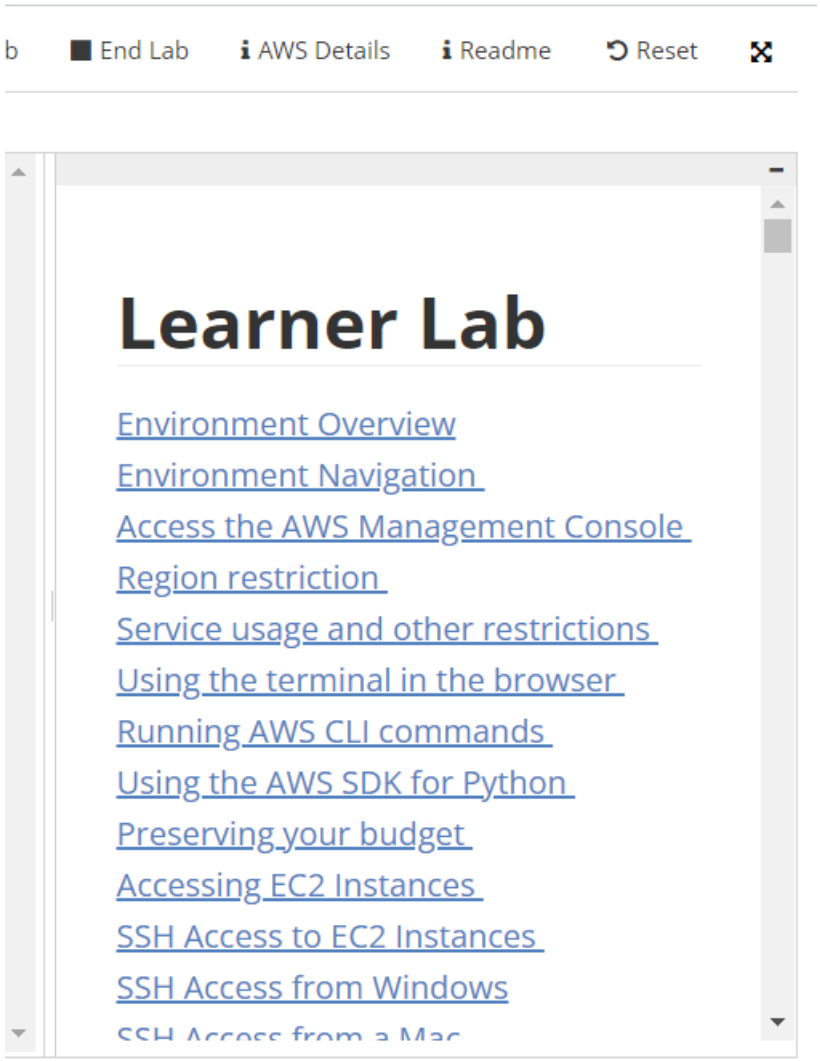


4. Once the lab is started, press the AWS button to view the AWS Management Console.

5. You will be transparently logged into a temporary AWS Account that you can use as long as the lab session timer is active (4 hours).

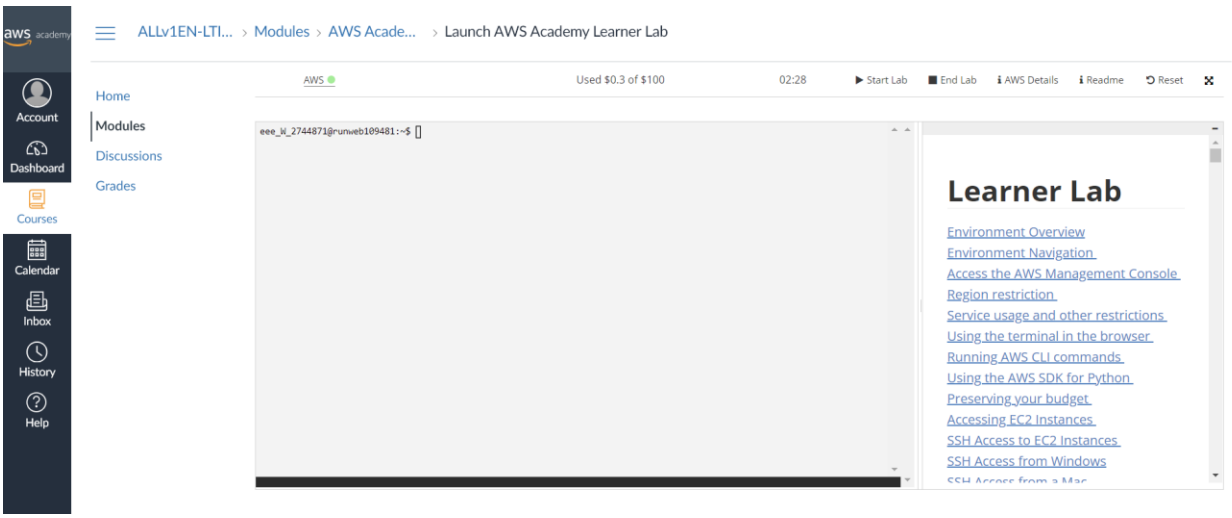


6. You can return to the LMS to see the lab instructions by using the Readme button.



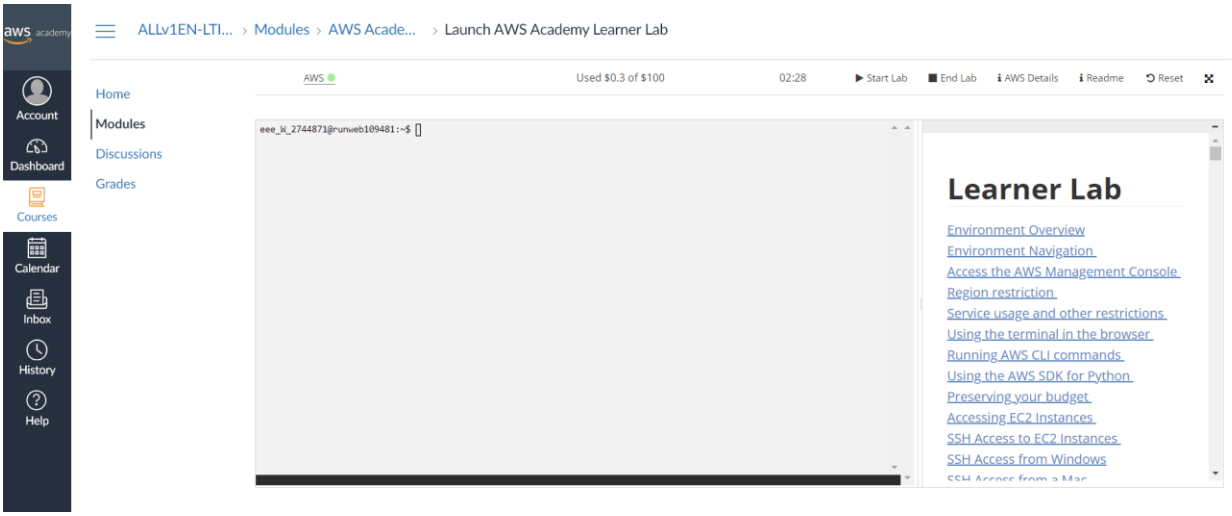
7. You can monitor your spending using the information at the top of the lab. Note: This information is provided by AWS Budgets and may be delayed by up to 8 hours. This is an approximate view of your spending.

8. You can monitor your remaining session time at the top of the lab. Note: If you are actively working and you need more time, you can reset your session timer by pressing “Start Lab” again.



