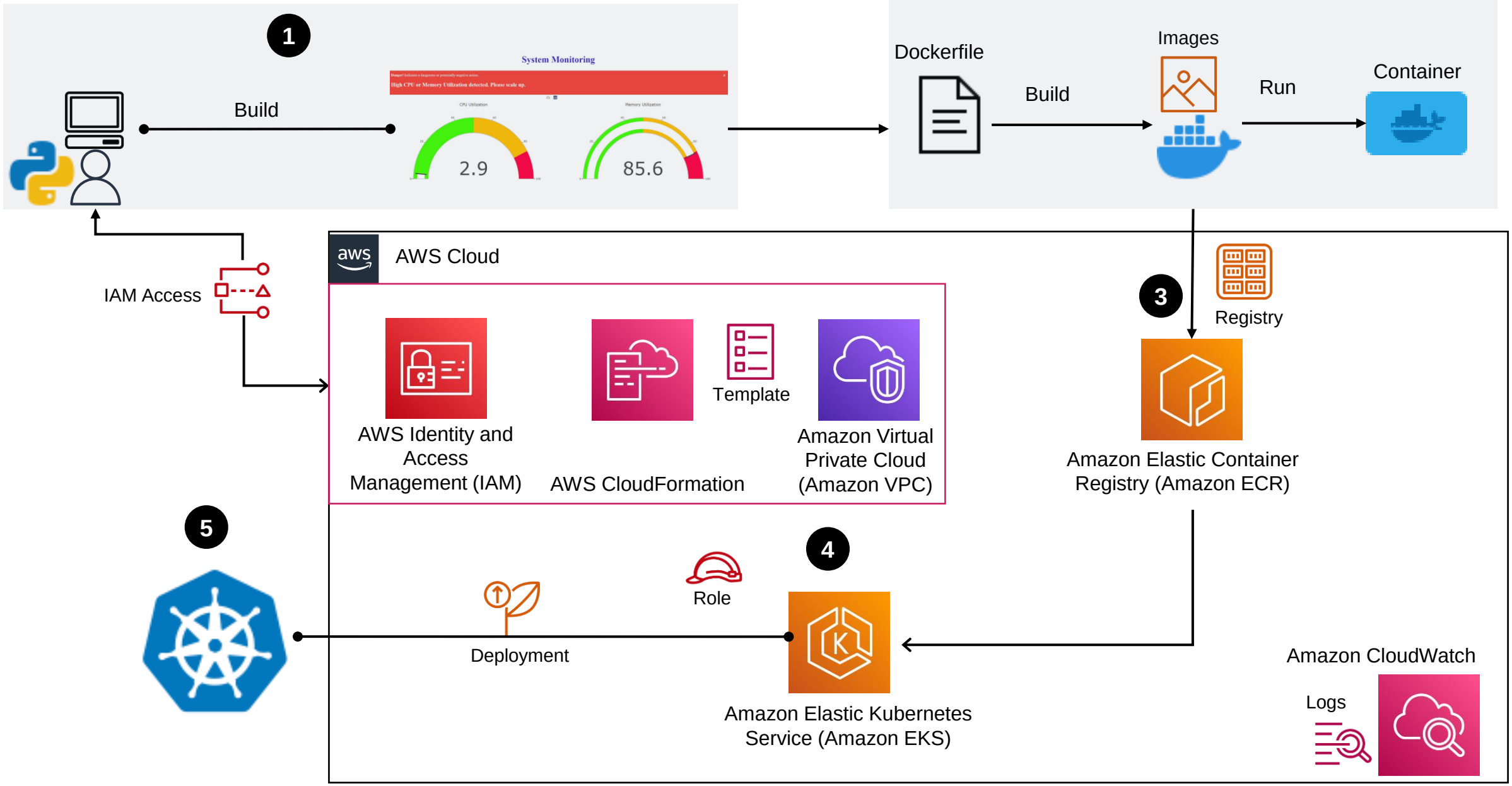
Cloud Native Monitoring Application Development and Containerization in AWS

Architecture of the project



AWS Services used: IAM, EKS, ECR, CloudWatch, Cloud Formation, VPC

Application Tools Used: Python, Flask, Docker Desktop, Git,

Libraries Used: Boto3 client, Plotly, psutil, kubectl, AWS CLI, Flask, render_template

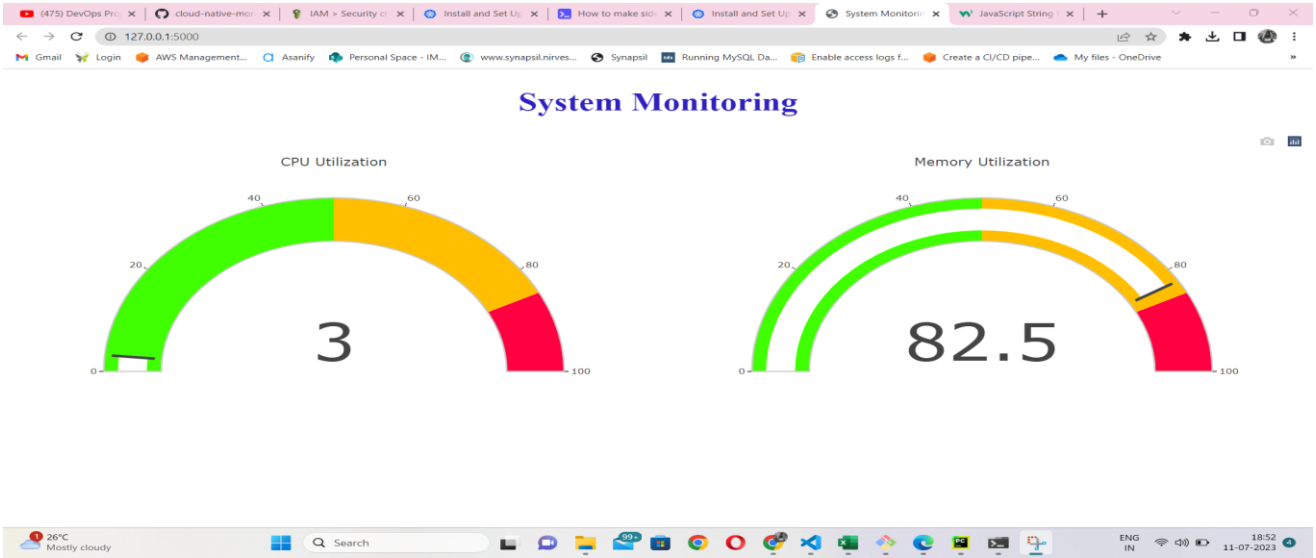
Step-1:

Deploy the Flask (Cloud-Native-Monitoring) application locally.

Run the application by using command:

$ Python app.py

This will start the Flask server on **`localhost:5000`**. Navigate to [http://localhost:5000/] (http://localhost:5000/) on your browser to access the application.



Step-2:

Dockerizing the Flask application (Cloud-Native-Monitoring)

Create a **`**Docker file**`** in the root directory of the project with the following contents:

Use the official python image as the base image

FROM python:3.10-slim-bullseye

#Set the working directory in the container

WORKDIR /app

#Copy the requirements file to the working directory

COPY requirements.txt .

#Install the required python packages

RUN pip install --no-cache-dir -r requirements.txt

#Copy the application code to the working directory

COPY . .

#Set the environment variables for the flask app

ENV FLASK_RUN_HOST=0.0.0.0

#Expose the port on which the flask app will run

EXPOSE 5000

#Start the Flask app when the container is run

CMD[“flask”, “run”]

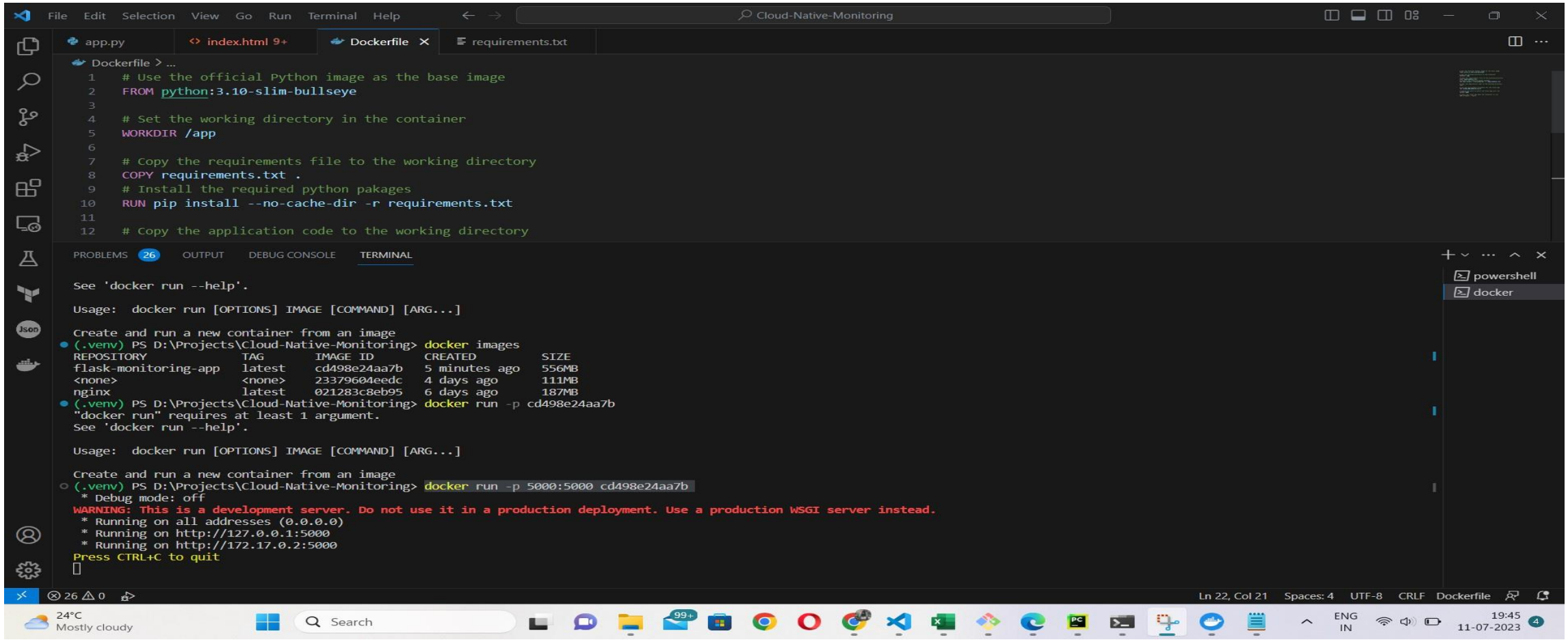
****Build the Docker image****

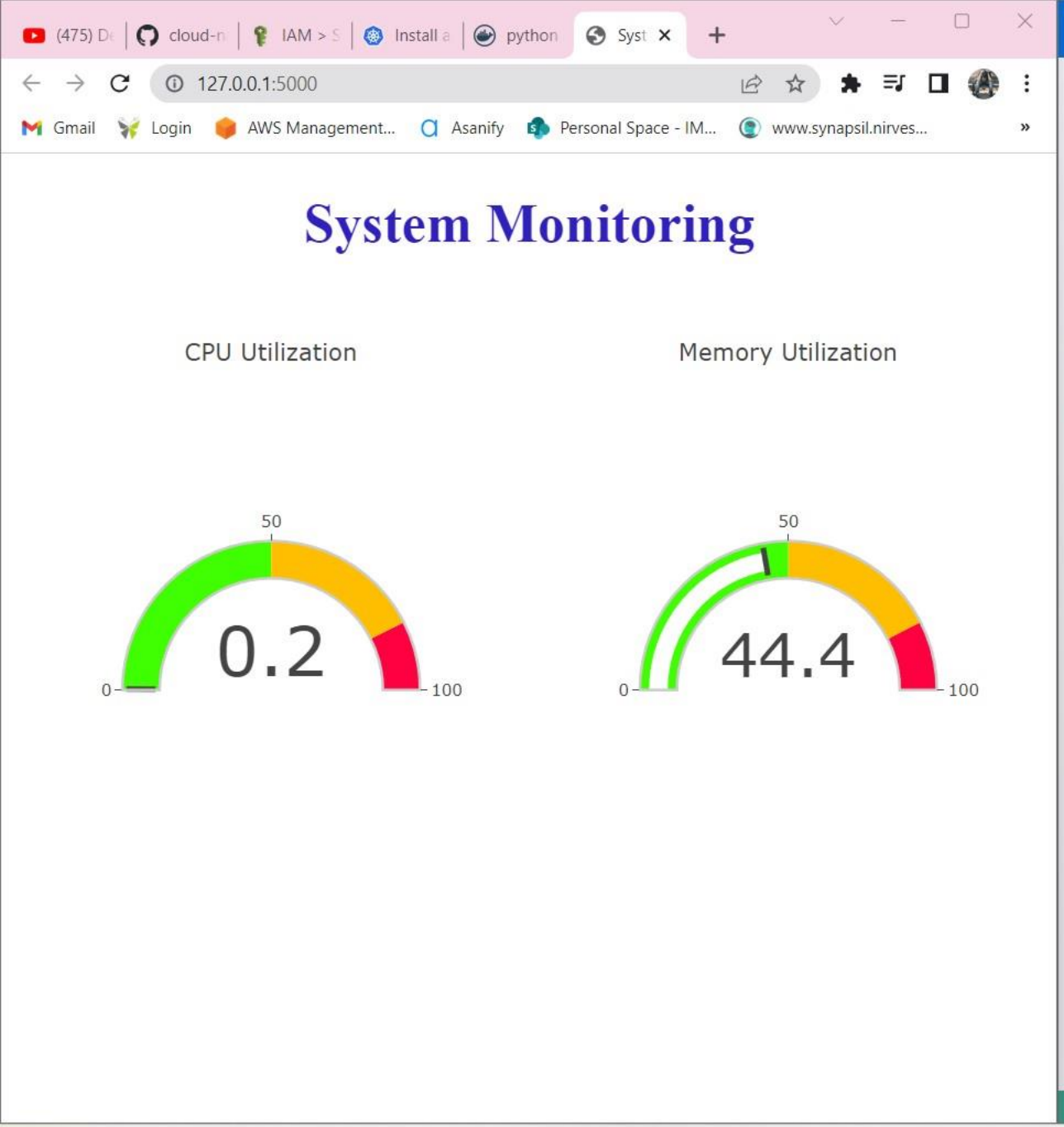
To build the Docker image, execute the following command:

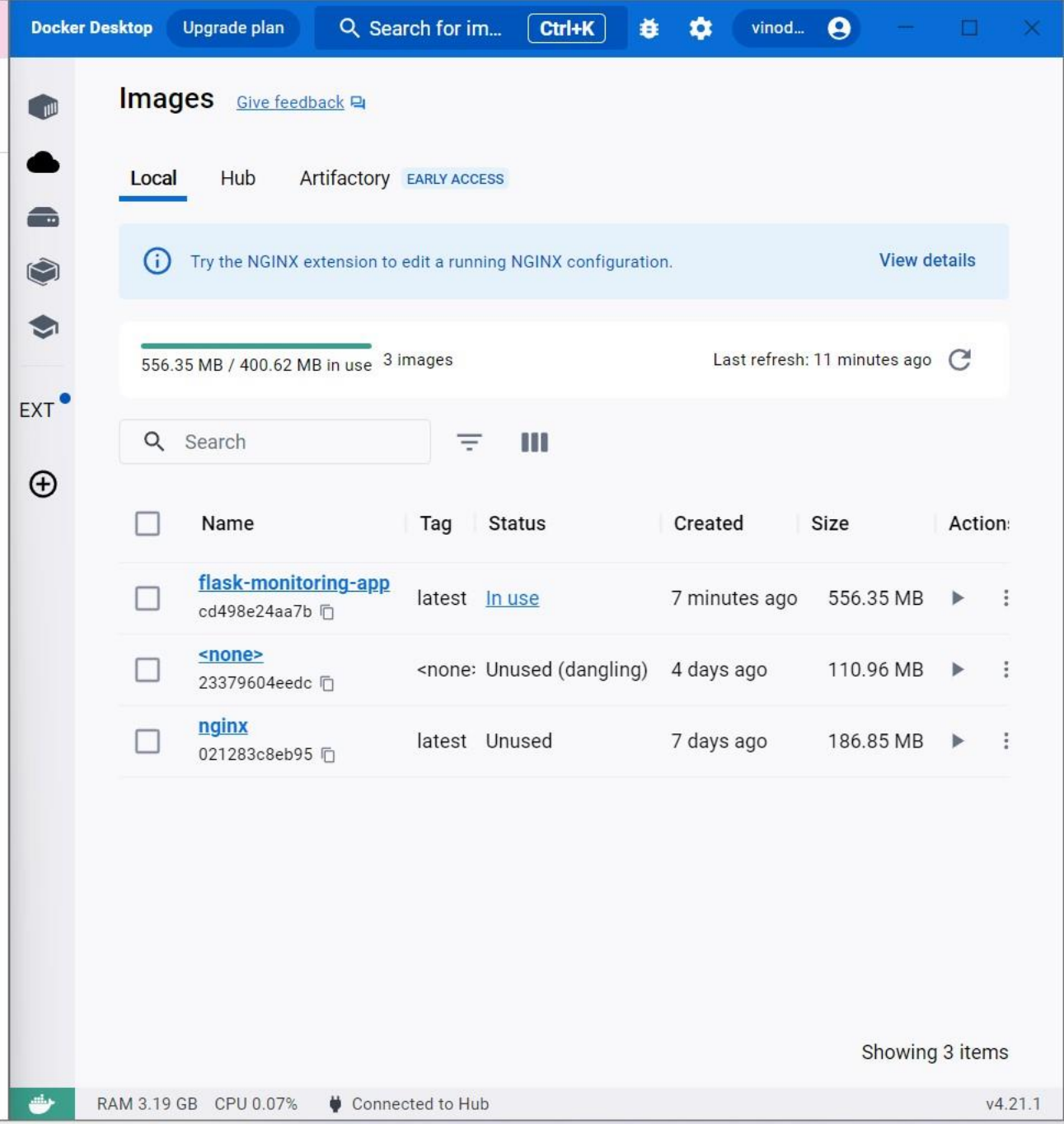
$ docker build -t flask-monitoring-app. #Here docker image name flask-monitoring-app

****Run the Docker image is it working or not****

$ docker run –p 5000:5000 cd498e24aa7b # Here this id(cd498e24aa7b) is docker image id




(475) D
cloud-n
IAM > S
Install a
python
Syst
127.0.0.1:5000
Gmail
Login
AWS Management...
Asanify
Personal Space - IM...
www.synapsil.nirves...
System Monitoring
CPU Utilization
50
0.2
0
100
Memory Utilization
50
44.4
0
100


Docker Desktop
Upgrade plan
Search for im...
Ctrl+K
vinod...
EXT
Images
Give feedback
Local
Hub
Artifactory
EARLY ACCESS
Try the NGINX extension to edit a running NGINX configuration.
View details
556.35 MB / 400.62 MB in use
3 images
Last refresh: 11 minutes ago
Search
Name
Tag
Status
Created
Size
Action
flask-monitoring-app
cd498e24aa7b
latest
In use
7 minutes ago
556.35 MB
<none>
23379604eedc
<none:
Unused (dangling)
4 days ago
110.96 MB
nginx
021283c8eb95
latest
Unused
7 days ago
186.85 MB
Showing 3 items
RAM 3.19 GB
CPU 0.07%
Connected to Hub
v4.21.1

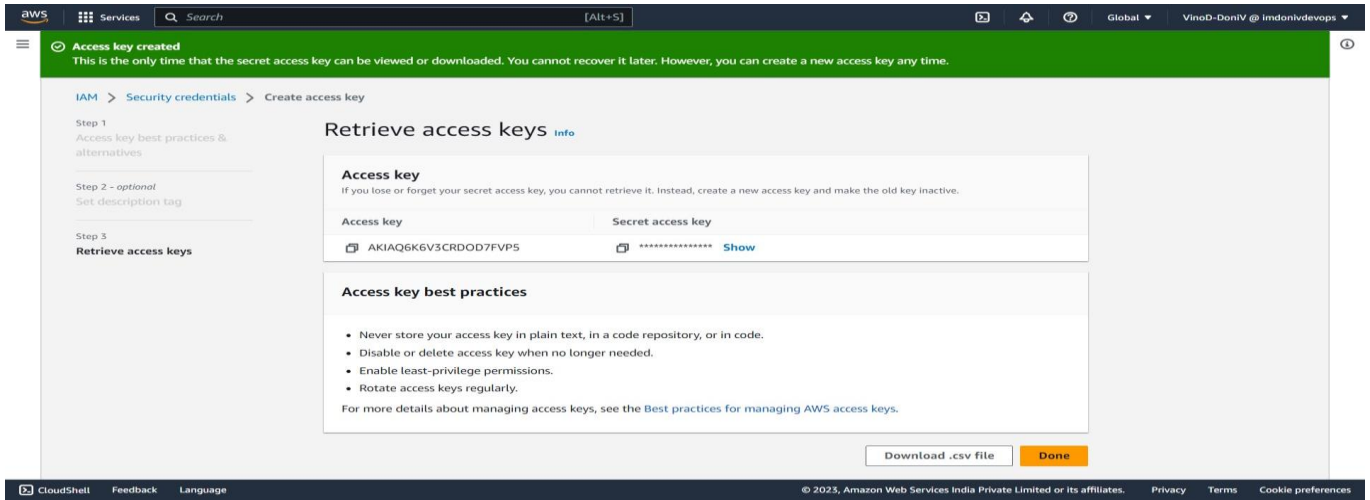

24°C
Mostly cloudy
Search
ENG
IN
19:47
11-07-2023

Step-3:

AWS Account and Access Keys

To get your access key ID and Secret access key

1. Open the IAM Console at https://console.aws.amazon.com/iam/
2. On the navigation menu, choose Users.
3. Open the Security credentials tab, and then choose Create access key.
4. On the Access Key best practices & alternatives, choose Command Line Interface (CLI)
5. Click next, To see the new access key, choose Show. Your credentials resemble the following.
    - Access Key ID: AKIAQ6V3CRDOD7FVPS
    - Secret access key: ----------
6. To download the key pair, choose download .csv file. Store the .csv file with keys In a secure location.

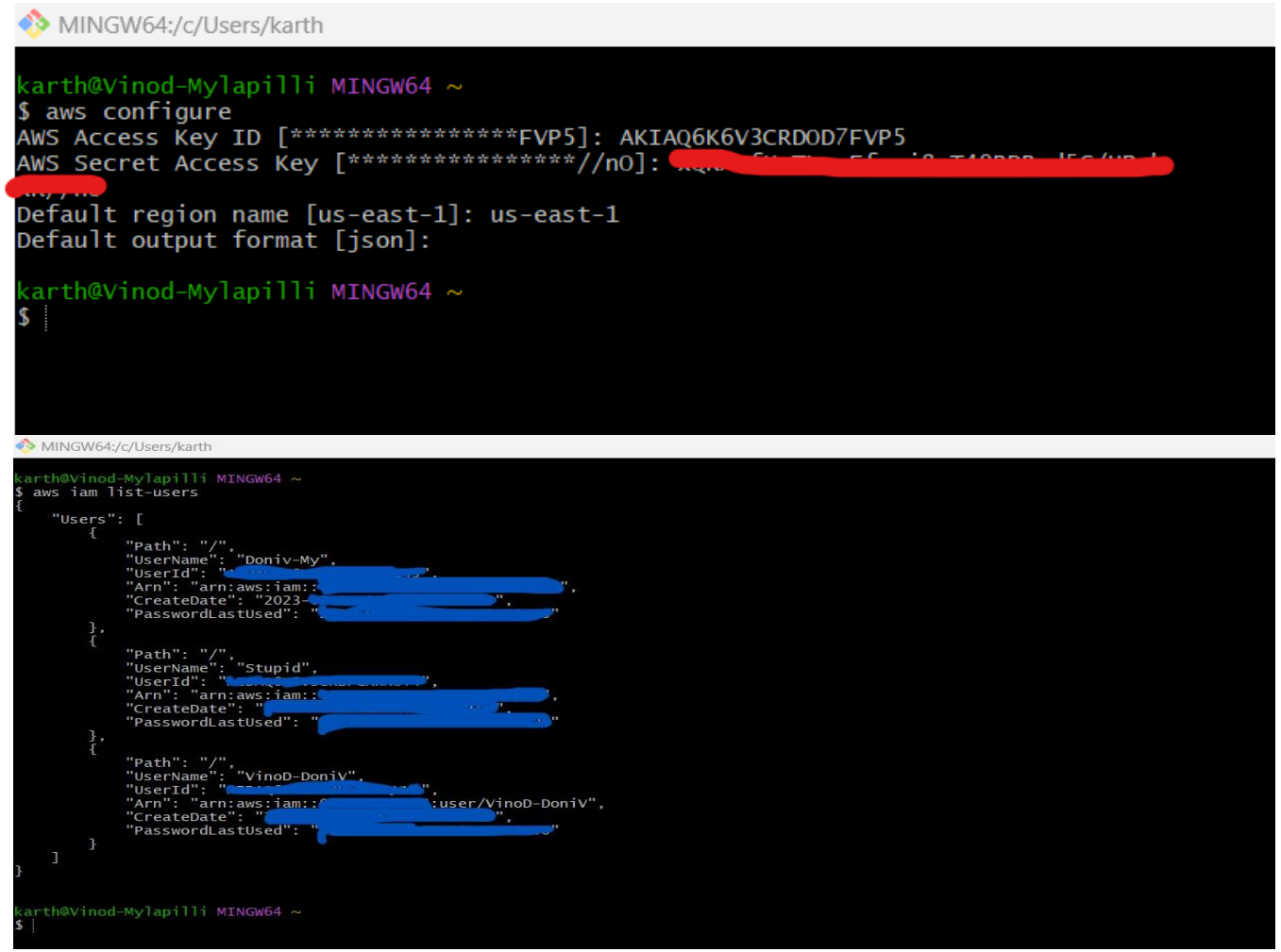


****AWS CLI installed and configured on your device (Here I am using Git bash)****

Run the following command in git bash

$ aws configure

Then give the AWS Access Key ID, Secret Access Key, Default region name and Default output format



Step-4:

Pushing the Docker image to ECR

Create an ECR by using either Management Console or using Python

Create an ECR repository using ECR Management Console:

1. Open the Amazon ECR console at (https://console.aws.amazon.com/ecr/).
2. Choose **Get Started**.
3. For Visibility settings, Choose **private**.
4. For Repository name, specify a name for the repository (flask-monitoring-app).
5. Rest all by default leave it.
6. **Choose Create repository**.

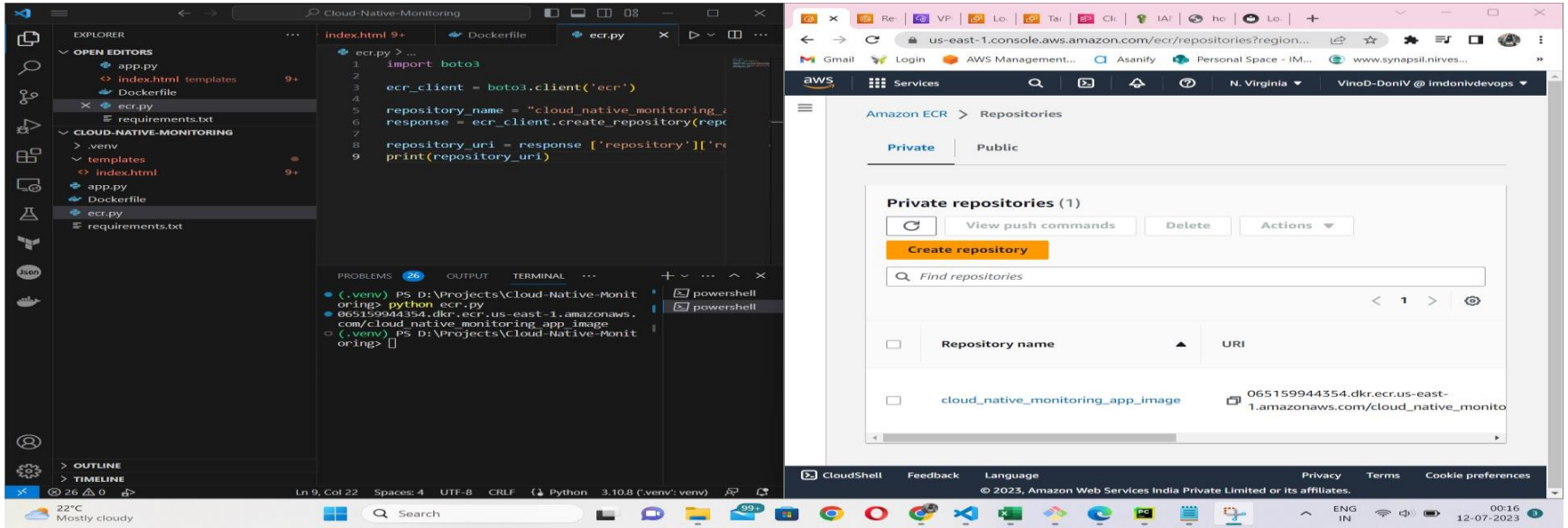
Build, tag, and push a Docker image

1. Select the repository you created and choose **View push commands** to view the steps to push an image to your new repository.
2. Run the login command that authenticates your Docker client to your registry by using the command from the console in a terminal window. This command provides an authorization token is valid for 12 hours.
3. Push the newly tagged image to your repository by using the docker push command in a terminal window.
4. Choose Close.

