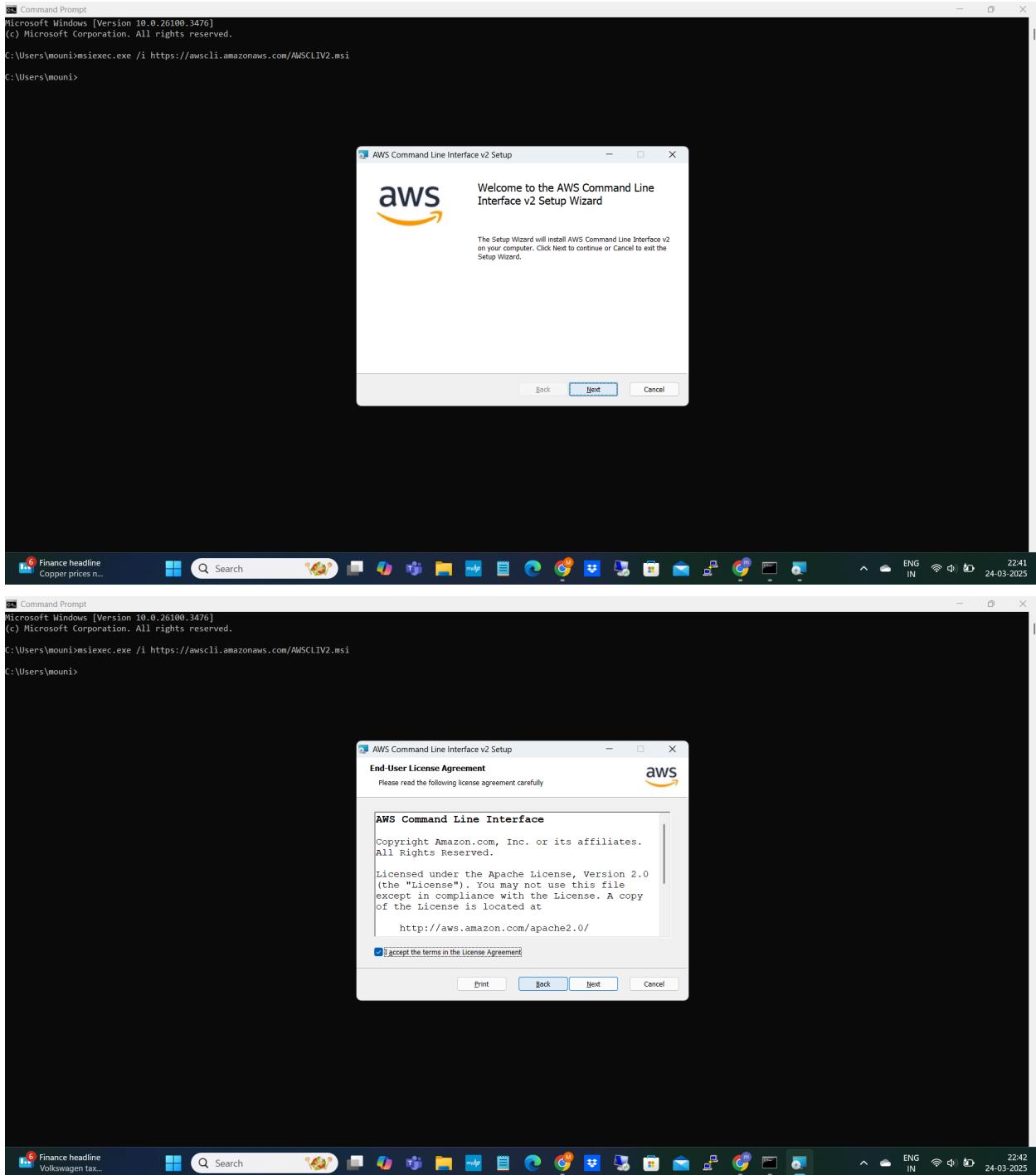
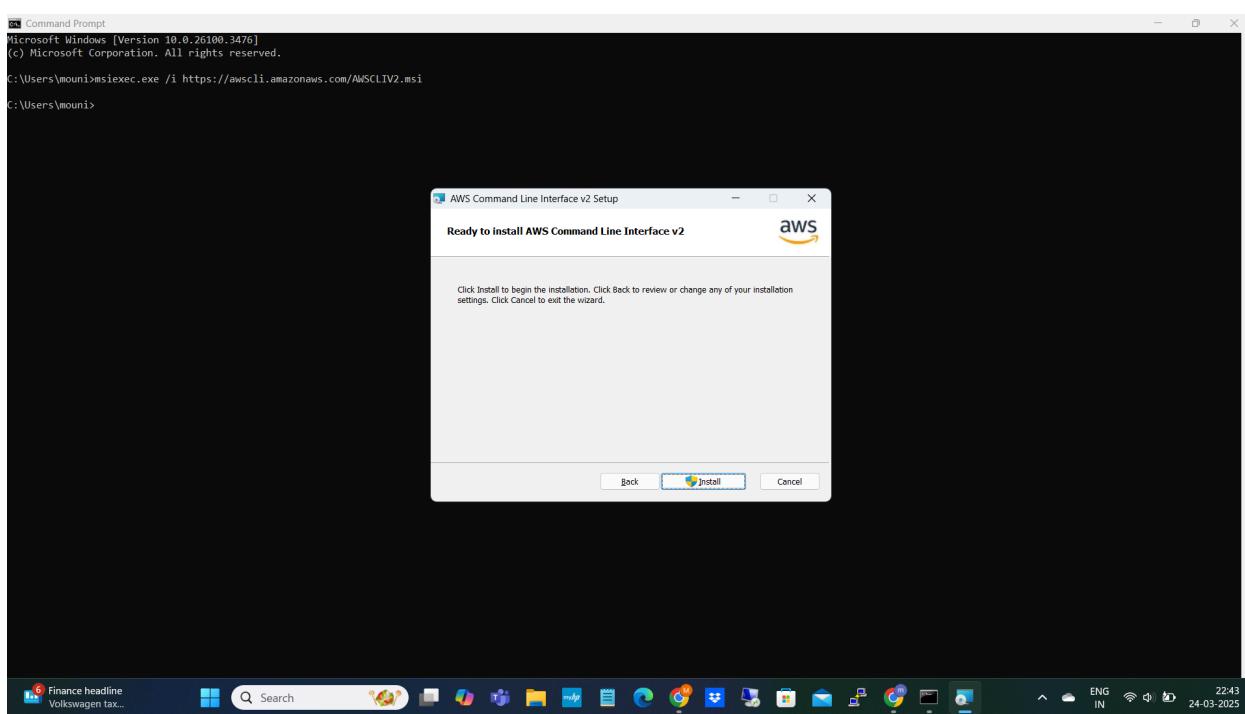
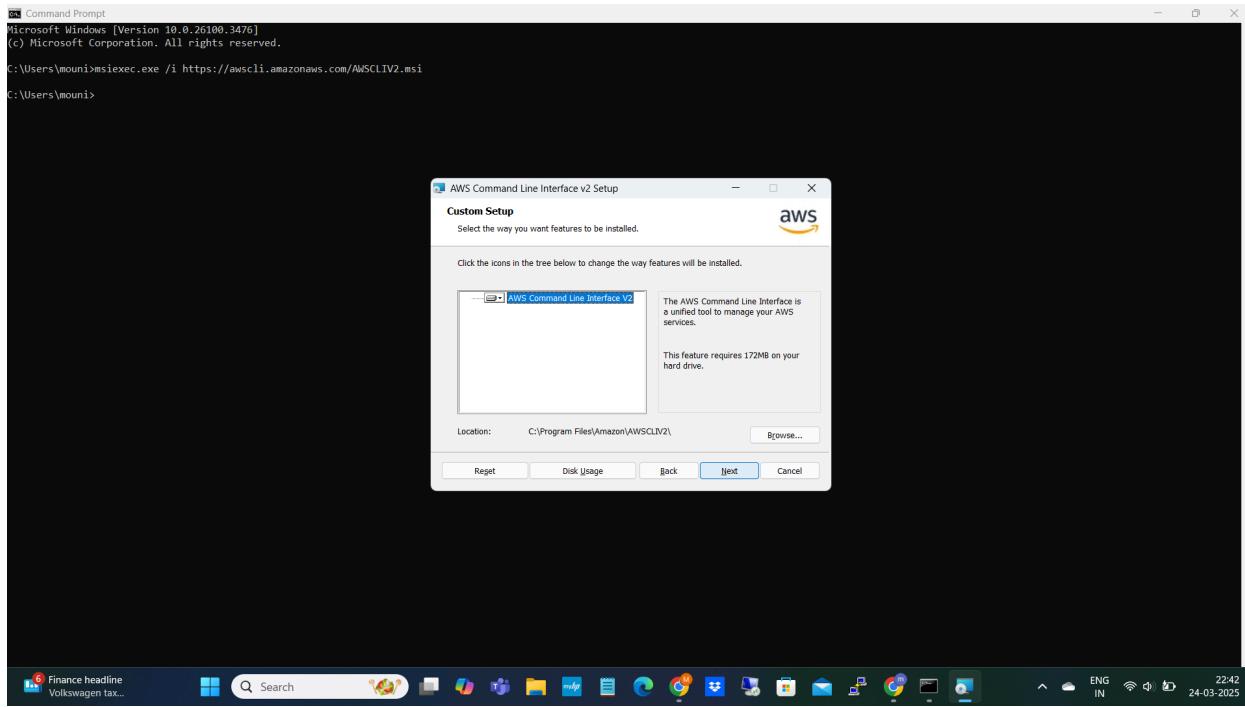


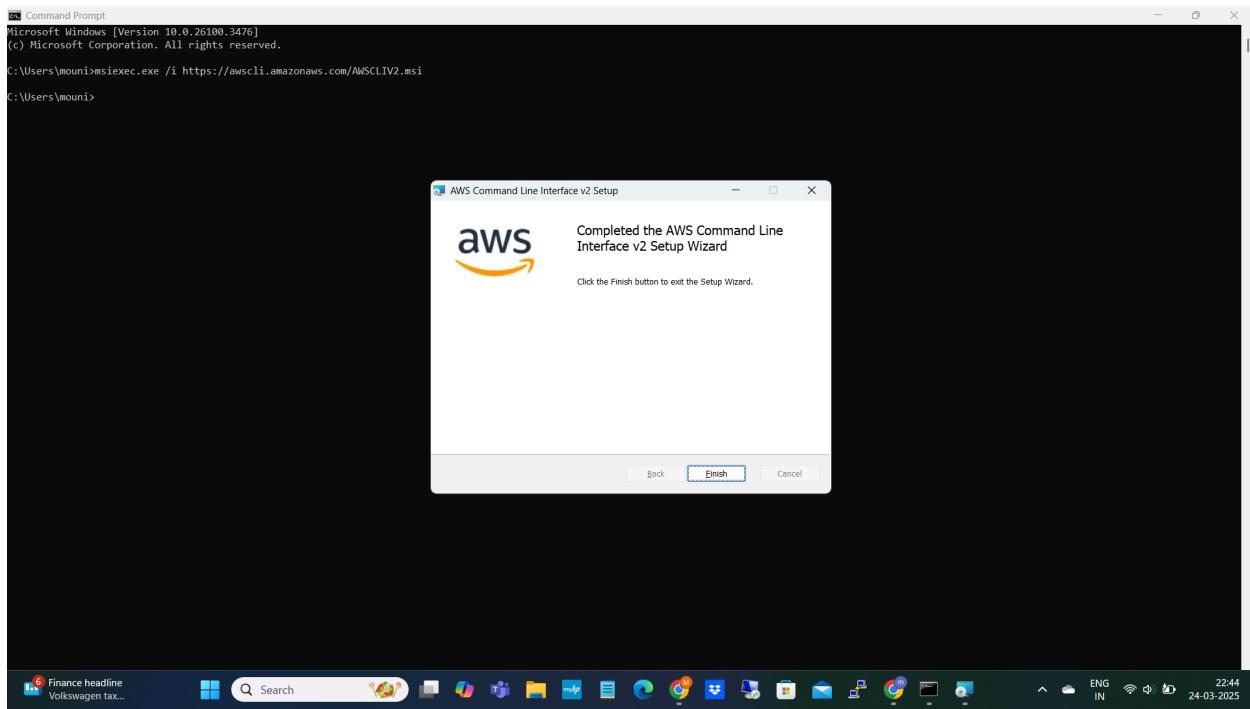
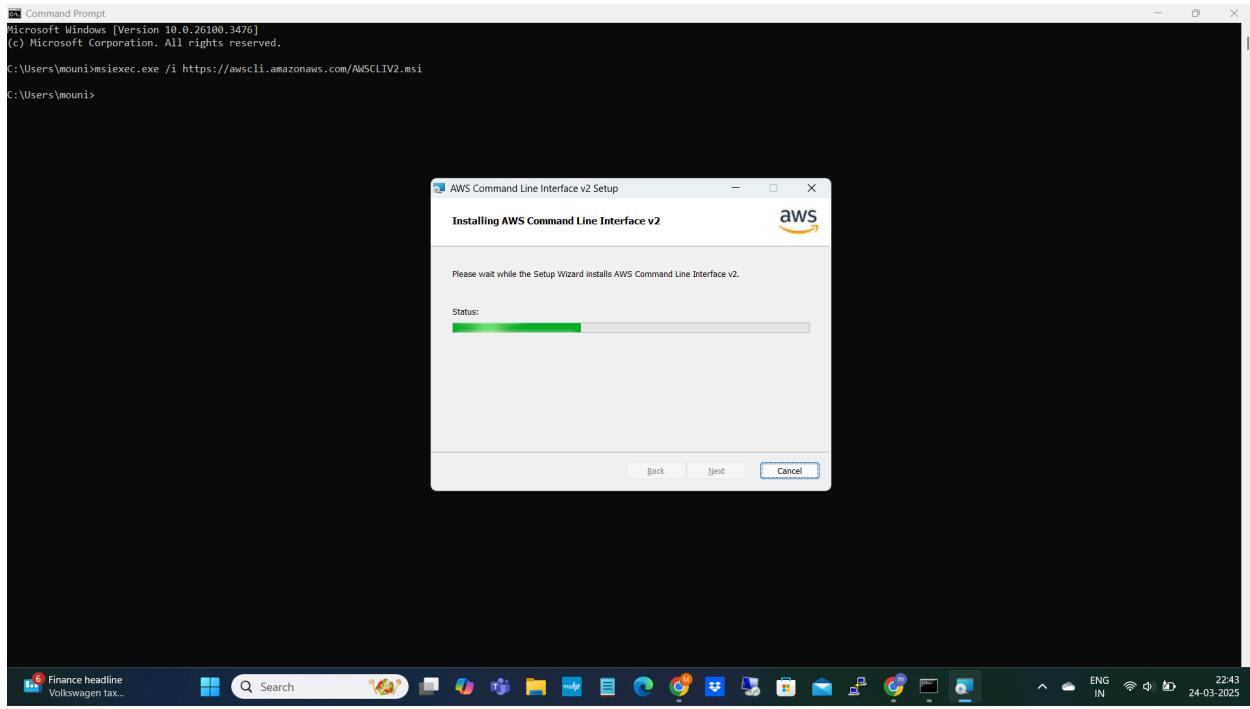
Kubernetes Task-2

