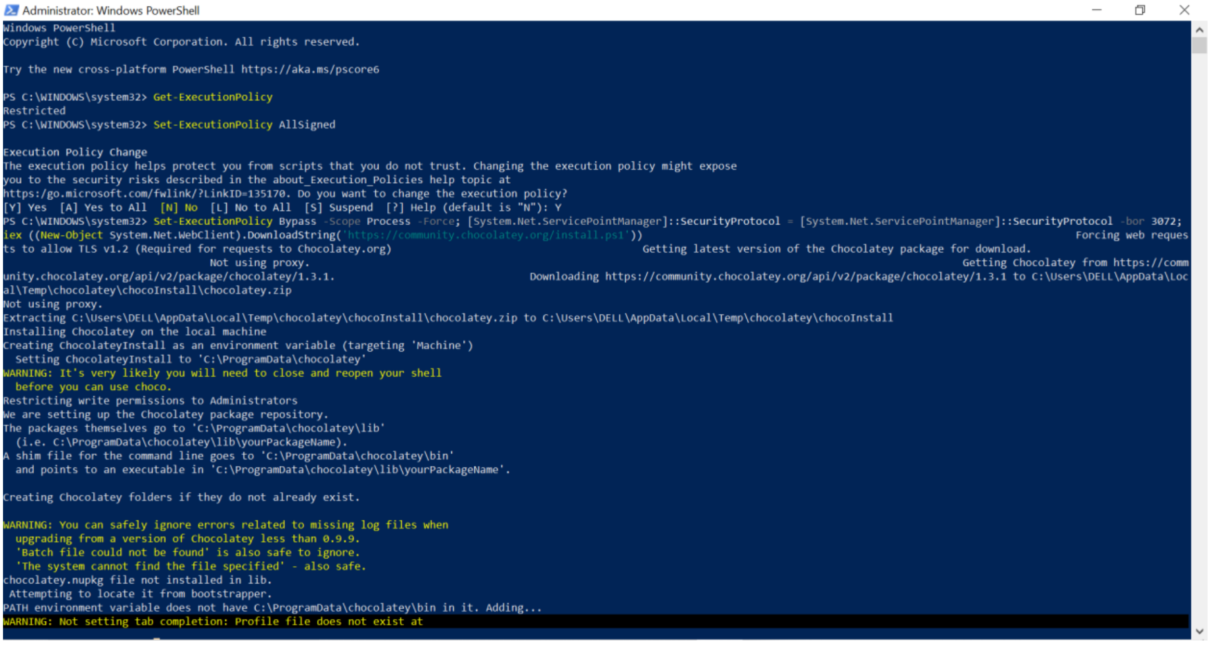
*Install chocolatey

First, ensure that you are using an ***administrative shell***

*To install the latest minikube stable release on x86-64 Windows using Chocolatey:

*With PowerShell, you must ensure Get-ExecutionPolicy is not Restricted. We suggest using **Bypass** to bypass the policy to get things installed or **AllSigned** for quite a bit more security.

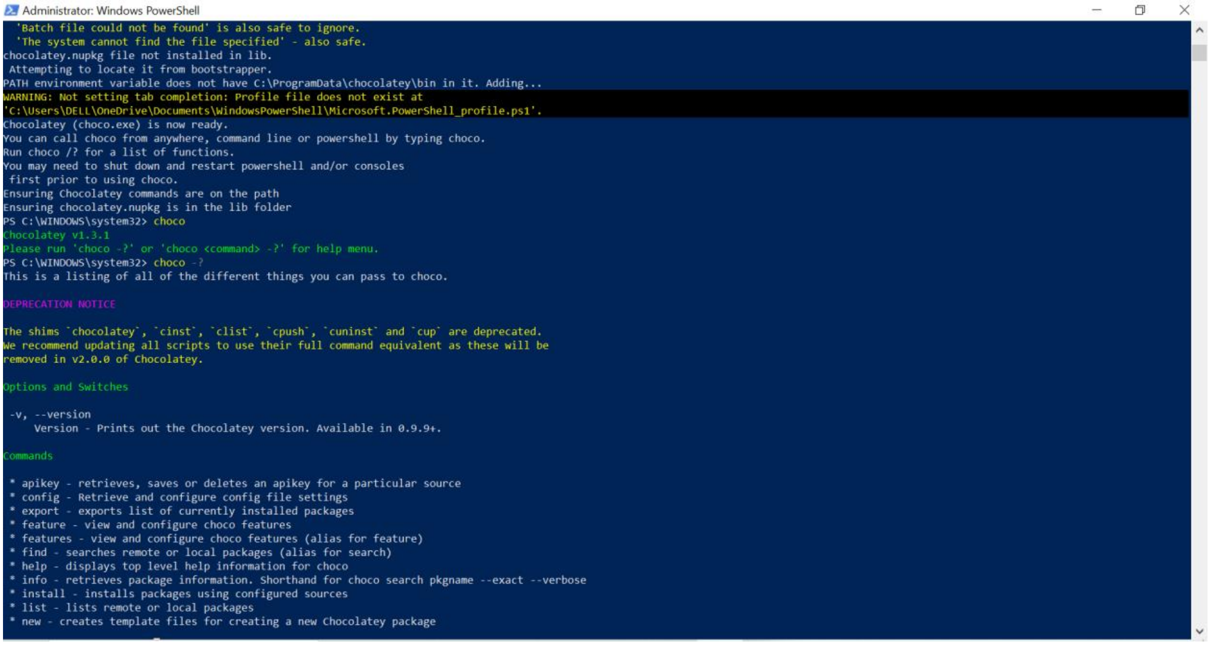
- Run **Get-ExecutionPolicy**. If it returns **Restricted**, then run **Set-ExecutionPolicy AllSigned** or **Set-ExecutionPolicy Bypass -Scope Process**.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> Get-ExecutionPolicy
Restricted
PS C:\WINDOWS\system32> Set-ExecutionPolicy AllSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
PS C:\WINDOWS\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072;
iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))                                          Forcing web reques
ts to allow TLS v1.2 (Required for requests to Chocolatey.org)                                   Getting latest version of the Chocolatey package for download.
                                Not using proxy.                                                                                                  Getting Chocolatey from https://comm
unity.chocolatey.org/api/v2/package/chocolatey/1.3.1.                              Downloading https://community.chocolatey.org/api/v2/package/chocolatey/1.3.1 to C:\Users\DELL\AppData\Loc
al\Temp\chocolatey\chocoInstall\chocolatey.zip
Not using proxy.
Extracting C:\Users\DELL\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\DELL\AppData\Local\Temp\chocolatey\chocoInstall
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
  Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
  before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
  (i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
  upgrading from a version of Chocolatey less than 0.9.9.
  'Batch file could not be found' is also safe to ignore.
  'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
 Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
```

Now run the following command:

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))



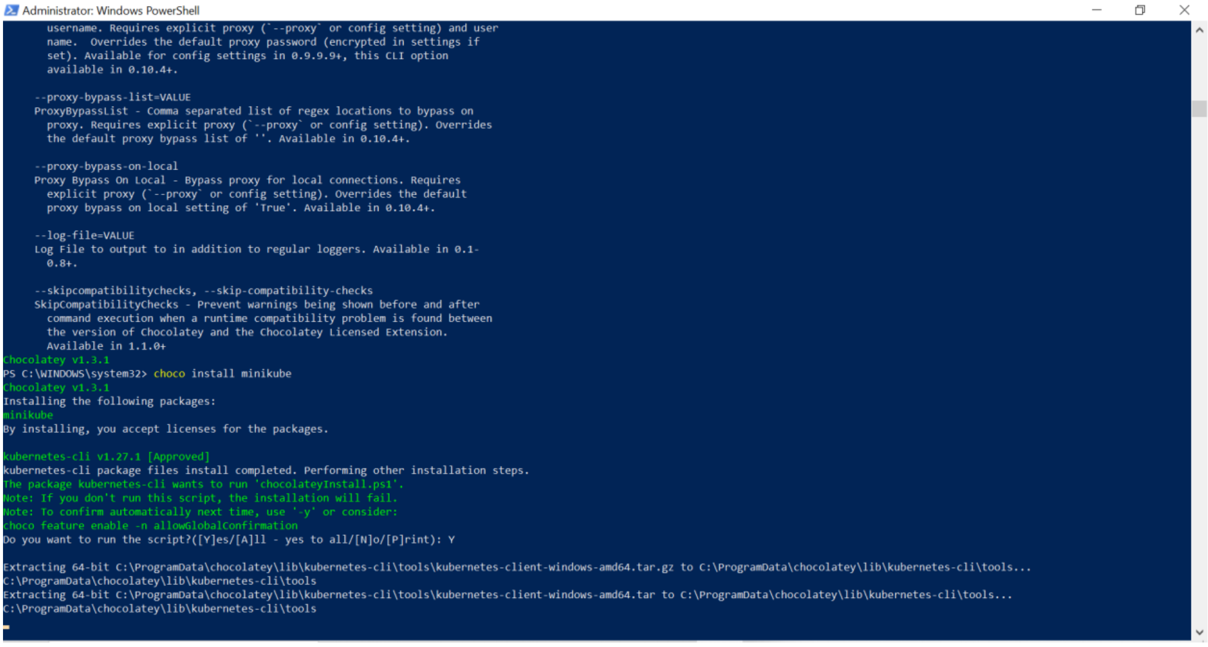
```
  'Batch file could not be found' is also safe to ignore.
  'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
 Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\DELL\OneDrive\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
 first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\WINDOWS\system32> choco
Chocolatey v1.3.1
Please run 'choco -?' or 'choco <command> -?' for help menu.
PS C:\WINDOWS\system32> choco -?
This is a listing of all of the different things you can pass to choco.

DEPRECATION NOTICE

The shims `chocolatey`, `cinst`, `clist`, `cpush`, `cuninst` and `cup` are deprecated.
We recommend updating all scripts to use their full command equivalent as these will be
removed in v2.0.0 of Chocolatey.

Options and Switches

 -v, --version
     Version - Prints out the Chocolatey version. Available in 0.9.9+.

Commands

 * apikey - retrieves, saves or deletes an apikey for a particular source
 * config - Retrieve and configure config file settings
 * export - exports list of currently installed packages
 * feature - view and configure choco features
 * features - view and configure choco features (alias for feature)
 * find - searches remote or local packages (alias for search)
 * help - displays top level help information for choco
 * info - retrieves package information. Shorthand for choco search pkgname --exact --verbose
 * install - installs packages using configured sources
 * list - lists remote or local packages
 * new - creates template files for creating a new Chocolatey package
```

*Now chocolatey is installed.

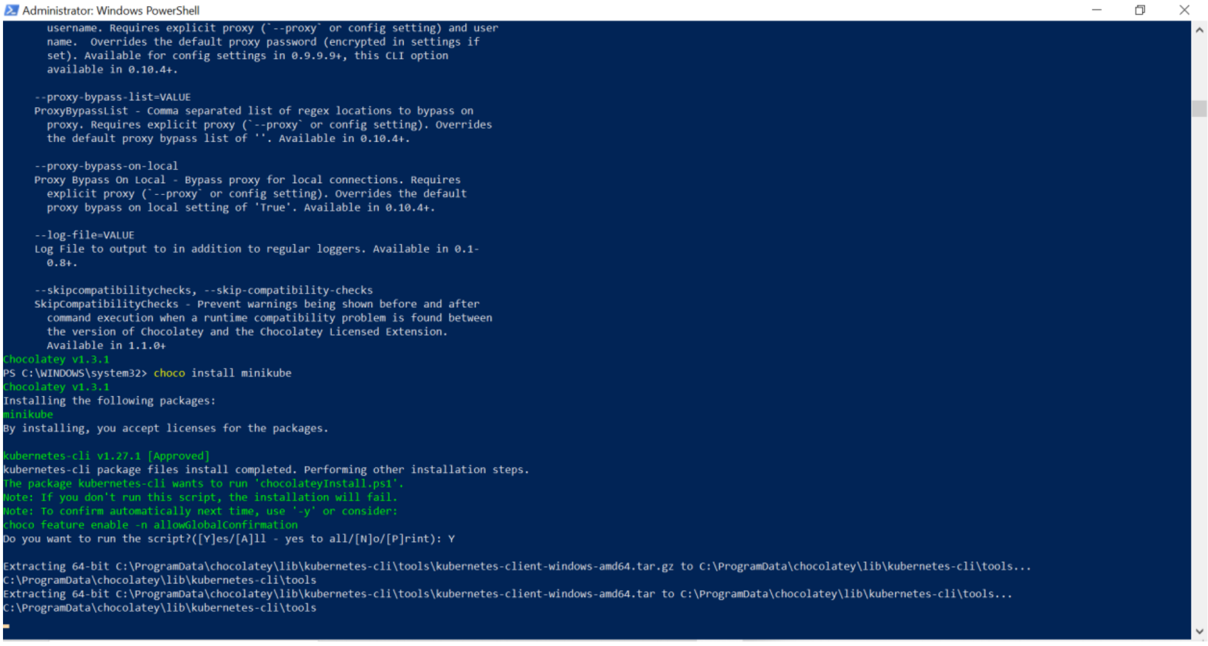
*You can now install minikube and Kubernetes-cli

To install minikube use the command:

choco install minikube



```
       username. Requires explicit proxy (`--proxy` or config setting) and user
       name.  Overrides the default proxy password (encrypted in settings if
       set). Available for config settings in 0.9.9.9+, this CLI option
       available in 0.10.4+.