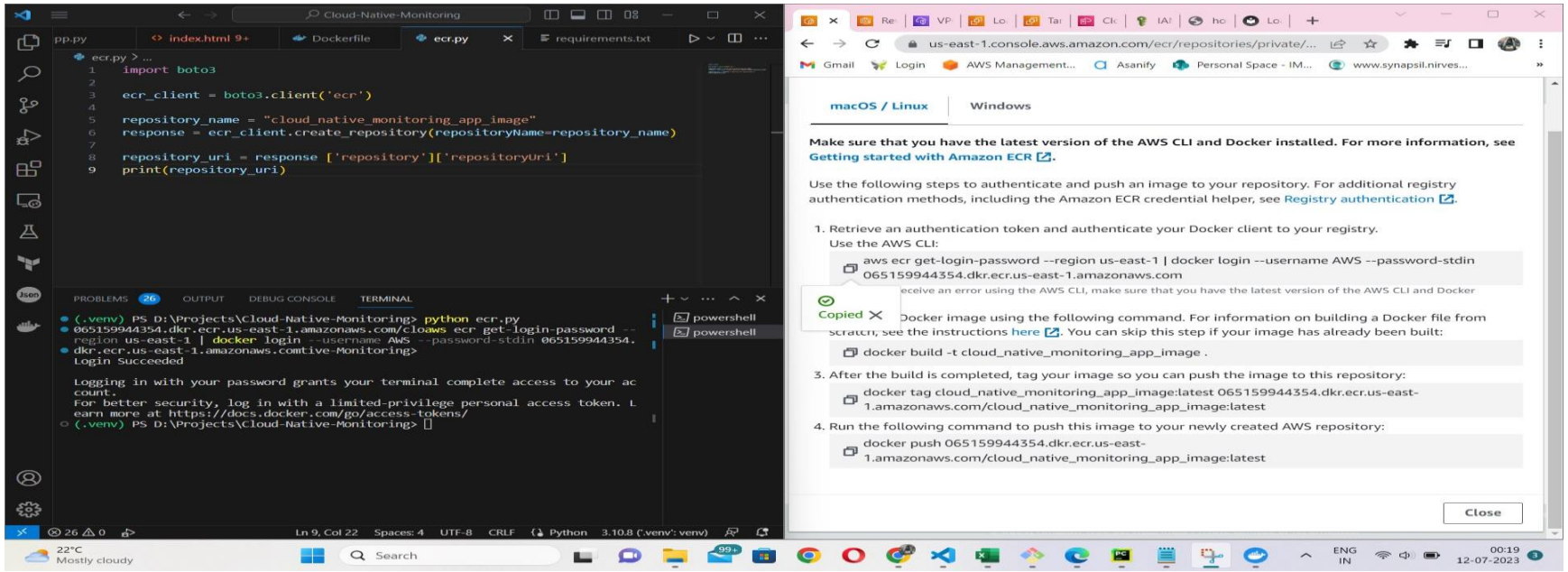
Create an ECR repository using Python:

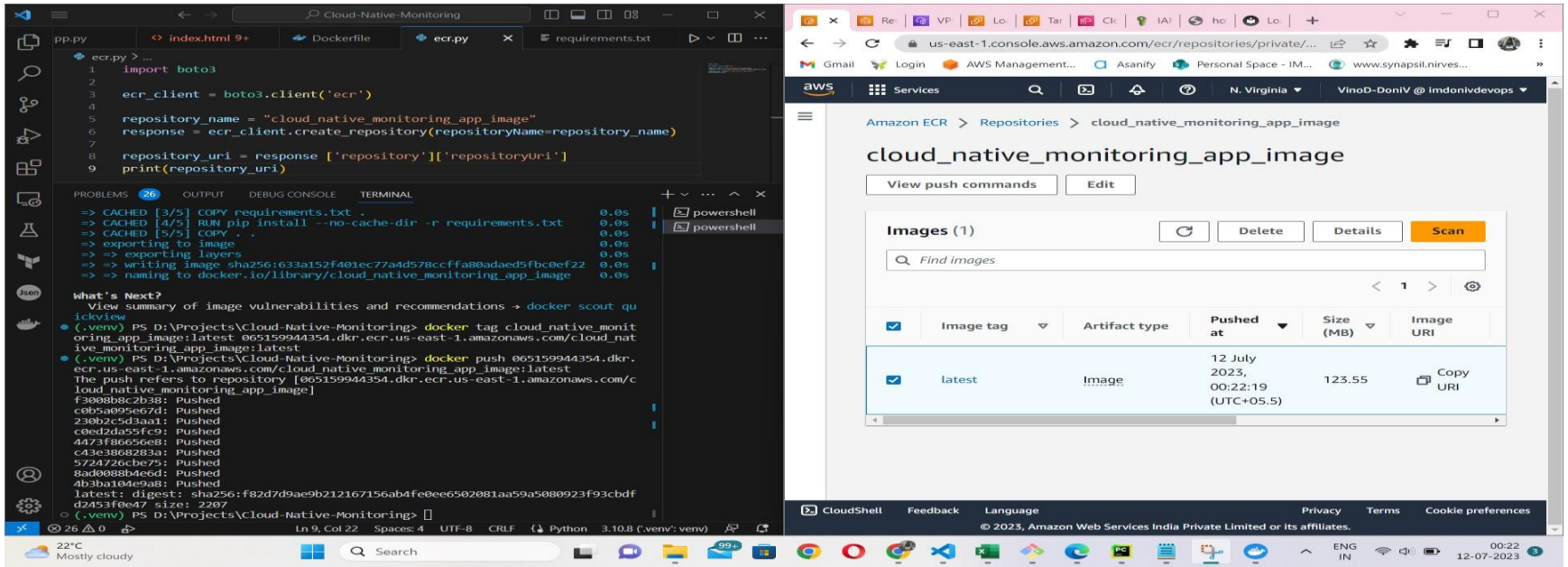
Create a **`**ecr.py**`** in the root directory of the project with the following contents:

```
import boto3
#Create an ECR Client
ecr_client = boto3.client(‘ecr’)
#Create a new ECR repository
repository_name = ‘cloud_native_monitoring_app_image’
response = ecr_client.create_repository(repositoryName=repository_name)
repository_uri = response [‘repository’][‘repositoryUri’]
print(repository_uri)
```



Push the Docker image to ECR using the push commands on the console:





Step 4:

Creating an Amazon EKS cluster and deploying the app using Python:

Before Creating the Cluster

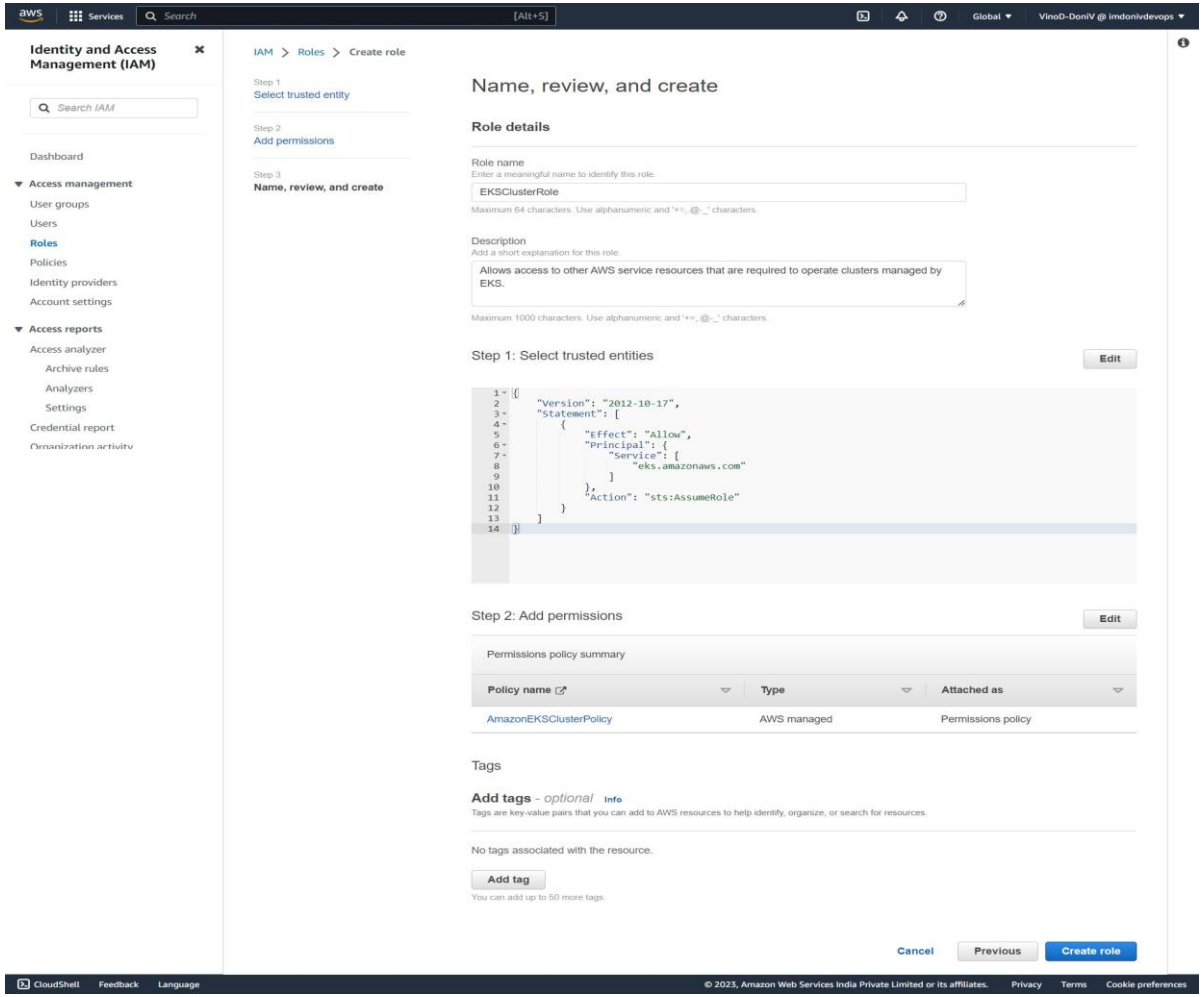
Prerequisites:

1. An IAM principal with permissions to create and describe an Amazon EKS cluster.
2. An existing VPC and subnets that meets Amazon EKS requirements.
3. AWS CLI installed and configured on your device.
4. The kubectl command line tool is installed on your device. The version can be the same as or up to one minor version earlier or later than the Kubernetes version of your cluster. For example, here I am use cluster version 1.27, we can use kubectl version 1.26, 1.27 or 1.28 with it.

To create an Amazon EKS cluster IAM role:

- Open the IAM console at https://console.aws.amazon.com/iam/
- In the navigation pane of the IAM console, Choose **Roles**, and then choose **Create role**.
- Under Trusted entity type, select **AWS service**.

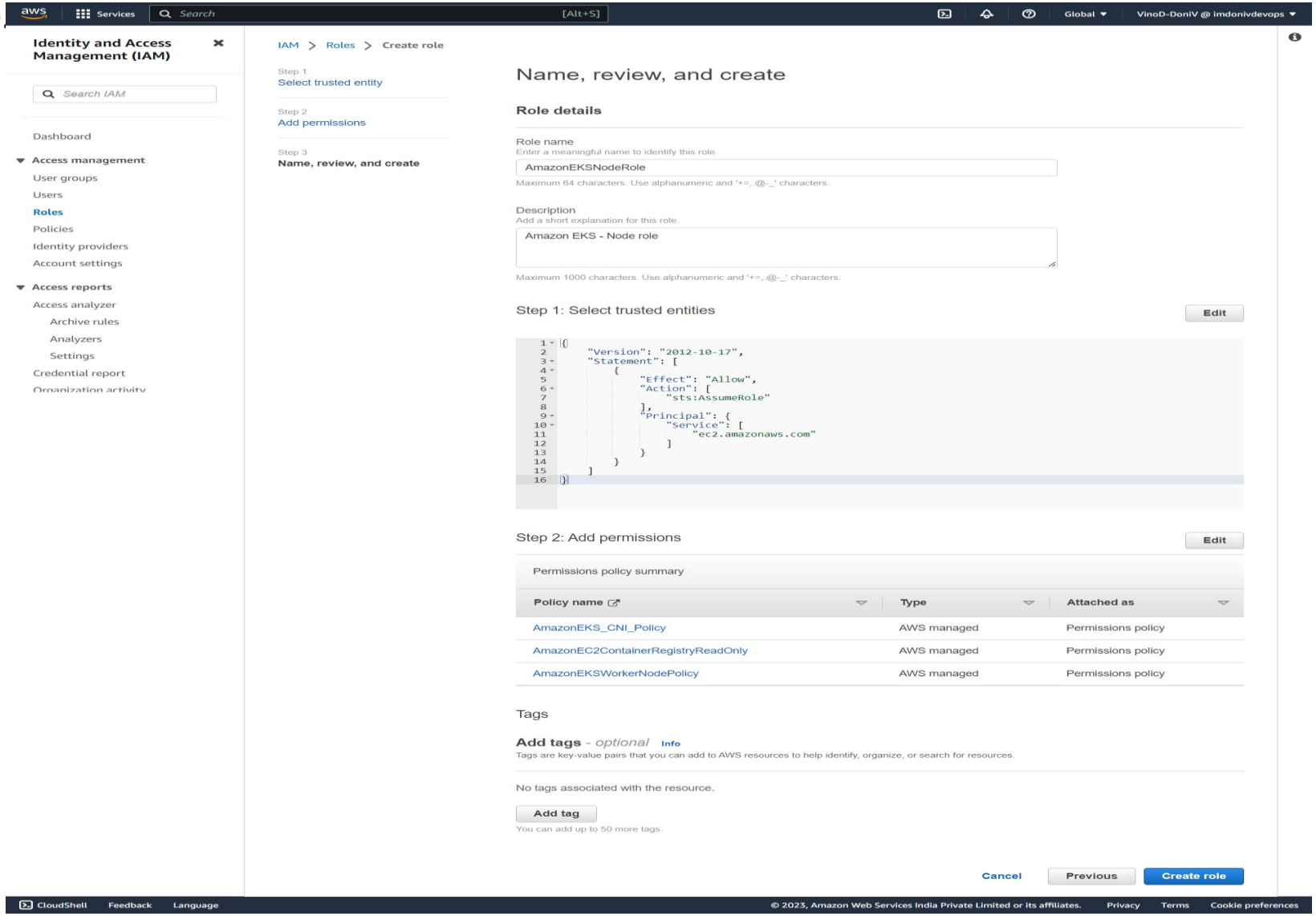
- From the Use cases for other AWS services dropdown list, choose **EKS**.
- Choose **EKS-Cluster** for your Use case, and then choose Next.
- On the Add permissions tab, choose Next.
- For Role name, enter a unique name for your role, such as **EKSClusterRole** .
- For Description, enter descriptive text.
- Chose **Create role**.