Install AWS CLI:

```
$ msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```







Check the AWS CLI version:

```
$ aws --version
```

```
on Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>aws --version
aws-cli/2.25.1 Python/3.12.9 Windows/11 exe/AMD64

C:\Windows\System32>
```

Kubectl already installed in my local:

```
on Select Administrator: Command Prompt
C:\Windows\System32>kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create      Create a resource from a file or from stdin
  expose     Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service
  run        Run a particular image on the cluster
  set        Set specific features on objects

Basic Commands (Intermediate):
  explain    Get documentation for a resource
  get        Display one or more resources
  edit       Edit a resource on the server
  delete    Delete resources by file names, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout   Manage the rollout of a resource
  scale     Set a new size for a deployment, replica set, or replication controller
  autoscale Auto-scale a deployment, replica set, stateful set, or replication controller

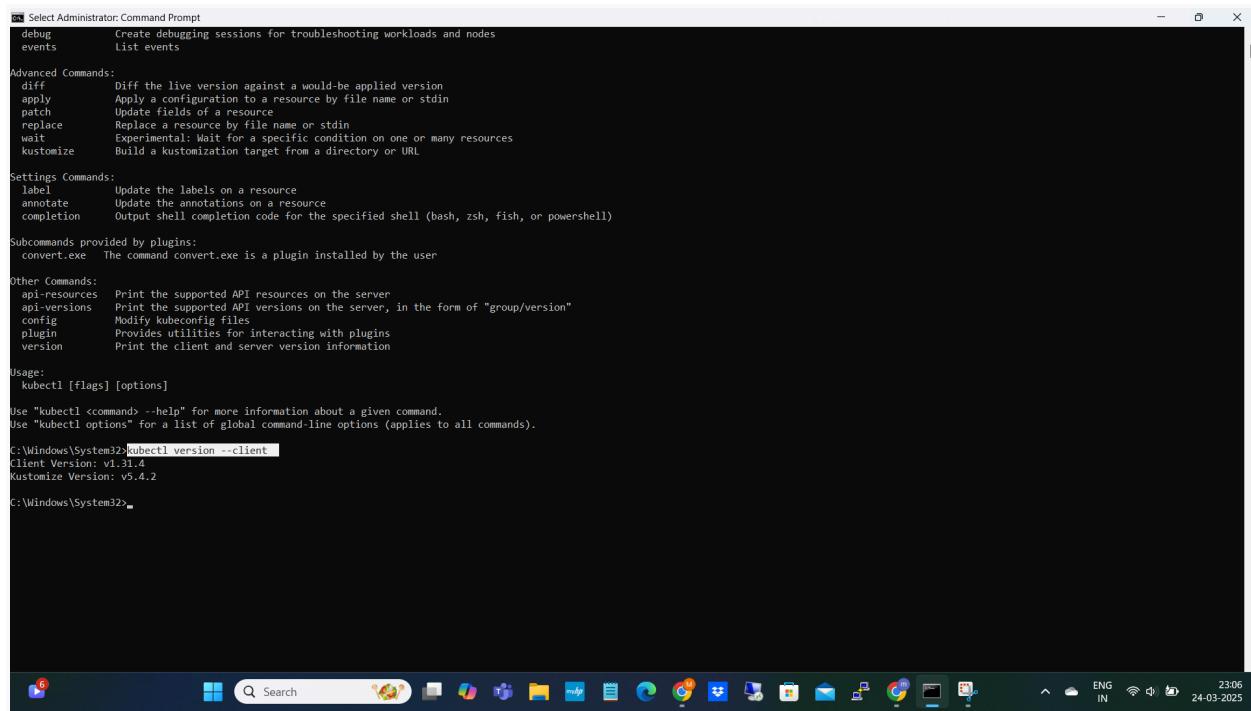
Cluster Management Commands:
  certificate Modify certificate resources
  cluster-info Display cluster information
  top        Display resource (CPU/memory) usage
  cordon    Mark node as unschedulable
  uncordon  Mark node as schedulable
  drain     Drain node in preparation for maintenance
  taint    Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe   Show details of a specific resource or group of resources
  logs       Print the logs for a container in a pod
  attach     Attach to a running container
  exec      Execute a command in a container
  port-forward Forward one or more local ports to a pod
  proxy     Run a proxy to the Kubernetes API server
  cp        Copy files and directories to and from containers
  auth      Inspect authorization
  debug    Create debugging sessions for troubleshooting workloads and nodes
  events   List events

Advanced Commands:
  diff      Diff the live version against a would-be applied version
  apply    Apply a configuration to a resource by file name or stdin
  patch   Update fields of a resource
  replace Replace a resource by file name or stdin
```



\$ kubectl version --client



```
PS Select Administrator: Command Prompt
debug      Create debugging sessions for troubleshooting workloads and nodes
events     List events

Advanced Commands:
diff       Diff the live version against a would-be applied version
apply     Apply a configuration to a resource by file name or stdin
patch     Update fields of a resource
replace   Replace a resource by file name or stdin
wait      Experimental: Wait for a specific condition on one or many resources
kustomize Build a kustomization target from a directory or URL

Settings Commands:
label     Update the labels on a resource
annotate  Update the annotations on a resource
completion Output shell completion code for the specified shell (bash, zsh, fish, or powershell)

Subcommands provided by plugins:
convert.exe The command convert.exe is a plugin installed by the user

Other Commands:
api-resources Print the supported API resources on the server
api-versions Print the supported API versions on the server, in the form of "group/version"
config     Modify kubeconfig files
plugin    Provides utilities for interacting with plugins
version   Print the client and server version information

Usage:
kubectl [flags] [options]

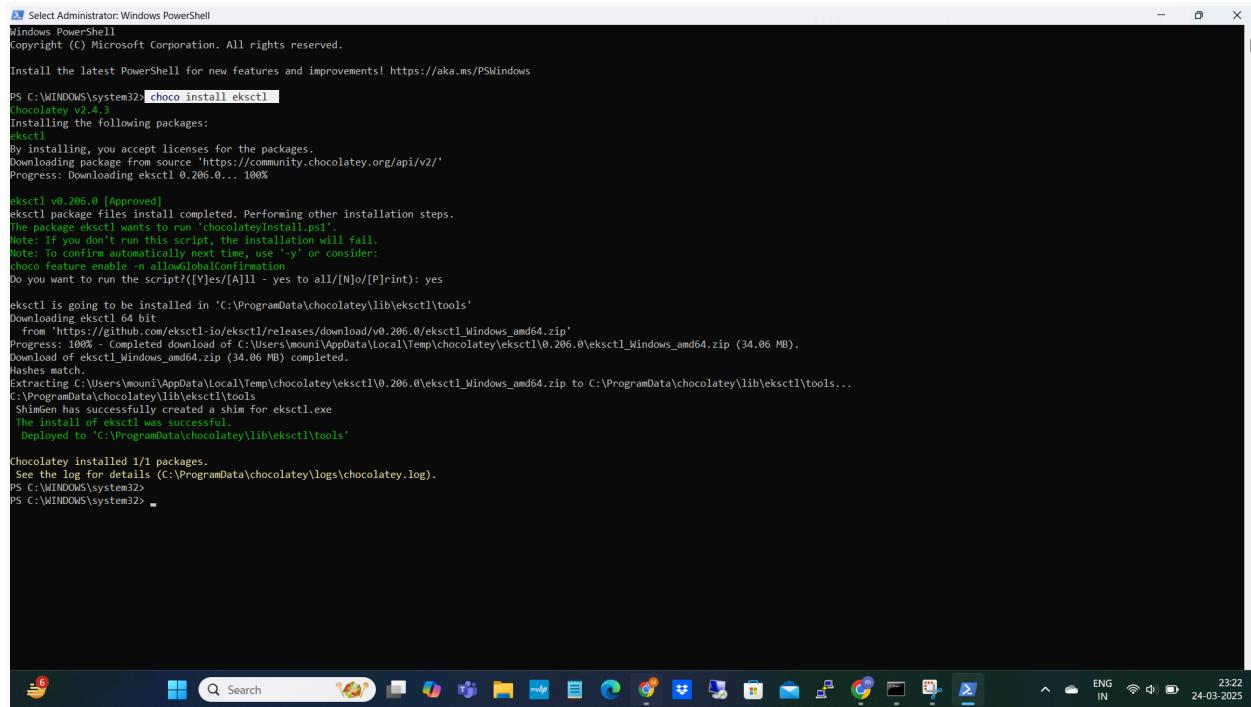
Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Windows\System32>kubectl version --client
Client Version: v1.31.4
Kustomize Version: v5.4.2

C:\Windows\System32>
```

Install eksctl :

\$ choco install eksctl



```
PS Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> choco install eksctl
Chocolatey v2.4.3
Installing the following packages:
eksctl
By installing, you accept licenses for the packages.
Downloading package from source "https://community.chocolatey.org/api/v2/"
Progress: Downloading eksctl 0.206.0... 100%
eksctl v0.206.0 [Approved]
eksctl package files install completed. Performing other installation steps.
The package eksctl wants to run "chocolateyInstall.ps1".
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): yes
eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
Downloading eksctl 64 bit
  from "https://github.com/ekscctl-io/ekscctl/releases/download/v0.206.0/eksctl_Windows_amd64.zip"
Progress: 100% Completed download of C:\Users\mounti\AppData\Local\Temp\chocolatey\eksctl\0.206.0\eksctl_Windows_amd64.zip (34.06 MB).
Download of eksctl_Windows_amd64.zip (34.06 MB) completed.
Hashes match.
Extracting C:\Users\mounti\AppData\Local\Temp\chocolatey\eksctl\0.206.0\eksctl_Windows_amd64.zip to C:\ProgramData\chocolatey\lib\eksctl\tools...
C:\ProgramData\chocolatey\lib\eksctl\tools
ShimGen has successfully created a shim for eksctl.exe
The install of eksctl was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\eksctl\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
```

To check version:

\$ eksctl version

```
PS Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> choco install eksctl
Chocolatey v2.4.3
Installing the following packages:
eksctl