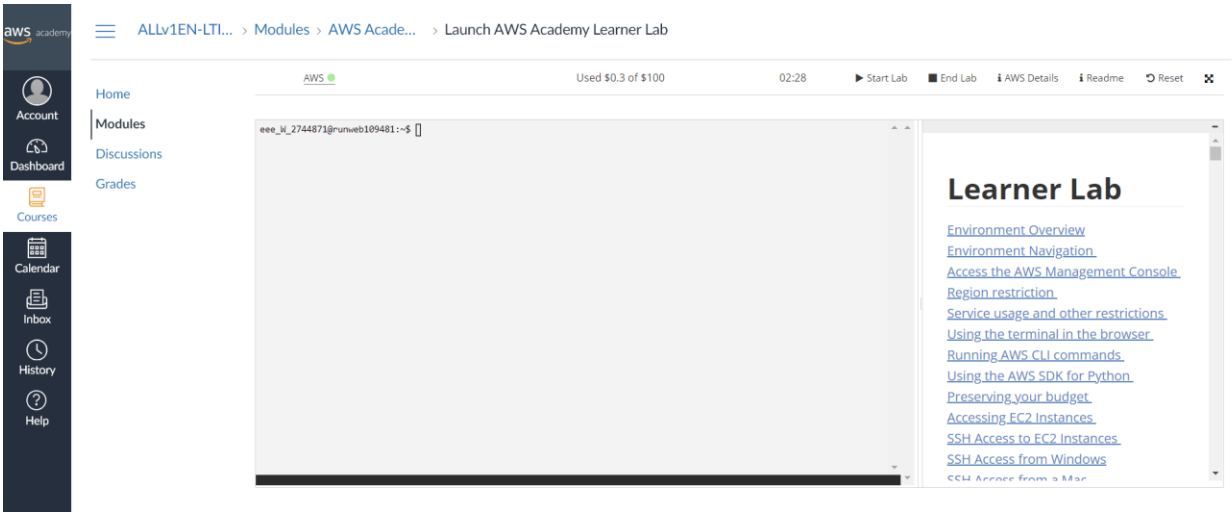
9. If you need to wipe out all of the resources that you have configured and start with a fresh AWS Account, press the Reset Button.

   NOTE: ALL WORK WILL BE LOST AND CANNOT BE RECOVERED.



10. When you are finished with this session, press the “End Lab” button. This will stop any running EC2 instances. When you return to your lab and restart it – the

EC2 instance will restart and any other resources you've configured will still be available.



Task 2: Explore AWS Credentials and our permissions in AWS Academy environment

To work with AWS APIs we need to have be authenticated and authorized. When working with AWS CLI, SDK and other deployment tools, our users need to have programmatic level access and our credentials should be available in our working environment.

Read more about different ways of setting AWS credentials in your environment.

1. From your AWS Academy environment, go to AWS Academy terminal window through steps 1 to 3 in Task 1. In the terminal window, type:
   `vi ~/.aws/credentials`
   `vi ~/.aws/config`

1

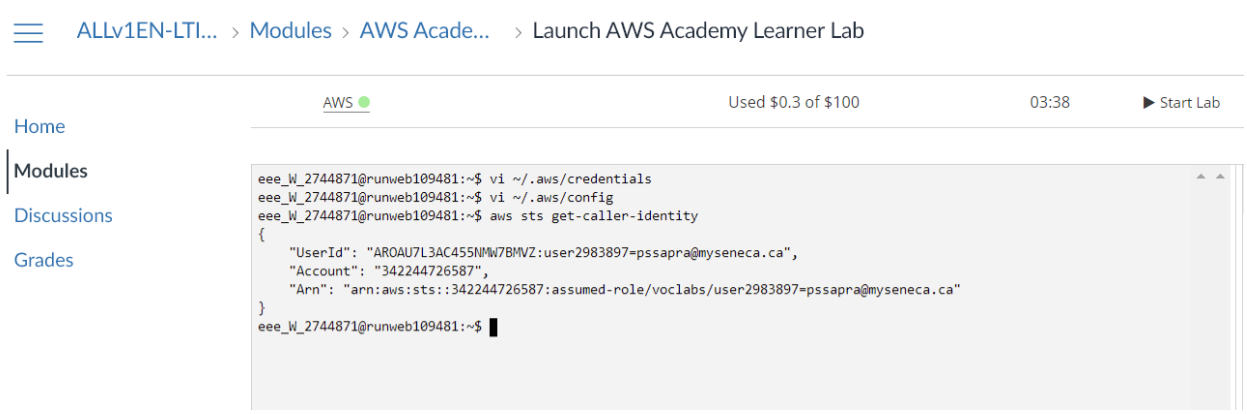
[1] Temporary credentials provisioned by AWS Academy for the duration of the Lab session

Note: The credentials below are temporary and are invalidated when I click "End Lab". Please do not try using these credentials.

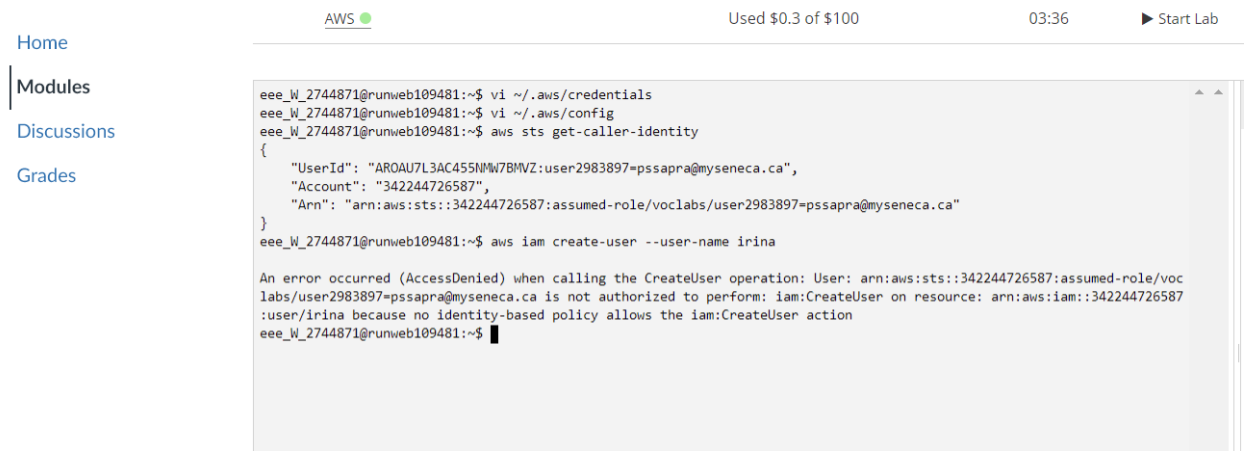
2. Read README file on the left side panel. README clarifies limitations of the AWS Academy environment and provides the basic set of instructions.