To create an Amazon EKS node IAM role:

- Open the IAM console at https://console.aws.amazon.com/iam/
- In the left navigation pane of the IAM console, Choose **Roles**, and then choose **Create role**.
- Under Trusted entity type, select **AWS service**.
- Under Use case, Choose **EC2**, Choose Next.
- On the Add permissions page , do the following:
    i. In the Filter policies box, enter **AmazonEKSWorkerNodePolicy**.
    ii. Select the check box to the left of **AmazonEKSWorkerNodePolicy** in the search results.
    iii. Choose Clear filters.
    iv. In the Filter policies box, enter **AmazonEC2ContainerRegistryReadOnly** in the search results. Either the **AmazonEKS_CNI_Policy** managed policy, or an IPv6 policy that you create must also be attached to either this role or to a different role that's mapped to the aws-node Kubernetes service account.
    v. Choose Next.

- On the Add permissions tab, choose Next.
- For Role name, enter a unique name for your role, such as **AmazonEKSNodeRole**.
- For Description, enter a descriptive text such as Amazon EKS-Node role.
- Choose Create role.

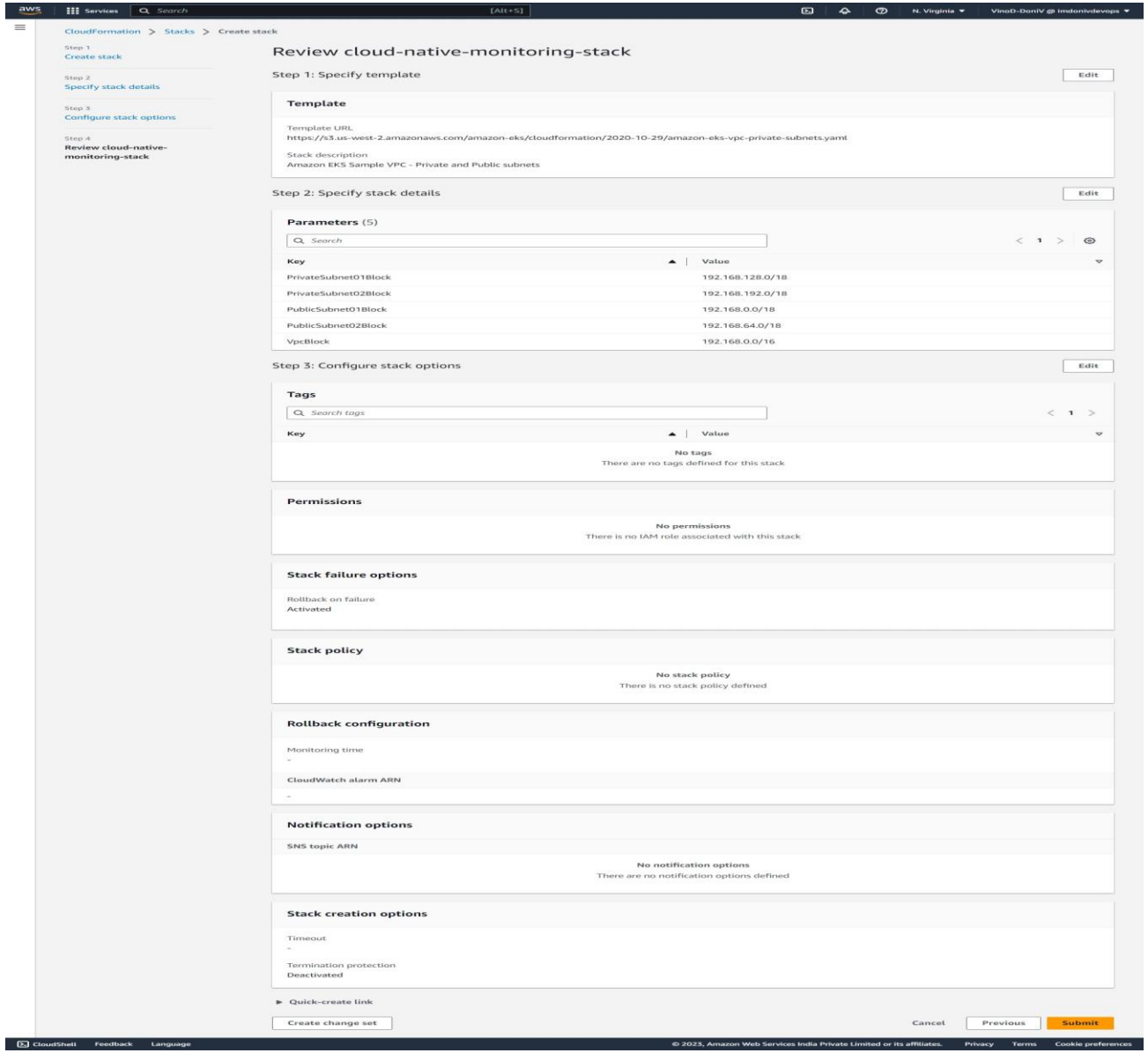


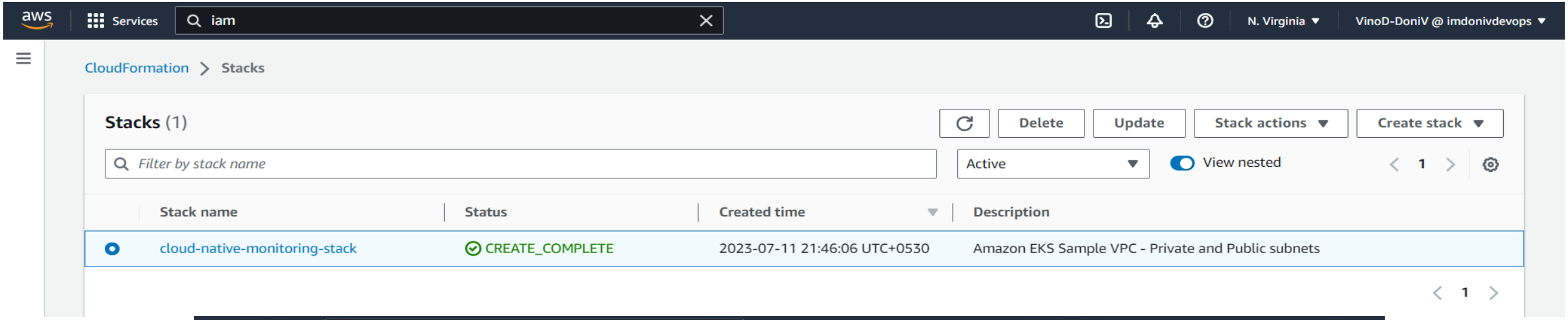
To create an Amazon VPC by using AWS CloudFormation Template:

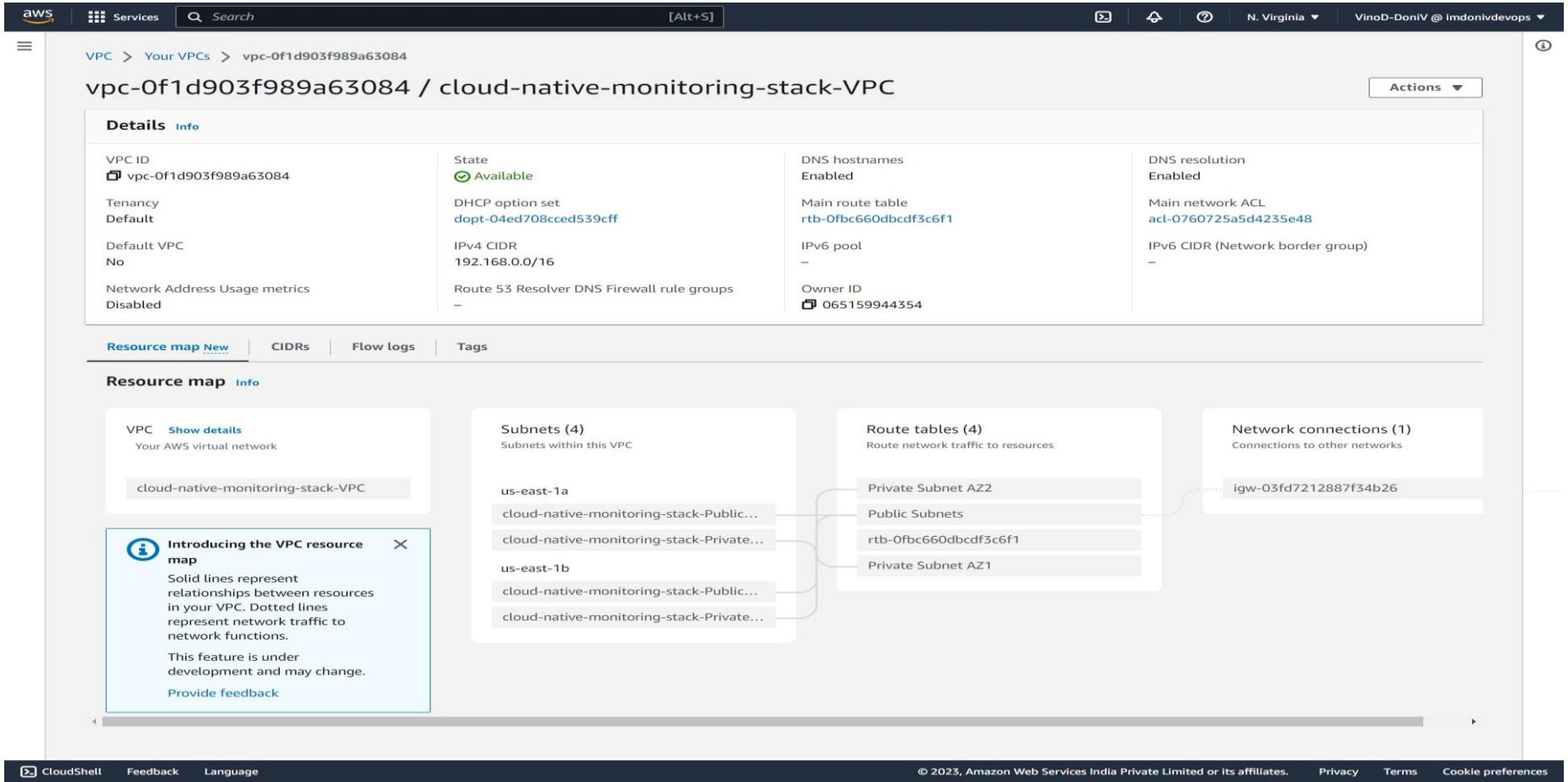
Deploy the VPC infrastructure :

- Open the CloudFormation console at https://console.aws.amazon.com/cloudformation/.
- Click Create Stack, then with **new resources(standard).**
- For Prerequisite-prepare template, Choose **Template ready**.
- For Specify template, Choose **Amazon S3 URL**.
- Give the URL **https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml.**
- Click Next.
- Enter the following details:
    i. For stack name, enter a name of the stack such as **cloud-native-monitoring-stack**.
    ii. For parameters: parameters may be left as defaults.
- At the bottom of the page click Next.
- Review the information for the stack, Check **I acknowledge that AWS CloudFormation might create IAM resources with custom names** then click **Create stack**

- After a few minutes the final stack status should change from CREATE_IN_PROGRESS to CREATE_COMPLETE. You can click the refresh button to check on the current status. We have now created the VPC stack.
- When the stack status is CREATED_COMPLETE, Go and see your VPC is ready.