By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading eksctl 0.206.0... 100%
eksctl v0.206.0 [Approved]
eksctl package files install completed. Performing other installation steps.
The package eksctl wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script? ([Y]es/[A]ll - yes to all/[N)o/[P]rint): yes

eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
Downloaded eksctl 64 bit
  from "https://github.com/eksctl-io/eksctl/releases/download/v0.206.0/eksctl_Windows_amd64.zip"
Progress: 100% Completed download of C:\Users\mouni\AppData\Local\Temp\chocolatey\lib\eksctl\0.206.0\eksctl_Windows_amd64.zip (34.06 MB).
Download of eksctl_Windows_amd64.zip (34.06 MB) completed.
Hashes match.
Extracting C:\Users\mouni\AppData\Local\Temp\chocolatey\lib\eksctl\0.206.0\eksctl_Windows_amd64.zip to C:\ProgramData\chocolatey\lib\eksctl\tools...
C:\ProgramData\chocolatey\lib\eksctl\tools
ShimGen has successfully created a shim for eksctl.exe
The install of eksctl was successful.
Deployed to: C:\ProgramData\chocolatey\lib\eksctl\tools

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> eksctl version
0.206.0
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
```

creating AWS Accesskey and Secret Key to configure through AWS CLI

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar contains navigation links for EC2, Instances, Images, and Elastic Block Store. The main content area displays various EC2 resources with their counts: Instances (running) 0, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 3, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. Below this, there are sections for Launch instance, Service health, and Zones. A modal window is open on the right, showing account details like Account ID (9054-1820-1986), Account attributes, Default VPC (vpc-07f2dc38977...), Settings, and a sign-out button. The status bar at the bottom shows the URL https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#security_credential, the date 25-03-2025, and the time 01:58.

Click on security credentials:

The screenshot shows the AWS IAM Security Credentials page for a root user. The left sidebar includes sections for Identity and Access Management (IAM), Access management, Access reports, and Organization activity. The main content area displays account details (Account name: srikanth kolluri, AWS account ID: 905418201986), a table for Multi-factor authentication (MFA) devices (one virtual device listed), and a section for Access keys (0). A 'Create access key' button is visible. The bottom of the screen shows a Windows taskbar with various icons.

Click on create access key:

The screenshot shows the same AWS IAM Security Credentials page as above, but the 'Access keys (0)' section is more prominent. It includes a note about avoiding long-term credentials and a 'Create access key' button. Below this, there is a section for CloudFront key pairs (0) with a 'Create CloudFront key pair' button. The bottom of the screen shows a Windows taskbar with various icons.

The screenshot shows the AWS IAM Access Key Wizard. The current step is "Alternatives to root user access keys". A warning message states: "Root user access keys are not recommended. We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice. Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials." Below this, there's a note: "If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user." At the bottom, there's a checkbox: "I understand creating a root access key is not a best practice, but I still want to create one." The "Create access key" button is highlighted in orange.

The screenshot shows the AWS IAM Access Key Wizard. The current step is "Retrieve access key". It displays the generated access key and secret access key. The access key is shown as "AKIA5FTZBLOBFWX4POPW" and the secret access key is shown as a masked string. Below this, there's a section titled "Access key best practices" with the following bullet points:

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

At the bottom, there are "Download .csv file" and "Done" buttons. The status bar at the bottom indicates the date and time as 25-03-2025 02:01.

Click on download csv: csv file downloaded with accesskey and secret key

The screenshot shows the AWS IAM Access Key Wizard interface. A green success message at the top states: "Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time." Below this, the "Retrieve access key" step is selected. It displays two fields: "Access key" (containing "AKIA5FTZBLOBFWX4POPW") and "Secret access key" (containing a long string of characters). A "Show" link is available for the secret key. The "Access key best practices" section lists several guidelines. At the bottom right are "Download .csv file" and "Done" buttons.

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Alternatives to root user access keys

Step 2
Retrieve access key

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key | Secret access key

AKIA5FTZBLOBFWX4POPW ***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file Done

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 02:01 25-03-2025

rootkey - Excel mounika kalluri MR

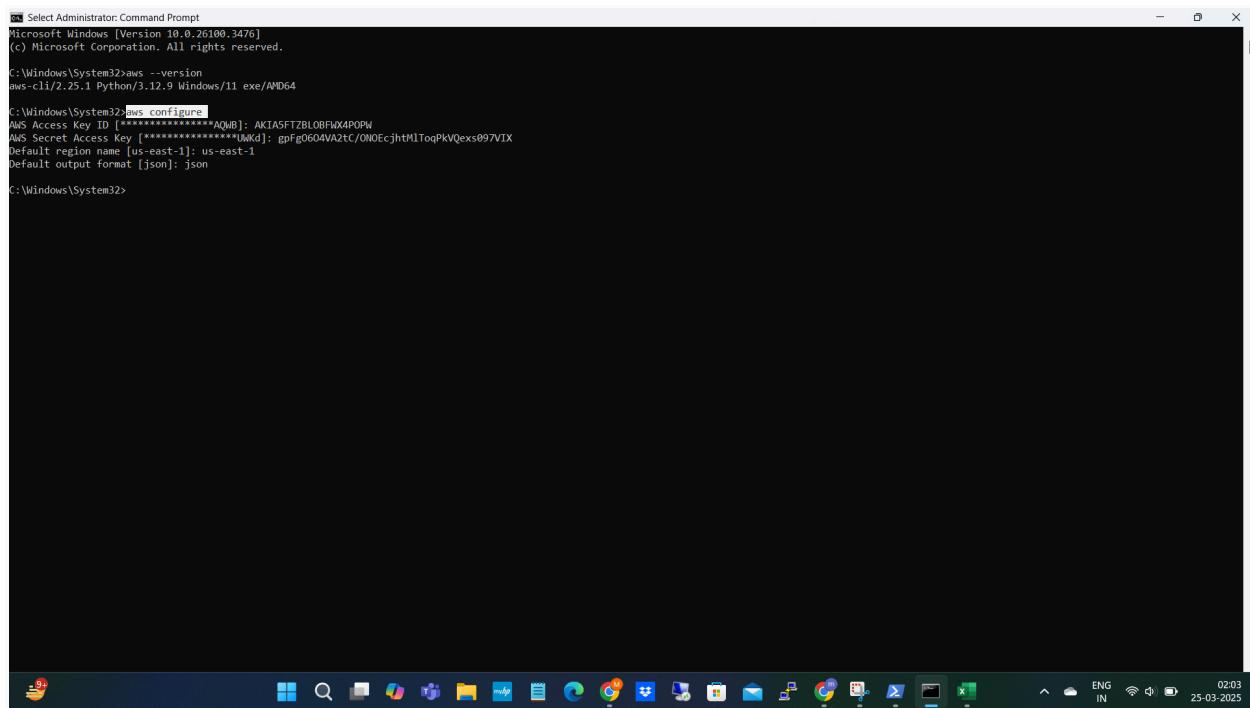
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1 Access key ID	Secret access key															
2 AKIA5FTZBLOBFWX4POPW	gpFg0604VA2tC/ONOEcjhtMIToqPkVQexs097VIX															
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																

rootkey Ready Accessibility: Unavailable 02:02 25-03-2025

Now configure the AWS with AWS CLI:

```
$ aws configure
```

```
Enter Access Key,secret key ,region and output format
```



```
Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>aws --version
aws-cli/2.25.1 Python/3.12.9 Windows/11 exe/AMD64

C:\Windows\System32>aws configure
AWS Access Key ID [*****]: AKIA5FTZBLO8ENXAP0DW
AWS Secret Access Key [*****]: gpFrG0604VA2tC/ONoEcjhM1ToqPkVQexs097VIX
Default region name [us-east-1]: us-east-1
Default output format [json]: json

C:\Windows\System32>
```

Now create cluster:

```
$ eksctl create cluster --name my-cluster-nginx --region us-east- --nodegroup-name
my-nodegrp-nginx --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 4
--managed
```

```
Administrator: Command Prompt
C:\Windows\System32\ekscctl create cluster --name my-cluster-nginx --region us-east-1 --nodegroup-name my-nodegrp-nginx --node-type t3.medium --nodes 3 --nodes-min 1 --nodes-max 4 --managed
2025-03-25 02:43:34 [ ] using region us-east-1
2025-03-25 02:43:35 [ ] setting availability zones to [us-east-1b us-east-1c]
2025-03-25 02:43:35 [ ] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2025-03-25 02:43:35 [ ] subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2025-03-25 02:43:35 [ ] nodegroup "my-nodegrp-nginx" will use "" [AmazonLinux2/1.32]
2025-03-25 02:43:35 [ ] using Kubernetes version 1.32
2025-03-25 02:43:35 [ ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-03-25 02:43:35 [ ] if you encounter any issues, check CloudFormation console or try 'ekscctl utils describe-stacks --region=us-east-1 --cluster=my-cluster-nginx'
2025-03-25 02:43:35 [ ] Kubernetes API endpoint access will use default of [publicAccess=true, privateAccess=false] for cluster "my-cluster-nginx" in "us-east-1"
2025-03-25 02:43:35 [ ] CloudWatch logging will not be enabled for cluster "my-cluster-nginx" in "us-east-1"
2025-03-25 02:43:35 [ ] you can enable it with 'ekscctl utils update-cluster-logging --enable-types=[SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)] --region=us-east-1 --cluster=my-cluster-nginx'
2025-03-25 02:43:35 [ ] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-03-25 02:43:35 [ ] 2 sequential tasks: { create cluster control plane "my-cluster-nginx",
  2 sequential sub-tasks: {
    1 task: { create addons },
    wait for control plane to become ready,
  },
  create managed nodegroup "my-nodegrp-nginx",
}
}
2025-03-25 02:43:35 [ ] building cluster stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:43:37 [ ] deploying stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:44:07 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:44:39 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:45:41 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:46:43 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:47:44 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:48:46 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:49:47 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:50:18 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-cluster"
2025-03-25 02:51:18 [ ] successfully created addon: metrics-server
2025-03-25 02:51:18 [ ] successfully created addon: metrics-server
2025-03-25 02:51:20 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, ekscctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under "addon.PodIdentityAssociations", and run 'ekscctl update addon'
2025-03-25 02:51:20 [ ] creating addon: vpc-cni
2025-03-25 02:51:20 [ ] successfully created addon: vpc-cni
2025-03-25 02:51:21 [ ] creating addon: kube-proxy
2025-03-25 02:51:21 [ ] successfully created addon: kube-proxy
2025-03-25 02:51:22 [ ] creating addon: coredns
2025-03-25 02:51:23 [ ] successfully created addon: coredns
2025-03-25 02:53:28 [ ] building managed nodegroup stack "ekscctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 02:53:30 [ ] deploying stack "ekscctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 02:54:02 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 02:54:53 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 02:57:27 [ ] waiting for CloudFormation stack "ekscctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 02:57:27 [ ] waiting for the control plane to become ready
2025-03-25 02:57:28 [✓] saved kubeconfig as "C:\\\\Users\\\\mouni\\\\.kube\\\\config"
2025-03-25 02:57:28 [ ] no tasks
2025-03-25 02:57:28 [✓] all EKS cluster resources for "my-cluster-nginx" have been created
2025-03-25 02:57:29 [ ] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [ ] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [ ] waiting for at least 1 node(s) to become ready in "my-nodegrp-nginx"
2025-03-25 02:57:29 [ ] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [ ] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [✓] created 1 managed nodegroup(s) in cluster "my-cluster-nginx"
2025-03-25 02:57:43 [ ] kubectl command should work with "C:\\\\Users\\\\mouni\\\\.kube\\\\config", try 'kubectl get nodes'
2025-03-25 02:57:43 [✓] EKS cluster "my-cluster-nginx" in "us-east-1" region is ready
```

```
Administrator: Command Prompt
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>

