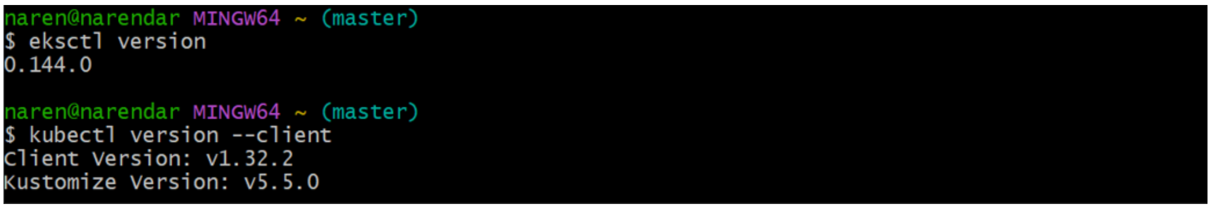
EKS

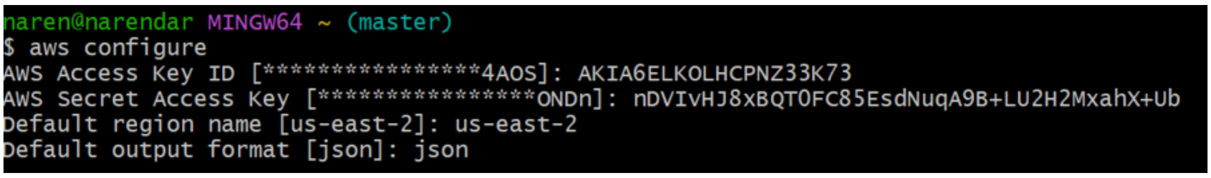
1) Setup eks cluster using eksctl.

--install eksctl and check version



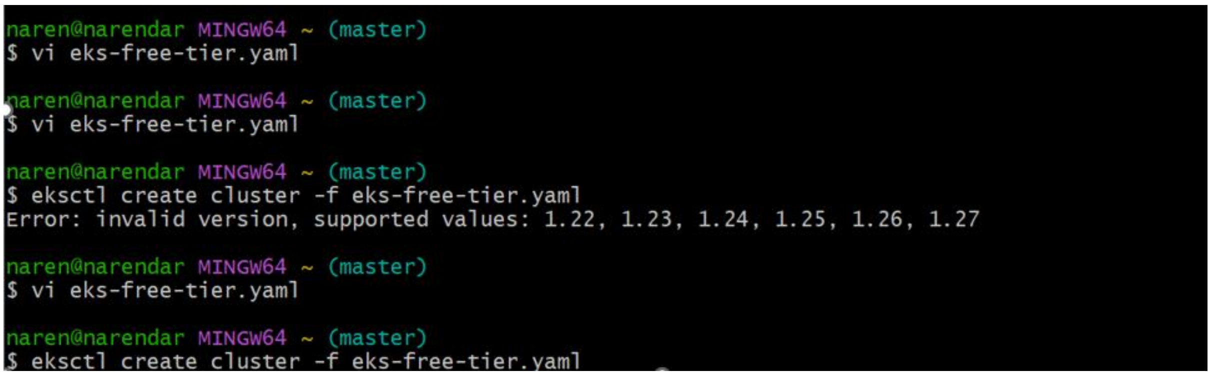
```
naren@narendar MINGW64 ~ (master)
$ eksctl version
0.144.0

naren@narendar MINGW64 ~ (master)
$ kubectl version --client
Client Version: v1.32.2
Kustomize Version: v5.5.0
```

--now set-up AWS configure



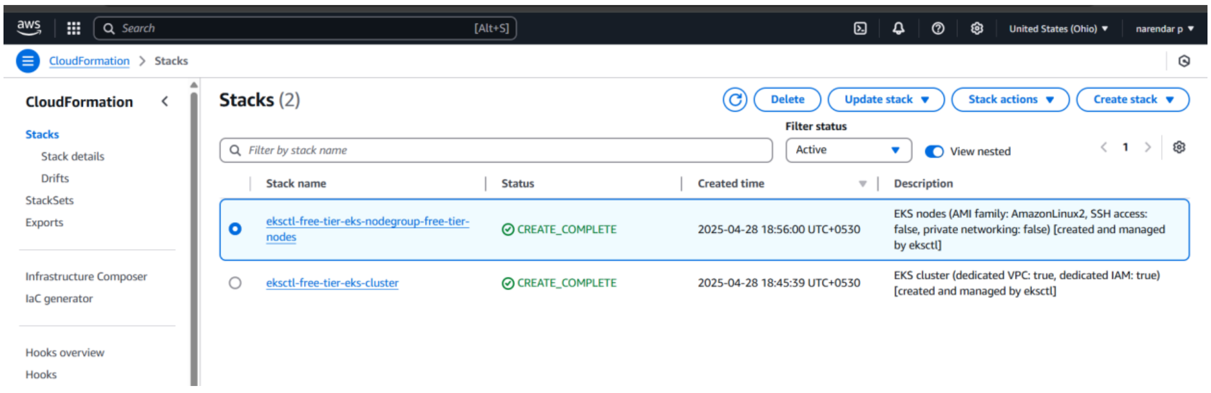
```
naren@narendar MINGW64 ~ (master)
$ aws configure
AWS Access Key ID [****************4AOS]: AKIA6ELKOLHCPNZ33K73
AWS Secret Access Key [****************ONDn]: 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
Default region name [us-east-2]: us-east-2
Default output format [json]: json
```