     --proxy-bypass-list=VALUE
     ProxyBypassList - Comma separated list of regex locations to bypass on
       proxy. Requires explicit proxy (`--proxy` or config setting). Overrides
       the default proxy bypass list of ''. Available in 0.10.4+.

     --proxy-bypass-on-local
     Proxy Bypass On Local - Bypass proxy for local connections. Requires
       explicit proxy (`--proxy` or config setting). Overrides the default
       proxy bypass on local setting of 'True'. Available in 0.10.4+.

     --log-file=VALUE
     Log File to output to in addition to regular loggers. Available in 0.1-
       0.8+.

     --skipcompatibilitychecks, --skip-compatibility-checks
     SkipCompatibilityChecks - Prevent warnings being shown before and after
       command execution when a runtime compatibility problem is found between
       the version of Chocolatey and the Chocolatey Licensed Extension.
       Available in 1.1.0+
Chocolatey v1.3.1
PS C:\WINDOWS\system32> choco install minikube
Chocolatey v1.3.1
Installing the following packages:
minikube
By installing, you accept licenses for the packages.

kubernetes-cli v1.27.1 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli 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): Y

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
```

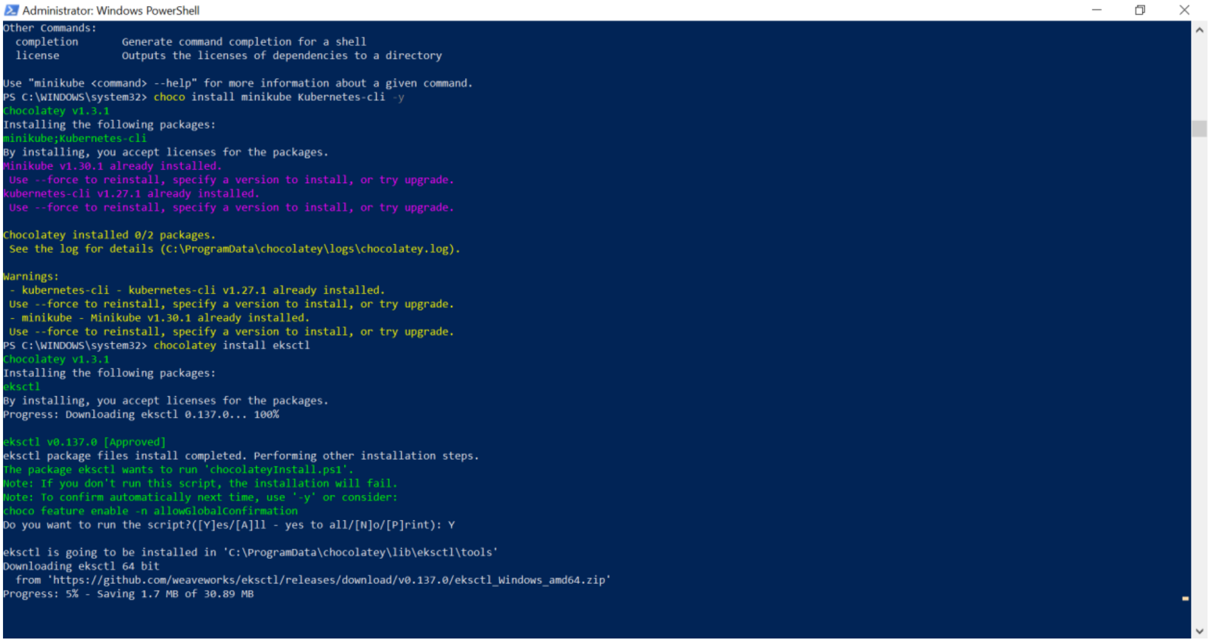


```
Administrator: Windows PowerShell
      username. Requires explicit proxy (`--proxy` or config setting) and user
      name.  Overrides the default proxy password (encrypted in settings if
      set). Available for config settings in 0.9.9.9+, this CLI option
      available in 0.10.4+.

    --proxy-bypass-list=VALUE
    ProxyBypassList - Comma separated list of regex locations to bypass on
      proxy. Requires explicit proxy (`--proxy` or config setting). Overrides
      the default proxy bypass list of ''. Available in 0.10.4+.

    --proxy-bypass-on-local
    Proxy Bypass On Local - Bypass proxy for local connections. Requires
      explicit proxy (`--proxy` or config setting). Overrides the default
      proxy bypass on local setting of 'True'. Available in 0.10.4+.

    --log-file=VALUE
    Log File to output to in addition to regular loggers. Available in 0.1-
      0.8+.

    --skipcompatibilitychecks, --skip-compatibility-checks
    SkipCompatibilityChecks - Prevent warnings being shown before and after
      command execution when a runtime compatibility problem is found between
      the version of Chocolatey and the Chocolatey Licensed Extension.
      Available in 1.1.0+
Chocolatey v1.3.1
PS C:\WINDOWS\system32> choco install minikube
Chocolatey v1.3.1
Installing the following packages:
minikube
By installing, you accept licenses for the packages.

kubernetes-cli v1.27.1 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli 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): Y

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
```

*Both minikube and Kubernetes-cli are installed successfully.

*Now install eksctl

chocolatey install eksctl



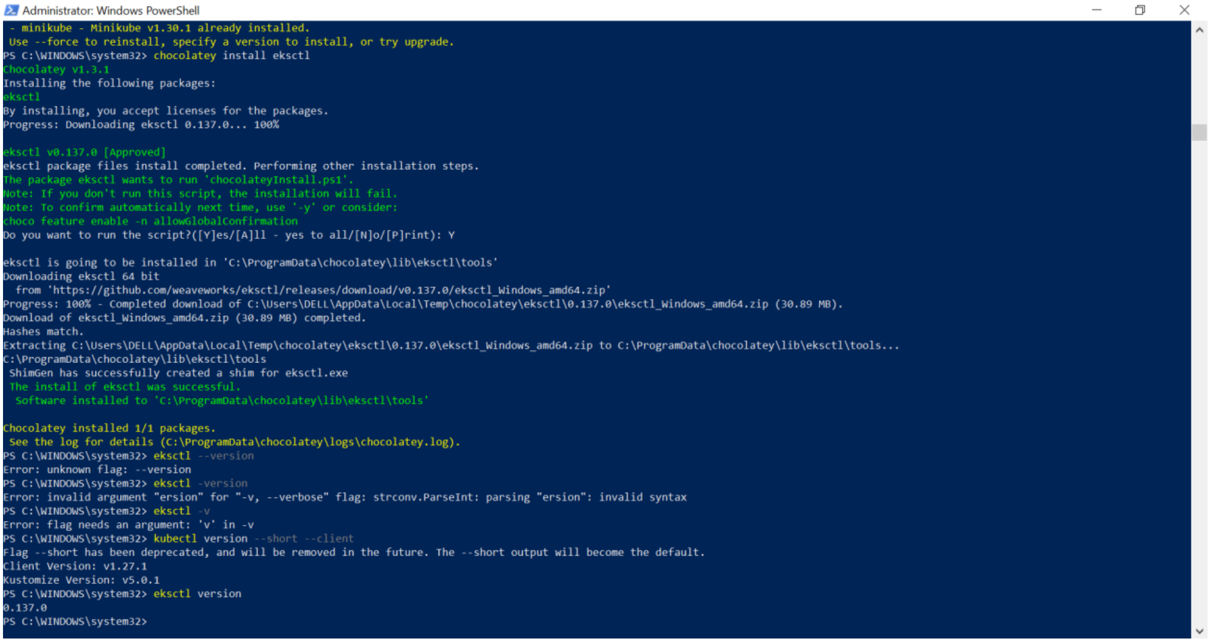
```
Administrator: Windows PowerShell
Other Commands:
  completion       Generate command completion for a shell
  license          Outputs the licenses of dependencies to a directory

Use "minikube <command> --help" for more information about a given command.
PS C:\WINDOWS\system32> choco install minikube Kubernetes-cli -y
Chocolatey v1.3.1
Installing the following packages:
minikube;Kubernetes-cli
By installing, you accept licenses for the packages.
Minikube v1.30.1 already installed.
 Use --force to reinstall, specify a version to install, or try upgrade.
kubernetes-cli v1.27.1 already installed.
 Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/2 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
 - kubernetes-cli - kubernetes-cli v1.27.1 already installed.
 Use --force to reinstall, specify a version to install, or try upgrade.
 - minikube - Minikube v1.30.1 already installed.
 Use --force to reinstall, specify a version to install, or try upgrade.
PS C:\WINDOWS\system32> chocolatey install eksctl
Chocolatey v1.3.1
Installing the following packages:
eksctl
By installing, you accept licenses for the packages.
Progress: Downloading eksctl 0.137.0... 100%

eksctl v0.137.0 [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): Y

eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
Downloading eksctl 64 bit
  from 'https://github.com/weaveworks/eksctl/releases/download/v0.137.0/eksctl_Windows_amd64.zip'
Progress: 5% - Saving 1.7 MB of 30.89 MB
```

*Once the eksctl is installed check the version using below command:

eksctl version

*to check the version of kubectl use the below command.

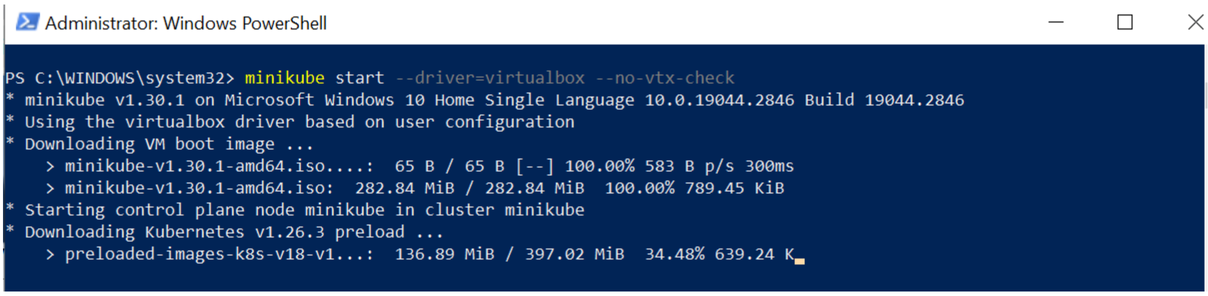
kubectl version –short –client



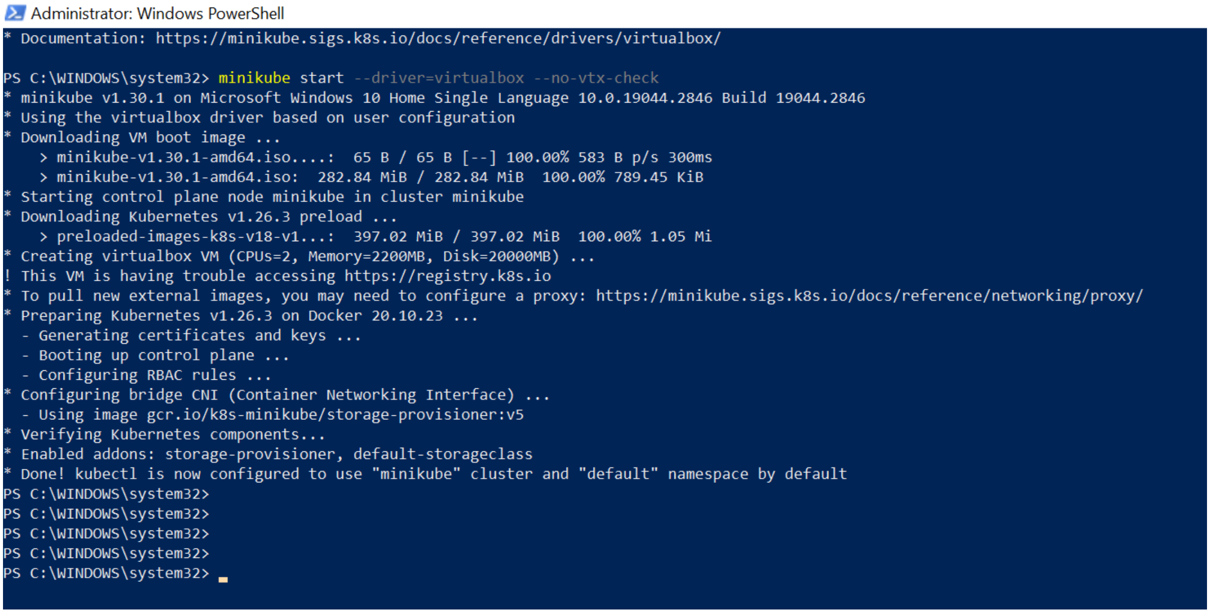
```
 - minikube - Minikube v1.30.1 already installed.
 Use --force to reinstall, specify a version to install, or try upgrade.
PS C:\WINDOWS\system32> chocolatey install eksctl
Chocolatey v1.3.1
Installing the following packages:
eksctl
By installing, you accept licenses for the packages.
Progress: Downloading eksctl 0.137.0... 100%

eksctl v0.137.0 [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): Y

eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
Downloading eksctl 64 bit
  from 'https://github.com/weaveworks/eksctl/releases/download/v0.137.0/eksctl_Windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\DELL\AppData\Local\Temp\chocolatey\eksctl\0.137.0\eksctl_Windows_amd64.zip (30.89 MB).
Download of eksctl_Windows_amd64.zip (30.89 MB) completed.
Hashes match.
Extracting C:\Users\DELL\AppData\Local\Temp\chocolatey\eksctl\0.137.0\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.
  Software installed 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> eksctl --version
Error: unknown flag: --version
PS C:\WINDOWS\system32> eksctl -version
Error: invalid argument "ersion" for "-v, --verbose" flag: strconv.ParseInt: parsing "ersion": invalid syntax
PS C:\WINDOWS\system32> eksctl -v
Error: flag needs an argument: 'v' in -v
PS C:\WINDOWS\system32> kubectl version --short --client
Flag --short has been deprecated, and will be removed in the future. The --short output will become the default.
Client Version: v1.27.1
Kustomize Version: v5.0.1
PS C:\WINDOWS\system32> eksctl version
0.137.0
PS C:\WINDOWS\system32>
```

*Start minikube

minikube start --driver=virtualbox --no-vtx-check