3. Try running AWS CLI command to see if you have permissions
`aws sts get-caller-identity`

*



4. Try running the command below. What does the error tell us?
`aws iam create-user --user-name irina`

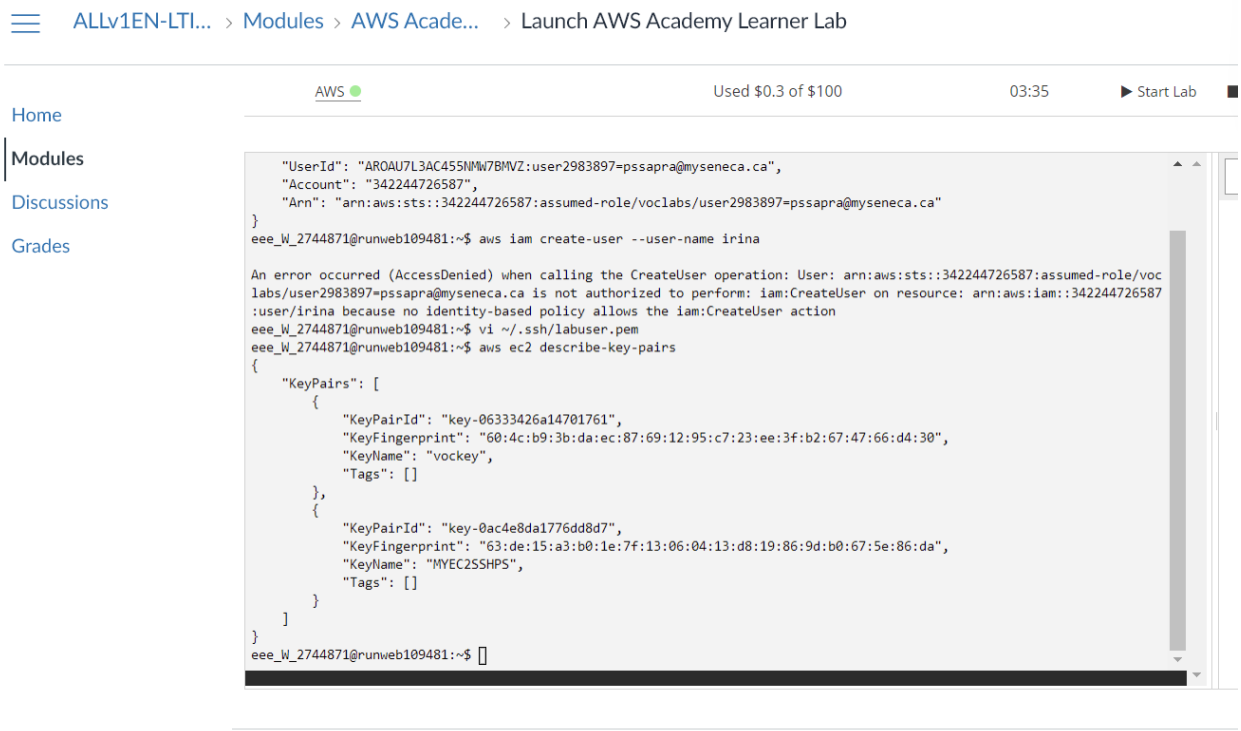


Other resources blocked by AWS Academy include EMR and EKS clusters, ECS and everything related to IAM.

5. AWS Academy pre-created an SSH key pair in our AWS environment. The private key of the key pair is located in you AWS Academy environment in the location below:
   `vi ~/.ssh/labuser.pem`

6. To see the SSH keypair provisioned in our environment, run the command below, Note the name of the key. We will use it to provision EC2 instance.

   `aws ec2 describe-key-pairs`

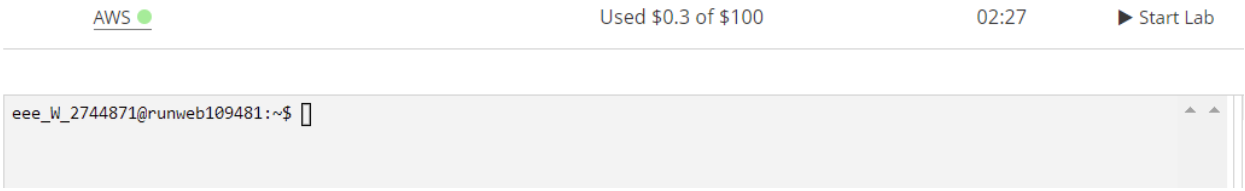


Task 3: Creating Amazon Cloud9 IDE

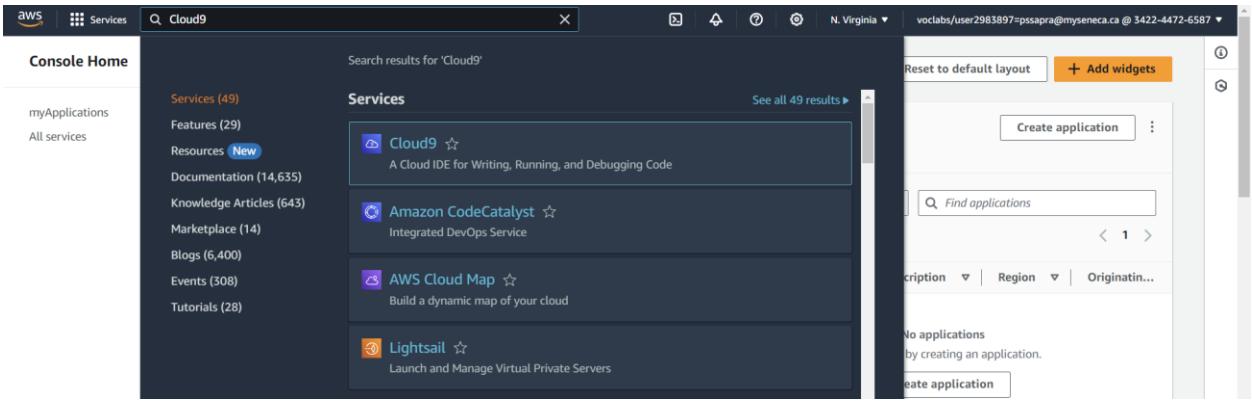
We will use Amazon Cloud9 as our IDE throughout the course. The Cloud9 environment is pre-provisioned with AWS CLI, AWS credentials, Terraform CLI, python, git and other tools that we will use throughout the course.

Please note that you can use other IDE, e.g. Visual Source code, but you will have to renew your credentials manually every few hours. We will discuss the local environment setup in Task 6.

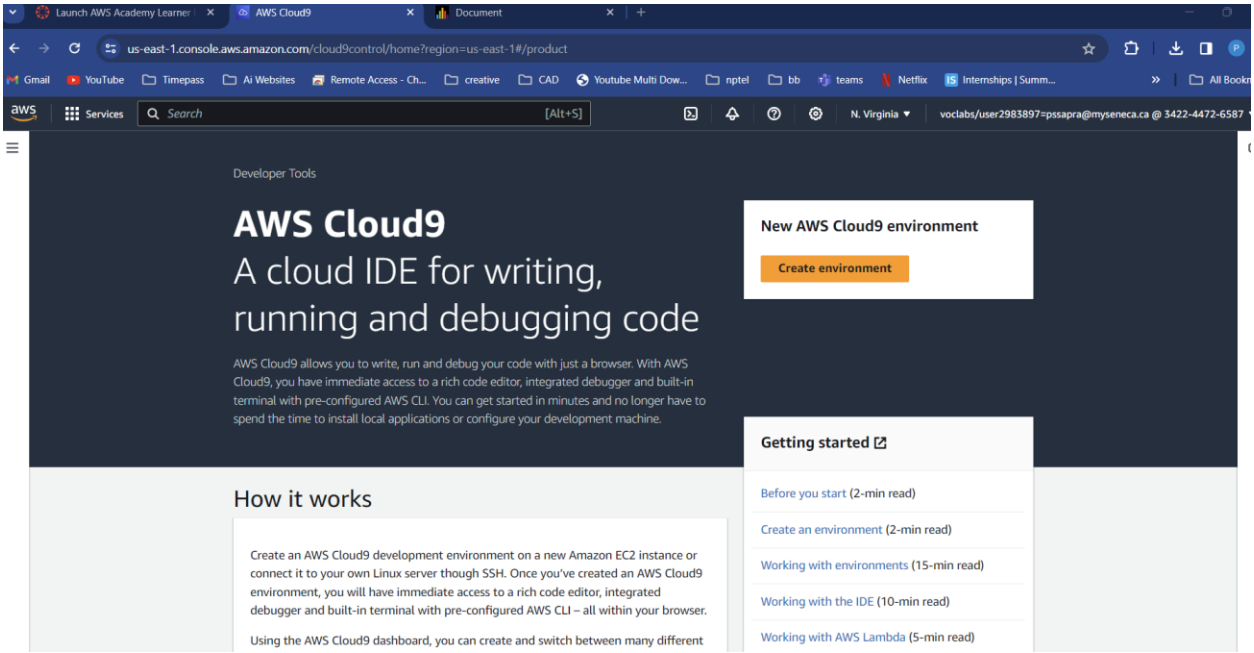
1. Open AWS Management Console by clicking on AWS with a green dot