Administrator: Command Prompt
2025-03-25 02:57:28 [ ] no tasks
2025-03-25 02:57:28 [✓] all EKS cluster resources for "my-cluster-nginx" have been created
2025-03-25 02:57:29 [ ] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [ ] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [ ] waiting for at least 1 node(s) to become ready in "my-nodegrp-nginx"
2025-03-25 02:57:29 [ ] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [ ] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [ ] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [✓] created 1 managed nodegroup(s) in cluster "my-cluster-nginx"
2025-03-25 02:57:43 [ ] kubectl command should work with "C:\\\\Users\\\\mouni\\\\.kube\\\\config", try 'kubectl get nodes'
2025-03-25 02:57:43 [✓] EKS cluster "my-cluster-nginx" in "us-east-1" region is ready
```

Verify the cluster creation in AWS console:

My-cluster-nginx is created

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links for 'Metrics' (selected), 'Logs', 'CloudWatch Metrics Insights', 'CloudWatch Metrics Metrics Insights', and 'CloudWatch Metrics Metrics Insights'. The main area displays a chart titled 'my-cluster-nginx' with a single data series named 'CPU Utilization'. The chart shows a sharp peak at approximately 10:00 UTC on March 25, 2025, reaching a value of about 100%. Below the chart, there's a table with columns: Metric, Unit, Value, and Last Update. The table shows the following data:

Metric	Unit	Value	Last Update
my-cluster-nginx CPU Utilization	%	~100	2025-03-25T10:00:00Z

The screenshot shows the AWS Elastic Kubernetes Service Clusters page. The left sidebar includes 'Clusters' (selected), 'Settings', 'Amazon EKS Anywhere', and 'Related services'. The main content area shows a table titled 'Clusters (1)'. The table has columns: Cluster name, Status, Kubernetes version, Support period, Upgrade policy, Created, and Provider. The data in the table is as follows:

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
my-cluster-nginx	Active	1.32	Standard support until March 21, 2026	Extended	21 minutes ago	EKS

Below the table, there's a detailed view for the 'my-cluster-nginx' cluster. It shows 'Cluster info' with status 'Active', Kubernetes version '1.32', support period 'Standard support until March 21, 2026', provider 'EKS', and sections for 'Cluster health issues', 'Upgrade insights', and 'Node health issues'. The 'Overview' tab is selected, followed by 'Resources', 'Compute', 'Networking', 'Add-ons', 'Access', 'Observability', 'Update history', and 'Tags'. The 'Details' section provides API server endpoint (<https://CA8EEA12D865BBA76E33432A9BD0ABA5.gr7.us-east-1.eks.amazonaws.com>), OpenID Connect provider URL (<https://oidc.eks.us-east-1.amazonaws.com/id/CA8EEA12D865BBA76E33432A9BD0ABA5>), Certificate authority ([View in IAM](#)), Cluster IAM role ARN ([arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-cluster-ServiceRole-HEN852BXUwlP](#)), and Platform version 'eks.5'. The bottom of the screen shows the AWS navigation bar and system status.

Resources:

The screenshot shows the AWS EKS Cluster Resources page for the 'my-cluster-nginx' cluster. The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services (Amazon ECR, AWS Batch). The main content area is titled 'Resource types' and shows a list of workloads. Under 'Workloads', it lists PodTemplates, Pods, Deployments, StatefulSets, DaemonSets, Jobs, CronJobs, PriorityClasses, and HorizontalPodAutoscalers. A detailed view of the 'Pods' section is shown, titled 'Workloads: Pods (10)'. The table lists ten pods, all created 11 minutes ago, with names like aws-node-2jlc, aws-node-4rxx6, aws-node-sj2lg, coredns-6b9575c64c-6v98k, coredns-6b9575c64c-786pt, kube-proxy-bwrn8, kube-proxy-pcb9s, and kube-proxy-t2lps.

Name	Created
aws-node-2jlc	11 minutes ago
aws-node-4rxx6	11 minutes ago
aws-node-sj2lg	11 minutes ago
coredns-6b9575c64c-6v98k	15 minutes ago
coredns-6b9575c64c-786pt	15 minutes ago
kube-proxy-bwrn8	11 minutes ago
kube-proxy-pcb9s	11 minutes ago
kube-proxy-t2lps	11 minutes ago

Below the table, there is a note: 'Pod is the smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster.' with a 'Learn more' link.

At the bottom of the page, there are links for CloudShell and Feedback, and a standard Windows taskbar with various application icons.

Compute:

The screenshot shows the AWS EKS Compute tab for the cluster "my-cluster-nginx".

Nodes (3) Info

Node name	Instance type	Compute	Managed by	Created	Status
ip-192-168-24-38.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	14 minutes ago	Ready
ip-192-168-59-39.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	14 minutes ago	Ready
ip-192-168-7-44.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	14 minutes ago	Ready

Node groups (1) Info

Group name	Desired size	AMI release version	Launch template	Status
my-nodegrp-nginx	3	1.32.1-20250317	eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx (1)	Active

Fargate profiles (0) Info

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 03:09 25-03-2025

The screenshot shows the AWS EKS Networking tab for the cluster "my-cluster-nginx".

Cluster info

Status: Active	Kubernetes version: 1.32	Support period: Standard support until March 21, 2026	Provider: EKS
----------------	--------------------------	---	---------------

Networking

VPC	Subnets	Cluster security group	API server endpoint access
vpc-0c34ceb25dd36409e	subnet-0a72ea7d5491ef18c, subnet-082ddf76f97831f0, subnet-0cddb1cfeb43a929, subnet-0d840b6b863bc029b	sg-028890e4be0bd6903	Public
Cluster IP address family: IPv4	Additional security groups: sg-08702f5df31362808	0.0.0.0/0 (open to all traffic)	
Service IPv4 range: 10.100.0.0/16			

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 03:09 25-03-2025

Adding new Addons:

The screenshot shows the AWS EKS Add-ons page. A blue selection box highlights the 'Node monitoring agent' add-on. The page lists several other add-ons:

- Mountpoint for Amazon S3 CSI Driver**: Category storage, Compatible compute EC2, EKS Auto Mode.
- AWS Network Flow Monitor Agent**: Category observability, Compatible compute EC2, EKS Auto Mode.
- Node monitoring agent**: Category observability, Compatible compute EC2.
- Amazon EKS Pod Identity Agent**: Category security, Compatible compute EKS Auto Mode.
- CSI Snapshot Controller**: Category storage.

At the bottom, there are navigation links for CloudShell and Feedback, and a standard Windows taskbar.

The screenshot shows the 'Configure selected add-ons settings' step for the 'Node monitoring agent'. The 'Configure selected add-ons settings' link is highlighted with a blue selection box. The page displays the configuration details for the selected add-on:

- Node monitoring agent**: Listed by AWS, Category observability, Status Ready to install.
- Version**: v1.0.eksbuild.1
- Optional configuration settings**

Below this, the 'Amazon EKS Pod Identity Agent' is also listed with its details:

- Amazon EKS Pod Identity Agent**: Listed by AWS, Category security, Status Ready to install.

At the bottom, there are navigation links for CloudShell and Feedback, and a standard Windows taskbar.

Screenshot of the AWS Cloud Console showing the process of adding add-ons to an EKS cluster.

The top navigation bar shows tabs for GUUI, Kebernet manag, Inbox (94) - mou, Home - Google, Untitled document, Releases - eksctl, CreateVpc | VPC, Add add-ons | E, and others. The URL is us-east-1.console.aws.amazon.com/eks/clusters/my-cluster-nginx/create-add-ons?region=us-east-1&selectedTab=cluster-add-ons-tab. The location bar shows Amazon Elastic Kubernetes Service > Clusters > my-cluster-nginx > Add add-ons.

The main interface is titled "Review and add".

Step 1: Select add-ons

Shows two selected add-ons:

Add-on name	Type	Status
eks-node-monitoring-agent	observability	Ready to install
eks-pod-identity-agent	security	Ready to install

Step 2: Configure selected add-ons settings

Selected add-ons version (2)

Add-on name	Version	IAM role for service account (IRSA)
eks-node-monitoring-agent	v1.1.0-eksbuild.1	Not set

EKS Pod Identity (0)

No Pod Identity associations

None of the selected add-on(s) have Pod Identity associations.

Buttons at the bottom: Cancel, Previous, Create.

CloudShell and Feedback icons are visible in the bottom left. The bottom right shows system status: ENG IN, 03:19, 25-03-2025.

Now the updated addons:

The screenshot shows two instances of the AWS EKS Cluster Add-ons page. Both instances are for the same cluster, "my-cluster-nginx".

Top Instance (Original View):

- Add-ons (6) Info:** Shows 6 add-ons: Amazon EKS Pod Identity Agent, CoreDNS, Node monitoring agent, kube-proxy, and Amazon VPC CNI.
- Amazon EKS Pod Identity Agent:** Category: security, Status: Active, Version: v1.3.4-eksbuild.1, EKS Pod Identity: -, IAM role for service account (IRSA): Not set.
- CoreDNS:** Category: networking, Status: Active, Version: v1.11.4-eksbuild.2, EKS Pod Identity: -, IAM role for service account (IRSA): Not set.

Bottom Instance (Updated View):

- Add-ons (6) Info:** Shows 6 add-ons: Node monitoring agent, kube-proxy, and Amazon VPC CNI.
- Node monitoring agent:** Category: observability, Status: Active, Version: v1.1.0-eksbuild.1, EKS Pod Identity: -, IAM role for service account (IRSA): Not set.
- kube-proxy:** Category: networking, Status: Active, Version: v1.32.0-eksbuild.2, EKS Pod Identity: -, IAM role for service account (IRSA): Not set.
- Amazon VPC CNI:** Category: networking, Status: Active, Version: v1.19.2-eksbuild.1, EKS Pod Identity: -, IAM role for service account (IRSA): Not set.

Both instances show the same browser interface and system tray at the bottom.

Access:

The screenshot shows the EKS console interface. On the left, there's a sidebar with navigation links for Clusters, Settings (Console settings), Amazon EKS Anywhere (Enterprise Subscriptions), and Related services (Amazon ECR, AWS Batch). The main area is titled 'Access configuration' under 'IAM access entries'. It lists two entries:

IAM principal ARN	Type	Username	Group names	Access policies
arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-nodegroup--NodeInstanceRole-WAUtxVY43JO	EC2 Linux	system:node: {{EC2PrivateDNSName}}	system:nodes	-
arn:aws:iam::905418201986:root	Standard	arn:aws:iam::905418201986:root	-	AmazonEKSClusterAdminPolicy

Describe eks cluster:

```
$ aws eks describe-cluster --name my-cluster-nginx
```

```
2025-03-25 02:57:28 [✓] saved kubeconfig as "C:\Users\mouni\.kube\config"
2025-03-25 02:57:28 [!] no tasks
2025-03-25 02:57:28 [✓] all EKS cluster resources for "my-cluster-nginx" have been created
2025-03-25 02:57:29 [!] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [!] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [!] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [!] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [!] waiting for at least 1 node(s) to become ready in "my-nodegrp-nginx"
2025-03-25 02:57:29 [!] nodegroup "my-nodegrp-nginx" has 3 node(s)
2025-03-25 02:57:29 [!] node "ip-192-168-24-38.ec2.internal" is ready
2025-03-25 02:57:29 [!] node "ip-192-168-59-39.ec2.internal" is ready
2025-03-25 02:57:29 [!] node "ip-192-168-7-44.ec2.internal" is ready
2025-03-25 02:57:29 [!] created 1 managed nodegroup(s) in cluster "my-cluster-nginx"
2025-03-25 02:57:43 [!] kubectl command should work with "C:\Users\mouni\.kube\config", try 'kubectl get nodes'
2025-03-25 02:57:43 [✓] EKS cluster "my-cluster-nginx" in "us-east-1" region is ready

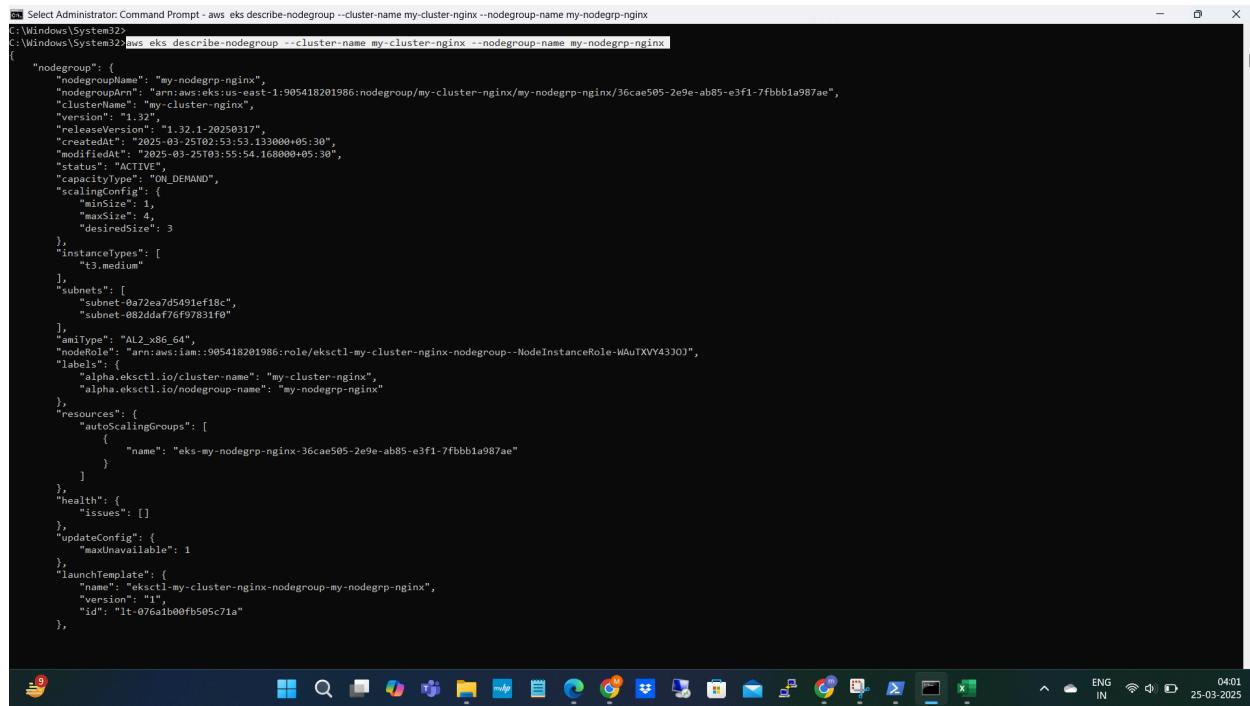
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>aws eks describe-cluster --name my-cluster-nginx
{
  "cluster": {
    "name": "my-cluster-nginx",
    "arn": "arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx",
    "createdAt": "2025-03-25T02:44:02.834000+05:30",
    "version": "1.32",
    "endpoint": "https://CABEEA120B658BA76E33432A9800A845.gr7.us-east-1.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-cluster-ServiceRole-HEN852XUwLp",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0a2eaf7d5491ef18c",
        "subnet-082ddaf76f97831f0",
        "subnet-0cd8b1feb43a929",
        "subnet-0d84bb6b863bc029b"
      ],
      "securityGroupIds": [
        "sg-087025df31362808"
      ],
      "clusterSecurityGroupId": "sg-028890e4be0bd6903",
      "vpcId": "vpc-0c34cb25d36409e",
      "endpointPublicAccess": true,
      "endpointPrivateAccess": false,
      "publicAccessScidrs": [
        "0.0.0.0/0"
      ]
    },
    "kubernetesNetworkConfig": {
      "serviceIpv4Cidr": "10.100.0.0/16",
      "ipFamily": "IPv4",
      "elasticLoadBalancing": {
        "loadBalancerType": "Cluster"
      }
    }
  }
}
```