--create yaml file and run it then cluster got created with two worker nodes



```
naren@narendar MINGW64 ~ (master)
$ vi eks-free-tier.yaml

naren@narendar MINGW64 ~ (master)
$ vi eks-free-tier.yaml

naren@narendar MINGW64 ~ (master)
$ eksctl create cluster -f eks-free-tier.yaml
Error: invalid version, supported values: 1.22, 1.23, 1.24, 1.25, 1.26, 1.27

naren@narendar MINGW64 ~ (master)
$ vi eks-free-tier.yaml

naren@narendar MINGW64 ~ (master)
$ eksctl create cluster -f eks-free-tier.yaml
```

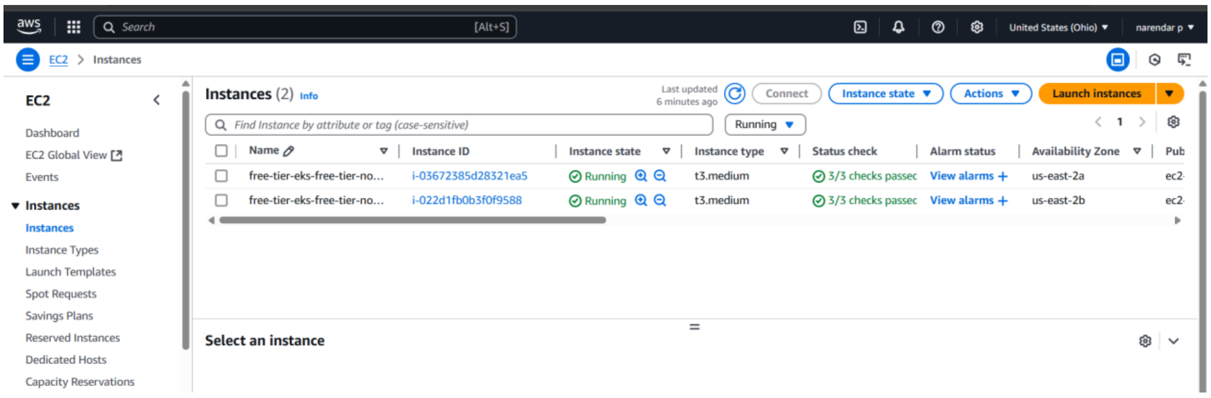
-check cloudformation



Stacks (2)

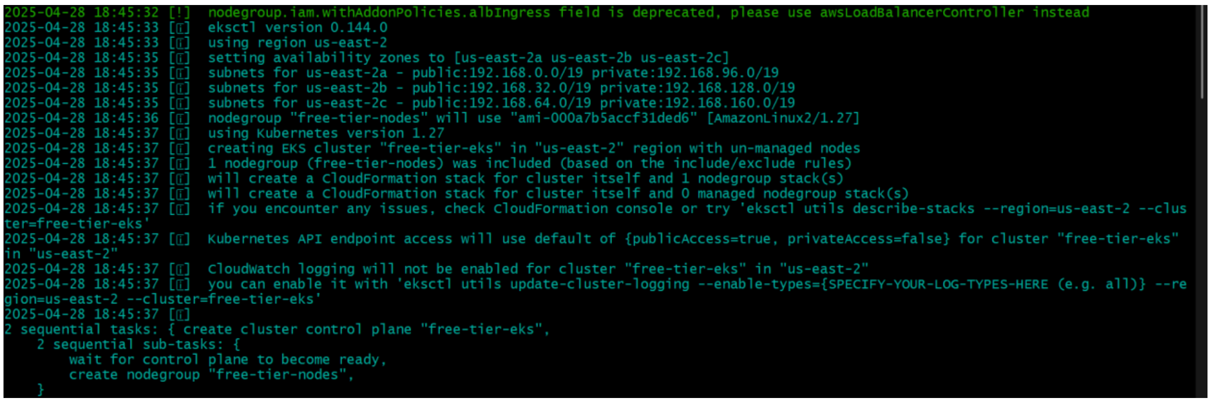
Filter status: Active

Stack name	Status	Created time	Description
eksctl-free-tier-eks-nodegroup-free-tier-nodes	CREATE_COMPLETE	2025-04-28 18:56:00 UTC+0530	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]
eksctl-free-tier-eks-cluster	CREATE_COMPLETE	2025-04-28 18:45:39 UTC+0530	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