```
PS C:\WINDOWS\system32> minikube start --driver=virtualbox --no-vtx-check
* minikube v1.30.1 on Microsoft Windows 10 Home Single Language 10.0.19044.2846 Build 19044.2846
* Using the virtualbox driver based on user configuration
* Downloading VM boot image ...
    > minikube-v1.30.1-amd64.iso....:  65 B / 65 B [--] 100.00% 583 B p/s 300ms
    > minikube-v1.30.1-amd64.iso:  282.84 MiB / 282.84 MiB  100.00% 789.45 KiB
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.26.3 preload ...
    > preloaded-images-k8s-v18-v1...:  136.89 MiB / 397.02 MiB  34.48% 639.24 K
```

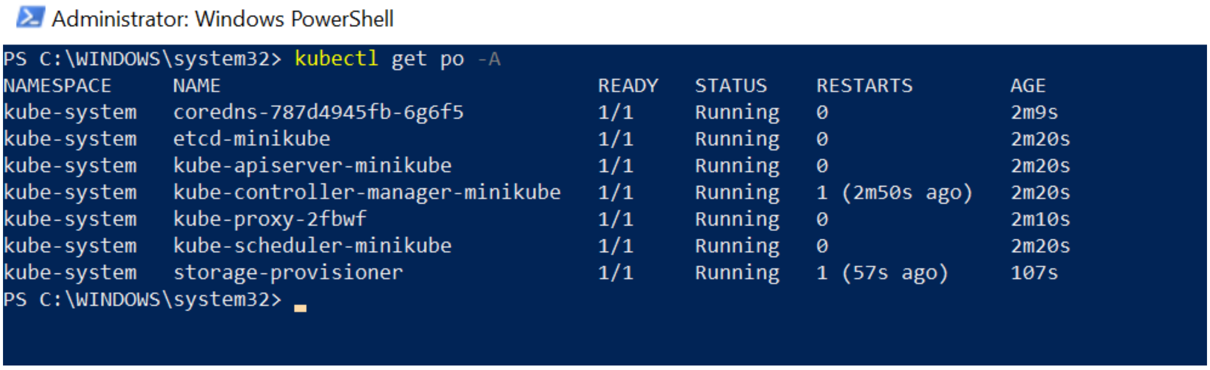


```
* Documentation: https://minikube.sigs.k8s.io/docs/reference/drivers/virtualbox/

PS C:\WINDOWS\system32> minikube start --driver=virtualbox --no-vtx-check
* minikube v1.30.1 on Microsoft Windows 10 Home Single Language 10.0.19044.2846 Build 19044.2846
* Using the virtualbox driver based on user configuration
* Downloading VM boot image ...
    > minikube-v1.30.1-amd64.iso....:  65 B / 65 B [--] 100.00% 583 B p/s 300ms
    > minikube-v1.30.1-amd64.iso:  282.84 MiB / 282.84 MiB  100.00% 789.45 KiB
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.26.3 preload ...
    > preloaded-images-k8s-v18-v1...:  397.02 MiB / 397.02 MiB  100.00% 1.05 Mi
* Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
! This VM is having trouble accessing https://registry.k8s.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.26.3 on Docker 20.10.23 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
```

*Interact with your cluster

If you already have kubectl installed, you can now use it to access your shiny new cluster:

kubectl get po -A

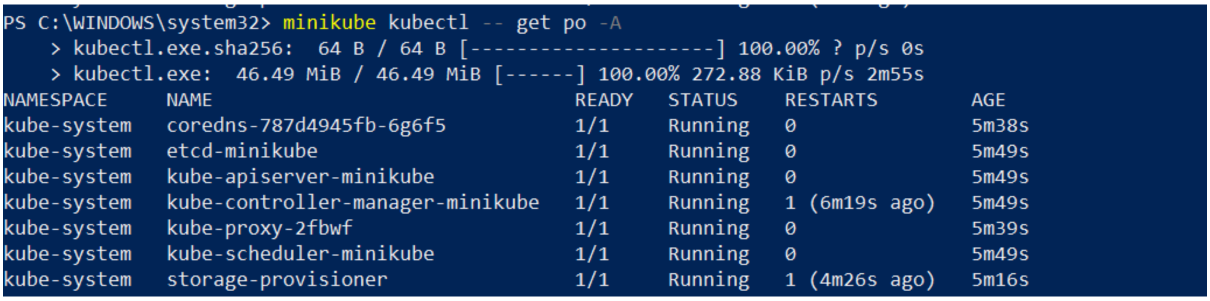


Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS        AGE
kube-system   coredns-787d4945fb-6g6f5           1/1     Running   0               2m9s
kube-system   etcd-minikube                      1/1     Running   0               2m20s
kube-system   kube-apiserver-minikube            1/1     Running   0               2m20s
kube-system   kube-controller-manager-minikube   1/1     Running   1 (2m50s ago)   2m20s
kube-system   kube-proxy-2fbwf                   1/1     Running   0               2m10s
kube-system   kube-scheduler-minikube            1/1     Running   0               2m20s
kube-system   storage-provisioner                1/1     Running   1 (57s ago)     107s
PS C:\WINDOWS\system32>
```

*Alternatively, minikube can download the appropriate version of kubectl and you should be able to use it like this:

minikube kubectl -- get po -A



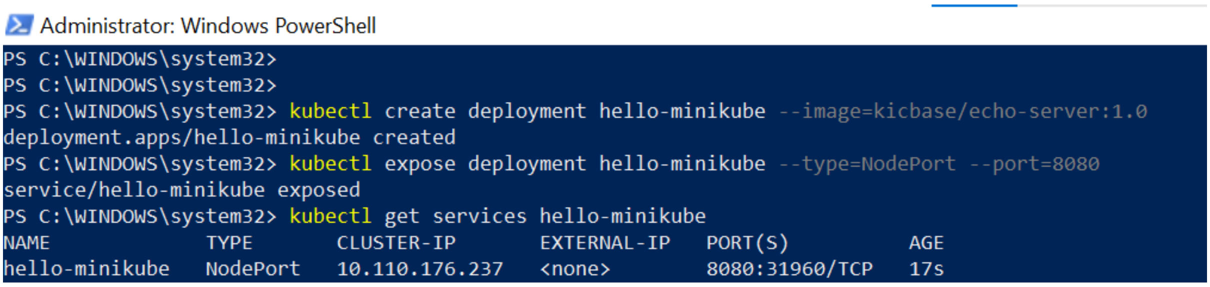
```
PS C:\WINDOWS\system32> minikube kubectl -- get po -A
    > kubectl.exe.sha256:  64 B / 64 B [--------------------] 100.00% ? p/s 0s
    > kubectl.exe:  46.49 MiB / 46.49 MiB [------] 100.00% 272.88 KiB p/s 2m55s
NAMESPACE     NAME                               READY   STATUS    RESTARTS        AGE
kube-system   coredns-787d4945fb-6g6f5           1/1     Running   0               5m38s
kube-system   etcd-minikube                      1/1     Running   0               5m49s
kube-system   kube-apiserver-minikube            1/1     Running   0               5m49s
kube-system   kube-controller-manager-minikube   1/1     Running   1 (6m19s ago)   5m49s
kube-system   kube-proxy-2fbwf                   1/1     Running   0               5m39s
kube-system   kube-scheduler-minikube            1/1     Running   0               5m49s
kube-system   storage-provisioner                1/1     Running   1 (4m26s ago)   5m16s
```

*Create a sample deployment and expose it on port 8080:

kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
kubectl expose deployment hello-minikube --type=NodePort --port=8080

It may take a moment, but your deployment will soon show up when you run:

kubectl get services hello-minikube



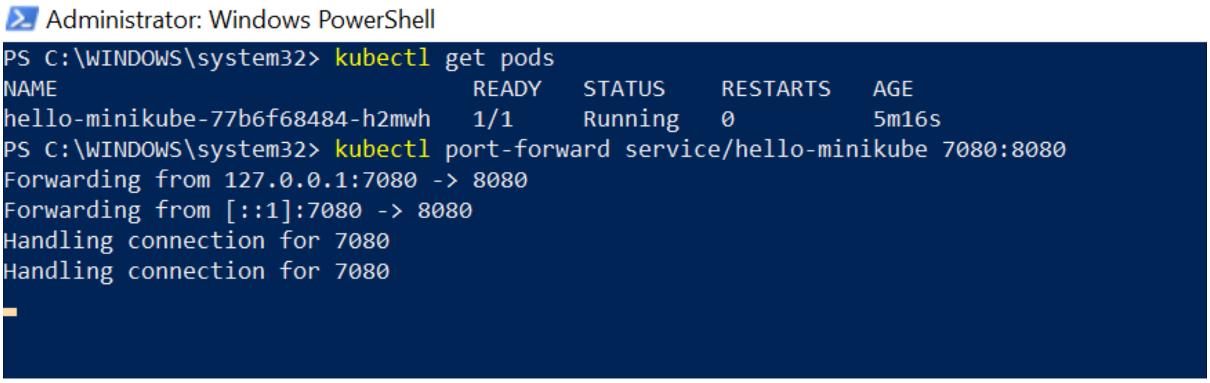
Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
PS C:\WINDOWS\system32> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS C:\WINDOWS\system32> kubectl get services hello-minikube
NAME             TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.110.176.237   <none>        8080:31960/TCP   17s
```

The **kubectl get pods** command is used to get the list of pods running in the default namespace of your Kubernetes cluster.

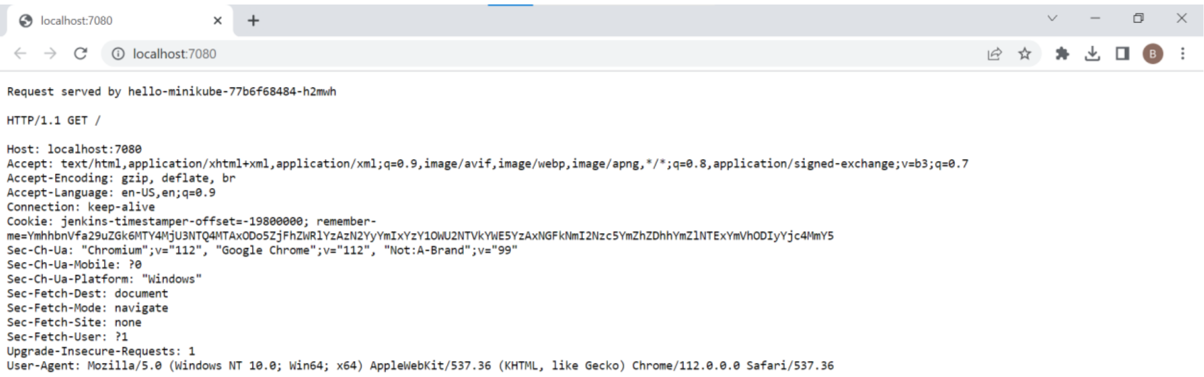
use kubectl to forward the port:

kubectl port-forward service/hello-minikube 7080:8080



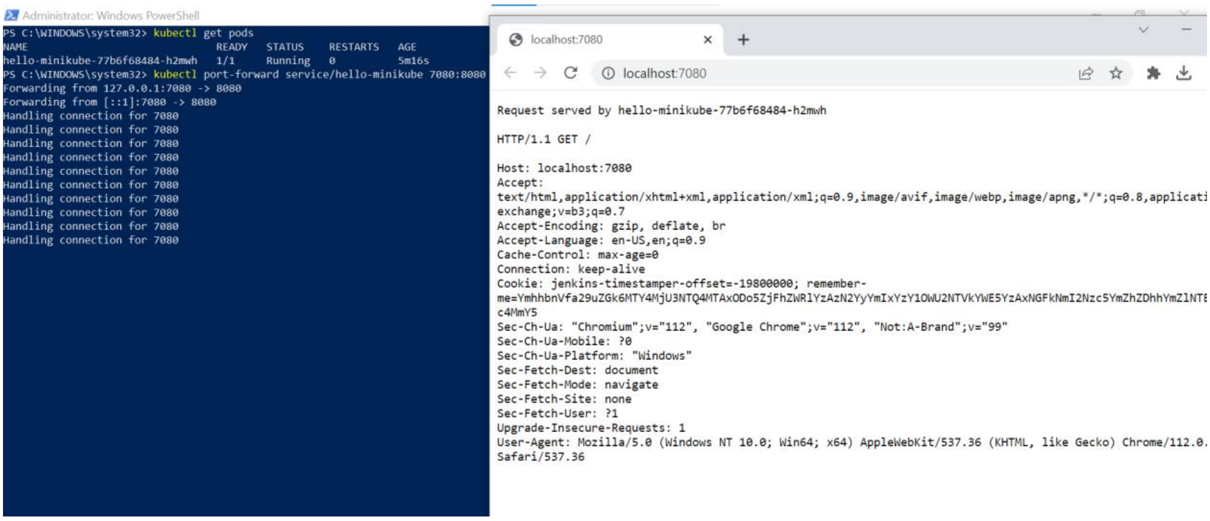
```
PS C:\WINDOWS\system32> kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-77b6f68484-h2mwh   1/1     Running   0          5m16s
PS C:\WINDOWS\system32> kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080
Handling connection for 7080
```

*Your application is now available at http://localhost:7080/



```
Request served by hello-minikube-77b6f68484-h2mwh

HTTP/1.1 GET /

Host: localhost:7080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Cookie: jenkins-timestamper-offset=-19800000; remember-
me=YmhhbnVfa29uZGk6MTY4MjU3NTQ4MTAxODo5ZjFhZWRlYzAzN2YyYmIxYzY1OWU2NTVkYWE5YzAxNGFkNmI2Nzc5YmZhZDhhYmZlNTExYmVhODIyYjc4MmY5
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
```

*You can be able to see the request metadata in the application output.