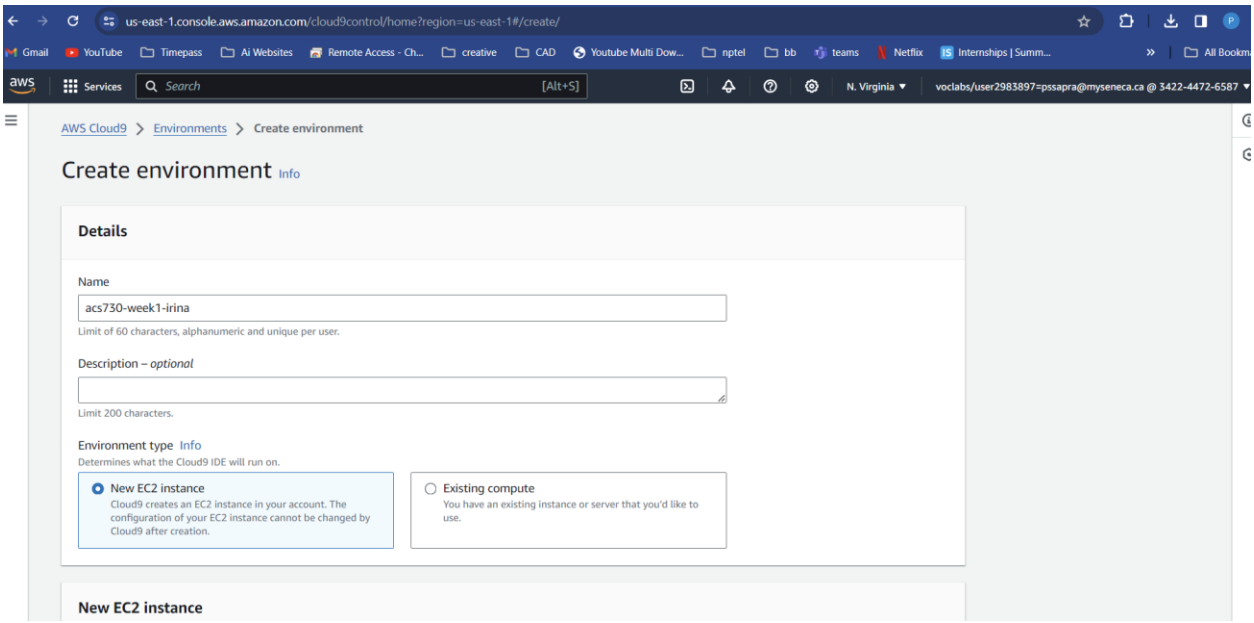
2. Type Cloud9 in the search bar and click on Cloud9



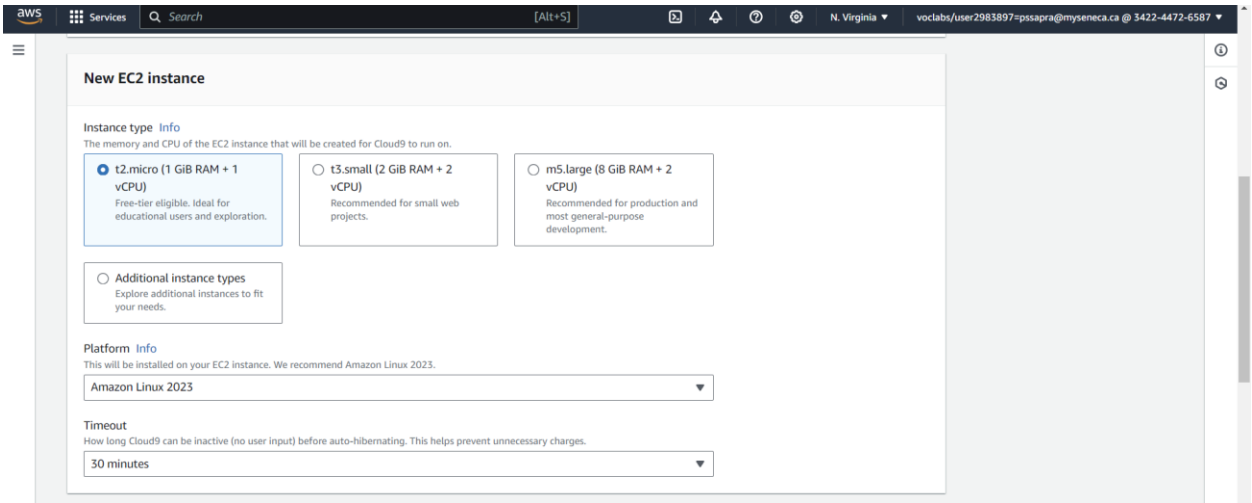
3. Click on Create environment button.