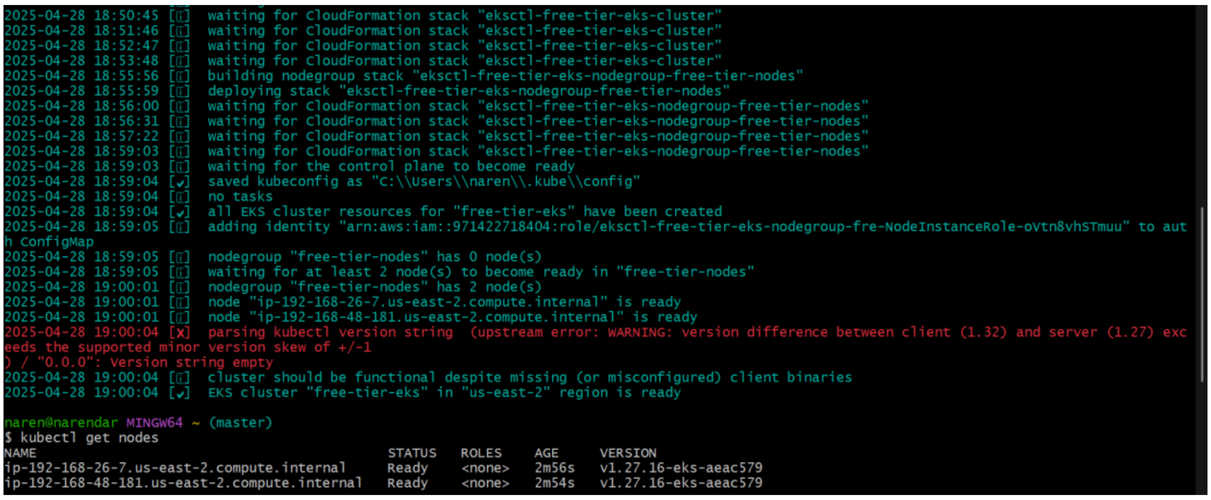


Instances (2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
free-tier-eks-free-tier-no...	i-03672385d28321ea5	Running	t3.medium	3/3 checks passec	View alarms +	us-east-2a	ec2-
free-tier-eks-free-tier-no...	i-022d1fb0b3f0f9588	Running	t3.medium	3/3 checks passec	View alarms +	us-east-2b	ec2-