```
PS C:\WINDOWS\system32> kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-77b6f68484-h2mwh   1/1     Running   0          5m16s
PS C:\WINDOWS\system32> kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
```

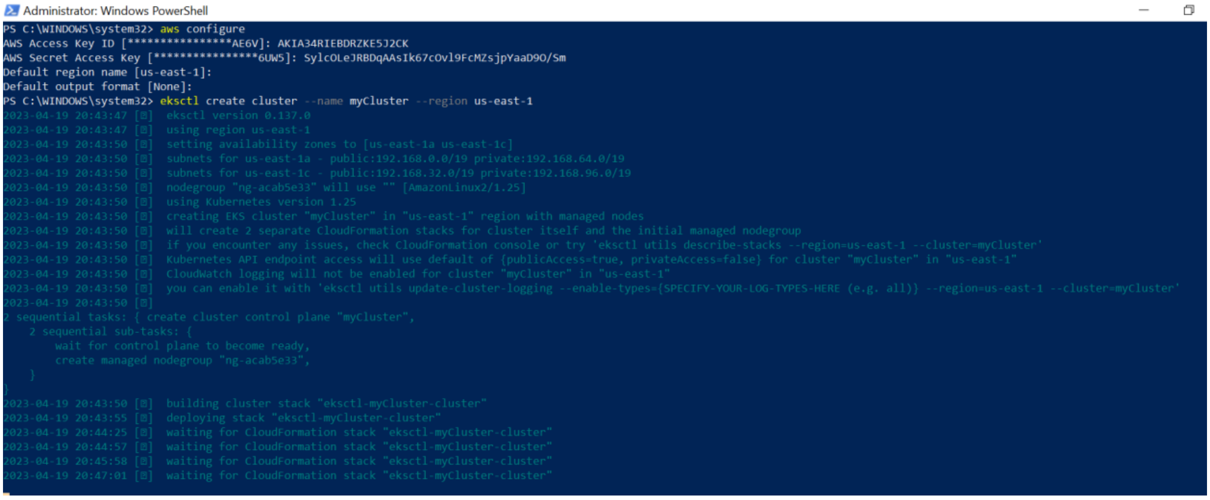
```
Request served by hello-minikube-77b6f68484-h2mwh

HTTP/1.1 GET /

Host: localhost:7080
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Cookie: jenkins-timestamper-offset=-19800000; remember-
me=YmhhbnVfa29uZGk6MTY4MjU3NTQ4MTAxODo5ZjFhZWRlYzAzN2YyYmIxYzY1OWU2NTVkYWE5YzAxNGFkNmI2Nzc5YmZhZDhhYmZlNTE
c4MmY5
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.
Safari/537.36
```

*Create an IAM user in your aws management console and create an access key.

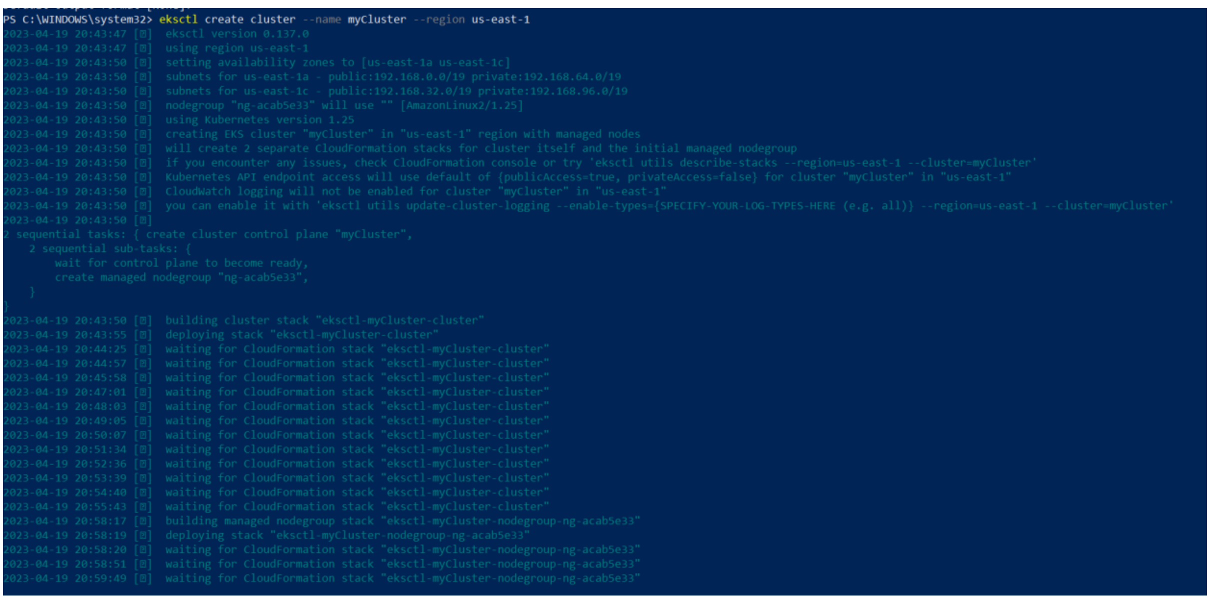
*Give **aws configure** command to configure and give access key id ,secret access key and region name.

*To create a cluster using eksctl give the command :

eksctl create cluster --name <cluster-name> --region <region>

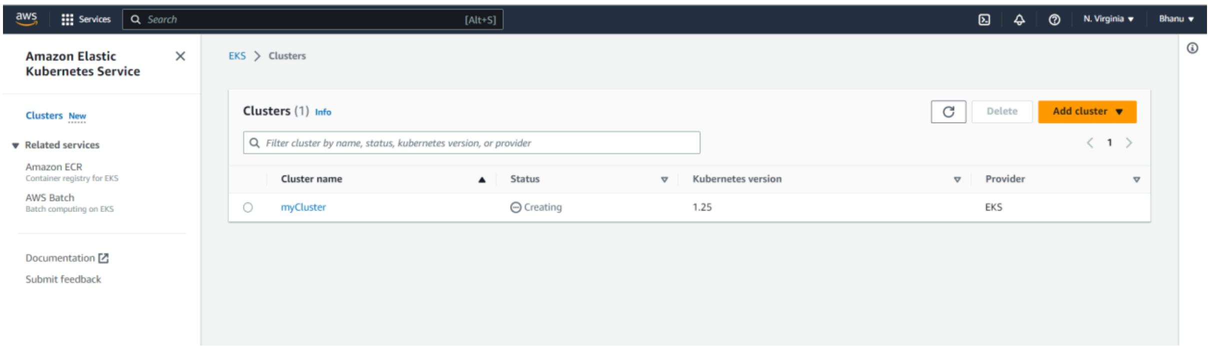