```
cmd Select Administrator: Command Prompt - aws eks describe-cluster --name my-cluster-nginx
{
    "createdAt": "2025-03-25T02:44:02.834800+05:30",
    "endpoint": "https://CABEEA12D865BBA76E33432A9BD0A85.gr7.us-east-1.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-cluster-ServiceRole-HEN852BXUwlP",
    "resourcesVpcConfig": {
        "subnetIds": [
            "subnet-0a72ea7d5491ef18c",
            "subnet-0a72ea7d5491ef10f",
            "subnet-0ad4b51fc643ad59",
            "subnet-0d40bd0b850c0290"
        ],
        "securityGroupIds": [
            "sg-08702f5df31362808"
        ],
        "clusterSecurityGroupId": "sg-028890e4be0bd6903",
        "vpcId": "vpc-0c34ce2b5d36409e",
        "endpointPublicAccess": true,
        "endpointPrivateAccess": false,
        "publicAccessCidrs": [
            "0.0.0.0/0"
        ]
    },
    "kubernetesNetworkConfig": {
        "serviceIpv4Cidr": "10.100.0.0/16",
        "ipFamily": "IPv4",
        "elasticLoadBalancing": {
            "enabled": false
        }
    },
    "logging": {
        "clusterLogging": [
            {
                "types": [
                    "api",
                    "audit",
                    "authenticator",
                    "controllerManager",
                    "scheduler"
                ],
                "enabled": false
            }
        ]
    },
    "identity": {
        ...
    }
}
-- More --
```

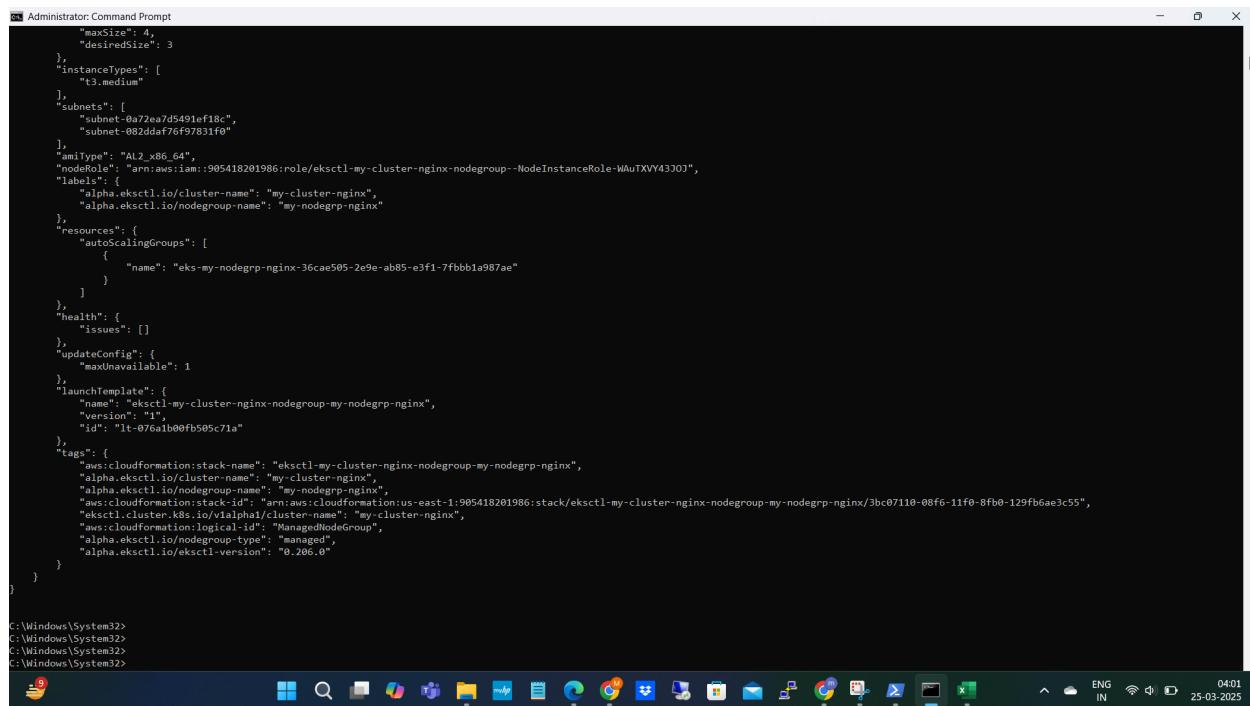
```
cmd Select Administrator: Command Prompt
{
    "authenticator",
    "controllerManager",
    "scheduler"
},
"enabled": false
}
],
"identity": {
    "oidc": {
        "issuer": "https://oidc.eks.us-east-1.amazonaws.com/id/CABEEA12D865BBA76E33432A9BD0A85"
    }
},
"status": "ACTIVE",
"certificateAuthority": {
    ...
},
"data": "LS0tLS1CRUkgT1BDRVJUJSUZJQ0FURS0tLS0tck1JSURCVRNQDmWzBt3SU1BZ01JSjdSmw5ShmlvaF13RF3ktvKvlodmNQFFTEJRjXgVEVUTUFR0ExXUUKX0N52E2Vl1wEp11hsBnGNOQwGzb5TIRbek1q1XjNVEWTmSpUZ23Ip0VEF61w
pJeUJURTRNammhTUVeApIekFSmdOVkIBTVRbQXwM1wEv1tyjbawE1322dfnRJU01M0RRRURJBUWVQ0tTRjQk3QxdnZvLClkrtvsUJBUURhakYZE9kTRkz0w8AvJvZZzQzNUHMwpUe41bfGbjk4d1Ry1W1n5wbhJvdihrnKPY190KgK91bR3V1YjVSyVFV
FnGa1MsXvRmSVxR014a3b2FTT0B1SjBtc2f1NFb2T1FD2g2c1ZkOfVKtH6u14efBTCAo0w1Bx02kVbMjVPUJU038jY95wQzDyc2loZD21pR0U1laWvVRMFrwBZhlfGen1N6DzTUNsZHRZU0h0cado6tRQlmwU1jYwQ00E8-Ng1mVG5c0110ER2Mp2UD1owlmZm1z
aU00Y0UpdrVp1Exgvav1VmBj0da2Nusckd90NmUNGpNzklXBxDdcSeck3YUREGz6bzxF21220NEVGtzG1ysG95VVJwazRdgT4L3ay1h1CUKNMcpdUvBN1NKdERy1h0Rgdtk9ExY1T2kNjVwQdNQkFBz2XvEY1T0E0RExWRlEdVCl3drRUF35UmREFQckn1t1Z
1Uk1CQy4RJUJU0QR8bvgU1J0RExVMR21FQXQJUEETL0t2ZhpafZ6UjY3eZtPMHJMUpSM26gQVYQmD0VhsRUVsAfkHnz2dmfXsmxjb1VsZEde1kMEd0d3Hu01M0RRR0Tjd1B0tRQkRQ2Ng0hu1zBCC0QoedU1nh0M1F5k2krUEshy2Zfb1puMfhndG22z0X
hJG1lobdh2a1BmKfKVYPhedIn40ptTTMs13RUUHehCKNShMnTV08xdnkIMepuSGY1RfdgM46QdWNYR82k1UDEhKCNzRFXz0EFBUzNr-tVJ2MKhRQJhQ201Ry8YKZKL1haeDV1Smzv1ZxeptbDmz13VVQUm-EK1Jhb2N1Q60a015NGfhZENZ2VRRfBm1IXG9m
IVNjGR1QDwRFLS0tL50k",
"platformVersion": "eks.5",
"tags": {
    ...
},
"aws:cloudformation:stack-name": "eksctl-my-cluster-nginx-cluster",
"alpha.eksctl.io/cluster-name": "my-cluster-nginx",
"aws:cloudformation:stack-id": "arn:aws:cloudformation:us-east-1:905418201986:stack/eksctl-my-cluster-nginx-cluster/da659c20-08f4-11f0-9406-1271428e233b",
"eksctl.cluster.k8s.io/vlalphi/c/cluster-name": "my-cluster-nginx",
"alpha.eksctl.io/cluster-oidc-enabled": "true",
"aws:cloudformation:logical-id": "ControlPlane",
"alpha.eksctl.io/eksctl-version": "0.206.0",
"Name": "eksctl-my-cluster-nginx-cluster/ControlPlane"
},
"health": {
    "issues": []
},
"accessConfig": {
    "authenticationMode": "API_AND_CONFIG_MAP"
},
"upgradePolicy": {
    "supportType": "EXTENDED"
}
}
```

Describe node group

```
$ aws eks describe-nodegroup --cluster-name my-cluster-nginx --nodegroup-name my-nodegrp-nginx
```



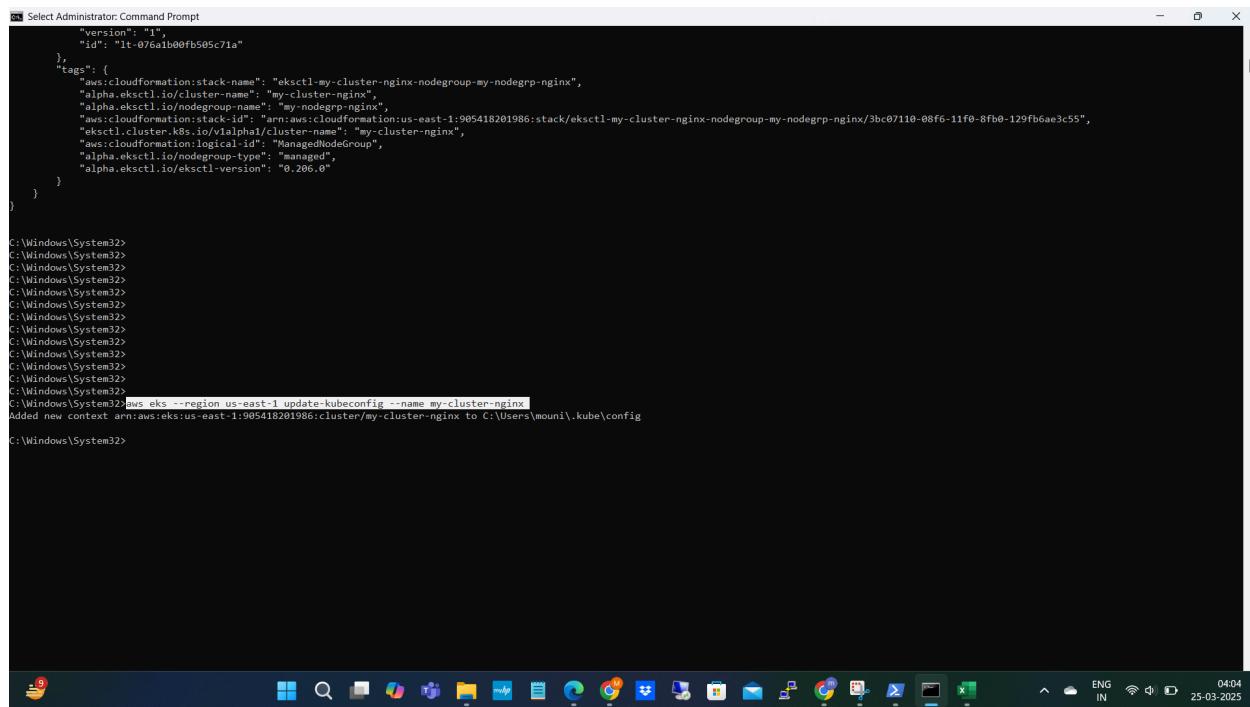
```
Administrator: Command Prompt - aws eks describe-nodegroup --cluster-name my-cluster-nginx --nodegroup-name my-nodegrp-nginx
C:\Windows\System32>
C:\Windows\System32>aws eks describe-nodegroup --cluster-name my-cluster-nginx --nodegroup-name my-nodegrp-nginx
{
    "nodegroup": {
        "nodegroupName": "my-nodegrp-nginx",
        "nodegroupArn": "arn:aws:eks:us-east-1:905418201986:nodegroup/my-cluster-nginx/my-nodegrp-nginx/36cae505-2e9e-ab85-e3f1-7fb8b1a987ae",
        "clusterName": "my-cluster-nginx",
        "version": "1.32",
        "releaseVersion": "1.32.1-20250317",
        "createdAt": "2025-03-25T02:53:53.133000+05:30",
        "modifiedAt": "2025-03-25T03:55:54.168000+05:30",
        "status": "ACTIVE",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 1,
            "maxSize": 4,
            "desiredSize": 3
        },
        "instanceTypes": [
            "t3.medium"
        ],
        "subnets": [
            "subnet-0a72ea7d5491ef18c",
            "subnet-082ddaf76f97831f0"
        ],
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-nodegroup--NodeInstanceRole-WAuTXVY43J0J",
        "labels": {
            "alpha.eksctl.io/cluster-name": "my-cluster-nginx",
            "alpha.eksctl.io/nodegroup-name": "my-nodegrp-nginx"
        },
        "resources": {
            "autoScalingGroups": [
                {
                    "name": "eks-my-nodegrp-nginx-36cae505-2e9e-ab85-e3f1-7fb8b1a987ae"
                }
            ]
        },
        "health": {
            "issues": []
        },
        "updateConfig": {
            "maxUnavailable": 1
        },
        "launchTemplate": {
            "name": "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx",
            "version": "1",
            "id": "lt-076ab0fb0f505c71a"
        }
    }
}
```