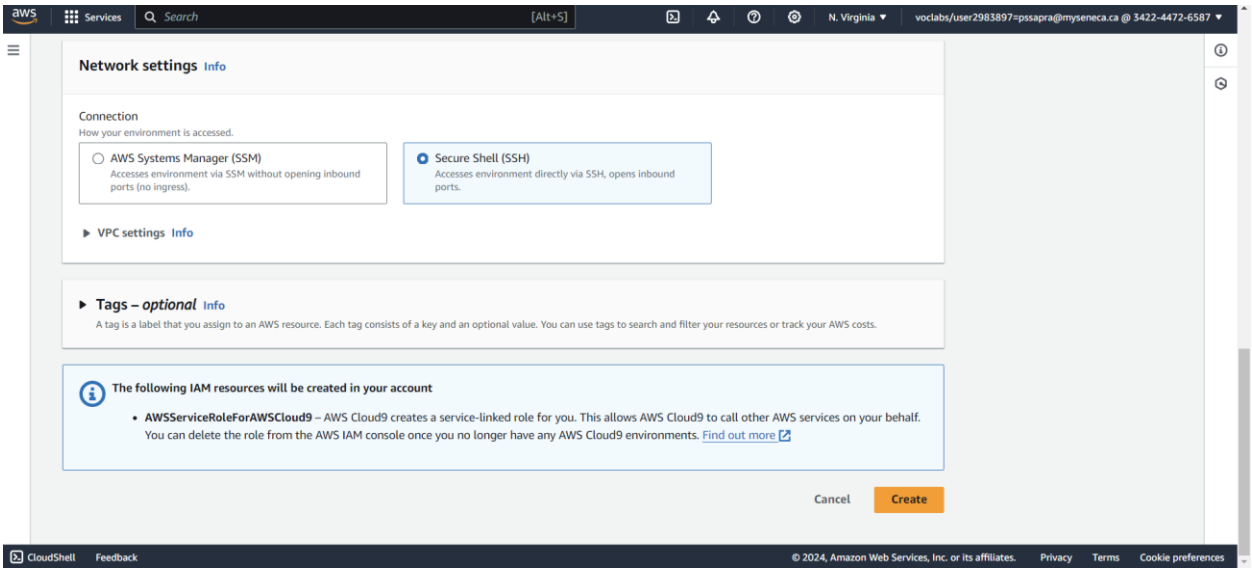


4. Name the environment and click on Next step



5. Leave all the defaults and click on the Next step again, then click Create environment



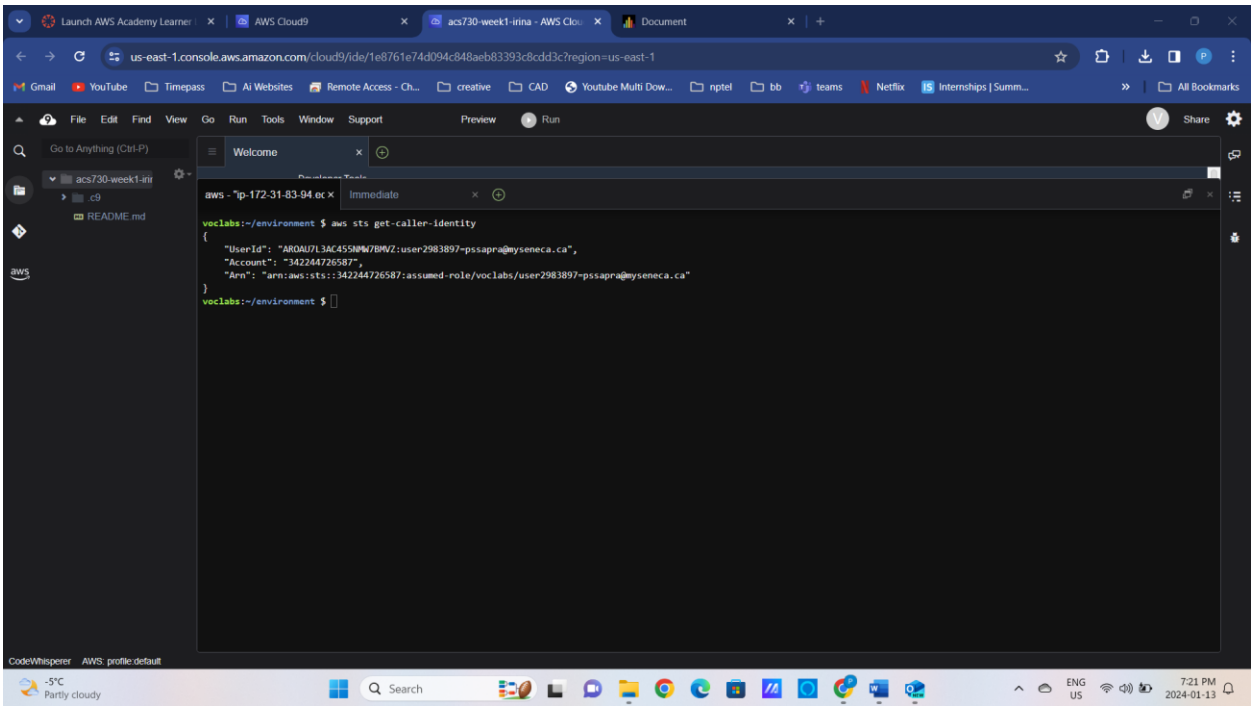


6.

The Cloud9 environment will take a couple of minutes to provision.

Try running AWS CLI command in the Cloud9 Terminal. Do you see any differences compared to Task 2, section 3?

```
aws sts get-caller-identity
```



7. Run `aws --v`, `terraform version` and `git version` to ensure the tools are installed.
   Install Terraform from the steps given in the link:

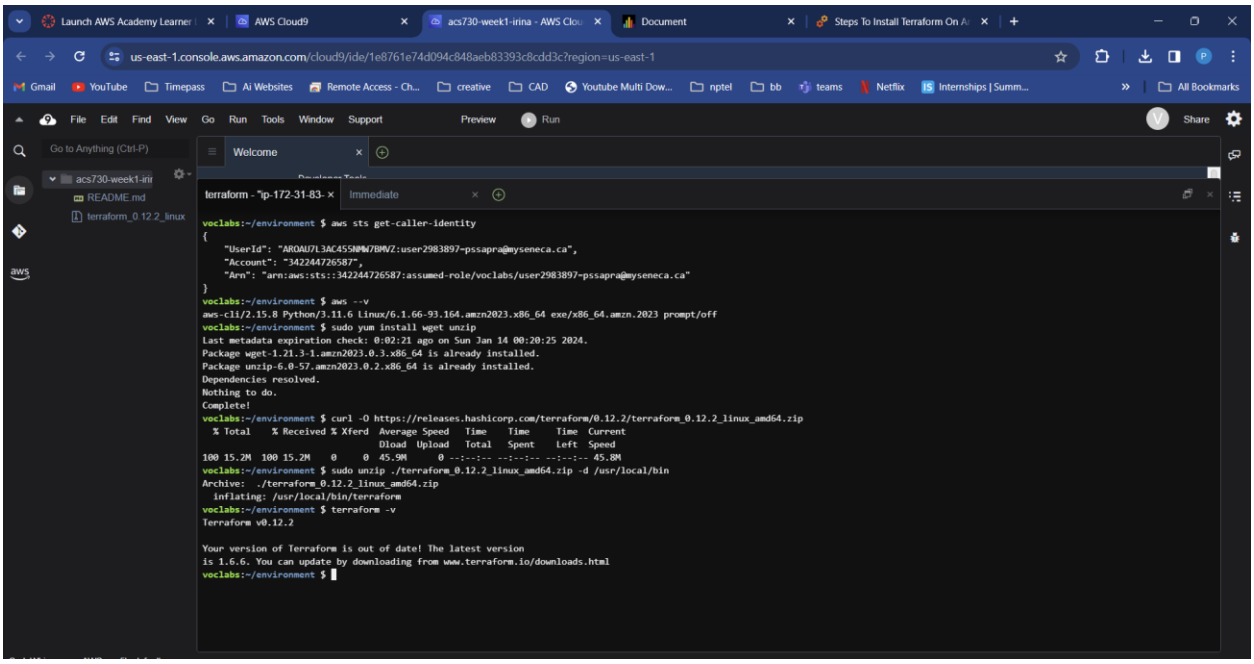
   https://sysadminxpert.com/steps-to-install-terraform-on-amazon-linux/#:~:text=Steps%20to%20Install%20Terraform%20on%20Amazon%20Linux%201,...%205%20Step%205%3A%20Terraform%20help%20command%20

   sudo yum install wget unzip

   curl -O https://releases.hashicorp.com/terraform/0.12.2/terraform_0.12.2_linux_amd64.zip

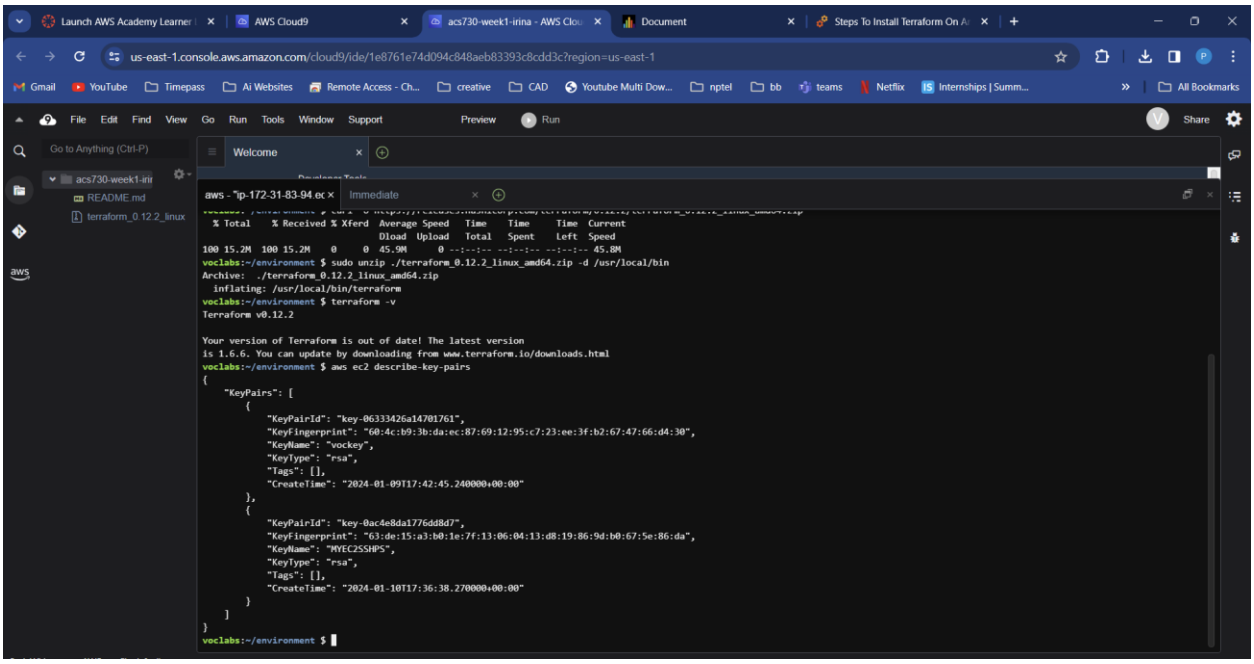
sudo unzip ./terraform_0.12.2_linux_amd64.zip -d /usr/local/bin