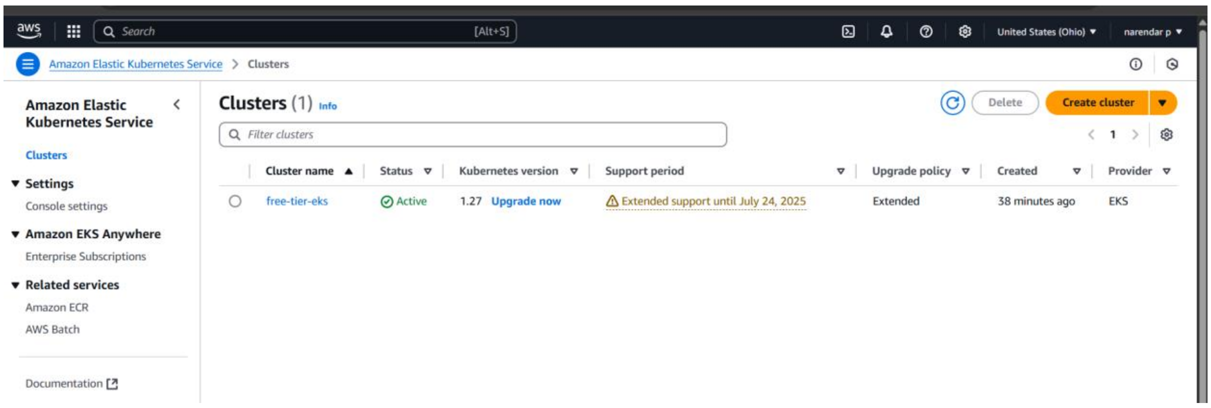


```
2025-04-28 18:45:32 [!]  nodegroup.iam.withAddonPolicies.albIngress field is deprecated, please use awsLoadBalancerController instead
2025-04-28 18:45:33 [ℹ]  eksctl version 0.144.0
2025-04-28 18:45:33 [ℹ]  using region us-east-2
2025-04-28 18:45:35 [ℹ]  setting availability zones to [us-east-2a us-east-2b us-east-2c]
2025-04-28 18:45:35 [ℹ]  subnets for us-east-2a - public:192.168.0.0/19 private:192.168.96.0/19
2025-04-28 18:45:35 [ℹ]  subnets for us-east-2b - public:192.168.32.0/19 private:192.168.128.0/19
2025-04-28 18:45:35 [ℹ]  subnets for us-east-2c - public:192.168.64.0/19 private:192.168.160.0/19
2025-04-28 18:45:36 [ℹ]  nodegroup "free-tier-nodes" will use "ami-000a7b5accf31ded6" [AmazonLinux2/1.27]
2025-04-28 18:45:37 [ℹ]  using Kubernetes version 1.27
2025-04-28 18:45:37 [ℹ]  creating EKS cluster "free-tier-eks" in "us-east-2" region with un-managed nodes
2025-04-28 18:45:37 [ℹ]  1 nodegroup (free-tier-nodes) was included (based on the include/exclude rules)
2025-04-28 18:45:37 [ℹ]  will create a CloudFormation stack for cluster itself and 1 nodegroup stack(s)
2025-04-28 18:45:37 [ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
2025-04-28 18:45:37 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --clus
ter=free-tier-eks'
2025-04-28 18:45:37 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "free-tier-eks"
in "us-east-2"
2025-04-28 18:45:37 [ℹ]  CloudWatch logging will not be enabled for cluster "free-tier-eks" in "us-east-2"
2025-04-28 18:45:37 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --re
gion=us-east-2 --cluster=free-tier-eks'
2025-04-28 18:45:37 [ℹ]
2 sequential tasks: { create cluster control plane "free-tier-eks",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create nodegroup "free-tier-nodes",
    }
```



```
2025-04-28 18:50:45 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-cluster"
2025-04-28 18:51:46 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-cluster"
2025-04-28 18:52:47 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-cluster"
2025-04-28 18:53:48 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-cluster"
2025-04-28 18:55:56 [ℹ]  building nodegroup stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:55:59 [ℹ]  deploying stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:56:00 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:56:31 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:57:22 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:59:03 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-nodegroup-free-tier-nodes"
2025-04-28 18:59:03 [ℹ]  waiting for the control plane to become ready
2025-04-28 18:59:04 [✔]  saved kubeconfig as "C:\\Users\\naren\\.kube\\config"
2025-04-28 18:59:04 [ℹ]  no tasks
2025-04-28 18:59:04 [✔]  all EKS cluster resources for "free-tier-eks" have been created
2025-04-28 18:59:05 [ℹ]  adding identity "arn:aws:iam::971422718404:role/eksctl-free-tier-eks-nodegroup-fre-NodeInstanceRole-oVtn8vhSTmuu" to aut
h ConfigMap
2025-04-28 18:59:05 [ℹ]  nodegroup "free-tier-nodes" has 0 node(s)
2025-04-28 18:59:05 [ℹ]  waiting for at least 2 node(s) to become ready in "free-tier-nodes"
2025-04-28 19:00:01 [ℹ]  nodegroup "free-tier-nodes" has 2 node(s)
2025-04-28 19:00:01 [ℹ]  node "ip-192-168-26-7.us-east-2.compute.internal" is ready
2025-04-28 19:00:01 [ℹ]  node "ip-192-168-48-181.us-east-2.compute.internal" is ready
2025-04-28 19:00:04 [x]  parsing kubectl version string  (upstream error: WARNING: version difference between client (1.32) and server (1.27) exc
eeds the supported minor version skew of +/-1
) / "0.0.0": Version string empty
2025-04-28 19:00:04 [ℹ]  cluster should be functional despite missing (or misconfigured) client binaries
2025-04-28 19:00:04 [✔]  EKS cluster "free-tier-eks" in "us-east-2" region is ready

naren@narendar MINGW64 ~ (master)
$ kubectl get nodes
NAME                                           STATUS   ROLES    AGE     VERSION
ip-192-168-26-7.us-east-2.compute.internal     Ready    <none>   2m56s   v1.27.16-eks-aeac579
ip-192-168-48-181.us-east-2.compute.internal   Ready    <none>   2m54s   v1.27.16-eks-aeac579
```