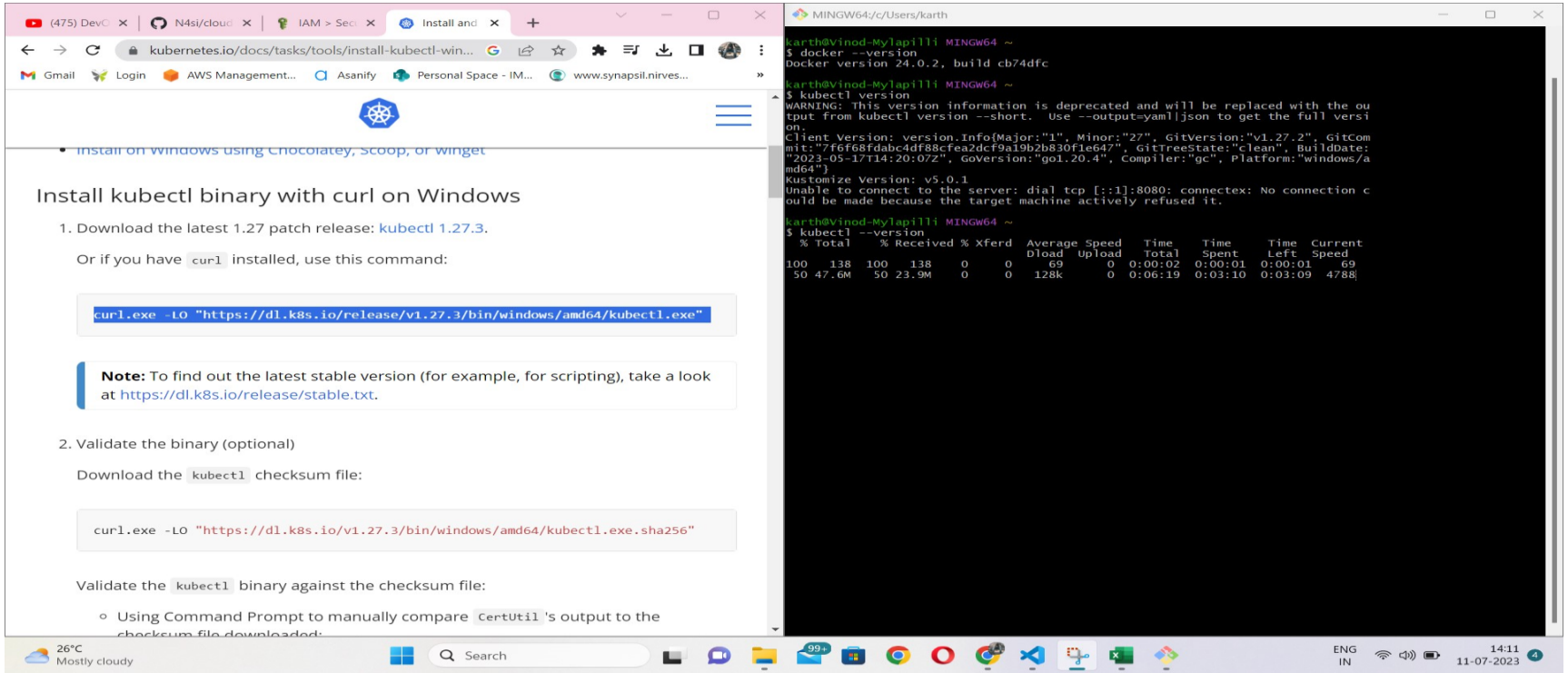



aws
Services
iam
N. Virginia
VinoD-DoniV @ imdonivdevops
CloudFormation > Stacks
Stacks (1)
Delete
Update
Stack actions
Create stack
Filter by stack name
Active
View nested
1
Stack name
Status
Created time
Description
cloud-native-monitoring-stack
CREATE_COMPLETE
2023-07-11 21:46:06 UTC+0530
Amazon EKS Sample VPC - Private and Public subnets
1


aws
Services
Search
[Alt+S]
N. Virginia
VinoD-DoniV @ imdonivdevops
VPC > Your VPCs > vpc-0f1d903f989a63084
vpc-0f1d903f989a63084 / cloud-native-monitoring-stack-VPC
Actions
Details Info
VPC ID
vpc-0f1d903f989a63084
State
Available
DNS hostnames
Enabled
DNS resolution
Enabled
Tenancy
Default
DHCP option set
dopt-04ed708cced539cff
Main route table
rtb-0fbc660dbcdf3c6f1
Main network ACL
acl-0760725a5d4235e48
Default VPC
No
IPv4 CIDR
192.168.0.0/16
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
065159944354
Resource map New
CIDRs
Flow logs
Tags
Resource map Info
VPC
Show details
Your AWS virtual network
cloud-native-monitoring-stack-VPC
Introducing the VPC resource map
Solid lines represent relationships between resources in your VPC. Dotted lines represent network traffic to network functions.
This feature is under development and may change.
Provide feedback
Subnets (4)
Subnets within this VPC
us-east-1a
cloud-native-monitoring-stack-Public...
cloud-native-monitoring-stack-Private...
us-east-1b
cloud-native-monitoring-stack-Public...
cloud-native-monitoring-stack-Private...
Route tables (4)
Route network traffic to resources
Private Subnet AZ2
Public Subnets
rtb-0fbc660dbcdf3c6f1
Private Subnet AZ1
Network connections (1)
Connections to other networks
igw-03fd7212887f34b26
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Kubectl Download and install on Windows:

- Download the latest version 1.2.7: **kubectl 1.27.3.**
- Create a folder in C-drive name as **kube**, move the **kubectl.exe** file in that folder name kube.
- Go to System properties, Choose **Environment Variables**
- For user variables, click new
    i. Give the **Variable name : Path** and **Variable value : kube**
    ii. Click ok.
- For System variables, click new
    i. Give the **Variable name : Path** and **Variable value : kube**
    ii. Click ok.
- Click OK.



**** Create an AWS EKS Cluster ****

1. Open the Amazon EKS Console at https://console.aws.amazon.com/eks/home#/clusters.
2. Choose Add Cluster and then choose Create.
3. On the Configure cluster page, enter the following fields:
   - For Cluster Configuration.
     - Name : cloud-native-monitoring-cluster
     - Kubernetes version : 1.27
     - Cluster Service role : EKSClusterRole
   - Rest all default leave it.
   - Click Next.
4. On the Specify networking, enter the following fields:
   - For Networking.
     - VPC : Existing VPC (cloud-native-monitoring-stack-vpc)
     - Subnets : Existing 4 subnets (2 public, 2 private)
     - Security Group : Existing Security group
     - Choose cluster Ip address family : IPv4
     - Configure Kubernetes service IP address range : Default
   - For Cluster endpoint access, Choose Public and private
   - Advance settings, by default leave it
5. Rest all leave it by default. (configure logging, select add-ons, configure selected add-ons settings), Click Create.

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Select add-ons

Step 5
Configure selected add-ons settings

Step 6
Review and create

Configure cluster

Cluster configuration Info

Name
Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

cloud-native-monitoring-cluster

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

Kubernetes version Info
Select the Kubernetes version for this cluster.

1.27

Cluster service role Info
Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This property cannot be changed after the cluster is created. To create a new role, follow the instructions in the Amazon EKS User Guide.

EKSClusterRole

Secrets encryption Info

Once turned on, secrets encryption cannot be modified or removed.

Turn on envelope encryption of Kubernetes secrets using KMS
Envelope encryption provides an additional layer of encryption for your Kubernetes secrets.

Tags (0) Info

This cluster does not have any tags.

Add tag

Remaining tags available to add: 50

Cancel Next

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Select add-ons

Step 5
Configure selected add-ons settings

Step 6
Review and create

Specify networking

Networking Info

These properties cannot be changed after the cluster is created.

VPC Info
Select a VPC to use for your EKS cluster resources.To create a new VPC, go to the VPC console.

vpc-0f1d903f989a63084 | cloud-native-monitoring-stack-VPC

Subnets Info
Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the VPC console.

Select subnets

subnet-043abd9162d0662f9 subnet-00e4d461f84b2716c
subnet-01db2fd91d39a4745 subnet-09a17c72c1e39342b

Security groups Info
Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker node subnets. To create a new security group, go to the corresponding page in the VPC console.

Select security groups

sg-0091a633913bc4aec

Choose cluster IP address family Info
Specify the IP address type for pods and services in your cluster.

- (•) IPv4
- ( ) IPv6

Configure Kubernetes service IP address range Info
Specify the range from which cluster services will receive IP addresses.

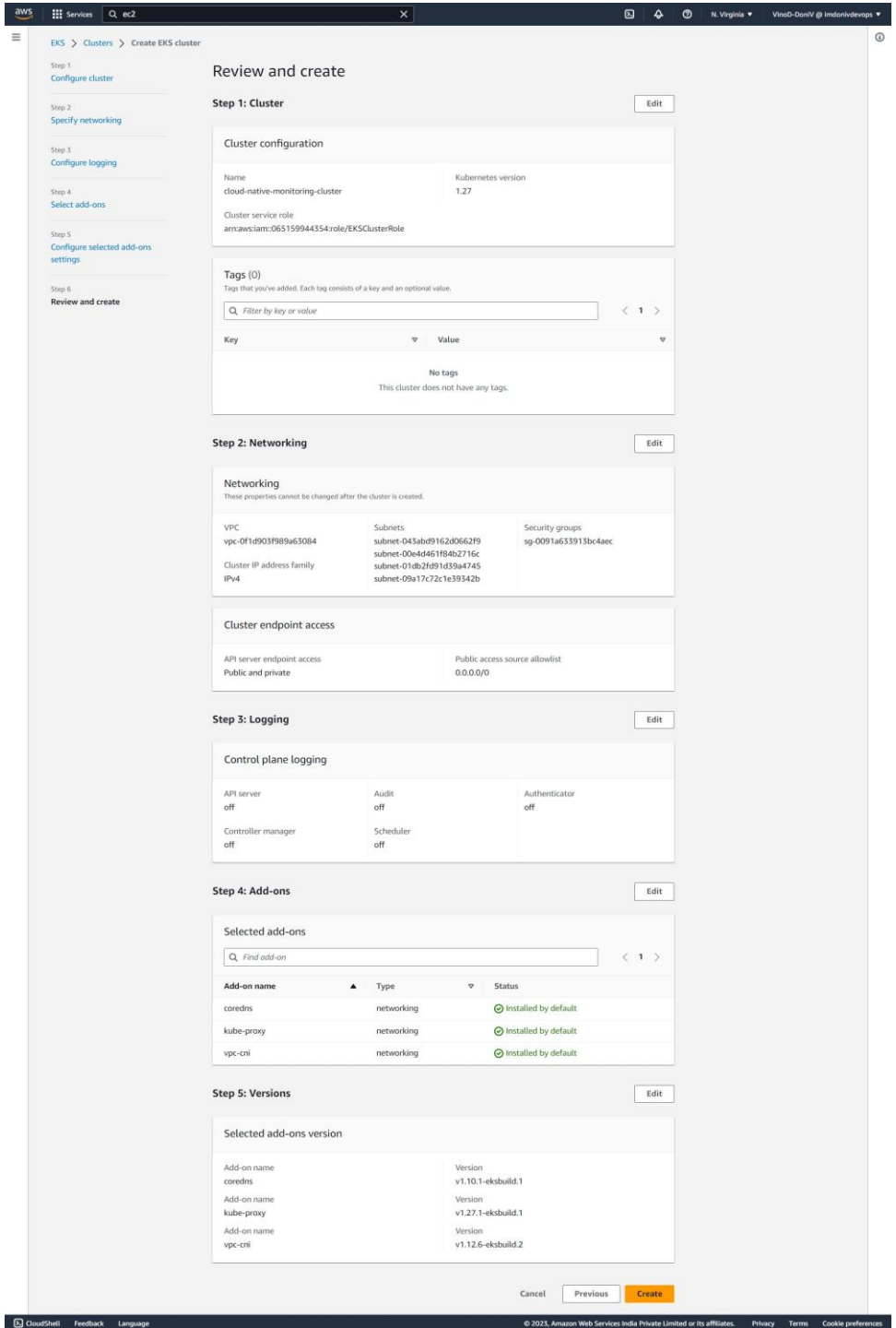
Cluster endpoint access Info

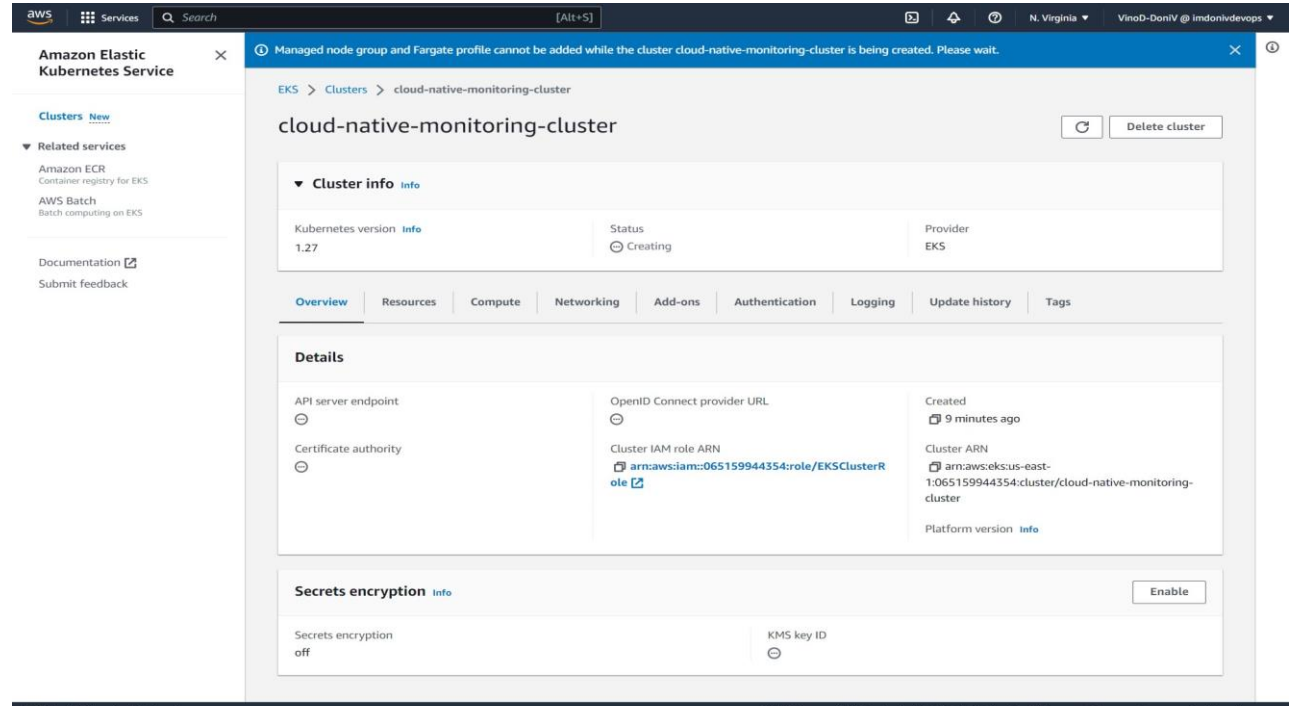
Configure access to the Kubernetes API server endpoint.