terraform -v


Launch AWS Academy Learner
AWS Cloud9
acs730-week1-irina - AWS Clou
Document
Steps To Install Terraform On A
us-east-1.console.aws.amazon.com/cloud9/ide/1e8761e74d094c848aeb83393c8cdd3c?region=us-east-1
Gmail
YouTube
Timepass
Ai Websites
Remote Access - Ch...
creative
CAD
Youtube Multi Dow...
nptel
bb
teams
Netflix
Internships | Summ...
All Bookmarks
File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
acs730-week1-iri
README.md
terraform_0.12.2_linux
Welcome
terraform - "ip-172-31-83-
Immediate
voclabs:~/environment $ aws sts get-caller-identity
{
"UserId": "AROAU7L3AC45NMW7BMVZ:user2983897=pssapra@myseneca.ca",
"Account": "342244726587",
"Arn": "arn:aws:sts::342244726587:assumed-role/voclabs/user2983897=pssapra@myseneca.ca"
}
voclabs:~/environment $ aws --v
aws-cli/2.15.8 Python/3.11.6 Linux/6.1.66-93.164.amzn2023.x86_64 exe/x86_64.amzn.2023 prompt/off
voclabs:~/environment $ sudo yum install wget unzip
Last metadata expiration check: 0:02:21 ago on Sun Jan 14 00:20:25 2024.
Package wget-1.21.3-1.amzn2023.0.3.x86_64 is already installed.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
voclabs:~/environment $ curl -O https://releases.hashicorp.com/terraform/0.12.2/terraform_0.12.2_linux_amd64.zip
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 15.2M 100 15.2M 0 0 45.9M 0 --:--:-- --:--:-- --:--:-- 45.8M
voclabs:~/environment $ sudo unzip ./terraform_0.12.2_linux_amd64.zip -d /usr/local/bin
Archive: ./terraform_0.12.2_linux_amd64.zip
inflating: /usr/local/bin/terraform
voclabs:~/environment $ terraform -v
Terraform v0.12.2
Your version of Terraform is out of date! The latest version
is 1.6.6. You can update by downloading from www.terraform.io/downloads.html
voclabs:~/environment $

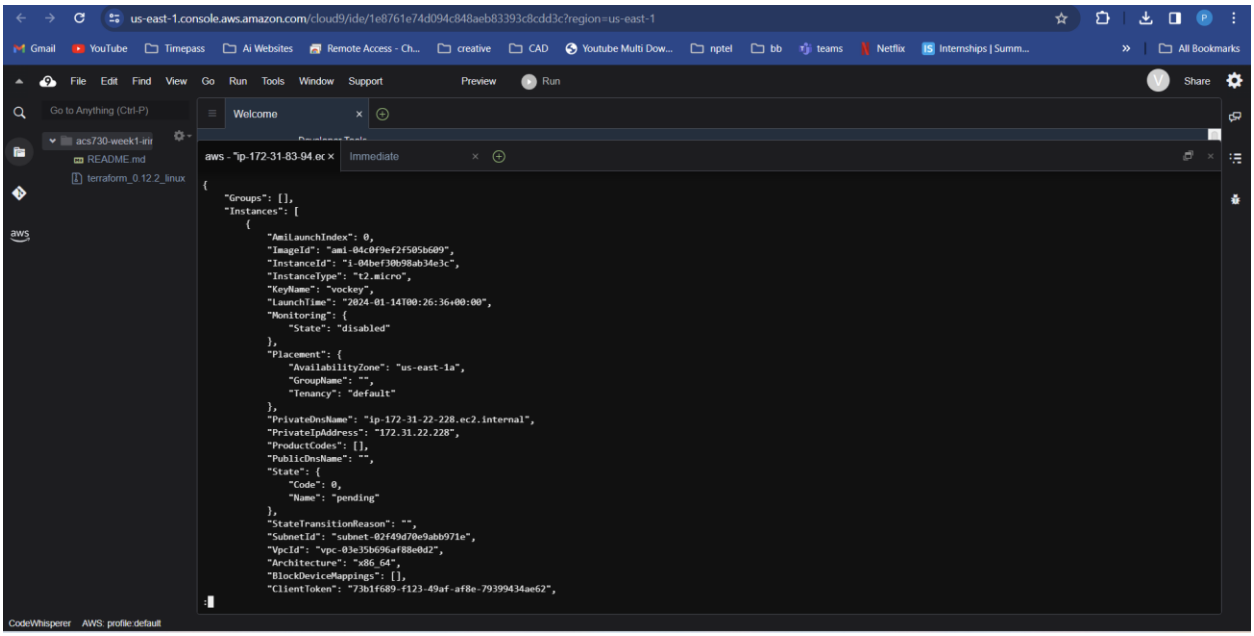
Task 4: Provisioning Amazon EC2 with AWS CLI

1. Get the key pair name provisioned by AWS Academy by running the AWS CLI command below.
   `aws ec2 describe-key-pairs`

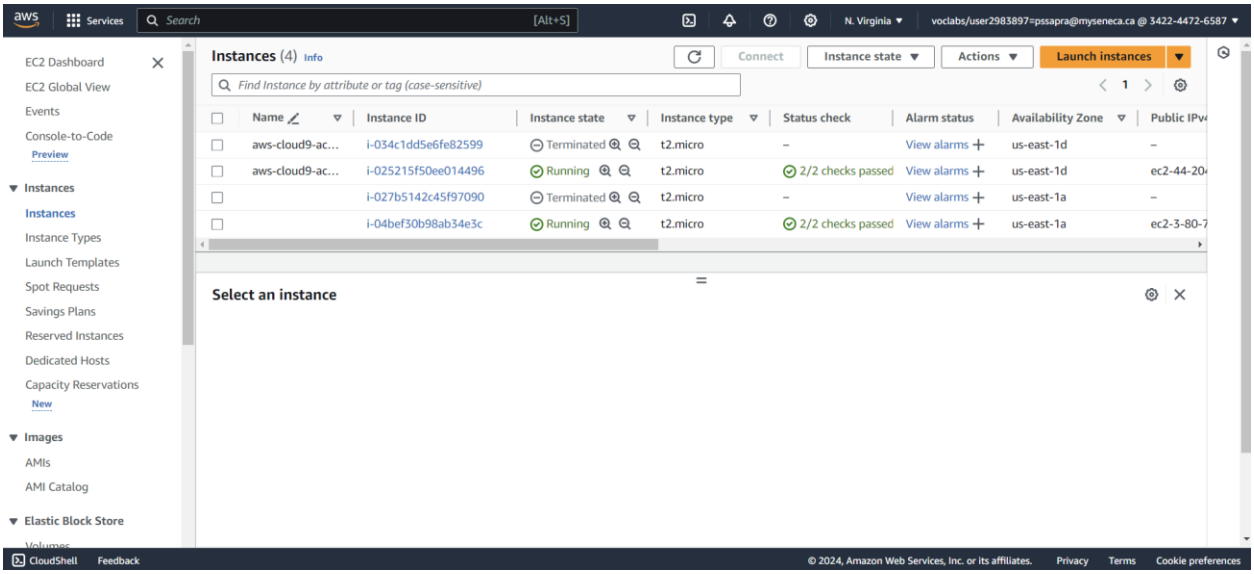


2. Provision and Amazon EC2 instance with the SSH key pair from the previous command. Notice dynamic resolution of the AMI id (VM template).

```
aws ec2 run-instances --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 --instance-type t2.micro --key-name vockey
```