```
PS C:\WINDOWS\system32> aws configure
AWS Access Key ID [****************AE6V]: AKIA34RIEBDRZKE5J2CK
AWS Secret Access Key [****************6UW5]: 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
Default region name [us-east-1]:
Default output format [None]:
PS C:\WINDOWS\system32> eksctl create cluster --name myCluster --region us-east-1
2023-04-19 20:43:47 [ℹ]  eksctl version 0.137.0
2023-04-19 20:43:47 [ℹ]  using region us-east-1
2023-04-19 20:43:50 [ℹ]  setting availability zones to [us-east-1a us-east-1c]
2023-04-19 20:43:50 [ℹ]  subnets for us-east-1a - public:192.168.0.0/19 private:192.168.64.0/19
2023-04-19 20:43:50 [ℹ]  subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2023-04-19 20:43:50 [ℹ]  nodegroup "ng-acab5e33" will use "" [AmazonLinux2/1.25]
2023-04-19 20:43:50 [ℹ]  using Kubernetes version 1.25
2023-04-19 20:43:50 [ℹ]  creating EKS cluster "myCluster" in "us-east-1" region with managed nodes
2023-04-19 20:43:50 [ℹ]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-04-19 20:43:50 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [ℹ]  CloudWatch logging will not be enabled for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [ℹ]
2 sequential tasks: { create cluster control plane "myCluster",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng-acab5e33",
    }
}
2023-04-19 20:43:50 [ℹ]  building cluster stack "eksctl-myCluster-cluster"
2023-04-19 20:43:55 [ℹ]  deploying stack "eksctl-myCluster-cluster"
2023-04-19 20:44:25 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:44:57 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:45:58 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:47:01 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
```



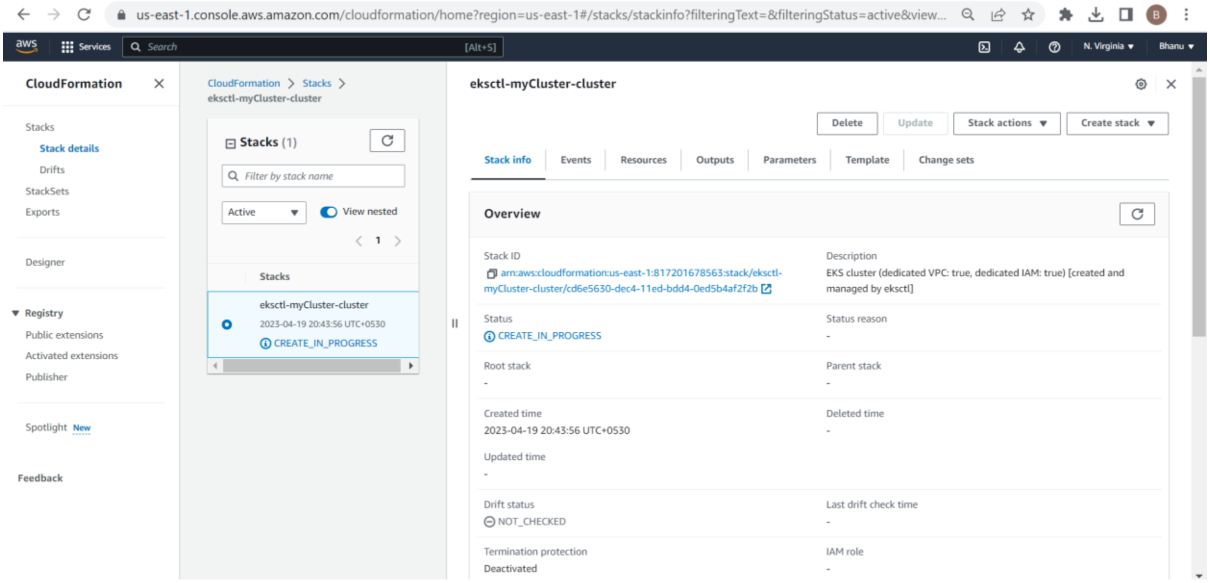
```
PS C:\WINDOWS\system32> eksctl create cluster --name myCluster --region us-east-1
2023-04-19 20:43:47 [ℹ]  eksctl version 0.137.0
2023-04-19 20:43:47 [ℹ]  using region us-east-1
2023-04-19 20:43:50 [ℹ]  setting availability zones to [us-east-1a us-east-1c]
2023-04-19 20:43:50 [ℹ]  subnets for us-east-1a - public:192.168.0.0/19 private:192.168.64.0/19
2023-04-19 20:43:50 [ℹ]  subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2023-04-19 20:43:50 [ℹ]  nodegroup "ng-acab5e33" will use "" [AmazonLinux2/1.25]
2023-04-19 20:43:50 [ℹ]  using Kubernetes version 1.25
2023-04-19 20:43:50 [ℹ]  creating EKS cluster "myCluster" in "us-east-1" region with managed nodes
2023-04-19 20:43:50 [ℹ]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-04-19 20:43:50 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [ℹ]  CloudWatch logging will not be enabled for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [ℹ]
2 sequential tasks: { create cluster control plane "myCluster",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng-acab5e33",
    }
}
2023-04-19 20:43:50 [ℹ]  building cluster stack "eksctl-myCluster-cluster"
2023-04-19 20:43:55 [ℹ]  deploying stack "eksctl-myCluster-cluster"
2023-04-19 20:44:25 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:44:57 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:45:58 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:47:01 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:48:03 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:49:05 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:50:07 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:51:34 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:52:36 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:53:39 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:54:40 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:55:43 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:58:17 [ℹ]  building managed nodegroup stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:19 [ℹ]  deploying stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:20 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:51 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:59:49 [ℹ]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
```

*Now go to EKS in AWS management console,then you will find the cluster.



	Cluster name ▲	Status	Kubernetes version	Provider
○	myCluster	⊖ Creating	1.25	EKS

*Open cloud formation to check the process of creation of a cluster.



us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/stackinfo?filteringText=&filteringStatus=active&view...

CloudFormation > Stacks > eksctl-myCluster-cluster

eksctl-myCluster-cluster

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Overview

Stack ID
arn:aws:cloudformation:us-east-1:817201678563:stack/eksctl-myCluster-cluster/cd6e5630-dec4-11ed-bdd4-0ed5b4af2f2b

Description
EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

Status
CREATE_IN_PROGRESS

Status reason
-

Root stack
-

Parent stack
-

Created time
2023-04-19 20:43:56 UTC+0530

Deleted time
-

Updated time
-

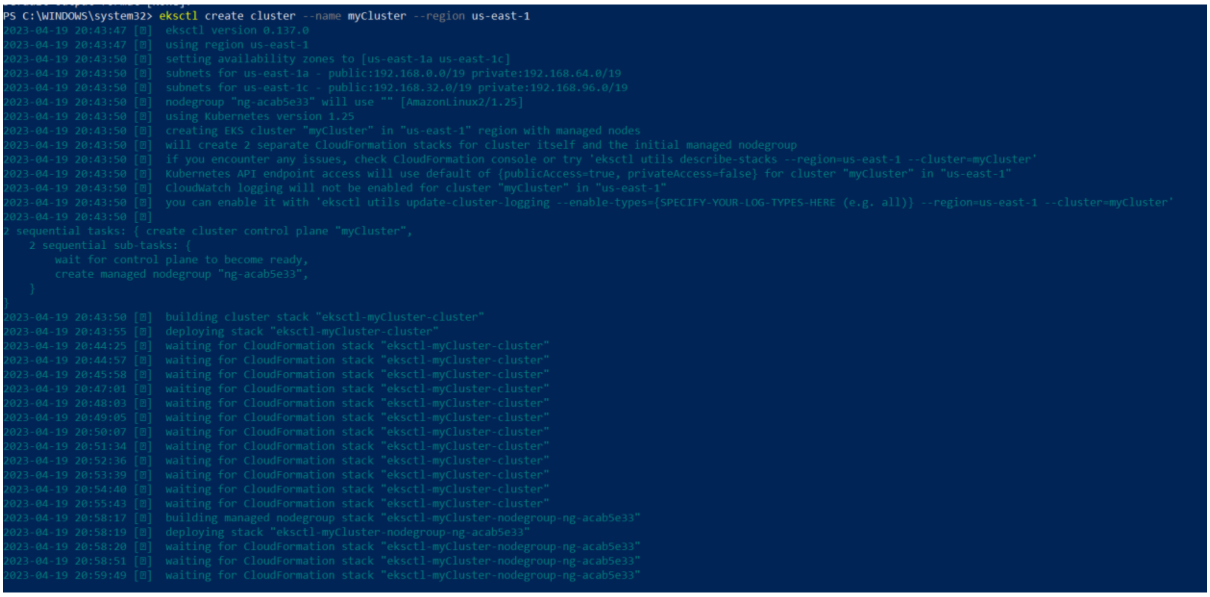
Drift status
NOT_CHECKED

Last drift check time
-

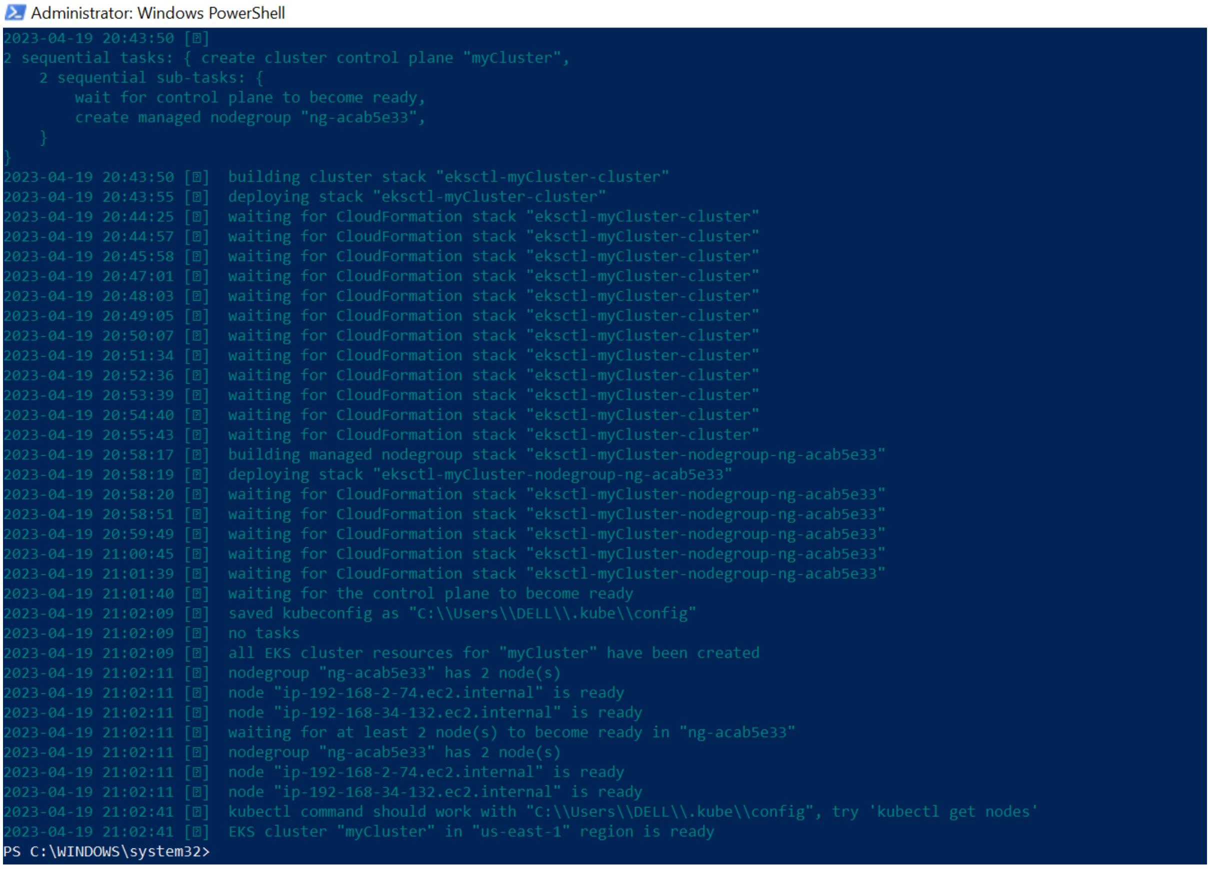
Termination protection
Deactivated

IAM role
-

*Now the cluster is active and we have successfully created a cluster using eksctl.

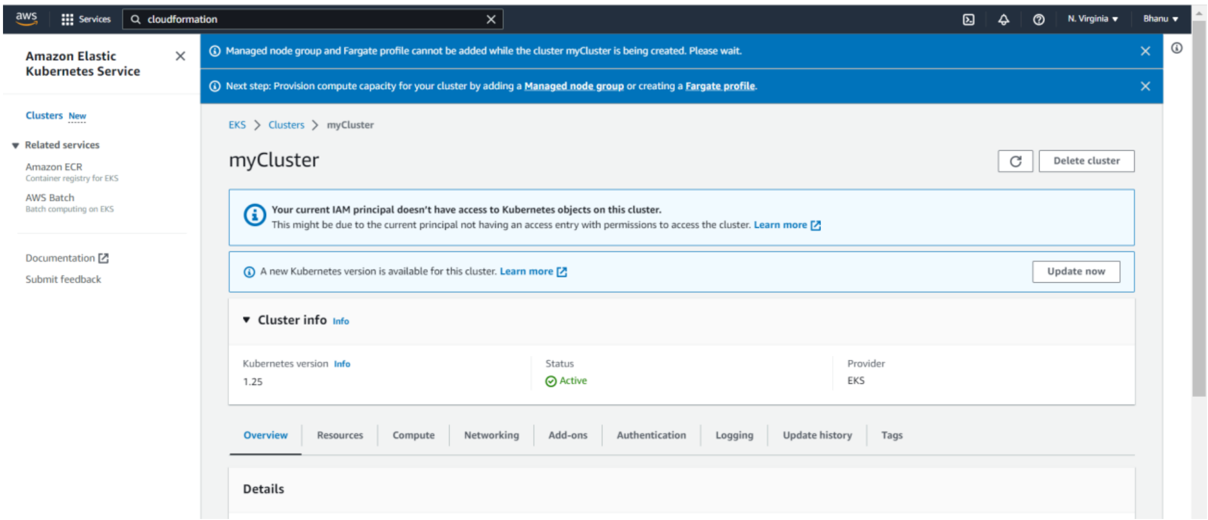


```
PS C:\WINDOWS\system32> eksctl create cluster --name myCluster --region us-east-1
2023-04-19 20:43:47 [□]  eksctl version 0.137.0
2023-04-19 20:43:47 [□]  using region us-east-1
2023-04-19 20:43:50 [□]  setting availability zones to [us-east-1a us-east-1c]
2023-04-19 20:43:50 [□]  subnets for us-east-1a - public:192.168.0.0/19 private:192.168.64.0/19
2023-04-19 20:43:50 [□]  subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2023-04-19 20:43:50 [□]  nodegroup "ng-acab5e33" will use "" [AmazonLinux2/1.25]
2023-04-19 20:43:50 [□]  using Kubernetes version 1.25
2023-04-19 20:43:50 [□]  creating EKS cluster "myCluster" in "us-east-1" region with managed nodes
2023-04-19 20:43:50 [□]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-04-19 20:43:50 [□]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [□]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [□]  CloudWatch logging will not be enabled for cluster "myCluster" in "us-east-1"
2023-04-19 20:43:50 [□]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=myCluster'
2023-04-19 20:43:50 [□]
2 sequential tasks: { create cluster control plane "myCluster",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng-acab5e33",
    }
}
2023-04-19 20:43:50 [□]  building cluster stack "eksctl-myCluster-cluster"
2023-04-19 20:43:55 [□]  deploying stack "eksctl-myCluster-cluster"
2023-04-19 20:44:25 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:44:57 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:45:58 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:47:01 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:48:03 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:49:05 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:50:07 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:51:34 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:52:36 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:53:39 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:54:40 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:55:43 [□]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:58:17 [□]  building managed nodegroup stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:19 [□]  deploying stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:20 [□]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:51 [□]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:59:49 [□]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
```



Administrator: Windows PowerShell