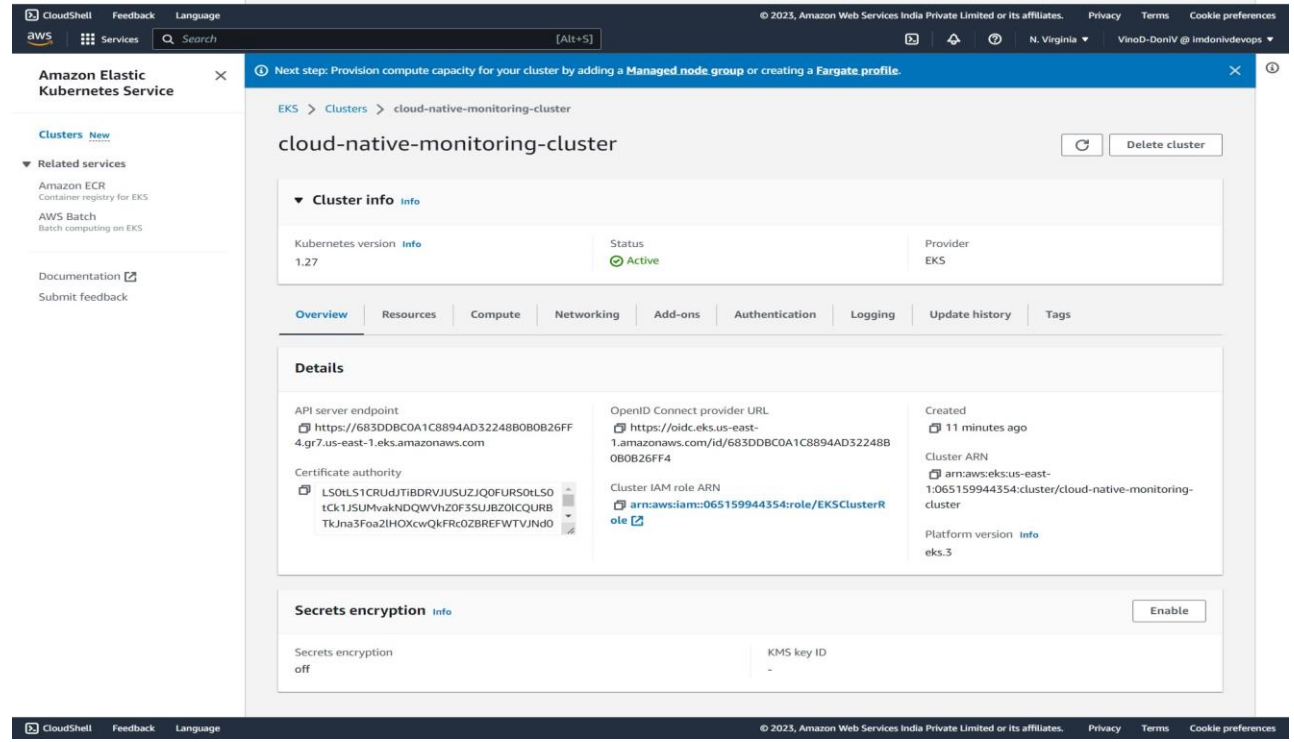
- ( ) Public
  The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.
- (•) Public and private
  The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.
- ( ) Private
  The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

▸ Advanced settings

Cancel Previous Next


Services
ec2
N. Virginia
VinoD-DoniV @ imdonivdevops
EKS > Clusters > Create EKS cluster
Step 1
Configure cluster
Step 2
Specify networking
Step 3
Configure logging
Step 4
Select add-ons
Step 5
Configure selected add-ons settings
Step 6
Review and create
Review and create
Step 1: Cluster
Edit
Cluster configuration
Name
cloud-native-monitoring-cluster
Kubernetes version
1.27
Cluster service role
arn:aws:iam::065159944354:role/EKSClusterRole
Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.
Filter by key or value
Key
Value
No tags
This cluster does not have any tags.
Step 2: Networking
Edit
Networking
These properties cannot be changed after the cluster is created.
VPC
vpc-0f1d903f989a63084
Subnets
subnet-043abd9162d0662f9
subnet-00e4d461f84b2716c
subnet-01db2fd91d39a4745
subnet-09a17c72c1e39342b
Security groups
sg-0091a633913bc4aec
Cluster IP address family
IPv4
Cluster endpoint access
API server endpoint access
Public and private
Public access source allowlist
0.0.0.0/0
Step 3: Logging
Edit
Control plane logging
API server
off
Audit
off
Authenticator
off
Controller manager
off
Scheduler
off
Step 4: Add-ons
Edit
Selected add-ons
Find add-on
Add-on name
Type
Status
coredns
networking
Installed by default
kube-proxy
networking
Installed by default
vpc-cni
networking
Installed by default
Step 5: Versions
Edit
Selected add-ons version
Add-on name
coredns
Version
v1.10.1-eksbuild.1
Add-on name
kube-proxy
Version
v1.27.1-eksbuild.1
Add-on name
vpc-cni
Version
v1.12.6-eksbuild.2
Cancel
Previous
Create
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
N. Virginia
VinoD-DoniV @ imdonivdevops
Amazon Elastic Kubernetes Service
Clusters New
Related services
Amazon ECR
Container registry for EKS
AWS Batch
Batch computing on EKS
Documentation
Submit feedback
Managed node group and Fargate profile cannot be added while the cluster cloud-native-monitoring-cluster is being created. Please wait.
EKS > Clusters > cloud-native-monitoring-cluster
cloud-native-monitoring-cluster
Delete cluster
Cluster info Info
Kubernetes version Info
1.27
Status
Creating
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Authentication
Logging
Update history
Tags
Details
API server endpoint
Certificate authority
OpenID Connect provider URL
Cluster IAM role ARN
arn:aws:iam::065159944354:role/EKSClusterRole
Created
9 minutes ago
Cluster ARN
arn:aws:eks:us-east-1:065159944354:cluster/cloud-native-monitoring-cluster
Platform version Info
Secrets encryption Info
Enable
Secrets encryption
off
KMS key ID


CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Services
Search
[Alt+S]
N. Virginia
VinoD-DoniV @ imdonivdevops
Amazon Elastic Kubernetes Service
Clusters New
Related services
Amazon ECR
Container registry for EKS
AWS Batch
Batch computing on EKS
Documentation
Submit feedback
Next step: Provision compute capacity for your cluster by adding a Managed node group or creating a Fargate profile.
EKS > Clusters > cloud-native-monitoring-cluster
cloud-native-monitoring-cluster
Delete cluster
Cluster info Info
Kubernetes version Info
1.27
Status
Active
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Authentication
Logging
Update history
Tags
Details
API server endpoint
https://683DDBC0A1C8894AD32248B0B0B26FF4.gr7.us-east-1.eks.amazonaws.com
Certificate authority
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0
OpenID Connect provider URL
https://oidc.eks.us-east-1.amazonaws.com/id/683DDBC0A1C8894AD32248B0B0B26FF4
Cluster IAM role ARN
arn:aws:iam::065159944354:role/EKSClusterRole
Created
11 minutes ago
Cluster ARN
arn:aws:eks:us-east-1:065159944354:cluster/cloud-native-monitoring-cluster
Platform version Info
eks.3
Secrets encryption Info
Enable
Secrets encryption
off
KMS key ID
-
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Create a managed node group

1. Wait for the cluster status to show as Active. Once its active.
2. Choose the name of the cluster that want to create a managed node group in.
3. Select the Compute tab.
4. Choose Add node group.
5. On the Configure node group page, enter the following fields.
    - For Node group configuration
        - Name : system-monitoring-node
        - Node IAM role : AmazonEKSNodeRole
    - Rest all leave as a default.
6. Click Next
7. On the Set compute and scaling configuration
    - For Node group compute configuration
        - AMI type : Amazon Linux 2(AL2_x86_64)
        - Capacity type : On-Demand
        - Instance types : t2.micro
        - Disk size : 20 GiB
    - For Node group scaling configuration
        - Desired size : 2 nodes
        - Manimum size : 2 nodes
        - Maximum size : 2 nodes
    - For Node group update configuration
        - For Maximum unavailable, Choose Number, Value : 1 node
8. Click Next, Rest all Leave default., Click Create.

Step 1
Configure node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

system-monitoring-node

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

AmazonEKSNodeRole

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

Launch template Info

These properties cannot be changed after the node group is created.

Use launch template

Configure this node group using an EC2 launch template.

Kubernetes labels Info

This node group does not have any labels.

Add label

Remaining labels available to add: 50

Kubernetes taints Info

This node group does not have any taints.

Add taint

Remaining taints available to add: 50

Tags (0) Info

This node group does not have any tags.

Add tag

Remaining tags available to add: 50

Cancel Next

Step 1
Configure node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Set compute and scaling configuration

Node group compute configuration

These properties cannot be changed after the node group is created.

AMI type Info

Select the EKS-optimized Amazon Machine Image for nodes.

Amazon Linux 2 (AL2_x86_64)

Capacity type

Select the capacity purchase option for this node group.

On-Demand

Instance types Info

Select instance types you prefer for this node group.

Select

t2.micro

vCPU: 1 vCPU Memory: 1 GiB Network: Low to Moderate Max ENI: 2 Max IPs: 4

Disk size

Select the size of the attached EBS volume for each node.

20 GiB

Node group scaling configuration

Desired size

Set the desired number of nodes that the group should launch with initially.

2 nodes

Minimum size

Set the minimum number of nodes that the group can scale in to.

2 nodes

Maximum size

Set the maximum number of nodes that the group can scale out to.