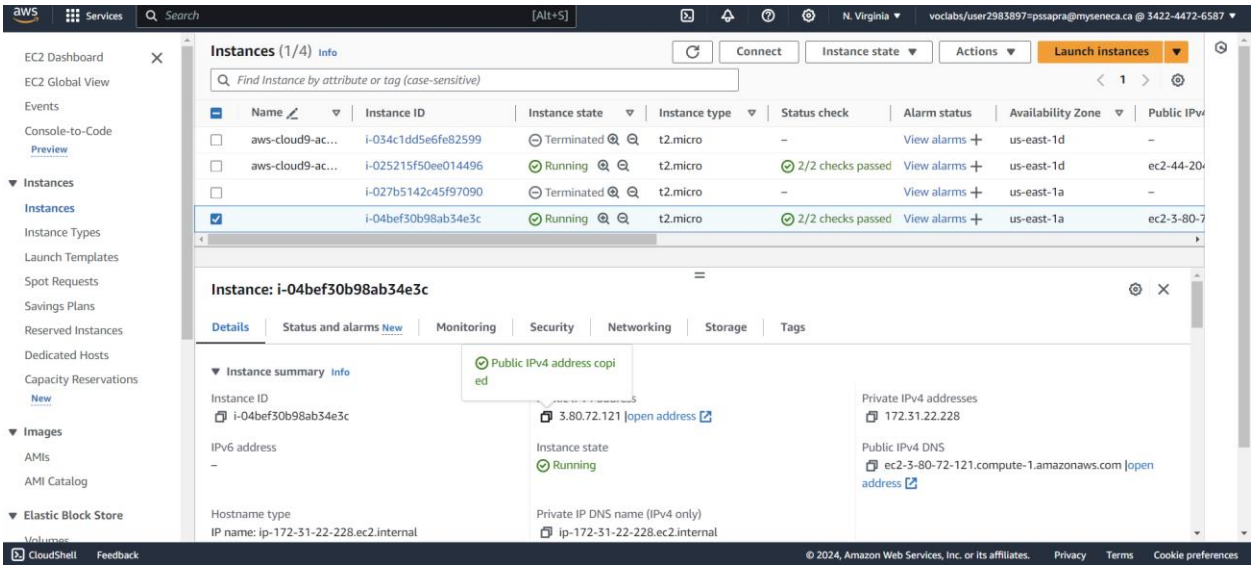


3. In the AWS Management console, verify that the instance got provisioned as expected. Save the instance id i-xxxxxxxxxx for the future use



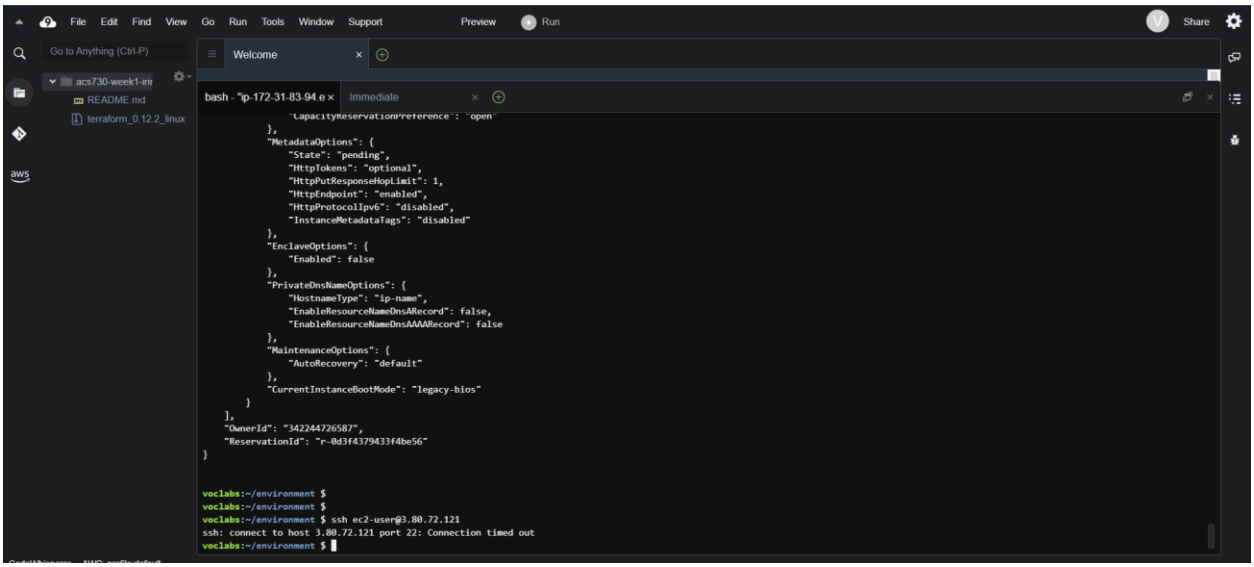
Task 5: Securely connecting Amazon EC2 with SSH (Secure Shell)

1. Select the provisioned instance and mark it's public IP.



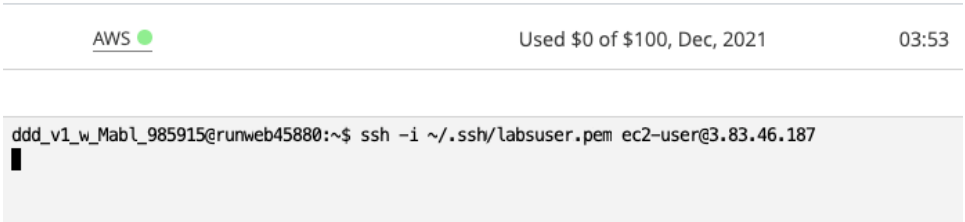
2. Attempt connecting to the EC2 from the Cloud9 terminal using the command below:=. What is the result?
   ssh ec2-user@3.83.46.187

   ssh ec2-user@3.80.72.121

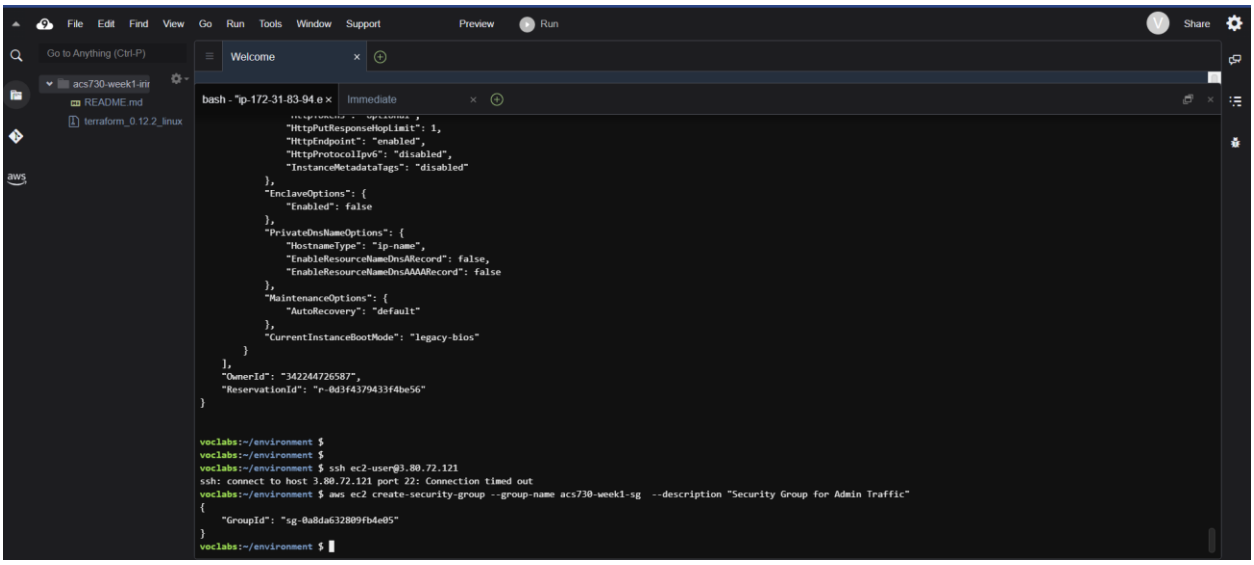


3. Attempt connecting to the EC2 from the AWS Academy Terminal using the command below. What is the result?
   ssh -i ~/.ssh/labsuser.pem ec2-user@3.83.46.187