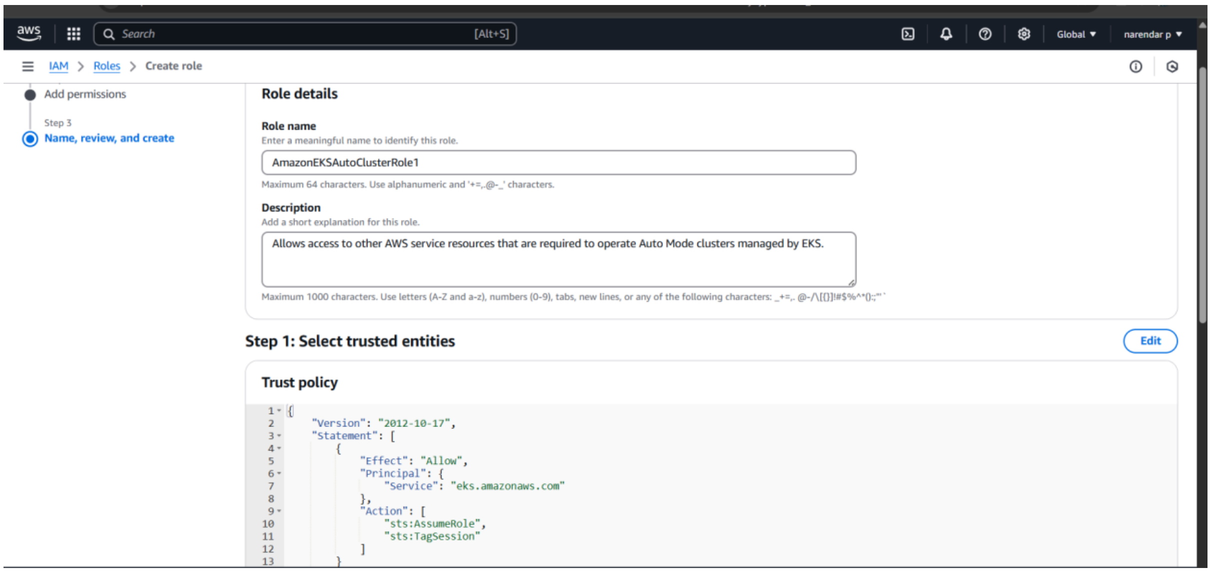
--cluster got created



Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
free-tier-eks	Active	1.27 Upgrade now	Extended support until July 24, 2025	Extended	38 minutes ago	EKS

2) setup eks cluster using console.

--add iam roles while creating cluster



Role details

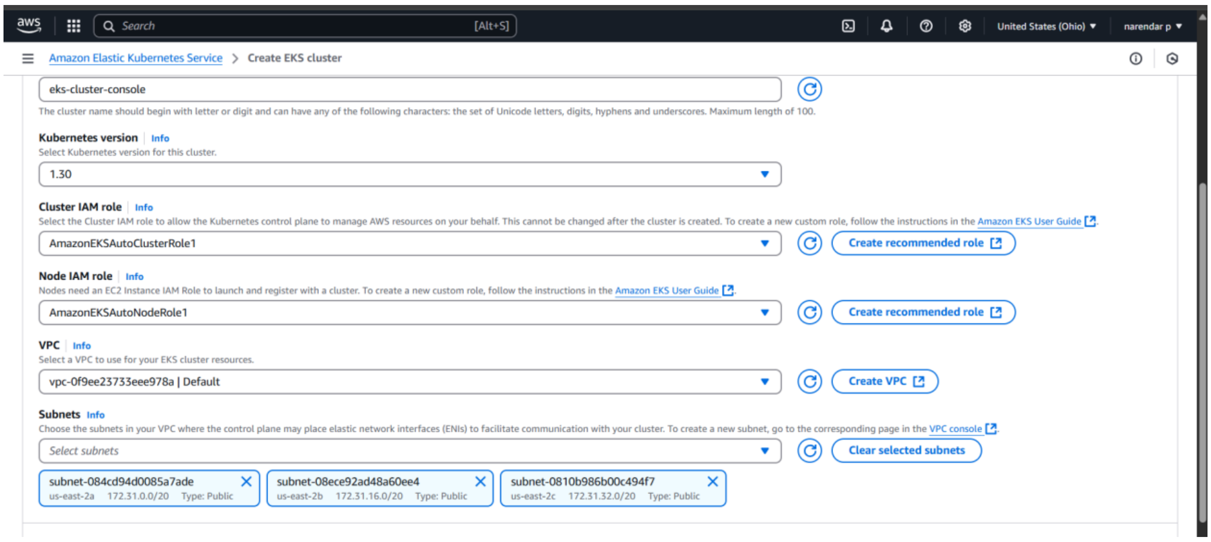
Role name
Enter a meaningful name to identify this role.
AmazonEKSAutoClusterRole1
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows access to other AWS service resources that are required to operate Auto Mode clusters managed by EKS.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*():;"' `