```
2023-04-19 20:43:50 [☒]
2 sequential tasks: { create cluster control plane "myCluster",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng-acab5e33",
    }
}
2023-04-19 20:43:50 [☒]  building cluster stack "eksctl-myCluster-cluster"
2023-04-19 20:43:55 [☒]  deploying stack "eksctl-myCluster-cluster"
2023-04-19 20:44:25 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:44:57 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:45:58 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:47:01 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:48:03 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:49:05 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:50:07 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:51:34 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:52:36 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:53:39 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:54:40 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:55:43 [☒]  waiting for CloudFormation stack "eksctl-myCluster-cluster"
2023-04-19 20:58:17 [☒]  building managed nodegroup stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:19 [☒]  deploying stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:20 [☒]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:58:51 [☒]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 20:59:49 [☒]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:00:45 [☒]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:01:39 [☒]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:01:40 [☒]  waiting for the control plane to become ready
2023-04-19 21:02:09 [☒]  saved kubeconfig as "C:\\Users\\DELL\\.kube\\config"
2023-04-19 21:02:09 [☒]  no tasks
2023-04-19 21:02:09 [☒]  all EKS cluster resources for "myCluster" have been created
2023-04-19 21:02:11 [☒]  nodegroup "ng-acab5e33" has 2 node(s)
2023-04-19 21:02:11 [☒]  node "ip-192-168-2-74.ec2.internal" is ready
2023-04-19 21:02:11 [☒]  node "ip-192-168-34-132.ec2.internal" is ready
2023-04-19 21:02:11 [☒]  waiting for at least 2 node(s) to become ready in "ng-acab5e33"
2023-04-19 21:02:11 [☒]  nodegroup "ng-acab5e33" has 2 node(s)
2023-04-19 21:02:11 [☒]  node "ip-192-168-2-74.ec2.internal" is ready
2023-04-19 21:02:11 [☒]  node "ip-192-168-34-132.ec2.internal" is ready
2023-04-19 21:02:41 [☒]  kubectl command should work with "C:\\Users\\DELL\\.kube\\config", try 'kubectl get nodes'
2023-04-19 21:02:41 [☒]  EKS cluster "myCluster" in "us-east-1" region is ready
PS C:\WINDOWS\system32>
```



ⓘ Managed node group and Fargate profile cannot be added while the cluster myCluster is being created. Please wait.

ⓘ Next step: Provision compute capacity for your cluster by adding a Managed node group or creating a Fargate profile.

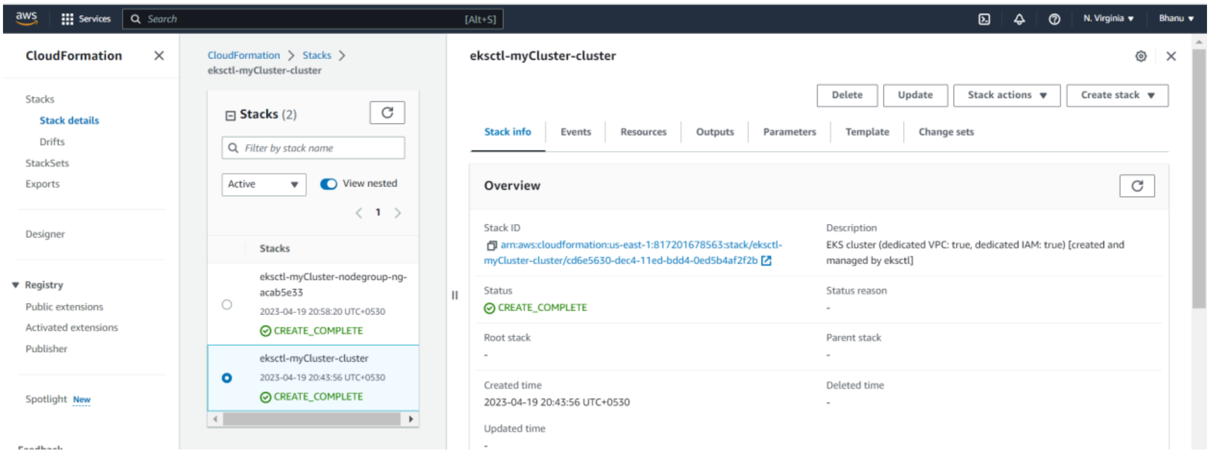
myCluster

Your current IAM principal doesn't have access to Kubernetes objects on this cluster.
This might be due to the current principal not having an access entry with permissions to access the cluster. Learn more

ⓘ A new Kubernetes version is available for this cluster. Learn more

▾ Cluster info Info

Kubernetes version Info	Status	Provider
1.25	Active	EKS



○ eksctl-myCluster-nodegroup-ng-acab5e33
2023-04-19 20:58:20 UTC+0530
CREATE_COMPLETE

● eksctl-myCluster-cluster
2023-04-19 20:43:56 UTC+0530
CREATE_COMPLETE

eksctl-myCluster-cluster

Overview

Stack ID
arn:aws:cloudformation:us-east-1:817201678563:stack/eksctl-myCluster-cluster/cd6e5630-dec4-11ed-bdd4-0ed5b4af2f2b

Description
EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

Status
CREATE_COMPLETE

Status reason
-

Root stack
-

Parent stack
-

Created time
2023-04-19 20:43:56 UTC+0530

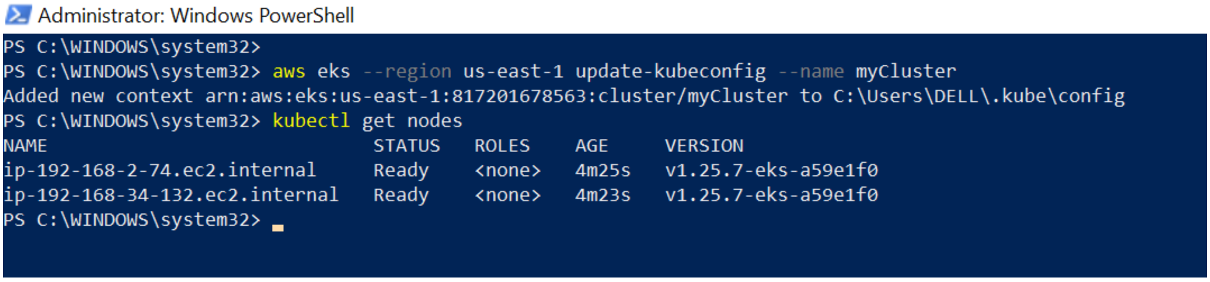
Deleted time
-

Updated time
-

*Open the Ec2 dashboard and you can see the running instances.

*Now Once the cluster is created, you can use the **kubectl** command-line tool to interact with it. Before you can do this, you need to update your **kubectl** configuration to point to the new EKS cluster. To do this, run the following command:

aws eks --region <region> update-kubeconfig --name <cluster-name>

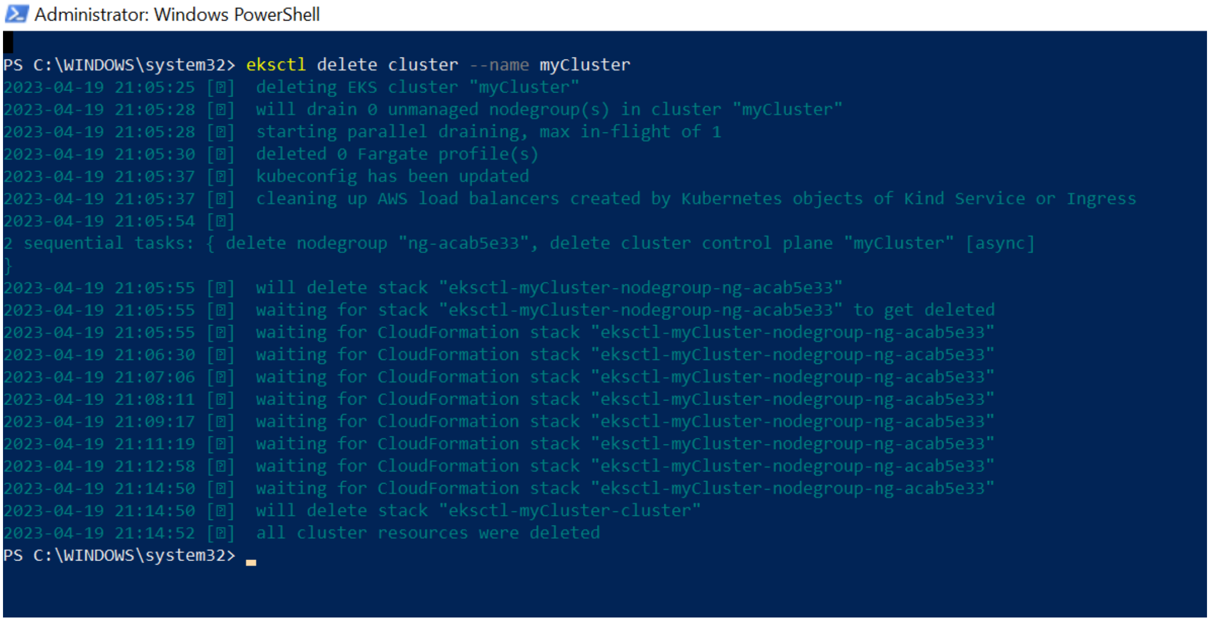


Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> aws eks --region us-east-1 update-kubeconfig --name myCluster
Added new context arn:aws:eks:us-east-1:817201678563:cluster/myCluster to C:\Users\DELL\.kube\config
PS C:\WINDOWS\system32> kubectl get nodes
NAME                             STATUS   ROLES    AGE     VERSION
ip-192-168-2-74.ec2.internal     Ready    <none>   4m25s   v1.25.7-eks-a59e1f0
ip-192-168-34-132.ec2.internal   Ready    <none>   4m23s   v1.25.7-eks-a59e1f0
PS C:\WINDOWS\system32>
```

*After you have tested your EKS cluster, you can delete it to avoid incurring any ongoing costs. To delete your EKS cluster, run the following command:

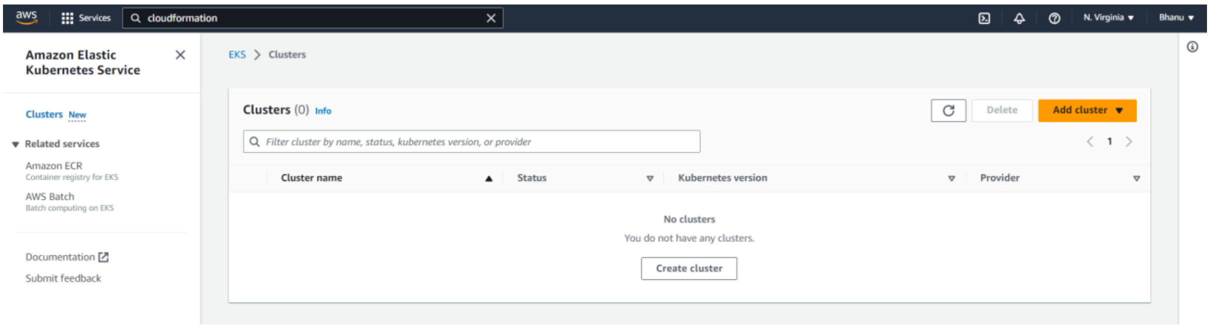
eksctl delete cluster --name <cluster-name>



Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> eksctl delete cluster --name myCluster
2023-04-19 21:05:25 [▯]  deleting EKS cluster "myCluster"
2023-04-19 21:05:28 [▯]  will drain 0 unmanaged nodegroup(s) in cluster "myCluster"
2023-04-19 21:05:28 [▯]  starting parallel draining, max in-flight of 1
2023-04-19 21:05:30 [▯]  deleted 0 Fargate profile(s)
2023-04-19 21:05:37 [▯]  kubeconfig has been updated
2023-04-19 21:05:37 [▯]  cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2023-04-19 21:05:54 [▯]
2 sequential tasks: { delete nodegroup "ng-acab5e33", delete cluster control plane "myCluster" [async]
}
2023-04-19 21:05:55 [▯]  will delete stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:05:55 [▯]  waiting for stack "eksctl-myCluster-nodegroup-ng-acab5e33" to get deleted
2023-04-19 21:05:55 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:06:30 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:07:06 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:08:11 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:09:17 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:11:19 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:12:58 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:14:50 [▯]  waiting for CloudFormation stack "eksctl-myCluster-nodegroup-ng-acab5e33"
2023-04-19 21:14:50 [▯]  will delete stack "eksctl-myCluster-cluster"
2023-04-19 21:14:52 [▯]  all cluster resources were deleted
PS C:\WINDOWS\system32> _
```

*Now go to AWS management console and go to EKS there are no clusters.



Clusters (0) Info

Filter cluster by name, status, kubernetes version, or provider

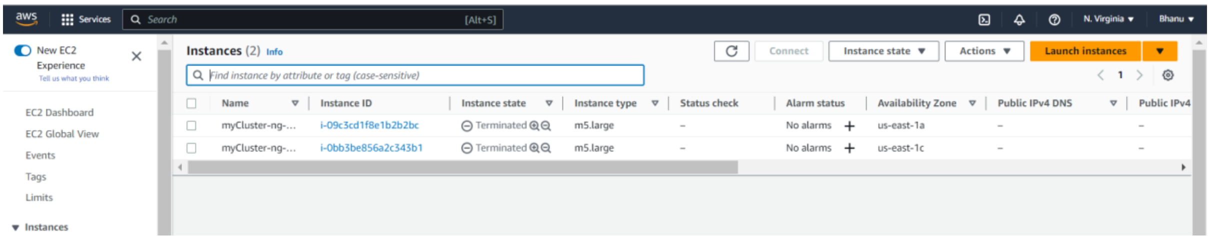
Cluster name	Status	Kubernetes version	Provider

No clusters

You do not have any clusters.

Create cluster

*Instances will terminated automatically once cluster is deleted.



Instances (2) Info

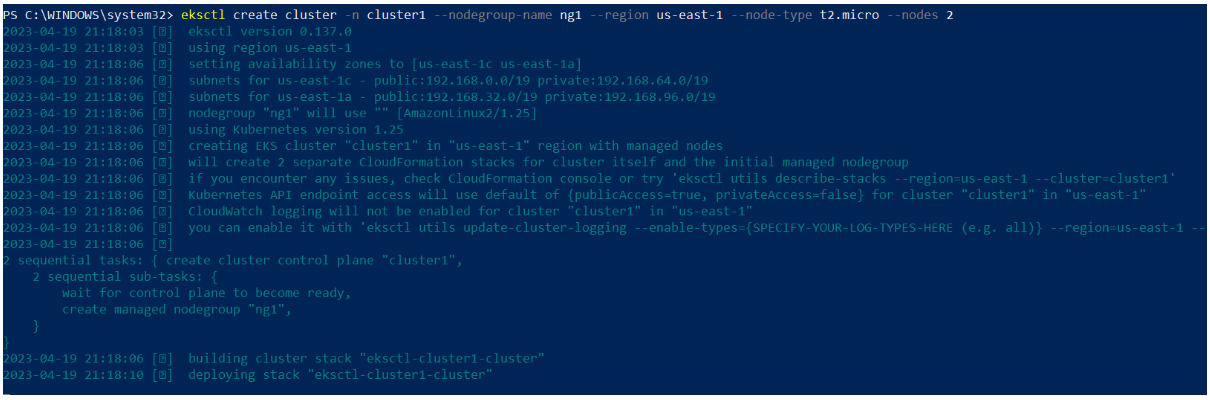
Find instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
□	myCluster-ng-...	i-09c3cd1f8e1b2b2bc	⊖ Terminated	m5.large	–	No alarms +	us-east-1a	–	–
□	myCluster-ng-...	i-0bb3be856a2c343b1	⊖ Terminated	m5.large	–	No alarms +	us-east-1c	–	–

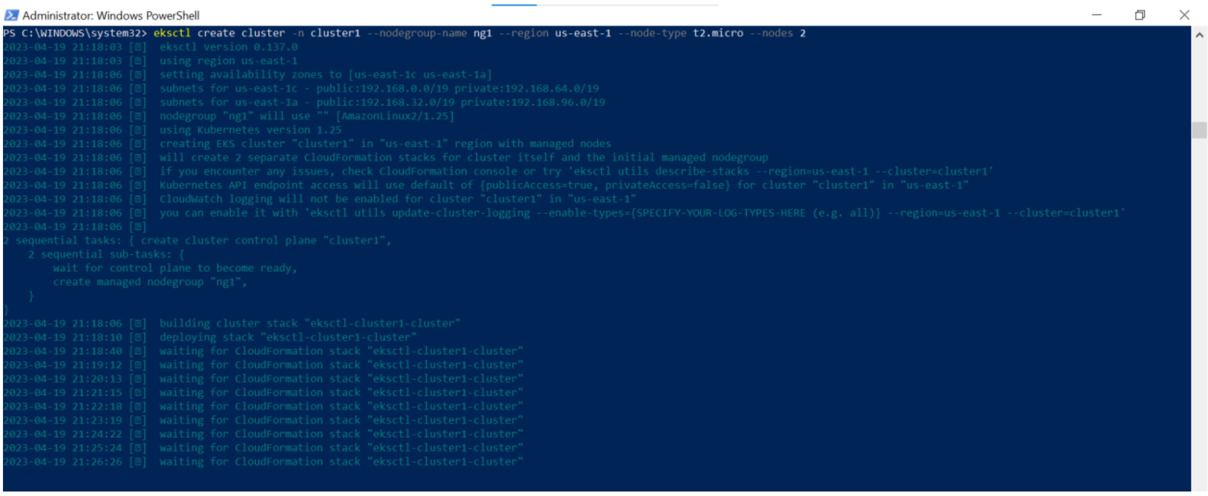
***Creating the second cluster.**

For creation of second cluster , use command mentioned below:

eksctl create cluster -n cluster1 --nodegroup-name ng1 --region us-east-1 --node-type t2.micro - -nodes 2



