

Assignment – 1

Step -1 : Login to AWS with IAM user always.

The image shows two screenshots related to AWS. The top screenshot is the AWS IAM user login page. It features the AWS logo, a 'Sign in as IAM user' section with input fields for Account ID (12 digits or account alias), IAM user name, and Password, and a 'Sign in' button. There are also links for 'Sign in using root user email' and 'Forgot password?'. To the right is a large banner for 'AWS Skill Builder' with the text 'Your new learning center to access 500+ free digital courses' and a 'GET STARTED' button. The bottom screenshot is the AWS Management Console home page. It shows the 'Console Home' with a 'Recently visited' list of services including CloudWatch, EC2, AWS Amplify, Route 53, Support, Billing and Cost Management, Lambda, API Gateway, VPC, IAM, Amazon EventBridge, and DynamoDB. There is also a 'Welcome to AWS' section with links for 'Getting started with AWS' and 'Training and certification'. The 'AWS Health' section shows 'Open issues' and 'Scheduled changes'. The 'Cost and usage' section displays 'Current month costs' as \$0.00, 'Forecasted month end costs' as \$0.01, and a bar chart for 'Total costs per month' showing a cost of \$4.00.

Sign in as IAM user

Account ID (12 digits) or account alias

313749178263

IAM user name

shreekar

Password

☐ Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

AWS Skill Builder

Your new learning center to access 500+ free digital courses

GET STARTED

English

[Terms of Use](#) [Privacy Policy](#) © 1996-2024, Amazon Web Services, Inc. or its affiliates.

Console Home

Reset to default layout **+ Add widgets**

Recently visited

- CloudWatch
- EC2
- AWS Amplify
- Route 53
- Support
- Billing and Cost Management
- Lambda
- API Gateway
- VPC
- IAM
- Amazon EventBridge
- DynamoDB

Applications (0)

Region: US East (N. Virginia)

us-east-1 (Current Region) Find applications

No applications
Get started by creating an application.
Create application

Welcome to AWS

Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
[Training and certification](#)

AWS Health

Open issues: 0 (Past 7 days)

Scheduled changes: 0 (Upcoming and past 7 days)