2 nodes

Node group update configuration Info

Maximum unavailable

Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

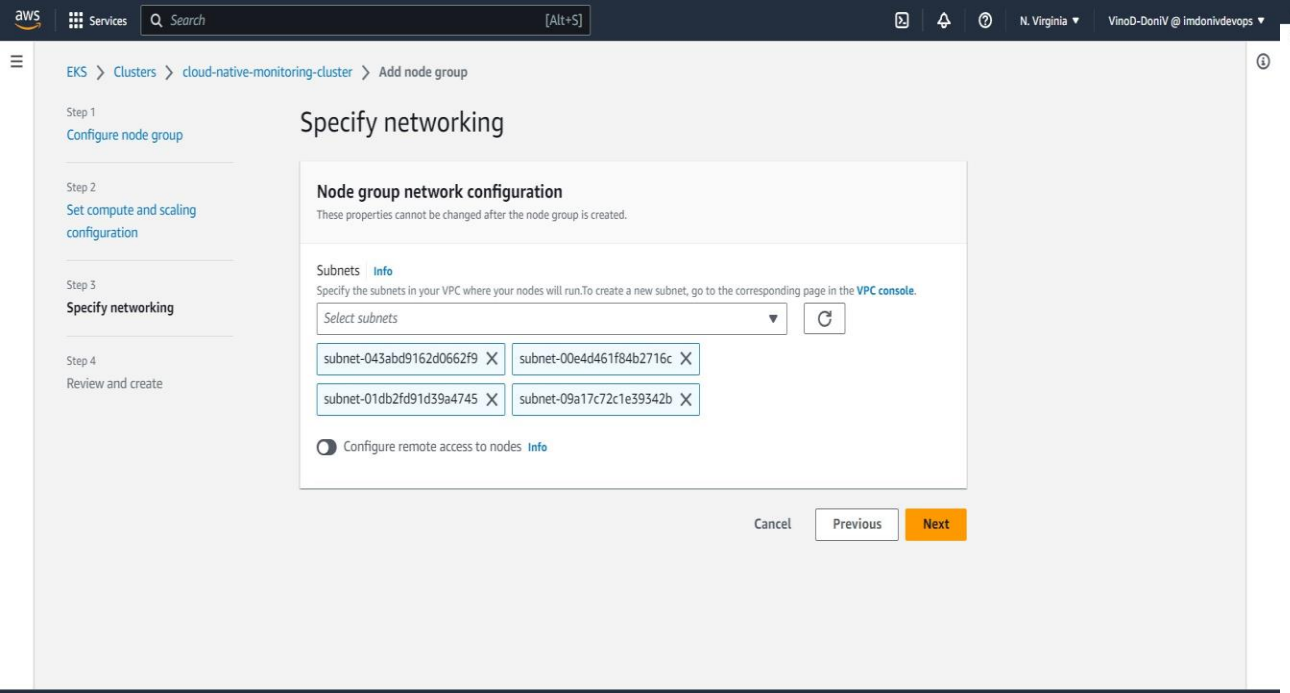
Number
Enter a number

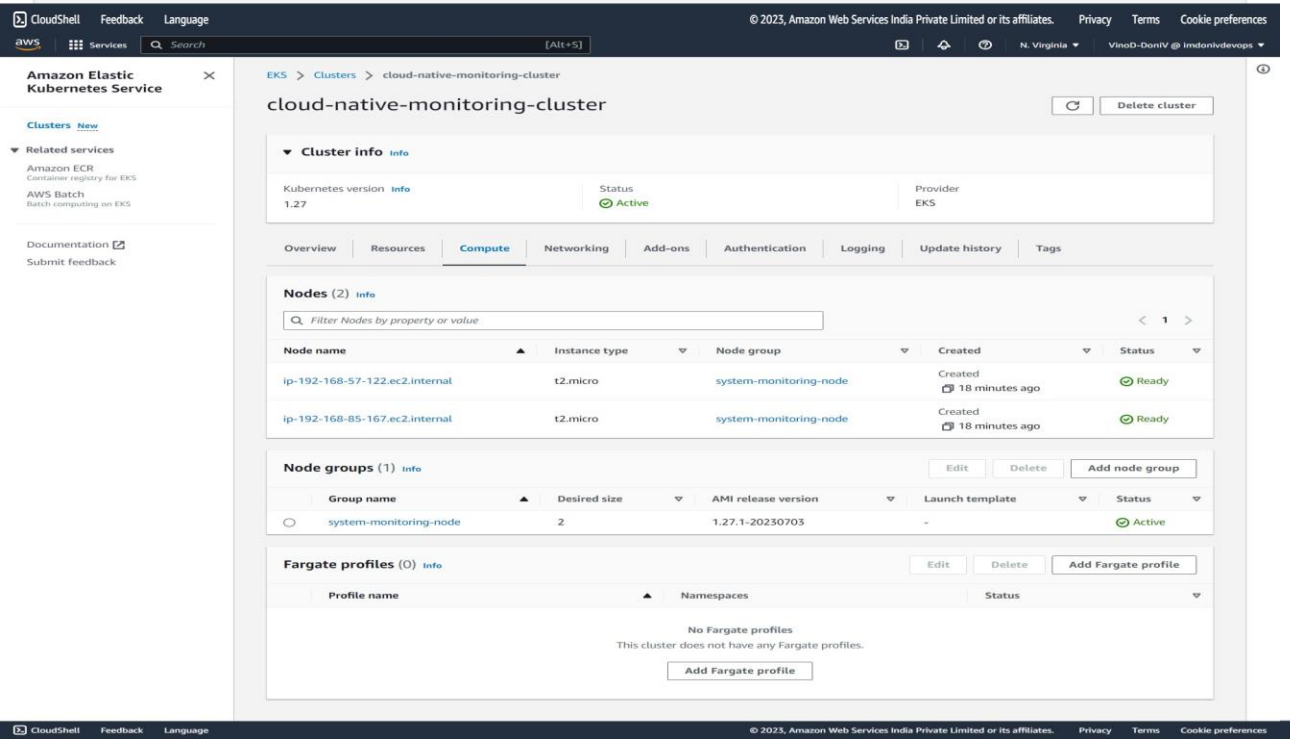
Percentage
Specify a percentage

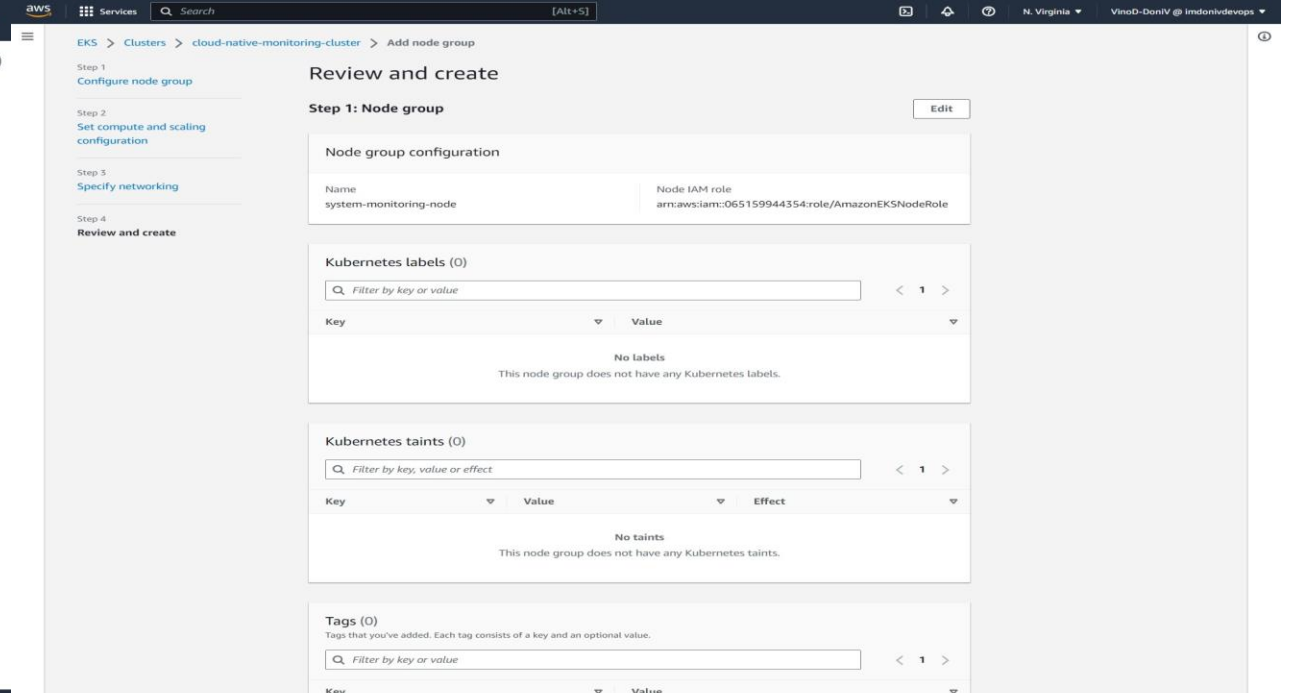
Value

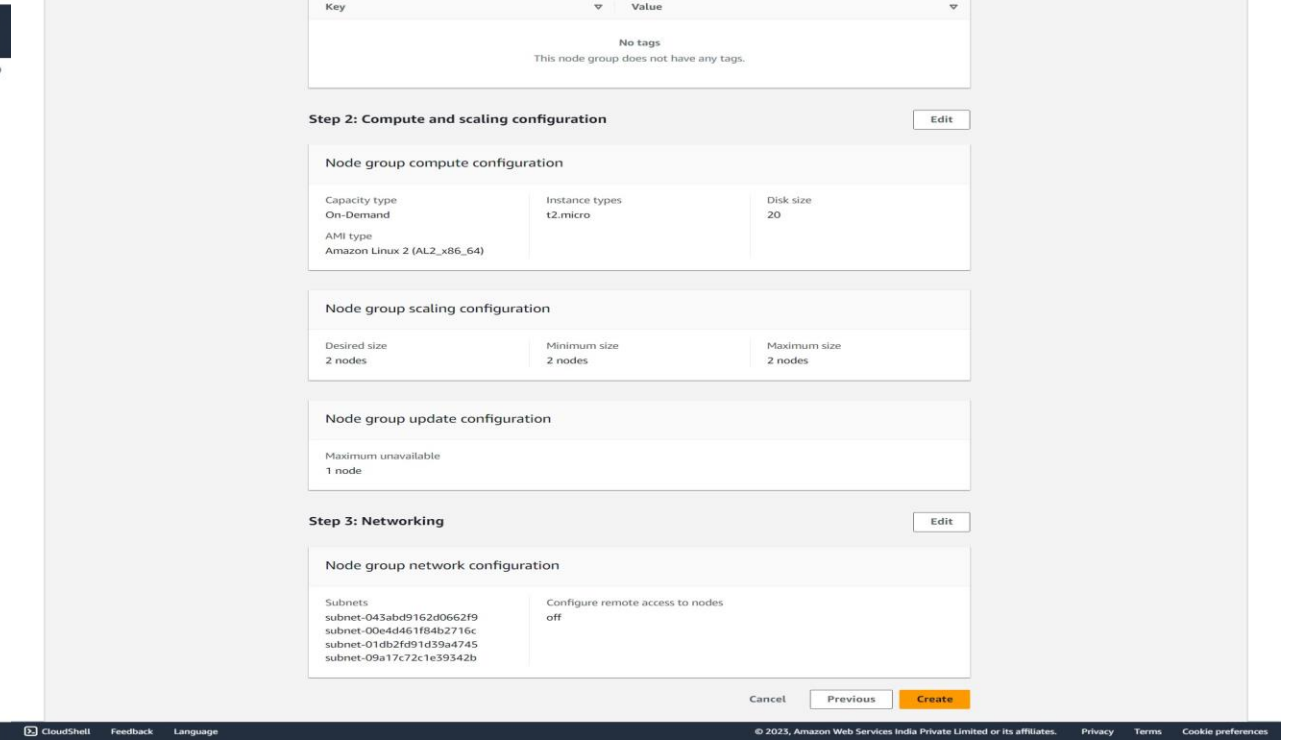
1 node

Cancel Previous Next


EKS > Clusters > cloud-native-monitoring-cluster > Add node group
Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create
Specify networking
Node group network configuration
These properties cannot be changed after the node group is created.
Subnets Info
Specify the subnets in your VPC where your nodes will run.To create a new subnet, go to the corresponding page in the VPC console.
Select subnets
subnet-043abd9162d0662f9
subnet-00e4d461f84b2716c
subnet-01db2fd91d39a4745
subnet-09a17c72c1e39342b
Configure remote access to nodes Info
Cancel
Previous
Next


CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
Amazon Elastic Kubernetes Service
Clusters New
Related services
Amazon ECR
Container registry for EKS
AWS Batch
Batch computing on EKS
Documentation
Submit feedback
EKS > Clusters > cloud-native-monitoring-cluster
cloud-native-monitoring-cluster
Delete cluster
Cluster info Info
Kubernetes version Info
1.27
Status
Active
Provider
EKS
Overview Resources Compute Networking Add-ons Authentication Logging Update history Tags
Nodes (2) Info
Filter Nodes by property or value
Node name
Instance type
Node group
Created
Status
ip-192-168-57-122.ec2.internal
t2.micro
system-monitoring-node
Created 18 minutes ago
Ready
ip-192-168-85-167.ec2.internal
t2.micro
system-monitoring-node
Created 18 minutes ago
Ready
Node groups (1) Info
Edit
Delete
Add node group
Group name
Desired size
AMI release version
Launch template
Status
system-monitoring-node
2
1.27.1-20230703
-
Active
Fargate profiles (0) Info
Edit
Delete
Add Fargate profile
Profile name
Namespaces
Status
No Fargate profiles
This cluster does not have any Fargate profiles.
Add Fargate profile


EKS > Clusters > cloud-native-monitoring-cluster > Add node group
Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create
Review and create
Step 1: Node group
Edit
Node group configuration
Name
system-monitoring-node
Node IAM role
arn:aws:iam::065159944354:role/AmazonEKSNodeRole
Kubernetes labels (0)
Filter by key or value
Key
Value
No labels
This node group does not have any Kubernetes labels.
Kubernetes taints (0)
Filter by key, value or effect
Key
Value
Effect
No taints
This node group does not have any Kubernetes taints.
Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.
Filter by key or value
Key
Value
No tags
This node group does not have any tags.
Step 2: Compute and scaling configuration
Edit
Node group compute configuration
Capacity type
On-Demand
Instance types
t2.micro
Disk size
20
AMI type
Amazon Linux 2 (AL2_x86_64)
Node group scaling configuration
Desired size
2 nodes
Minimum size
2 nodes
Maximum size
2 nodes
Node group update configuration
Maximum unavailable
1 node
Step 3: Networking
Edit
Node group network configuration
Subnets
subnet-043abd9162d0662f9
subnet-00e4d461f84b2716c
subnet-01db2fd91d39a4745
subnet-09a17c72c1e39342b
Configure remote access to nodes
off
Cancel
Previous
Create


CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

****Create deployment and service****

Create a **`**eks.py**`** in the root directory of the project with the following contents:

```
from kubernetes import client, config
# Load Kubernete configuration
config.load_kube_config()
# Create a Kubernetes API client
api_client=clientApiClient()
#Define the deployment
deployment = client.V1deployment(
        metadata = client.V1ObjectMeta(name="system-monitoring-app"),
        spec = client.V1DeploymentSpec(
                replicas=1,
                selector=client.V1LabelSelector(
                        match_labels={"app" : "system-monitoring-app"}
                ),
```

```
                template=client.V1PodTemplateSpec(
                        metadata=client.V1ObjectMeta(
                                labels={"app": "system-monitoring-app"}
                        ),
                        spec=client.V1PodSpec(
                                containers=[
                                        client.V1Container(
                                                name="system-monitoring-container",
                                                image="065159944354.dkr.ecr.us-east-
                                                1.amazonaws.com/cloud_native_monitoring_app_image:latest",
                                                ports=[client.V1ContainerPort(container_port=5000)]
                                        )
                                ]
                        )
                )
        )
)


# Create the deployment
api_instance = client.AppsV1Api(api_client)
api_instance.create_namespaced_deployment(
    namespace="default",
    body=deployment
)
```

```
# Define the service
service = client.V1Service(
    metadata=client.V1ObjectMeta(name="system-monitoring-service"),
    spec=client.V1ServiceSpec(
        selector={"app": "system-monitoring-app"},
        ports=[client.V1ServicePort(port=5000)]
    )
)



# Create the service
api_instance = client.CoreV1Api(api_client)
api_instance.create_namespaced_service(
    namespace="default",
    body=service
)
```

Once you run this file by running “**python eks.py**” deployment and service will be create.

$ python eks.py

Check by running following commands:

- **kubectl version –client**
- **kubectl get pods –n kube-system** # Get the pods of our system
- **aws eks update-kubeconfig –name cloud-native-monitoring-cluster**
- **kubectl get ns**
- **kubectl get deployment –n default** # Check Deployments
- **kubectl get service –n default** # Check service
- **kubectl get pods –n default –w** # Check the pods

Any Errors are come follow this command you may know what's the error:

- **kubectl describe pods system-monitoring-app-6874d98c77-4kfb5 –n default**

If you want edit anything in pods follow this command:

- **kubectl edit deployment system-monitoring-app –n default**

Finally Run the project follow this command:

- **kubectl port-forward svc/system-monitoring-service 5000:5000** # Here service name system-monitoring-service

MINGW64:/c/Users/karth

```
karth@Vinod-Mylapilli MINGW64 ~
$ kubectl get deployment -n default
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
system-monitoring-app   1/1     1            1           37s

karth@Vinod-Mylapilli MINGW64 ~
$
```

Cloud-Native-Monitoring

index.html 9+ | Dockerfile | ecr.py | eks.py | requirements.txt

eks.py > ...

```python
        )
    )

    # Create the deployment
    api_instance = client.AppsV1Api(api_client)
    api_instance.create_namespaced_deployment(
        namespace="default",
        body=deployment
    )

    # Define the service
    service = client.V1Service(
        metadata=client.V1ObjectMeta(name="system-monitoring-service"),
        spec=client.V1ServiceSpec(
            selector={"app": "system-monitoring-app"},
            ports=[client.V1ServicePort(port=5000)]
        )
    )

    # Create the service
    api_instance = client.CoreV1Api(api_client)
```

PROBLEMS 26 | OUTPUT | DEBUG CONSOLE | TERMINAL

```
(.venv) PS D:\Projects\Cloud-Native-Monitoring> python eks.py
(.venv) PS D:\Projects\Cloud-Native-Monitoring>
```

powershell
powershell

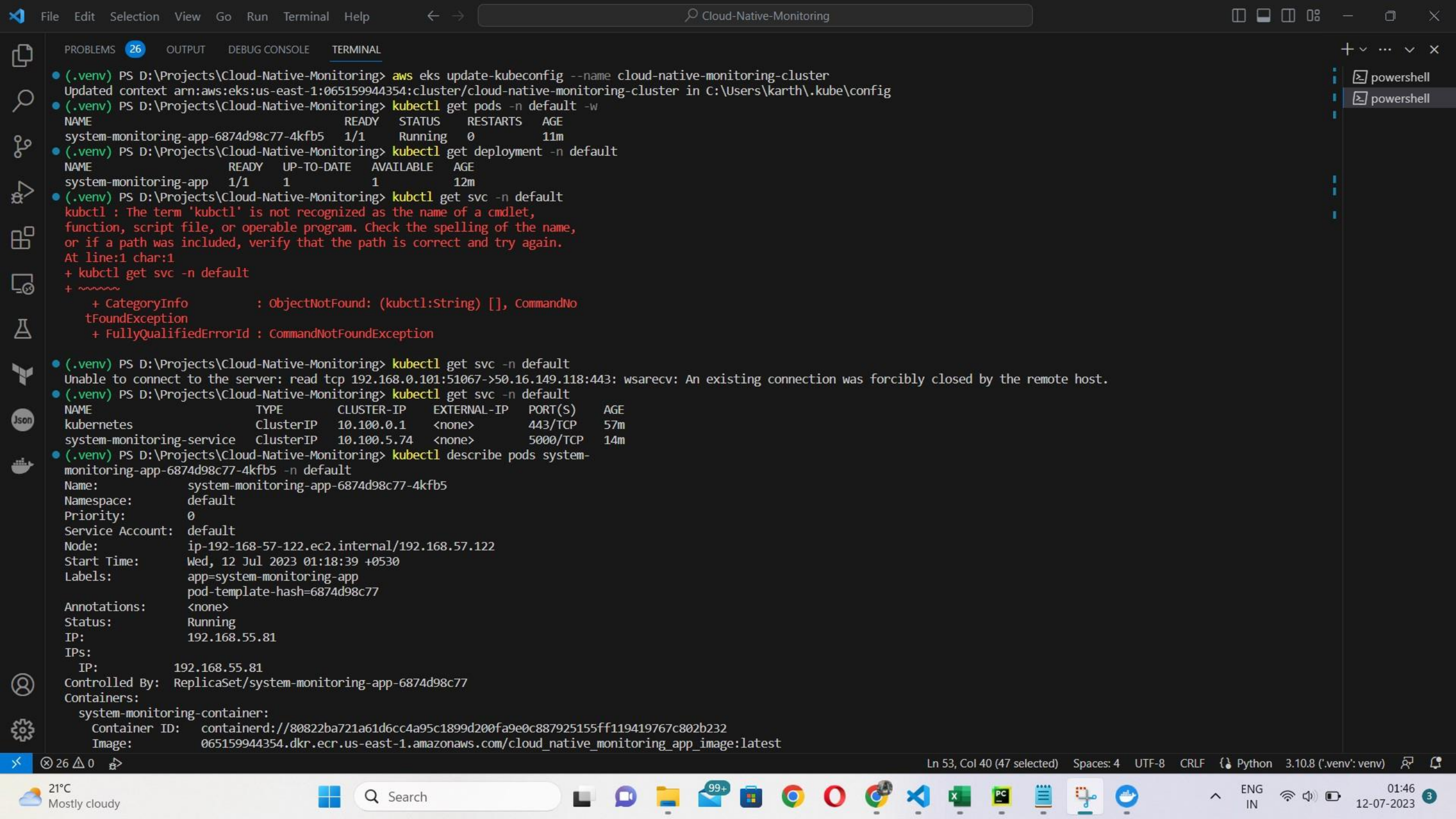
Ln 54, Col 25 | Spaces: 4 | UTF-8 | CRLF | Python | 3.10.8 ('.venv': venv)