```
PS C:\WINDOWS\system32> eksctl create cluster -n cluster1 --nodegroup-name ng1 --region us-east-1 --node-type t2.micro --nodes 2
2023-04-19 21:18:03 [▯]  eksctl version 0.137.0
2023-04-19 21:18:03 [▯]  using region us-east-1
2023-04-19 21:18:06 [▯]  setting availability zones to [us-east-1c us-east-1a]
2023-04-19 21:18:06 [▯]  subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2023-04-19 21:18:06 [▯]  subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2023-04-19 21:18:06 [▯]  nodegroup "ng1" will use "" [AmazonLinux2/1.25]
2023-04-19 21:18:06 [▯]  using Kubernetes version 1.25
2023-04-19 21:18:06 [▯]  creating EKS cluster "cluster1" in "us-east-1" region with managed nodes
2023-04-19 21:18:06 [▯]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-04-19 21:18:06 [▯]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=cluster1'
2023-04-19 21:18:06 [▯]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "cluster1" in "us-east-1"
2023-04-19 21:18:06 [▯]  CloudWatch logging will not be enabled for cluster "cluster1" in "us-east-1"
2023-04-19 21:18:06 [▯]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --
2023-04-19 21:18:06 [▯]
2 sequential tasks: { create cluster control plane "cluster1",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng1",
    }
}
2023-04-19 21:18:06 [▯]  building cluster stack "eksctl-cluster1-cluster"
2023-04-19 21:18:10 [▯]  deploying stack "eksctl-cluster1-cluster"
```

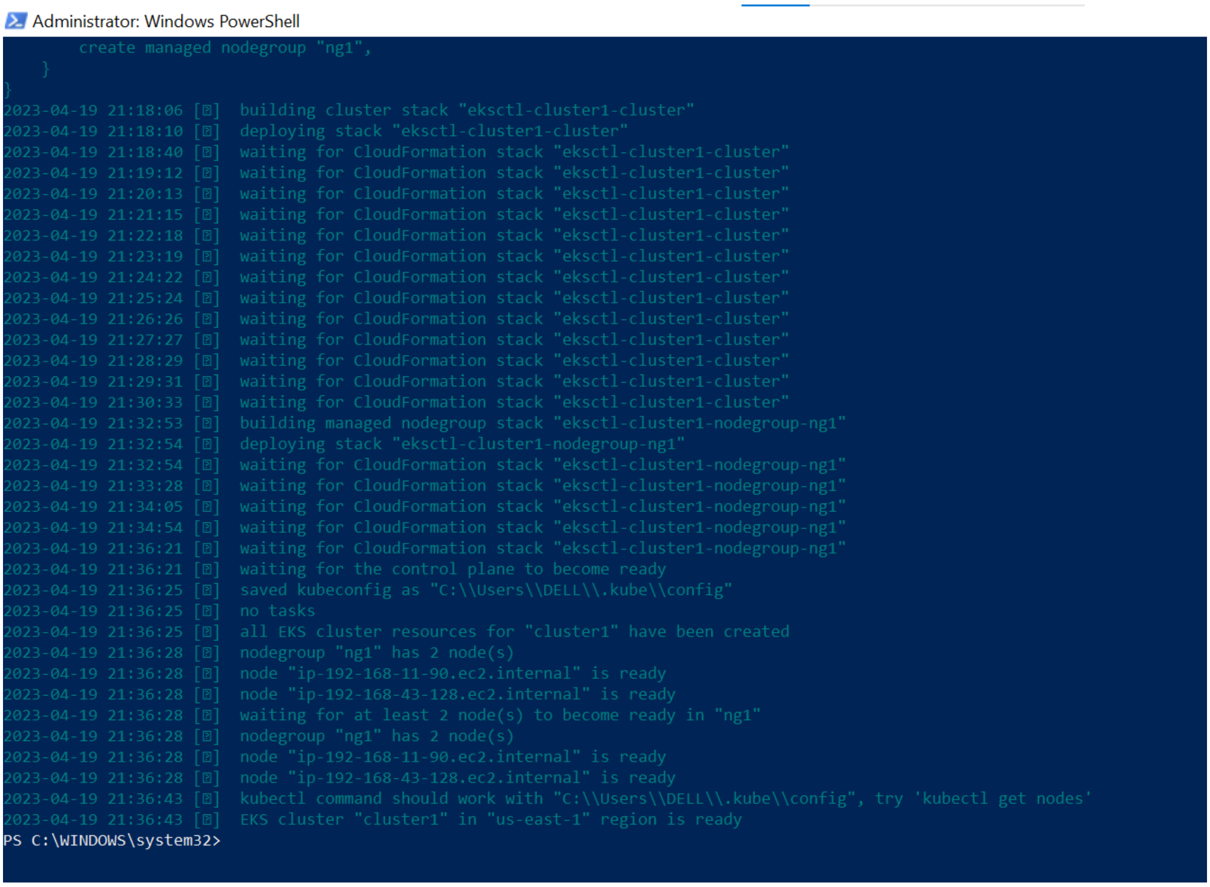


```
PS C:\WINDOWS\system32> eksctl create cluster -n cluster1 --nodegroup-name ng1 --region us-east-1 --node-type t2.micro --nodes 2
2023-04-19 21:18:03 [▯]  eksctl version 0.137.0
2023-04-19 21:18:03 [▯]  using region us-east-1
2023-04-19 21:18:06 [▯]  setting availability zones to [us-east-1c us-east-1a]
2023-04-19 21:18:06 [▯]  subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2023-04-19 21:18:06 [▯]  subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2023-04-19 21:18:06 [▯]  nodegroup "ng1" will use "" [AmazonLinux2/1.25]
2023-04-19 21:18:06 [▯]  using Kubernetes version 1.25
2023-04-19 21:18:06 [▯]  creating EKS cluster "cluster1" in "us-east-1" region with managed nodes
2023-04-19 21:18:06 [▯]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-04-19 21:18:06 [▯]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=cluster1'
2023-04-19 21:18:06 [▯]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "cluster1" in "us-east-1"
2023-04-19 21:18:06 [▯]  CloudWatch logging will not be enabled for cluster "cluster1" in "us-east-1"
```

```
2023-04-19 21:18:06 [▯]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=cluster1'
```

```
2023-04-19 21:18:06 [▯]
2 sequential tasks: { create cluster control plane "cluster1",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "ng1",
    }
}
2023-04-19 21:18:06 [▯]  building cluster stack "eksctl-cluster1-cluster"
2023-04-19 21:18:10 [▯]  deploying stack "eksctl-cluster1-cluster"
```

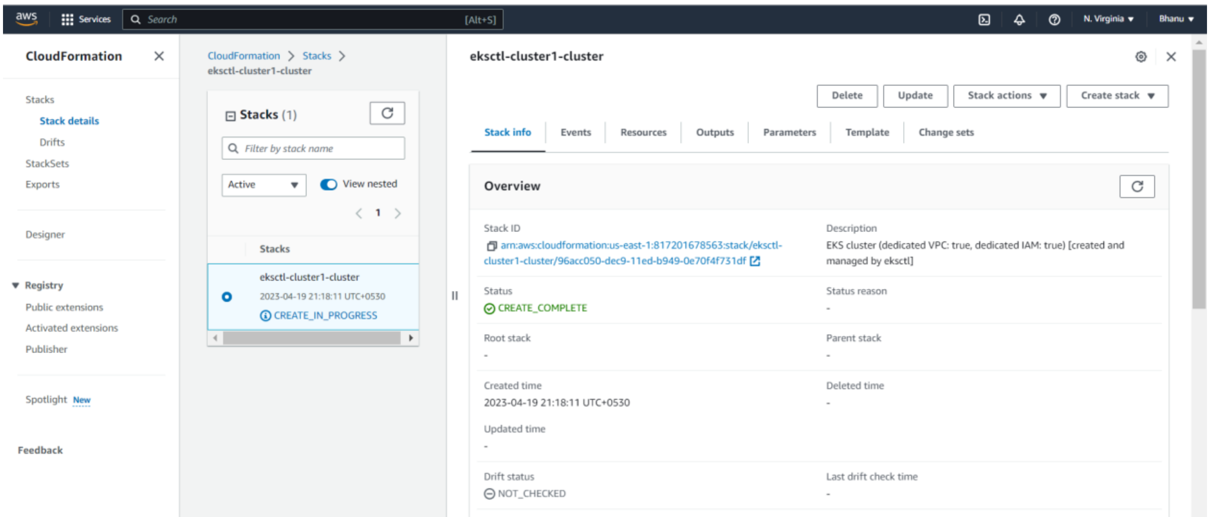
```
2023-04-19 21:18:40 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:19:12 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:20:13 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:21:15 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:22:18 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:23:19 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:24:22 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:25:24 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:26:26 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
```



```
        create managed nodegroup "ng1",
    }
}
2023-04-19 21:18:06 [▯]  building cluster stack "eksctl-cluster1-cluster"
2023-04-19 21:18:10 [▯]  deploying stack "eksctl-cluster1-cluster"
2023-04-19 21:18:40 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:19:12 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:20:13 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:21:15 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:22:18 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:23:19 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:24:22 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:25:24 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:26:26 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
```