Cost and usage

Current month costs: **\$0.00**

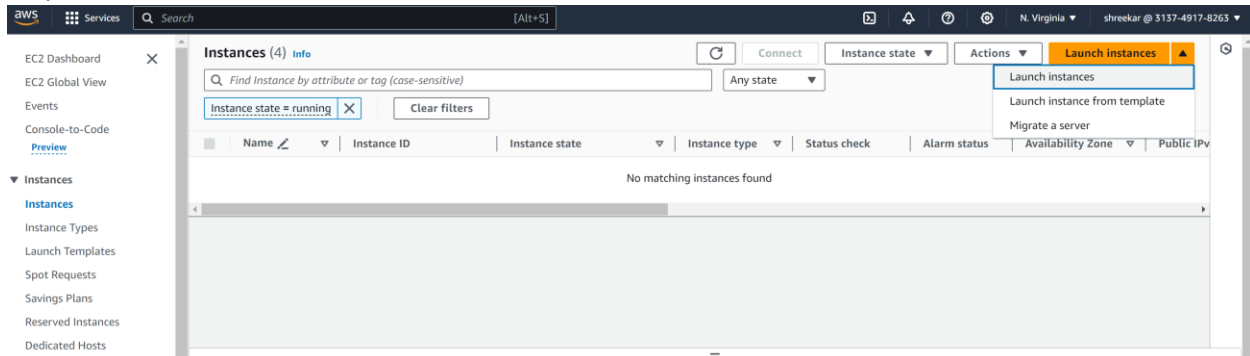
Forecasted month end costs: **\$0.01**

Total costs per month
Cost (USD)
\$6.00
\$4.00
\$2.00

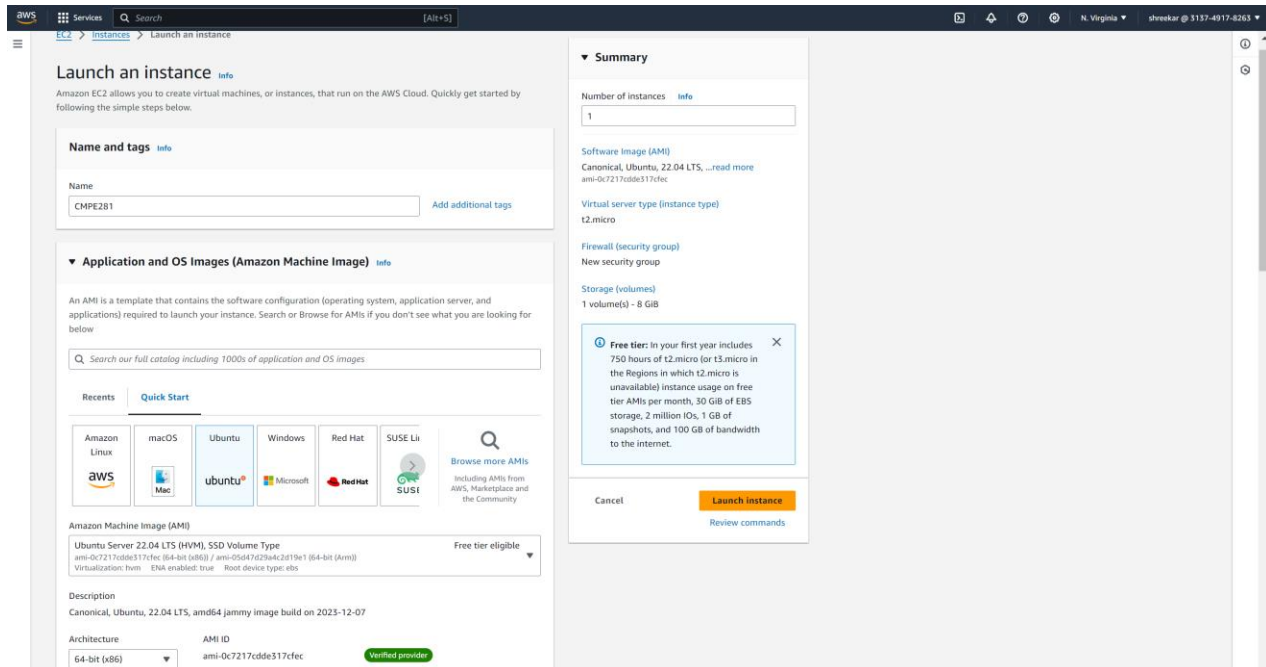
© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Logged in to the AWS account
From the Dashboard visit EC2

Step – 2: We should launch an EC2 Instance



I have named my ec2 instance as CMPE 281 and selected Ubuntu as my AMI image.



I have selected my instance as t2micro (free tier) and have created a key pair already named demo281.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

▼

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ 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*

▼

[Create new key pair](#)

I have configured a new security group and enabled all traffic but limited to my IP address.

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-0b81bb7dd64bf289a

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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

We'll create a new security group called 'launch-wizard-26' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

My IP 76.103.204.194/32 ▼

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

Add the Jenkins script to run on the Ubuntu server

The screenshot shows the AWS Management Console 'Launch Instance' wizard. The 'User data' field is populated with the following script:

```
sudo hostnamectl set-hostname Jenkins
/bin/bash
sudo apt update
sudo apt install openjdk-11-jre
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/ | sudo tee
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
```