```
Administrator: Command Prompt
{
    "nodegroup": {
        "maxSize": 4,
        "desiredSize": 3
    },
    "instanceTypes": [
        "t3.medium"
    ],
    "subnets": [
        "subnet-0a72ea7d5491ef18c",
        "subnet-082ddaf76f97831f0"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-nodegroup--NodeInstanceRole-WAuTXVY43J0J",
    "labels": {
        "alpha.eksctl.io/cluster-name": "my-cluster-nginx",
        "alpha.eksctl.io/nodegroup-name": "my-nodegrp-nginx"
    },
    "resources": {
        "autoScalingGroups": [
            {
                "name": "eks-my-nodegrp-nginx-36cae505-2e9e-ab85-e3f1-7fb8b1a987ae"
            }
        ]
    },
    "health": {
        "issues": []
    },
    "updateConfig": {
        "maxUnavailable": 1
    },
    "launchTemplate": {
        "name": "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx",
        "version": "1",
        "id": "lt-076ab0fb0f505c71a"
    },
    "tags": [
        {
            "aws:cloudformation:stack-name": "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx",
            "alpha.eksctl.io/cluster-name": "my-cluster-nginx",
            "alpha.eksctl.io/nodegroup-name": "my-nodegrp-nginx",
            "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-east-1:905418201986:stack/eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx/3bc07110-08f6-11f0-0fb0-129fb6ae3c55",
            "eksctl.cluster.k8s.io/viaIpha1/cluster-name": "my-cluster-nginx",
            "aws:cloudformation:logical-id": "ManagedNodeGroup",
            "alpha.eksctl.io/nodegroup-type": "managed",
            "alpha.eksctl.io/eksctl-version": "0.266.0"
        }
    ]
}
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>
```

Config kubectl to the EKS cluster:

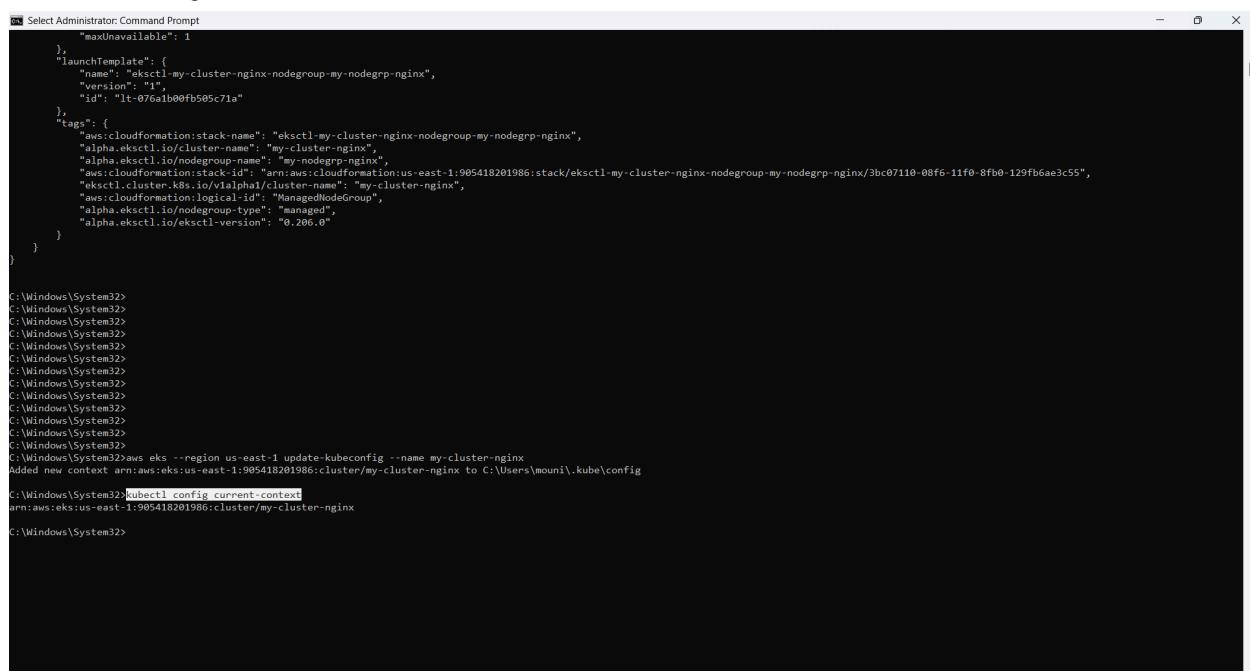
```
$ aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
```



```
C:\Windows\System32>
C:\Windows\System32> aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:\Users\mouni\.kube\config
C:\Windows\System32>
```

Current cluster access:

```
$ kubectl config current-context
```



```
C:\Windows\System32>
C:\Windows\System32> kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx
C:\Windows\System32>
```

Verify the cluster nodes:

```
$ kubectl get nodes
```

```
C:\Windows\System32>
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context [arn:aws:eks:us-east-1:1905418201986:cluster/my-cluster-nginx] to C:\Users\mouni\.kube\config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:1905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-192-168-24-38.ec2.internal   Ready    <none>    80m   v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready    <none>    79m   v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready    <none>    80m   v1.32.1-eks-5d632ec

C:\Windows\System32>
```

Deploy nginx application to the cluster:

```
$ notepad nginx-deployment.yaml
```

```

Select Administrator: Command Prompt
  "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-cluster-nginx",
  "aws:cloudformation:logical-id": "ManagedNodeGroup",
  "alpha.eksctl.io/nodegroup-type": "managed",
  "alpha.eksctl.io/eksctl-version": "0.206.0"
}

C:\Windows\System32>
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES   AGE    VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m   v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec

C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>

Ln 22, Col 1 | 341 characters          100%     Windows (CRLF) | UTF-8

```

Apply or execute the deployment

\$ kubectl apply -f nginx-deployment.yaml

```

Select Administrator: Command Prompt
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:\Users\mouni\.kube\config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES   AGE    VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m   v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec

C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>cat nginx-deployment.yaml
'cat' is not recognized as an internal or external command,
operable program or batch file.

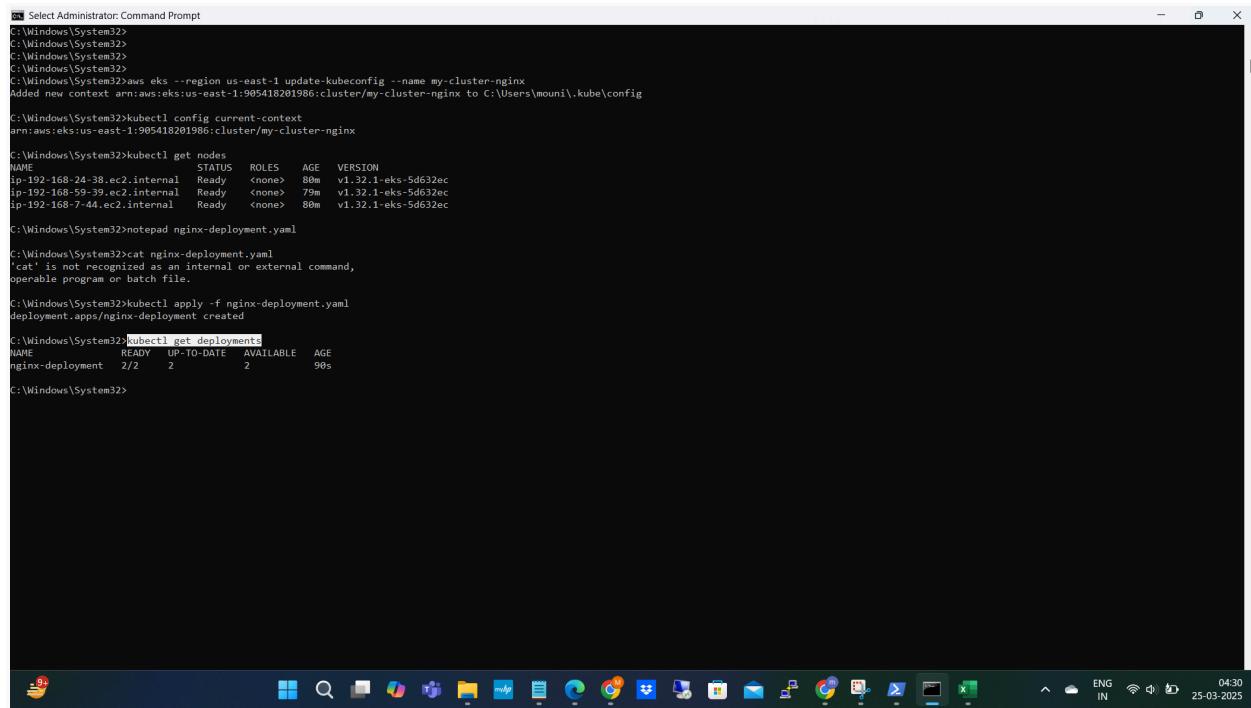
C:\Windows\System32>kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

C:\Windows\System32>

```

Verify the deployment:

```
$ kubectl get deployments
```



```
C:\Windows\System32>Select Administrator: Command Prompt
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:\Users\mouni\.kube\config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m   v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec

C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>cat nginx-deployment.yaml
'cat' is not recognized as an internal or external command,
operable program or batch file.