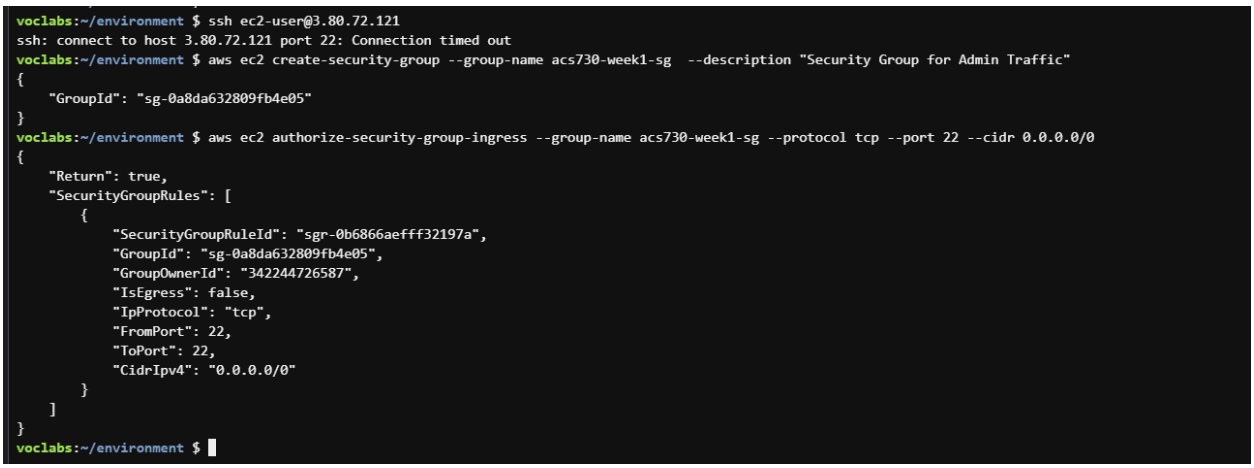
   ssh -i ~/.ssh/labsuser.pem ec2-user@3.80.72.121



4. Create firewall rules for our AWS EC2 to allow ingress on port 22.
   aws ec2 create-security-group --group-name acs730-week1-sg --description "Security Group for Admin Traffic"



5. Authorize ingress into EC2 instance from everywhere in the world:
aws ec2 authorize-security-group-ingress --group-name acs730-week1-sg --protocol tcp --port 22 --cidr 0.0.0.0/0



Note the group id (sg-XXXXXXXX) and use it as input for the next command. We will add this security group to our EC2 instance.

export INSATANCE_ID=i-039846ca048ac228a

export INSATANCE_ID=i-04bef30b98ab34e3c

export SG_IG=sg-03c8aacfc095f4070

export SG_IG=sg-05921ae694ad50992

```
voclabs:~/environment $ export INSATANCE_ID=i-04bef30b98ab34e3c
voclabs:~/environment $ export SG_IG=sg-05921ae694ad50992
voclabs:~/environment $
```

aws ec2 modify-instance-attribute --instance-id ${INSATANCE_ID} --groups ${SG_IG}

aws ec2 modify-instance-attribute --instance-id i-04bef30b98ab34e3c --groups sg-05921ae694ad50992

```
voclabs:~/environment $ export INSATANCE_ID=i-04bef30b98ab34e3c
voclabs:~/environment $ export SG_IG=sg-05921ae694ad50992
voclabs:~/environment $ aws ec2 modify-instance-attribute --instance-id i-04bef30b98ab34e3c --groups sg-05921ae694ad50992
voclabs:~/environment $
```

6. Verify that the security group is associated with the instance.

aws ec2 describe-instances --instance-ids ${INSTANCE_ID} --query "Reservations[].Instances[].SecurityGroups[].GroupId[]" --output text

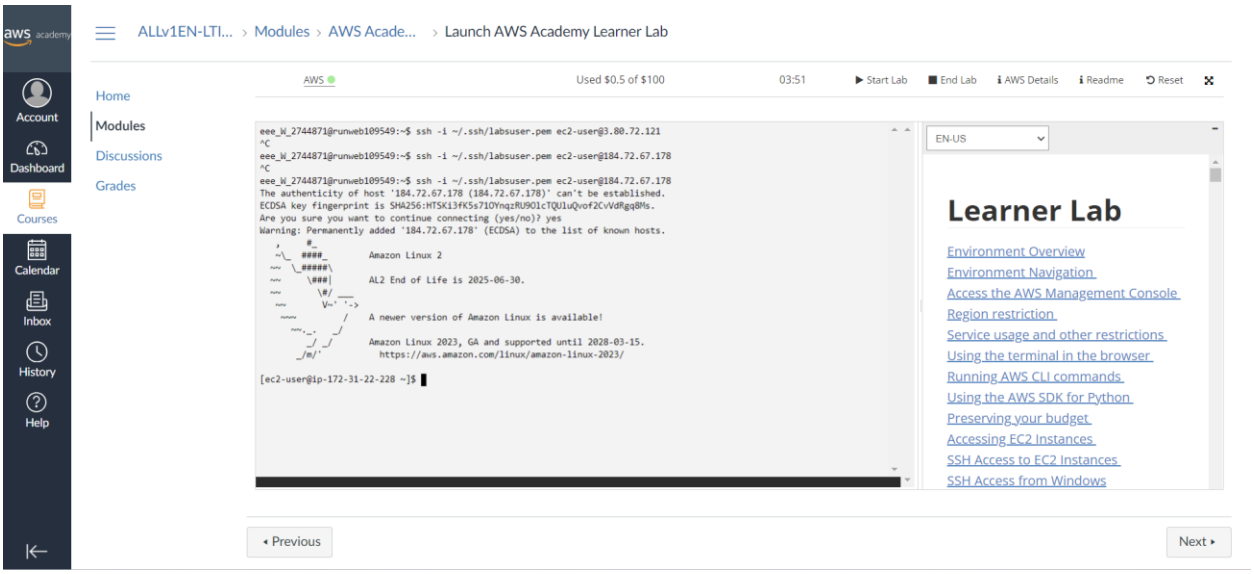
aws ec2 describe-instances --instance-ids i-04bef30b98ab34e3c --query "Reservations[].Instances[].SecurityGroups[].GroupId[]" --output text

```
}
voclabs:~/environment $ export INSATANCE_ID=i-04bef30b98ab34e3c
voclabs:~/environment $ export SG_IG=sg-05921ae694ad50992
voclabs:~/environment $ aws ec2 modify-instance-attribute --instance-id i-04bef30b98ab34e3c --groups sg-05921ae694ad50992
voclabs:~/environment $ aws ec2 describe-instances --instance-ids i-04bef30b98ab34e3c --query "Reservations[].Instances[].SecurityGroups[].GroupId[]" --output text
sg-05921ae694ad50992
voclabs:~/environment $
```

7. Attempt connecting to the EC2 from the AWS Academy Terminal using the command below.

ssh -i ~/.ssh/labsuser.pem ec2-user@3.83.46.187

ssh -i ~/.ssh/labsuser.pem ec2-user@184.72.67.178



Note: The public IP address might change between AWS Academy sessions. Make sure to verify the public IP address of the running instance.

8. Read more about connecting to Amazon EC2 from your local environment with SSH key pair and other options