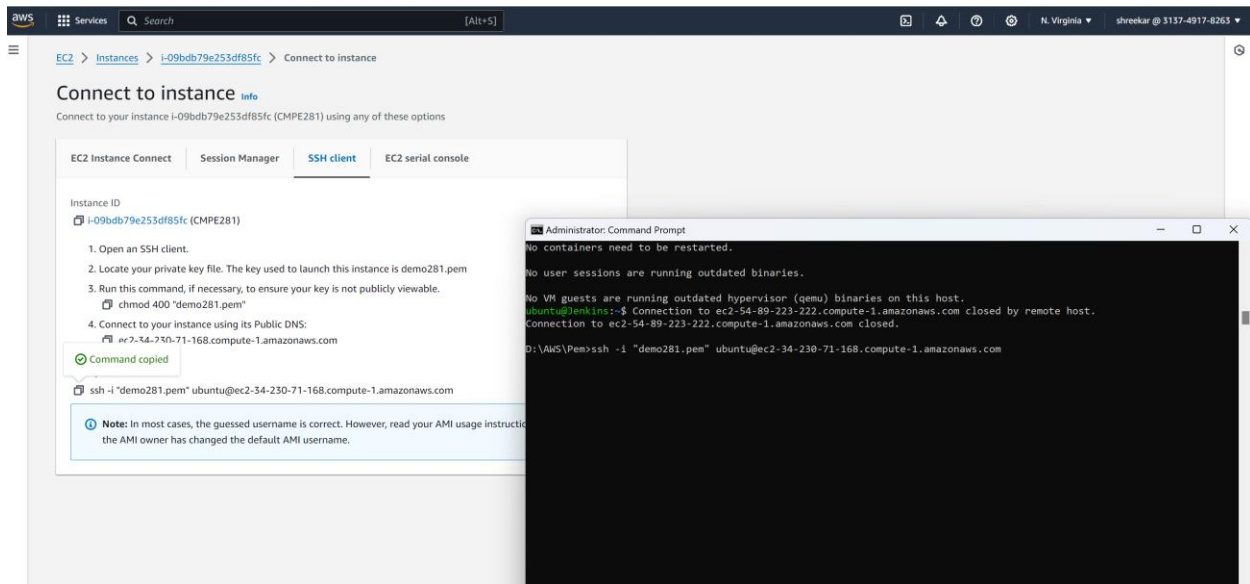
The 'Summary' panel on the right shows the instance configuration: 1 instance, Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, new security group, and 1 volume of 8 GiB. A 'Free tier' notification indicates that the first year includes 750 hours of t2.micro usage. The 'Launch instance' button is visible at the bottom right.

Now the instance is ready and running

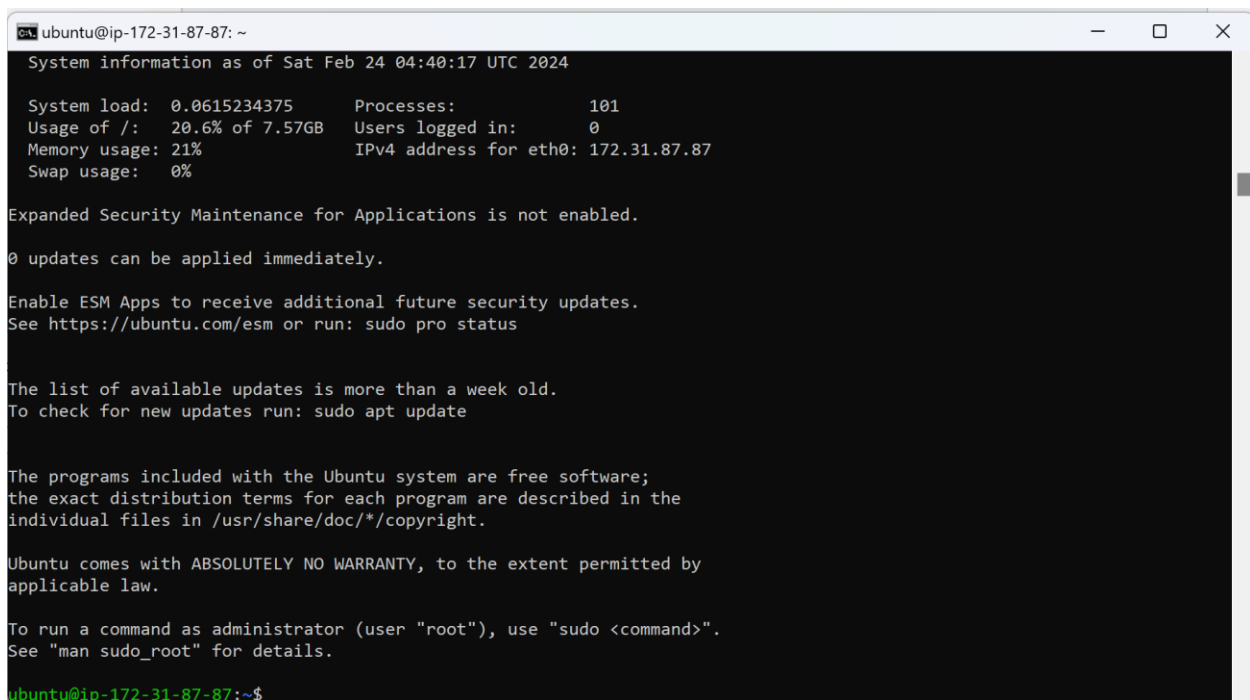
The screenshot shows the AWS Management Console 'Instances' page. A single instance, CMPE281 (i-09bdb79e253df85fc), is listed with a 'Running' status. The instance details are expanded below:

- Instance summary:** Instance ID i-09bdb79e253df85fc (CMPE281), IPV6 address -, Hostname type IP name: ip-172-31-87-87.ec2.internal, Answer private resource DNS name IPV4 (A) 34.230.71.168 [Public IP], Auto-assigned IP address 34.230.71.168 [Public IP].
- Public IPv4 address:** 34.230.71.168 [Open address].
- Private IPv4 address:** 172.31.87.87.
- Public IPv4 DNS:** ec2-34-230-71-168.compute-1.amazonaws.com [Open address].
- Instance state:** Running.
- Private IP DNS name (IPv4 only):** ip-172-31-87-87.ec2.internal.
- Instance type:** t2.micro.
- VPC ID:** vpc-0b81bb7dd64bf289a.
- Elastic IP addresses:** -.
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations.

Now we are accessing the ubuntu ec2 instance via cmd with the key pair



Now we have accessed the ubuntu ec2 instance



Exposing the default port on 8080 to run Jenkins on our instance

aws Services [Alt+5] N. Virginia shreekar @ 3137-4917-8263

EC2 > Security Groups > sg-0ed3f1251cca54f3d - launch-wizard-26 > Edit inbound rules

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	Description - optional Info	
sg-06d35fb2c17ac750e	SSH	TCP	22	Custom	Q 0.0.0.0/0 X	Delete
sg-0983bb98860033a80	HTTPS	TCP	443	Custom	Q 0.0.0.0/0 X	Delete
sg-0aeecdc35962e4897	HTTP	TCP	80	Custom	Q 0.0.0.0/0 X	Delete
-	Custom TCP	TCP	8080	Anywher...	Q 0.0.0.0/0 X	Delete

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

We now enable Jenkins, start the service and fetch the status.

```
ubuntu@Jenkins: ~  
Executing: /lib/systemd/systemd-sysv-install enable jenkins  
ubuntu@Jenkins:~$ sudo systemctl start jenkins  
ubuntu@Jenkins:~$ sudo systemctl status jenkins  
● jenkins.service - Jenkins Continuous Integration Server  
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2024-02-24 04:55:29 UTC; 42s ago  
     Main PID: 4751 (java)  
        Tasks: 43 (limit: 1121)  
       Memory: 310.7M  
          CPU: 46.926s  
      CGroup: /system.slice/jenkins.service  
              └─4751 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkin>  
  
Feb 24 04:54:53 Jenkins jenkins[4751]: 27f9f35612bc4c37a0d66d8e11aaec93  
Feb 24 04:54:53 Jenkins jenkins[4751]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword  
Feb 24 04:54:53 Jenkins jenkins[4751]: *****  
Feb 24 04:54:53 Jenkins jenkins[4751]: *****  
Feb 24 04:54:53 Jenkins jenkins[4751]: *****  
Feb 24 04:55:29 Jenkins jenkins[4751]: 2024-02-24 04:55:29.449+0000 [id=30] INFO jenkins.InitReactorRunne>  
Feb 24 04:55:29 Jenkins jenkins[4751]: 2024-02-24 04:55:29.479+0000 [id=22] INFO hudson.lifecycle.Lifecyc>  
Feb 24 04:55:29 Jenkins systemd[1]: Started Jenkins Continuous Integration Server.  
Feb 24 04:55:29 Jenkins jenkins[4751]: 2024-02-24 04:55:29.610+0000 [id=45] INFO h.m.DownloadService$Down>  
Feb 24 04:55:29 Jenkins jenkins[4751]: 2024-02-24 04:55:29.611+0000 [id=45] INFO hudson.util.Retrier#star>  
lines 1-20/20 (END)
```