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
```



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

Kubernetes version Info
Select Kubernetes version for this cluster.
1.30

Cluster IAM role Info
Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the Amazon EKS User Guide.
AmazonEKSAutoClusterRole1

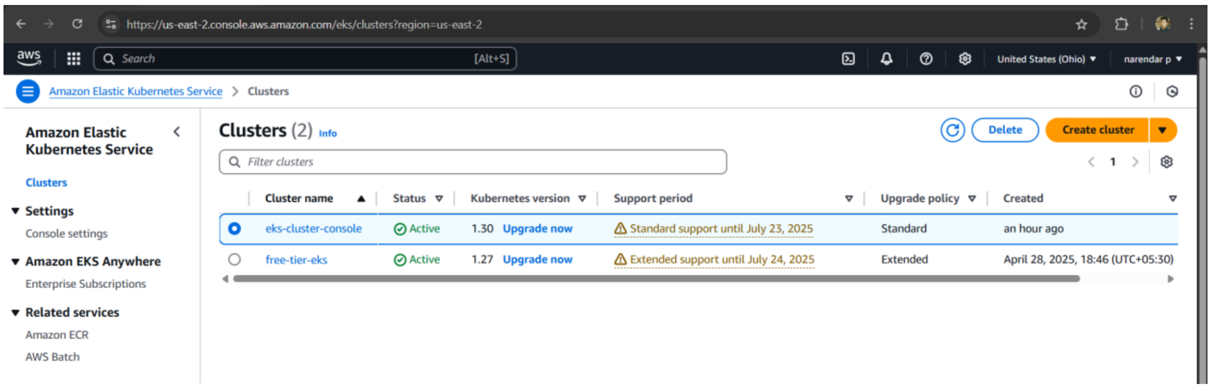
Node IAM role Info
Nodes need an EC2 Instance IAM Role to launch and register with a cluster. To create a new custom role, follow the instructions in the Amazon EKS User Guide.
AmazonEKSAutoNodeRole1

VPC Info
Select a VPC to use for your EKS cluster resources.
vpc-0f9ee23733eee978a | Default

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

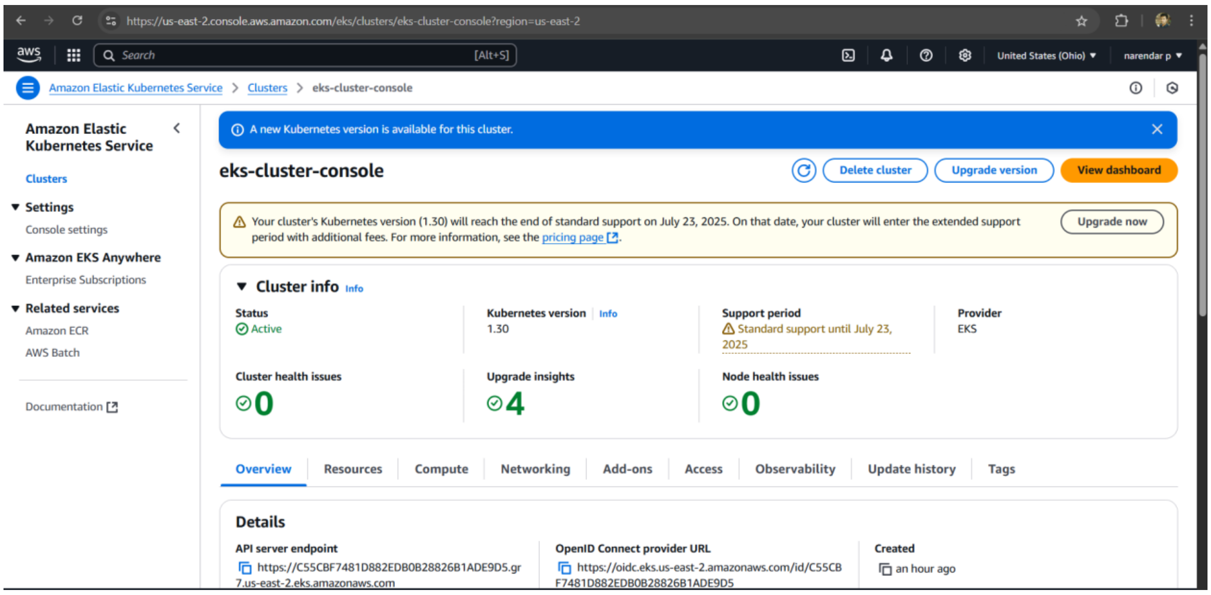
subnet-084cd94d0085a7ade us-east-2a 172.31.0.0/20 Type: Public	subnet-08ece92ad48a60ee4 us-east-2b 172.31.16.0/20 Type: Public	subnet-0810b986b00c494f7 us-east-2c 172.31.32.0/20 Type: Public

--created eks cluster using console



Clusters (2) Info

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created
eks-cluster-console	Active	1.30 Upgrade now	Standard support until July 23, 2025	Standard	an hour ago
free-tier-eks	Active	1.27 Upgrade now	Extended support until July 24, 2025	Extended	April 28, 2025, 18:46 (UTC+05:30)



https://us-east-2.console.aws.amazon.com/eks/clusters/eks-cluster-console?region=us-east-2

Amazon Elastic Kubernetes Service > Clusters > eks-cluster-console

A new Kubernetes version is available for this cluster.

eks-cluster-console