21°C Mostly cloudy

Search

ENG IN

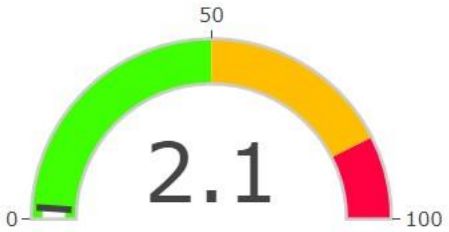
01:19 12-07-2023


File Edit Selection View Go Run Terminal Help
Cloud-Native-Monitoring
PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL
(.venv) PS D:\Projects\Cloud-Native-Monitoring> aws eks update-kubeconfig --name cloud-native-monitoring-cluster
Updated context arn:aws:eks:us-east-1:065159944354:cluster/cloud-native-monitoring-cluster in C:\Users\karth\.kube\config
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get pods -n default -w
NAME READY STATUS RESTARTS AGE
system-monitoring-app-6874d98c77-4kfb5 1/1 Running 0 11m
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get deployment -n default
NAME READY UP-TO-DATE AVAILABLE AGE
system-monitoring-app 1/1 1 1 12m
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubctl get svc -n default
kubctl : The term 'kubctl' is not recognized as the name of a cmdlet,
function, script file, or operable program. Check the spelling of the name,
or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ kubctl get svc -n default
+ ~~~~~~
+ CategoryInfo : ObjectNotFound: (kubctl:String) [], CommandNo
tFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get svc -n default
Unable to connect to the server: read tcp 192.168.0.101:51067->50.16.149.118:443: wsarecv: An existing connection was forcibly closed by the remote host.
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get svc -n default
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 57m
system-monitoring-service ClusterIP 10.100.5.74 <none> 5000/TCP 14m
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl describe pods system-
monitoring-app-6874d98c77-4kfb5 -n default
Name: system-monitoring-app-6874d98c77-4kfb5
Namespace: default
Priority: 0
Service Account: default
Node: ip-192-168-57-122.ec2.internal/192.168.57.122
Start Time: Wed, 12 Jul 2023 01:18:39 +0530
Labels: app=system-monitoring-app
pod-template-hash=6874d98c77
Annotations: <none>
Status: Running
IP: 192.168.55.81
IPs:
IP: 192.168.55.81
Controlled By: ReplicaSet/system-monitoring-app-6874d98c77
Containers:
system-monitoring-container:
Container ID: containerd://80822ba721a61d6cc4a95c1899d200fa9e0c887925155ff119419767c802b232
Image: 065159944354.dkr.ecr.us-east-1.amazonaws.com/cloud_native_monitoring_app_image:latest
powershell
powershell
Ln 53, Col 40 (47 selected) Spaces: 4 UTF-8 CRLF Python 3.10.8 ('.venv': venv)
21°C Mostly cloudy
Search
ENG IN
01:46 12-07-2023

```
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get svc -n default
NAME                        TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)
  AGE
kubernetes                  ClusterIP   10.100.0.1    <none>        443/TCP
  72m
system-monitoring-service   ClusterIP   10.100.5.74   <none>        5000/TCP
  29m
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl port-forward svc/syst
em-monitoring-service -p 5000:5000
error: unknown shorthand flag: 'p' in -p
See 'kubectl port-forward --help' for usage.
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl port-forward svc/syst
em-monitoring-service  5000:5000
Forwarding from 127.0.0.1:5000 -> 5000
Forwarding from [::1]:5000 -> 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
```

System Monitoring

CPU Utilization

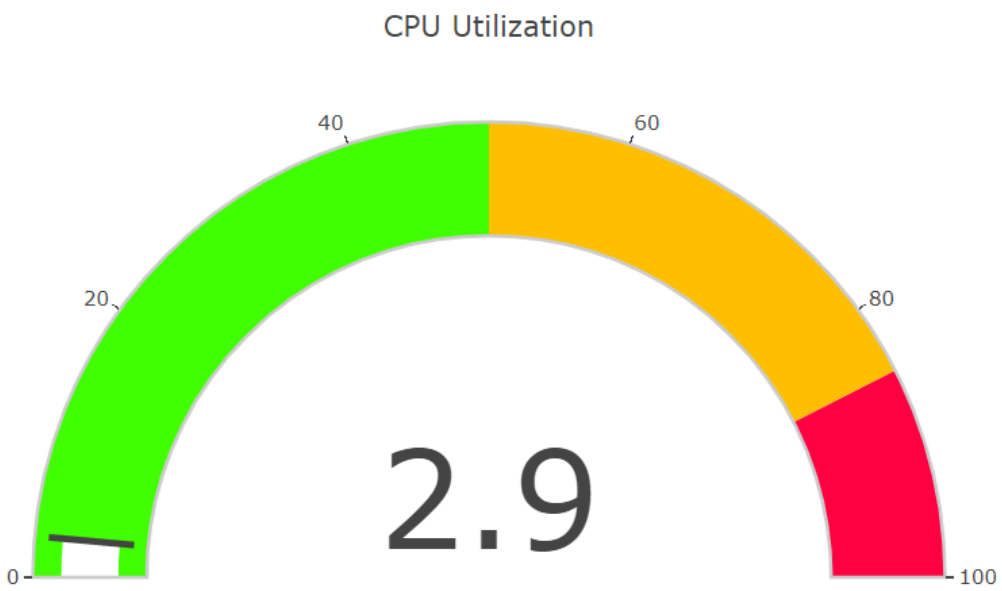


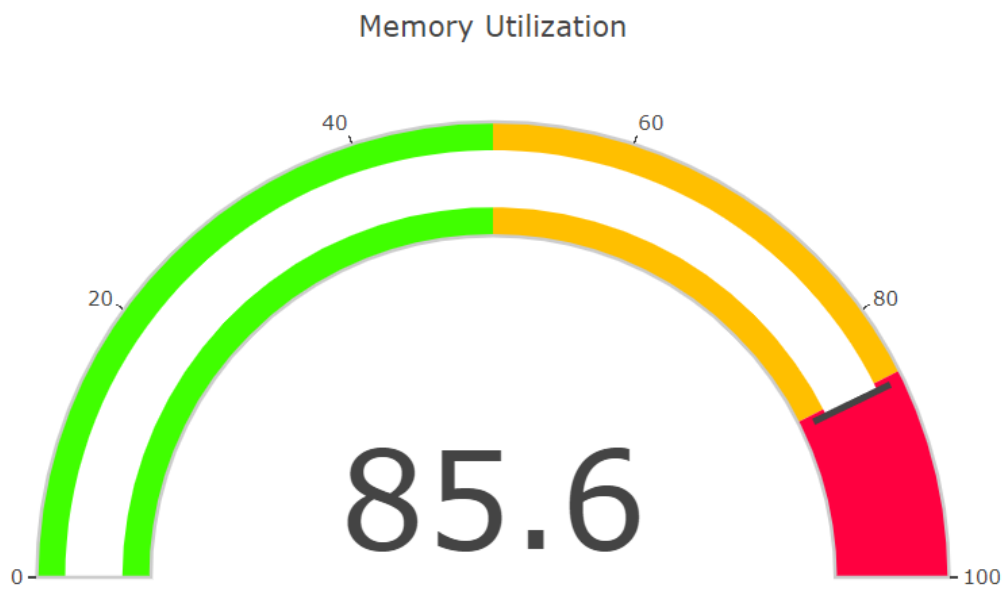
Memory Utilization

System Monitoring

Danger! Indicates a dangerous or potentially negative action.

High CPU or Memory Utilization detected. Please scale up.

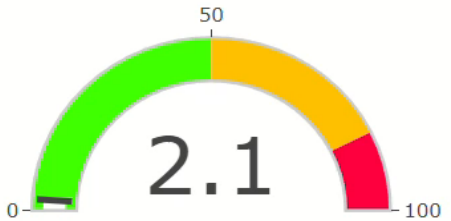




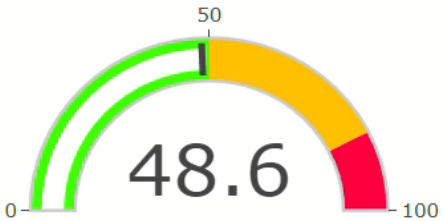
```
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl get svc -n default
NAME                        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)
  AGE
kubernetes                  ClusterIP   10.100.0.1     <none>        443/TCP
  72m
system-monitoring-service   ClusterIP   10.100.5.74    <none>        5000/TCP
  29m
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl port-forward svc/syst
em-monitoring-service -p 5000:5000
error: unknown shorthand flag: 'p' in -p
See 'kubectl port-forward --help' for usage.
(.venv) PS D:\Projects\Cloud-Native-Monitoring> kubectl port-forward svc/syst
em-monitoring-service  5000:5000
Forwarding from 127.0.0.1:5000 -> 5000
Forwarding from [::1]:5000 -> 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
Handling connection for 5000
```

System Monitoring

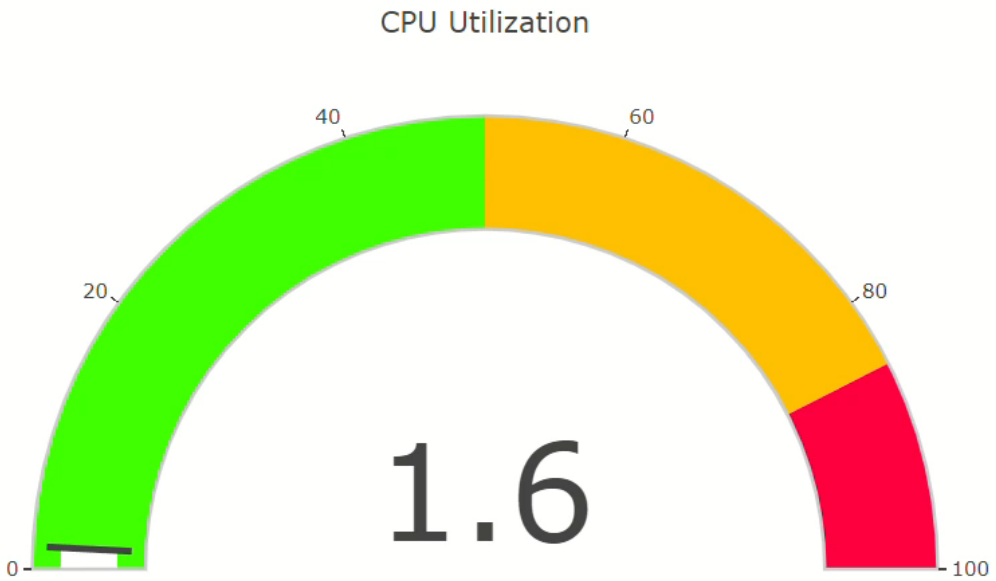
CPU Utilization

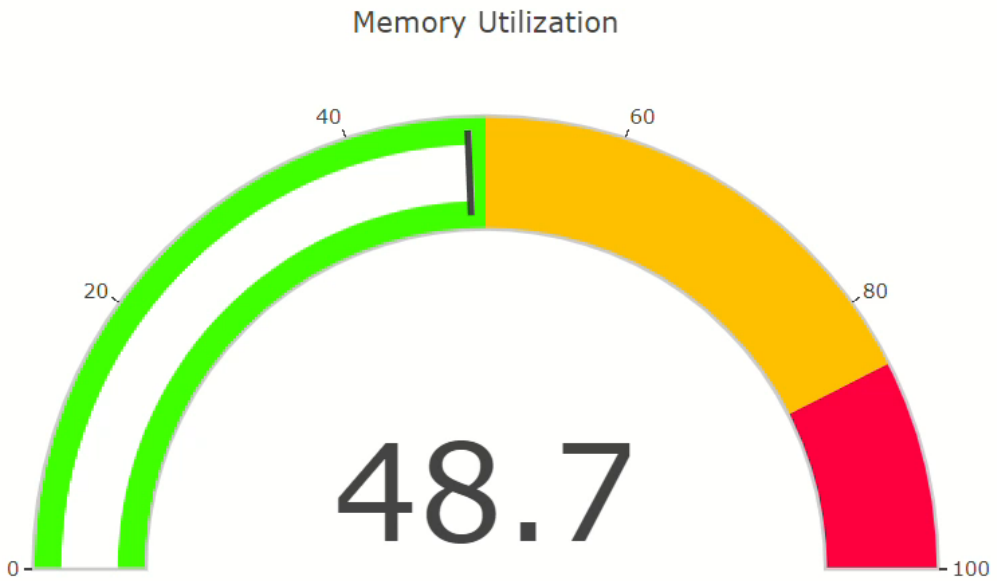


Memory Utilization



System Monitoring





Thank You !

See You Soon…….