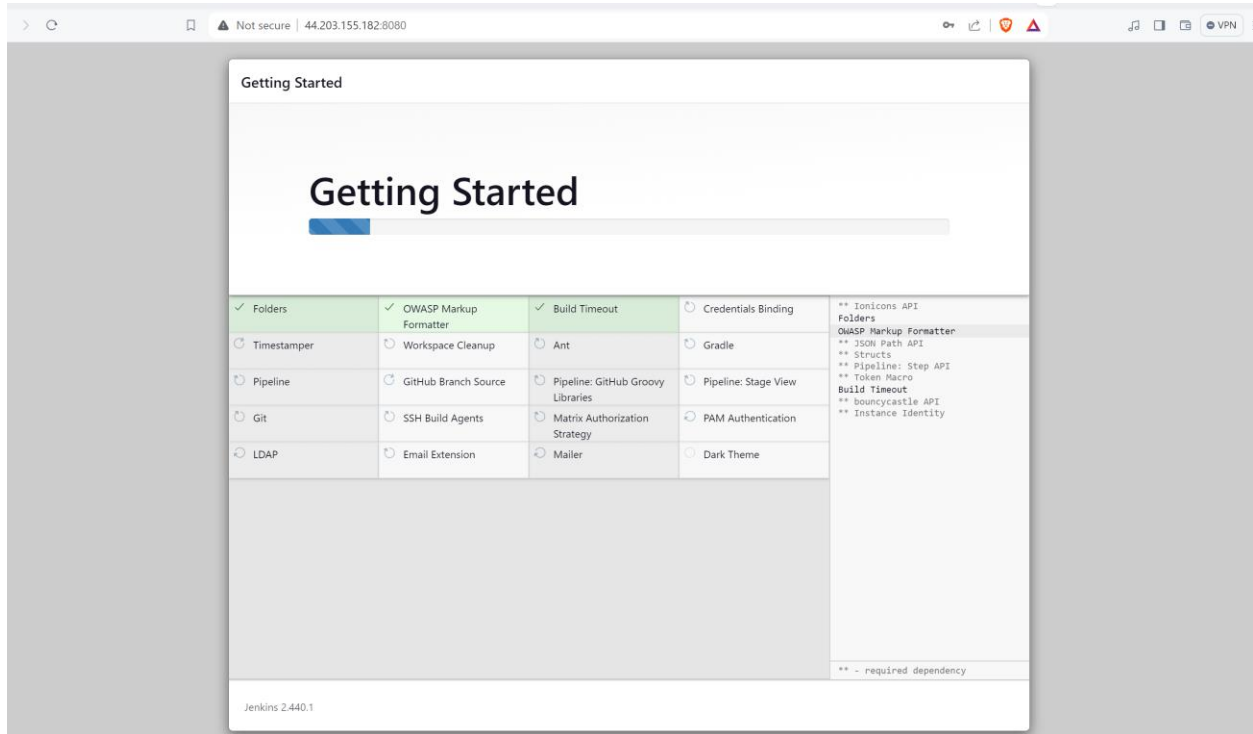
We now open the public ip address

The screenshot displays the AWS Management Console interface. On the left, there is a navigation menu with categories like EC2 Dashboard, Instances, Elastic Block Store, and Network & Security. The main content area shows the 'Instance summary' for the instance ID i-07f06e977c50eca93 (CMPE281). The instance is in a 'Running' state. Key details include the public IPv4 address 44.203.155.182, the instance type t2.micro, and the AMI Ubuntu (Inferred). The instance is located in the us-east-1 region. Below the summary, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is selected, showing further information about the instance, including the platform details (Linux/UNIX), the AMI name (ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20231207), and the launch time (Fri Feb 23 2024 20:50:37 GMT-0800 (Pacific Standard Time)).