Your cluster's Kubernetes version (1.30) will reach the end of standard support on July 23, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page.

Cluster info

Status	Kubernetes version	Support period	Provider
Active	1.30	Standard support until July 23, 2025	EKS

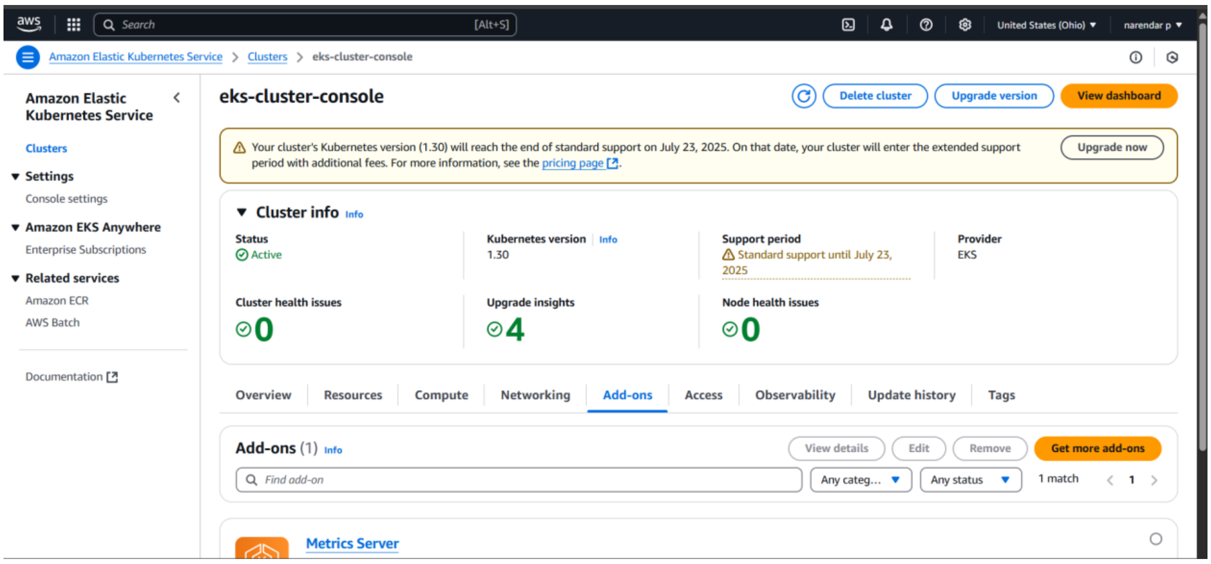
Cluster health issues	Upgrade insights	Node health issues
0	4	0

Overview | Resources | Compute | Networking | Add-ons | Access | Observability | Update history | Tags

Details

API server endpoint	OpenID Connect provider URL	Created
https://C55CBF7481D882EDB0B28826B1ADE9D5.gr7.us-east-2.eks.amazonaws.com	https://oidc.eks.us-east-2.amazonaws.com/id/C55CBF7481D882EDB0B28826B1ADE9D5	an hour ago

--check add-ons



Amazon Elastic Kubernetes Service > Clusters > eks-cluster-console

eks-cluster-console

Your cluster's Kubernetes version (1.30) will reach the end of standard support on July 23, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page.

Cluster info

Status	Kubernetes version	Support period	Provider
Active	1.30	Standard support until July 23, 2025	EKS

Cluster health issues	Upgrade insights	Node health issues
0	4	0

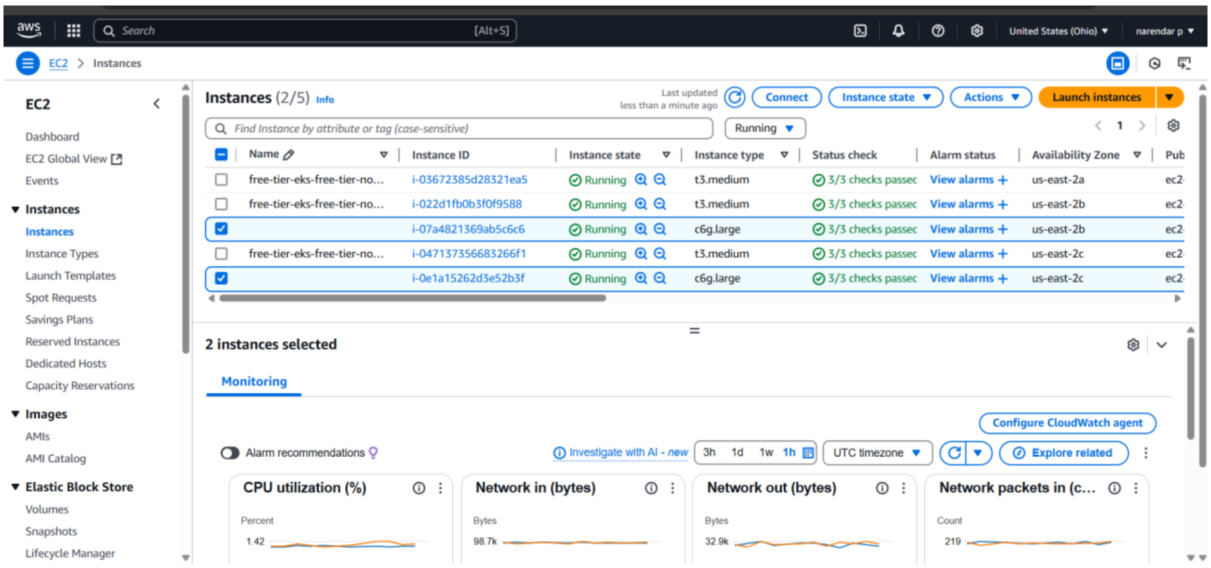
Overview | Resources | Compute | Networking | Add-ons | Access | Observability | Update history | Tags

Add-ons (1)

Find add-on | Any categ... | Any status | 1 match

Metrics Server

--two new ec2s are created along with cluster



EC2 > Instances

Instances (2/5)

Last updated less than a minute ago

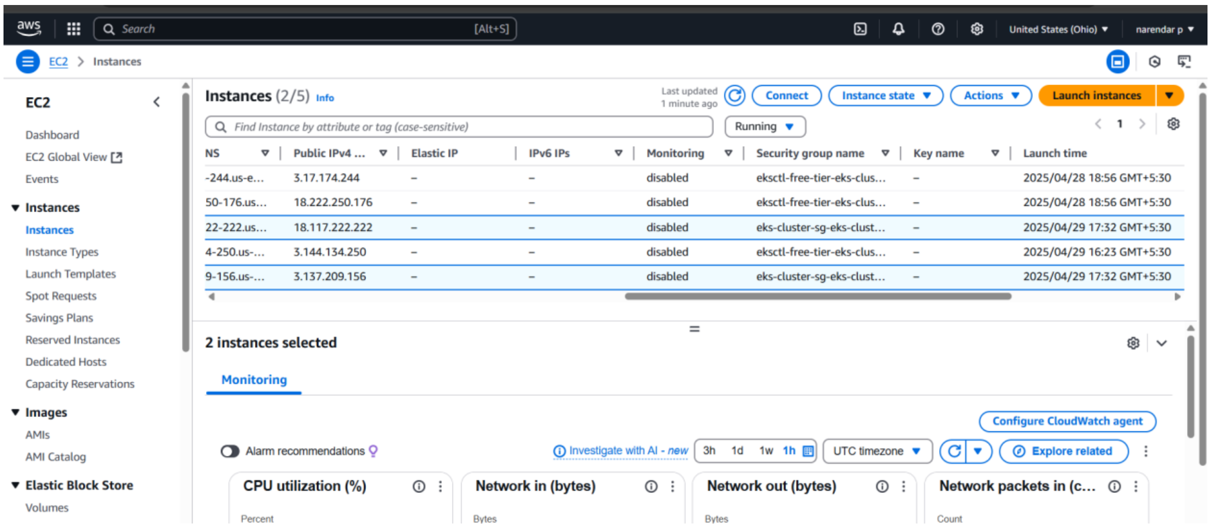