```
2023-04-19 21:27:27 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:28:29 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:29:31 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:30:33 [▯]  waiting for CloudFormation stack "eksctl-cluster1-cluster"
2023-04-19 21:32:53 [▯]  building managed nodegroup stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:32:54 [▯]  deploying stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:32:54 [▯]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:33:28 [▯]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:34:05 [▯]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:34:54 [▯]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:36:21 [▯]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:36:21 [▯]  waiting for the control plane to become ready
2023-04-19 21:36:25 [▯]  saved kubeconfig as "C:\\Users\\DELL\\.kube\\config"
2023-04-19 21:36:25 [▯]  no tasks
2023-04-19 21:36:25 [▯]  all EKS cluster resources for "cluster1" have been created
2023-04-19 21:36:28 [▯]  nodegroup "ng1" has 2 node(s)
2023-04-19 21:36:28 [▯]  node "ip-192-168-11-90.ec2.internal" is ready
2023-04-19 21:36:28 [▯]  node "ip-192-168-43-128.ec2.internal" is ready
2023-04-19 21:36:28 [▯]  waiting for at least 2 node(s) to become ready in "ng1"
2023-04-19 21:36:28 [▯]  nodegroup "ng1" has 2 node(s)
2023-04-19 21:36:28 [▯]  node "ip-192-168-11-90.ec2.internal" is ready
2023-04-19 21:36:28 [▯]  node "ip-192-168-43-128.ec2.internal" is ready
2023-04-19 21:36:43 [▯]  kubectl command should work with "C:\\Users\\DELL\\.kube\\config", try 'kubectl get nodes'
2023-04-19 21:36:43 [▯]  EKS cluster "cluster1" in "us-east-1" region is ready
PS C:\WINDOWS\system32>
```

*Once the cluster is created , go to EKS and check whether cluster is there or not.



eksctl-cluster1-cluster

Stacks (1)

Stacks
eksctl-cluster1-cluster 2023-04-19 21:18:11 UTC+0530 CREATE_IN_PROGRESS

eksctl-cluster1-cluster

Overview

Stack ID
arn:aws:cloudformation:us-east-1:817201678563:stack/eksctl-cluster1-cluster/96acc050-dec9-11ed-b949-0e70f4f731df

Description
EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

Status
CREATE_COMPLETE

Status reason
-

Root stack
-

Parent stack
-

Created time
2023-04-19 21:18:11 UTC+0530

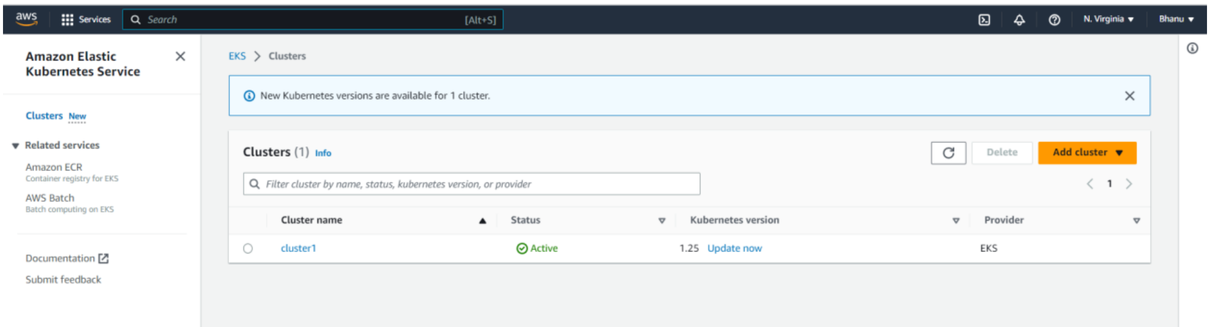
Deleted time
-

Updated time
-

Drift status
NOT_CHECKED

Last drift check time
-

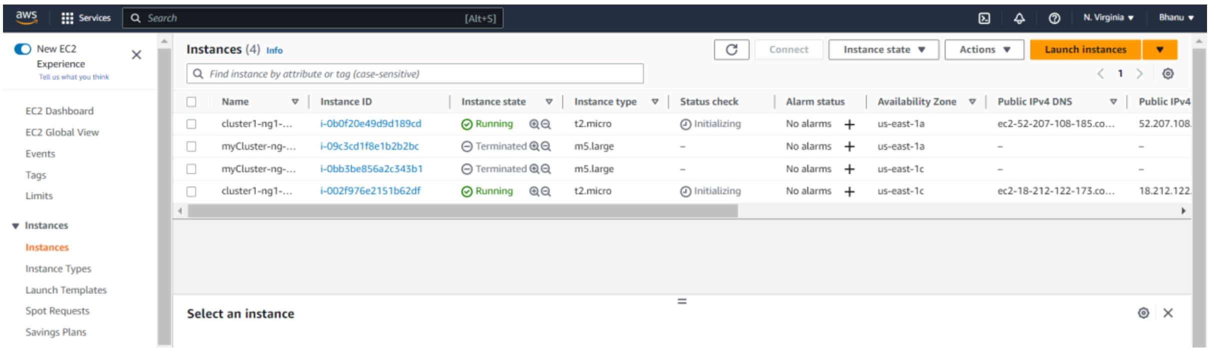
*The cluster1 is in active state ,that means the cluster is created successfully using eksctl.



New Kubernetes versions are available for 1 cluster.

Clusters (1) Info

Cluster name	Status	Kubernetes version	Provider
cluster1	Active	1.25 Update now	EKS



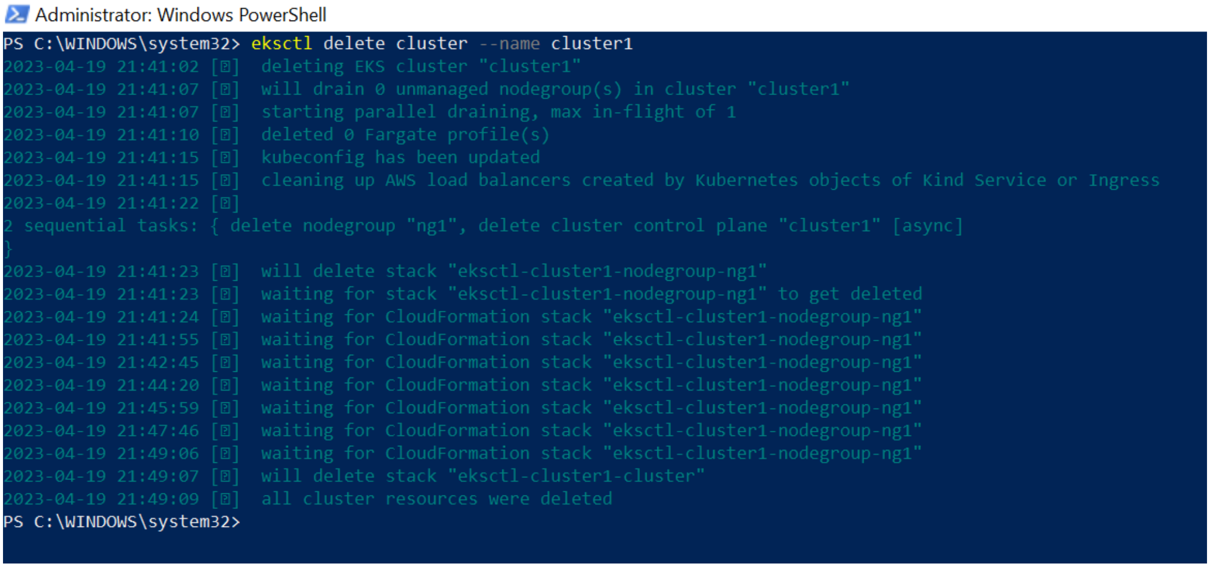
Instances (4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
cluster1-ng1-...	i-0b0f20e49d9d189cd	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-52-207-108-185.co...	52.207.108.
myCluster-ng-...	i-09c3cd1f8e1b2b2bc	Terminated	m5.large	–	No alarms	us-east-1a	–	–
myCluster-ng-...	i-0bb3be856a2c343b1	Terminated	m5.large	–	No alarms	us-east-1c	–	–
cluster1-ng1-...	i-002f976e2151b62df	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-18-212-122-173.co...	18.212.122.

Select an instance

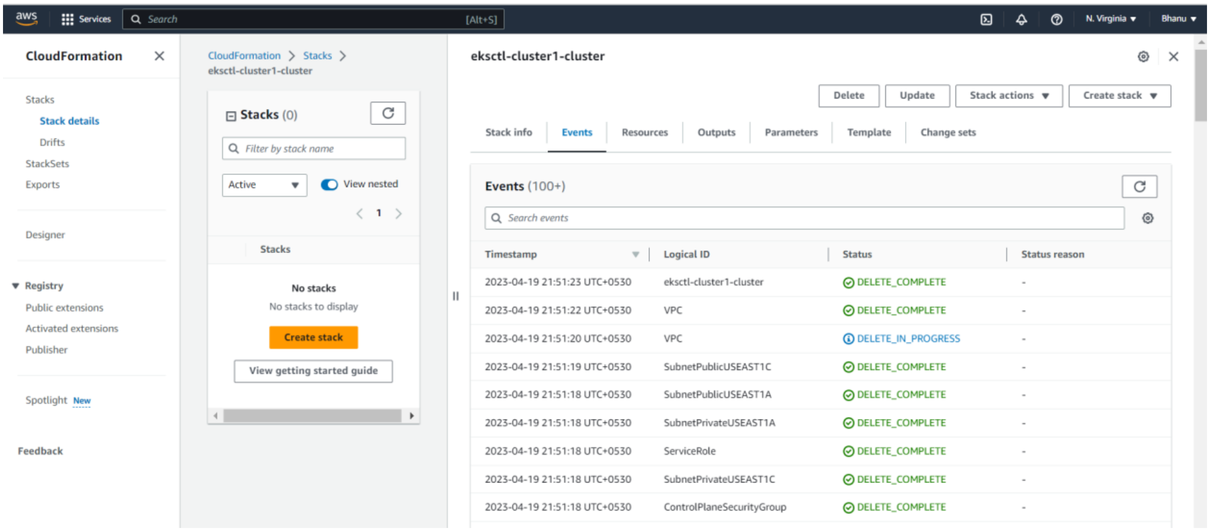
*delete the cluster using below command.

eksctl delete cluster --name <cluster-name>

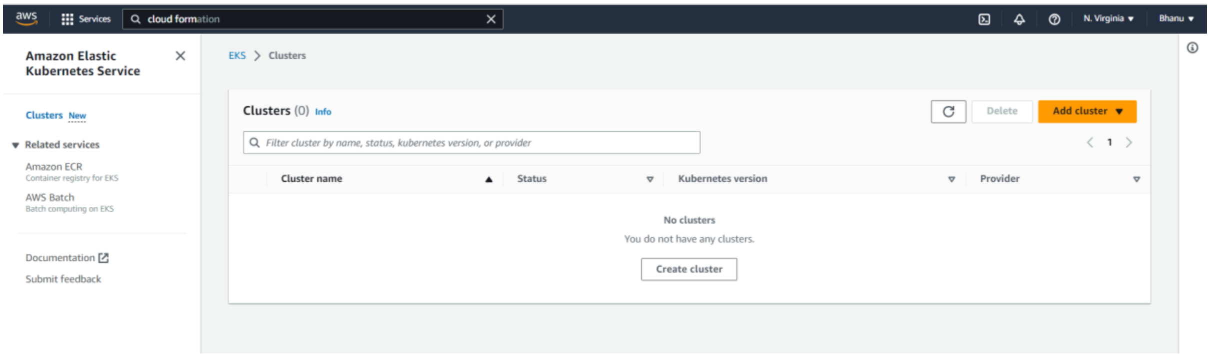


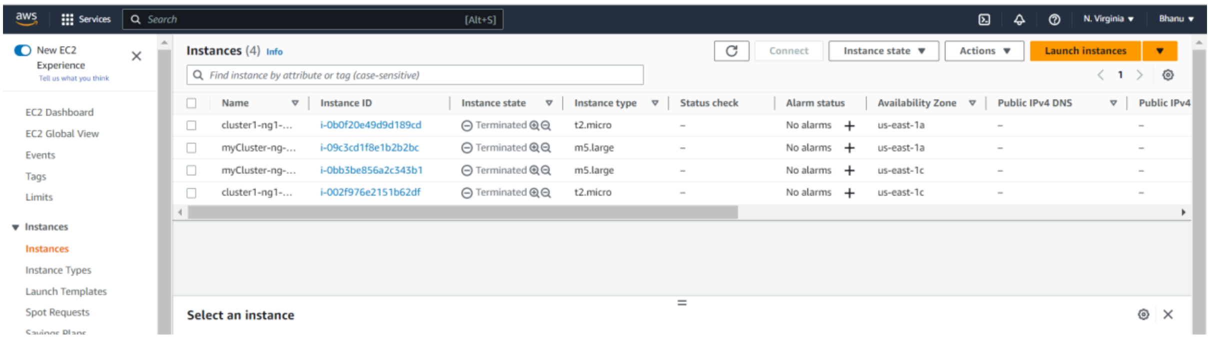
Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> eksctl delete cluster --name cluster1
2023-04-19 21:41:02 [⍰]  deleting EKS cluster "cluster1"
2023-04-19 21:41:07 [⍰]  will drain 0 unmanaged nodegroup(s) in cluster "cluster1"
2023-04-19 21:41:07 [⍰]  starting parallel draining, max in-flight of 1
2023-04-19 21:41:10 [⍰]  deleted 0 Fargate profile(s)
2023-04-19 21:41:15 [⍰]  kubeconfig has been updated
2023-04-19 21:41:15 [⍰]  cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2023-04-19 21:41:22 [⍰]
2 sequential tasks: { delete nodegroup "ng1", delete cluster control plane "cluster1" [async]
}
2023-04-19 21:41:23 [⍰]  will delete stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:41:23 [⍰]  waiting for stack "eksctl-cluster1-nodegroup-ng1" to get deleted
2023-04-19 21:41:24 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:41:55 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:42:45 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:44:20 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:45:59 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:47:46 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:49:06 [⍰]  waiting for CloudFormation stack "eksctl-cluster1-nodegroup-ng1"
2023-04-19 21:49:07 [⍰]  will delete stack "eksctl-cluster1-cluster"
2023-04-19 21:49:09 [⍰]  all cluster resources were deleted
PS C:\WINDOWS\system32>
```



Timestamp	Logical ID	Status	Status reason
2023-04-19 21:51:23 UTC+0530	eksctl-cluster1-cluster	DELETE_COMPLETE	-
2023-04-19 21:51:22 UTC+0530	VPC	DELETE_COMPLETE	-
2023-04-19 21:51:20 UTC+0530	VPC	DELETE_IN_PROGRESS	-
2023-04-19 21:51:19 UTC+0530	SubnetPublicUSEAST1C	DELETE_COMPLETE	-
2023-04-19 21:51:18 UTC+0530	SubnetPublicUSEAST1A	DELETE_COMPLETE	-
2023-04-19 21:51:18 UTC+0530	SubnetPrivateUSEAST1A	DELETE_COMPLETE	-
2023-04-19 21:51:18 UTC+0530	ServiceRole	DELETE_COMPLETE	-
2023-04-19 21:51:18 UTC+0530	SubnetPrivateUSEAST1C	DELETE_COMPLETE	-
2023-04-19 21:51:18 UTC+0530	ControlPlaneSecurityGroup	DELETE_COMPLETE	-





□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
□	cluster1-ng1-...	i-0b0f20e49d9d189cd	⊖ Terminated	t2.micro	–	No alarms +	us-east-1a	–	–
□	myCluster-ng-...	i-09c3cd1f8e1b2b2bc	⊖ Terminated	m5.large	–	No alarms +	us-east-1a	–	–
□	myCluster-ng-...	i-0bb3be856a2c343b1	⊖ Terminated	m5.large	–	No alarms +	us-east-1c	–	–
□	cluster1-ng1-...	i-002f976e2151b62df	⊖ Terminated	t2.micro	–	No alarms +	us-east-1c	–	–

Select an instance