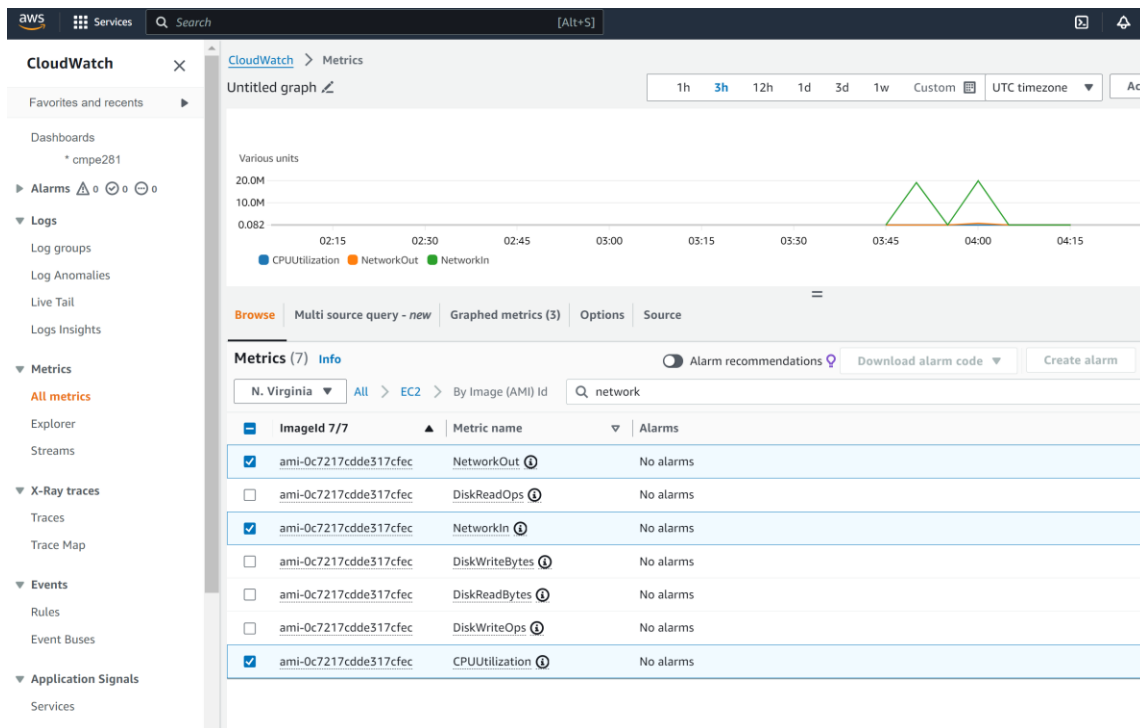
Now we access the ip address with port 8080

The screenshot shows a web browser window with the address bar displaying '44.203.155.182:8080/login?from=%2F'. The page content is titled 'Getting Started' and 'Unlock Jenkins'. It explains that to ensure Jenkins is securely set up, a password has been written to the log file /var/lib/jenkins/secrets/initialAdminPassword. The user is instructed to copy the password from either location and paste it into the 'Administrator password' field. The password field is currently empty, and a 'Continue' button is located at the bottom right of the form.

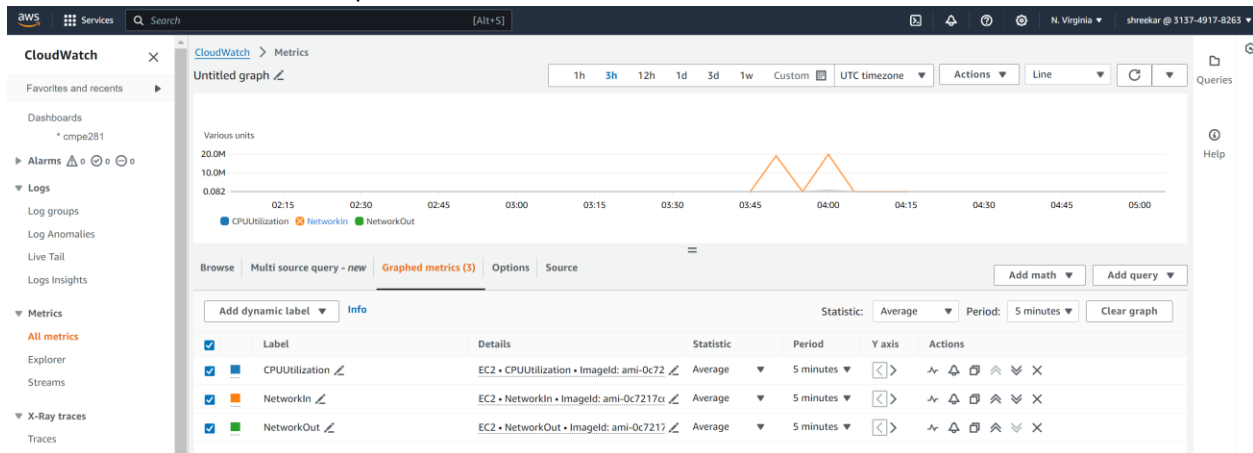
Jenkins is exposed to 8080 port. <http://44.203.155.182:8080>



Now we add cloudwatch metrics – Network out, Network In and CPU utilization to observe and monitor our instance.