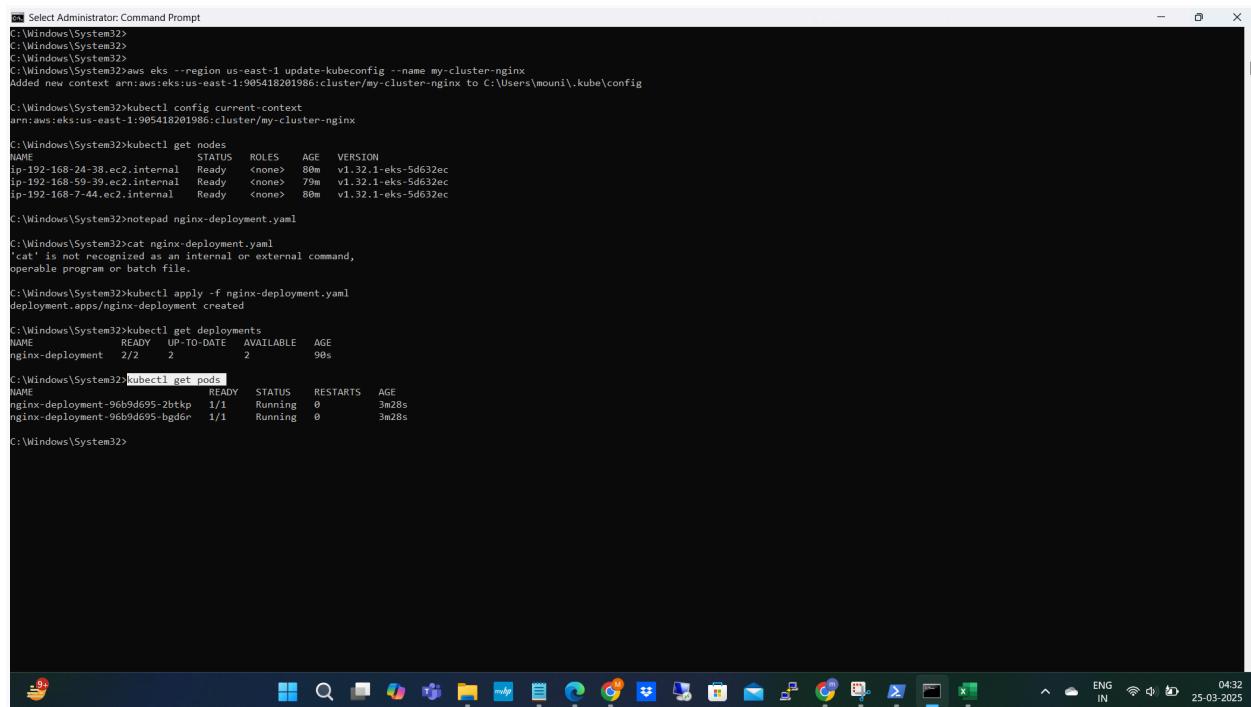
C:\Windows\System32>kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

C:\Windows\System32>kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2          2          90s

C:\Windows\System32>
```

Verify the pods:

```
$ kubectl get pods
```



```
C:\Windows\System32>Select Administrator: Command Prompt
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:\Users\mouni\.kube\config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m   v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m   v1.32.1-eks-5d632ec

C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>cat nginx-deployment.yaml
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

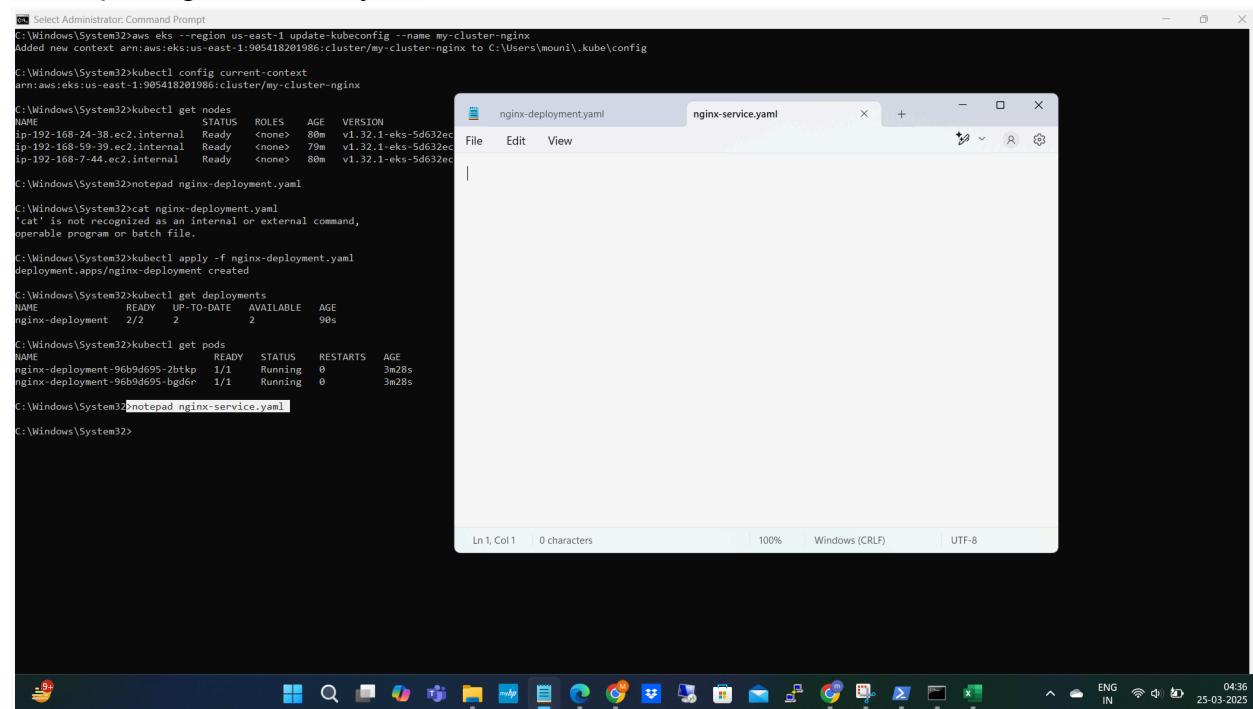
C:\Windows\System32>kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2          2          90s

C:\Windows\System32>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-deployment-96b9d695-2btkp   1/1     Running   0          3m28s
nginx-deployment-96b9d695-bgdr6   1/1     Running   0          3m28s

C:\Windows\System32>
```

Now expose Nginx deployment with a LoadBalancer:

\$ notepad nginx-service.yaml



```
cmd Select Administrator: Command Prompt
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:/Users/mouni/.kube/config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES    AGE     VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m    v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m    v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m    v1.32.1-eks-5d632ec

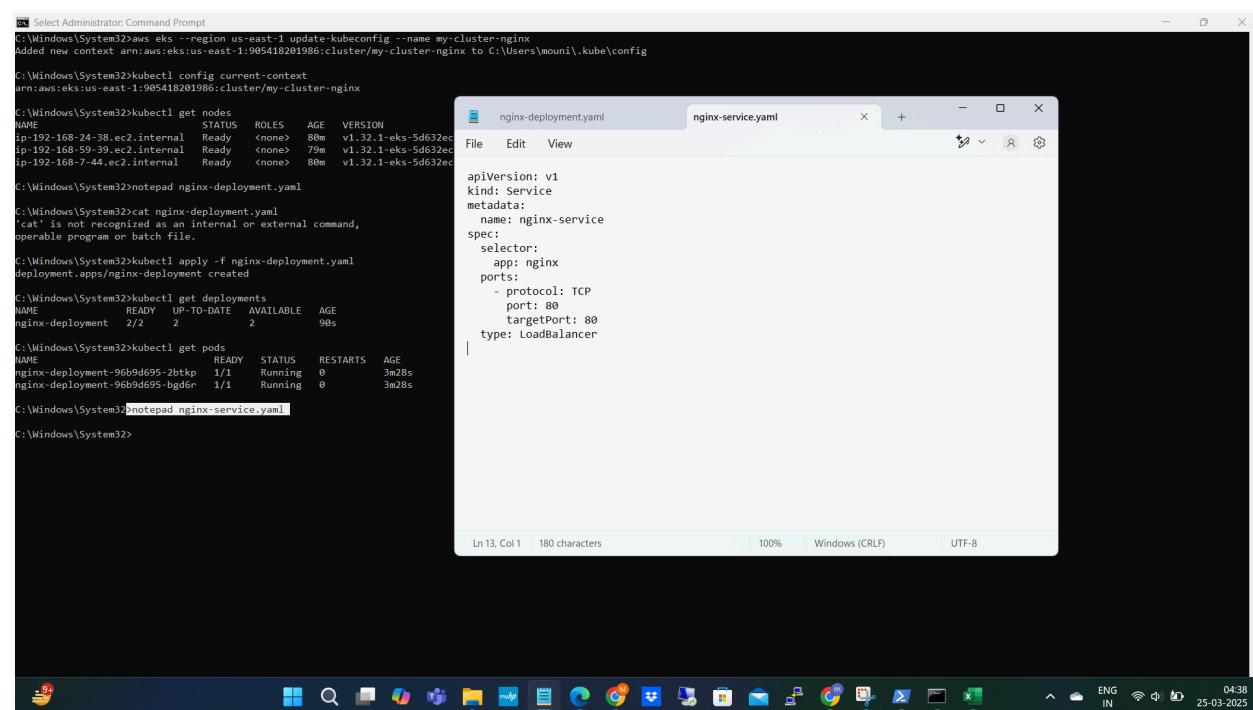
C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>cat nginx-deployment.yaml
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

E:\Windows\System32>kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2          98s

C:\Windows\System32>kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-9699d695-zbtkp   1/1     Running   0          3m28s
nginx-deployment-9699d695-bgd6r   1/1     Running   0          3m28s

C:\Windows\System32>notepad nginx-service.yaml
C:\Windows\System32>
```



```
cmd Select Administrator: Command Prompt
C:\Windows\System32>aws eks --region us-east-1 update-kubeconfig --name my-cluster-nginx
Added new context arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx to C:/Users/mouni/.kube/config

C:\Windows\System32>kubectl config current-context
arn:aws:eks:us-east-1:905418201986:cluster/my-cluster-nginx

C:\Windows\System32>kubectl get nodes
NAME           STATUS   ROLES    AGE     VERSION
ip-192-168-24-38.ec2.internal   Ready   <none>   80m    v1.32.1-eks-5d632ec
ip-192-168-59-39.ec2.internal   Ready   <none>   79m    v1.32.1-eks-5d632ec
ip-192-168-7-44.ec2.internal   Ready   <none>   80m    v1.32.1-eks-5d632ec

C:\Windows\System32>notepad nginx-deployment.yaml
C:\Windows\System32>cat nginx-deployment.yaml
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created

E:\Windows\System32>kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2          98s

C:\Windows\System32>kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-9699d695-zbtkp   1/1     Running   0          3m28s
nginx-deployment-9699d695-bgd6r   1/1     Running   0          3m28s

C:\Windows\System32>notepad nginx-service.yaml
C:\Windows\System32>
```

Apply or Execute the service:

```
$ kubectl apply -f nginx-service.yaml
```

```
C:\> Select Administrator: Command Prompt  
operable program or batch file.  
C:\Windows\System32>kubectl apply -f nginx-deployment.yaml  
deployment.apps/nginx-deployment created  
C:\Windows\System32>kubectl get deployments  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx-deployment 2/2 2 2 99s  
C:\Windows\System32>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-96b9d695-2btkp 1/1 Running 0 3m28s  
nginx-deployment-96b9d695-bg6fr 1/1 Running 0 3m28s  
C:\Windows\System32>notepad nginx-service.yaml  
C:\Windows\System32>kubectl apply -f nginx-service.yaml  
service/nginx-service created  
C:\Windows\System32>  
C:\Windows\System32>
```

Check the status and External IP address:

```
$ kubectl get services
```

```
C:\> Select Administrator: Command Prompt  
arniaws:eks:us-east-1:905418201986:cluster/my-cluster--nginx  
C:\Windows\System32>kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ip-192-168-24-38.ec2.internal Ready <none> 80m v1.32.1-eks-5d632ec  
ip-192-168-59-39.ec2.internal Ready <none> 79m v1.32.1-eks-5d632ec  
ip-192-168-7-44.ec2.internal Ready <none> 80m v1.32.1-eks-5d632ec  
C:\Windows\System32>notepad nginx-deployment.yaml  
C:\Windows\System32>cat nginx-deployment.yaml  
'cat' is not recognized as an internal or external command,  
operable program or batch file.  
C:\Windows\System32>kubectl apply -f nginx-deployment.yaml  
deployment.apps/nginx-deployment created  
C:\Windows\System32>kubectl get deployments  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx-deployment 2/2 2 2 99s  
C:\Windows\System32>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-96b9d695-2btkp 1/1 Running 0 3m28s  
nginx-deployment-96b9d695-bg6fr 1/1 Running 0 3m28s  
C:\Windows\System32>notepad nginx-service.yaml  
C:\Windows\System32>kubectl apply -f nginx-service.yaml  
service/nginx-service created  
C:\Windows\System32>  
C:\Windows\System32>kubectl get services  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 11m  
nginx-service LoadBalancer 10.100.68.28 a236a82a1d2964f3fa64ad641dd25ee4-1896677718.us-east-1.elb.amazonaws.com 80:31859/TCP 72s  
C:\Windows\System32>  
C:\Windows\System32>
```

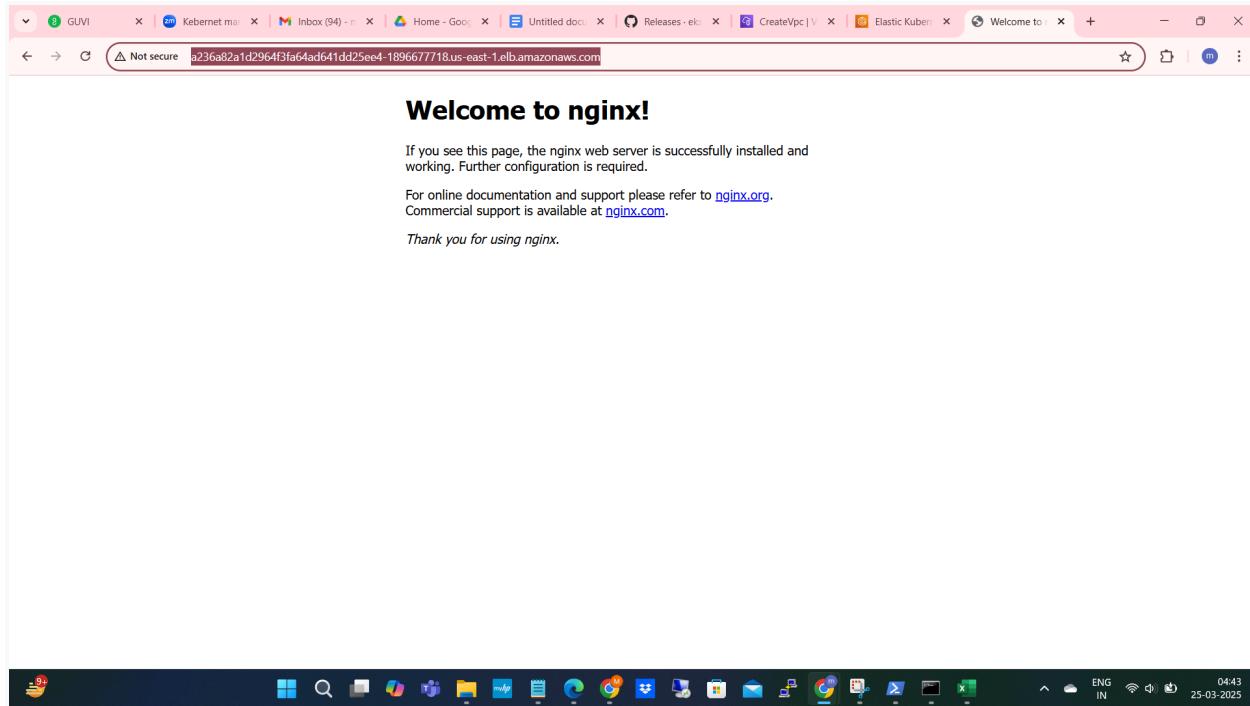
Nginx external ip is:

a236a82a1d2964f3fa64ad641dd25ee4-1896677718.us-east-1.elb.amazonaws.com

Now Access the Nginx application outside the cluster

url:

<http://a236a82a1d2964f3fa64ad641dd25ee4-1896677718.us-east-1.elb.amazonaws.com/>



Able to access the application outside the cluster outside

After deployment verify the cluster:

The screenshot shows the AWS EKS Cluster Overview page for the cluster 'my-cluster-nginx'. The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services like Amazon ECR and AWS Batch. The main content area displays 'Cluster info' with status 'Active', Kubernetes version '1.32', support period until March 21, 2026, and provider EKS. It also shows 'Cluster health issues' (0), 'Upgrade insights' (4), and 'Node health issues' (0). Below this are tabs for Overview, Resources, Compute, Networking, Add-ons, Access, Observability, Update history, and Tags. The 'Overview' tab is selected. A 'Details' section provides API server endpoint (<https://CA8EEA12D865BBA76E33432A9BD0AB5.gr7.us-east-1.eks.amazonaws.com>), OpenID Connect provider URL (<https://oidc.eks.us-east-1.amazonaws.com/id/CA8EEA12D865BBA76E33432A9BD0AB5>), Certificate authority (long string of characters), Cluster IAM role ARN (<arn:aws:iam::905418201986:role/eksctl-my-cluster-nginx-cluster-ServiceRole-HEN852BXUwLp>), and Platform version 'eks.5'. The bottom navigation bar includes CloudShell, Feedback, and links to various AWS services.

Deployments:

The screenshot shows the AWS EKS Resources page, specifically the 'Workloads: Deployments' section. The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services like Amazon ECR and AWS Batch. The main content area shows a table of deployments across all namespaces. The table has columns for Name, Namespace, Type, Created, Pod count, and Status. Three entries are listed: 'coredns' (kube-system, deployments, created 2 hours ago, 2 pods, 2 Ready | 0 Failed | 2 Desired), 'metrics-server' (kube-system, deployments, created 2 hours ago, 2 pods, 2 Ready | 0 Failed | 2 Desired), and 'nginx-deployment' (default, deployments, created 30 minutes ago, 2 pods, 2 Ready | 0 Failed | 2 Desired). The bottom navigation bar includes CloudShell, Feedback, and links to various AWS services.

Replicsets:

The screenshot shows the EKS console interface. The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services (Amazon ECR, AWS Batch). The main content area is titled "Workloads: ReplicaSets (3)". It displays a table with three entries:

Name	Namespace	Type	Created	Pod count	Status
coredns-6b9575c64c	kube-system	replicasets	2 hours ago	2	2 Ready 0 Failed 2 Desired
metrics-server-57b774cc8d	kube-system	replicasets	2 hours ago	2	2 Ready 0 Failed 2 Desired
nginx-deployment-96b9d695	default	replicasets	27 minutes ago	2	2 Ready 0 Failed 2 Desired

At the bottom, there are links for CloudShell and Feedback, and a standard Windows taskbar.

Daemon sets:

The screenshot shows the EKS console interface. The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services (Amazon ECR, AWS Batch). The main content area is titled "Workloads: DaemonSets (5)". It displays a table with five entries:

Name	Namespace	Type	Created	Pod count	Status
aws-node	kube-system	daemonsets	2 hours ago	3	3 Ready 0 Failed 3 Desired
dcmg-server	kube-system	daemonsets	2 hours ago	0	0 Ready 0 Failed 0 Desired
eks-node-monitoring-agent	kube-system	daemonsets	2 hours ago	3	3 Ready 0 Failed 3 Desired
eks-pod-identity-agent	kube-system	daemonsets	2 hours ago	3	3 Ready 0 Failed 3 Desired
kube-proxy	kube-system	daemonsets	2 hours ago	3	3 Ready 0 Failed 3 Desired

At the bottom, there are links for CloudShell and Feedback, and a standard Windows taskbar.

Nodes:

The screenshot shows the AWS EKS console interface. The left sidebar navigation includes 'Clusters', 'Settings' (Console settings), 'Amazon EKS Anywhere' (Enterprise Subscriptions), and 'Related services' (Amazon ECR, AWS Batch). The main content area displays 'Resource types' under 'Cluster' with 'Nodes' selected. A table titled 'Cluster: Nodes (3)' lists three worker machines. The columns are 'Node name', 'Instance type', 'Compute', 'Managed by', 'Created', and 'Status'. Each node entry includes a copy icon and a 'Read' status indicator.

Node name	Instance type	Compute	Managed by	Created	Status
ip-192-168-24-38.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	2 hours ago	Read
ip-192-168-59-39.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	2 hours ago	Read
ip-192-168-7-44.ec2.internal	t3.medium	Node group	my-nodegrp-nginx	2 hours ago	Read

Name spaces:

The screenshot shows the AWS EKS console interface. The left sidebar navigation includes 'Clusters', 'Settings' (Console settings), 'Amazon EKS Anywhere' (Enterprise Subscriptions), and 'Related services' (Amazon ECR, AWS Batch). The main content area displays 'Resource types' under 'Cluster' with 'Namespaces' selected. A table titled 'Cluster: Namespaces (4)' lists four namespaces: default, kube-node-lease, kube-public, and kube-system. The columns are 'Name' and 'Created'. Each namespace entry includes a copy icon and a 'Created' timestamp.

Name	Created
default	2 hours ago
kube-node-lease	2 hours ago
kube-public	2 hours ago
kube-system	2 hours ago

Services:

The screenshot shows the AWS EKS Services page for a cluster named "my-cluster-nginx". The left sidebar includes sections for Clusters, Settings, Amazon EKS Anywhere, and Related services (Amazon ECR, AWS Batch). The main content area displays a table titled "Service and networking: Services (5)".

Name	Created
eks-extension-metrics-api	2 hours ago
kube-dns	2 hours ago
kubernetes	2 hours ago
metrics-server	2 hours ago
nginx-service	24 minutes ago

End points:

The screenshot shows the AWS EKS Endpoints page for the same cluster "my-cluster-nginx". The left sidebar is identical to the Services page. The main content area displays a table titled "Service and networking: Endpoints (5)".

Name	Created
eks-extension-metrics-api	2 hours ago
kube-dns	2 hours ago
kubernetes	2 hours ago
metrics-server	2 hours ago
nginx-service	25 minutes ago

Config maps:

The screenshot shows the AWS Lambda function configuration page. The left sidebar lists functions: 'my-lambda-function' (selected), 'my-lambda-function-1', 'my-lambda-function-2', and 'my-lambda-function-3'. The main area shows the configuration for 'my-lambda-function', including the Handler ('lambda_function.lambda_handler'), Runtime ('python3.8'), and Memory ('128'). The 'Environment' tab is selected, showing the environment variables: 'AWS_LAMBDA_FUNCTION_NAME' (value: 'my-lambda-function'), 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE' (value: '128'), 'AWS_LAMBDA_FUNCTION_TIMEOUT' (value: '3'), and 'AWS_LAMBDA_HANDLER' (value: 'lambda_function.lambda_handler'). The 'Logs' tab is also visible.

Now delete the cluster:

```
$ eksctl delete cluster --name my-cluster-nginx --region us-east-1
```

```
C:\Windows\System32>Administrator: Command Prompt
C:\Windows\System32>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-deployment-96b9d695-2tbpk   1/1     Running   0          3m28s
nginx-deployment-96b9d695-bgd6r   1/1     Running   0          3m28s

C:\Windows\System32>notepad nginx-service.yaml
C:\Windows\System32>kubectl apply -f nginx-service.yaml
service/nginx-service created

C:\Windows\System32>
C:\Windows\System32>kubectl get services
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
Kubernetes     ClusterIP  10.109.8.1   <none>        443/TCP       111m
nginx-service  LoadBalancer 10.109.68.28  a236a82a1d2964f3fa64ad641dd25ee4-1896677718.us-east-1.elb.amazonaws.com  80:31859/TCP   72s

C:\Windows\System32>
C:\Windows\System32>eksctl delete cluster --name my-cluster-nginx --region us-east-1
2025-03-25 05:13:35 [ ] deleting EKS cluster "my-cluster-nginx"
2025-03-25 05:13:37 [ ] will drain 0 unmanaged nodegroup(s) in cluster "my-cluster-nginx"
2025-03-25 05:13:39 [ ] starting parallel draining, max in-flight of 1
2025-03-25 05:13:39 [ ] deleted 0 Fargate profile(s)
2025-03-25 05:13:41 [ ] kubeconfig has been updated
2025-03-25 05:13:41 [ ] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2025-03-25 05:14:27 [ ] 2 sequential tasks: { delete nodegroup "my-nodegrp-nginx", delete cluster control plane "my-cluster-nginx" [async] }
2025-03-25 05:14:27 [ ] will delete stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:14:27 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx" to get deleted
2025-03-25 05:14:59 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:15:49 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:17:24 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:19:06 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:20:51 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:22:19 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:23:14 [ ] waiting for CloudFormation stack "eksctl-my-cluster-nginx-nodegroup-my-nodegrp-nginx"
2025-03-25 05:23:14 [ ] Will delete stack "eksctl-my-cluster-nginx-cluster"
2025-03-25 05:23:16 [ ] all cluster resources were deleted

C:\Windows\System32>
```

Check the luster deleted or not in AWS console

The screenshot shows the AWS EKS Clusters page. The left sidebar includes sections for Amazon Elastic Kubernetes Service, Amazon EKS Anywhere, and Related services like Amazon ECR and AWS Batch. The main content area displays a table titled 'Clusters (0)'. A message states 'No clusters' and 'You do not have any clusters.' A 'Create cluster' button is visible. The top navigation bar shows the URL 'us-east-1.console.aws.amazon.com/eks/clusters?region=us-east-1' and the region 'United States (N. Virginia)'. The bottom status bar shows the date and time as 25-03-2025.

All the resources also deleted

The screenshot shows the AWS EC2 Home page. The left sidebar lists EC2 Global View, Instances, Images, and Elastic Block Store. The main content area features a 'Resources' section with tables for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. All counts are 0. It also includes sections for Launch instance, Service health (AWS Health Dashboard), and Zones. A message at the top says 'You can change your default landing page for EC2.' A 'Change landing page' button is available. The top navigation bar shows the URL 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home' and the region 'United States (N. Virginia)'. The bottom status bar shows the date and time as 25-03-2025.

Instances terminated

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays a table of terminated instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. There are three rows, each corresponding to a terminated instance named 'my-cluster-nginx'. The instance IDs are i-0b1ac3b4c249d9038, i-0c91d515112aa4ff1, and i-071cdce7d99104b. All three instances are in the 'terminated' state, t3.medium type, and located in us-east-1c, us-east-1b, and us-east-1b respectively. The status check is '-' for all. The alarm status is 'View alarms +'. The availability zone is 'us-east-1c' for the first and 'us-east-1b' for the others. The public IP is listed as '-'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
my-cluster-nginx...	i-0b1ac3b4c249d9038	Terminated	t3.medium	-	View alarms +	us-east-1c	-
my-cluster-nginx...	i-0c91d515112aa4ff1	Terminated	t3.medium	-	View alarms +	us-east-1b	-
my-cluster-nginx...	i-071cdce7d99104b	Terminated	t3.medium	-	View alarms +	us-east-1b	-