As of now we don't see and cpu utilization



We have downloaded the pem key in the location and used the path in the directory and connected to our instance via ssh/

```
D:\>cd D:\AWS\Pem

D:\AWS\Pem>ssh -i "demo281.pem" ubuntu@ec2-44-203-155-182.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

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

System information as of Sat Feb 24 05:14:49 UTC 2024

System load:  0.0           Processes:            100
Usage of /:   31.5% of 7.57GB Users logged in:        0
Memory usage: 52%          IPv4 address for eth0: 172.31.95.171
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

78 updates can be applied immediately.
43 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Sat Feb 24 04:51:27 2024 from 76.103.204.194
ubuntu@Jenkins:~$
```

Now we increase the load on the server by installing the package stress.

```
ubuntu@jenkins:~$ sudo apt install stress
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  stress
0 upgraded, 1 newly installed, 0 to remove and 76 not upgraded.
Need to get 18.4 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 stress amd64 1.0.5-1 [18.4 kB]
Fetched 18.4 kB in 0s (1083 kB/s)
Selecting previously unselected package stress.
(Reading database ... 66488 files and directories currently installed.)
Preparing to unpack .../stress_1.0.5-1_amd64.deb ...
Unpacking stress (1.0.5-1) ...
Setting up stress (1.0.5-1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

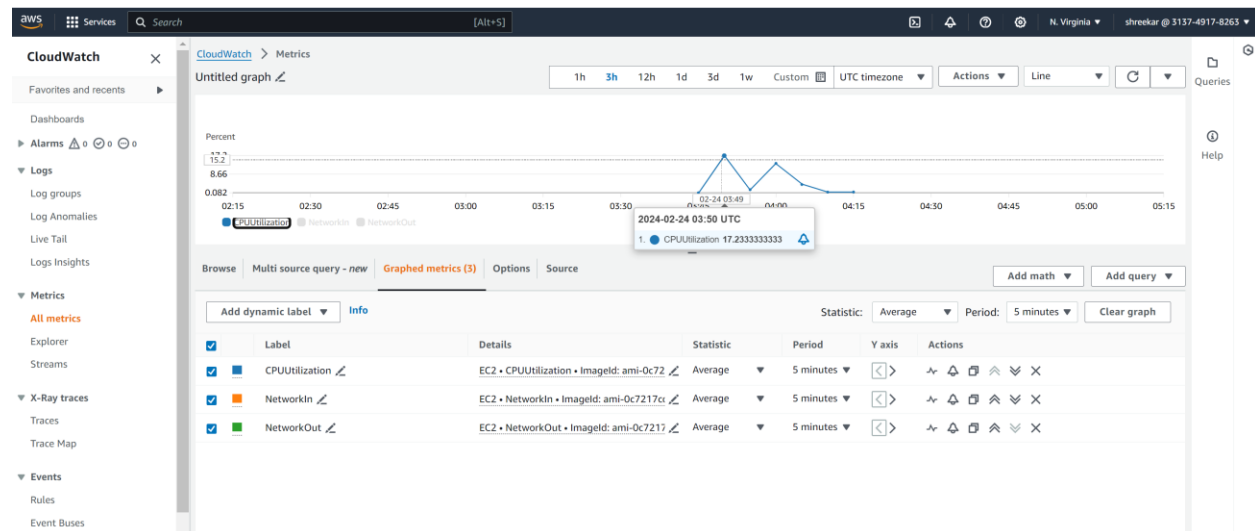
No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Now we monitor the logs on cloud watch and check the CPU utilization as we can see it has increased to 17%



Now we can see that the CPU utilization to less that 1%