Find Instance by attribute or tag (case-sensitive) | Running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
☐	free-tier-eks-free-tier-no...	i-03672385d28321ea5	Running	t3.medium	3/3 checks passec	View alarms +	us-east-2a	ec2
☐	free-tier-eks-free-tier-no...	i-022d1fb0b3f0f9588	Running	t3.medium	3/3 checks passec	View alarms +	us-east-2b	ec2
☑		i-07a4821369ab5c6c6	Running	c6g.large	3/3 checks passec	View alarms +	us-east-2b	ec2
☐	free-tier-eks-free-tier-no...	i-047137356683266f1	Running	t3.medium	3/3 checks passec	View alarms +	us-east-2c	ec2
☑		i-0e1a15262d3e52b3f	Running	c6g.large	3/3 checks passec	View alarms +	us-east-2c	ec2

2 instances selected

Monitoring

CPU utilization (%) | Network in (bytes) | Network out (bytes) | Network packets in (c...


EC2 > Instances

Instances (2/5) Info — Last updated 1 minute ago — Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) — Running

NS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Launch time
-244.us-e...	3.17.174.244	–	–	disabled	eksctl-free-tier-eks-clus...	–	2025/04/28 18:56 GMT+5:30
50-176.us...	18.222.250.176	–	–	disabled	eksctl-free-tier-eks-clus...	–	2025/04/28 18:56 GMT+5:30
22-222.us...	18.117.222.222	–	–	disabled	eks-cluster-sg-eks-clust...	–	2025/04/29 17:32 GMT+5:30
4-250.us-...	3.144.134.250	–	–	disabled	eksctl-free-tier-eks-clus...	–	2025/04/29 16:23 GMT+5:30
9-156.us-...	3.137.209.156	–	–	disabled	eks-cluster-sg-eks-clust...	–	2025/04/29 17:32 GMT+5:30

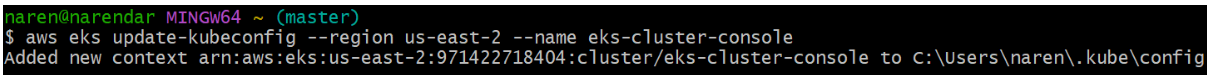
2 instances selected

Monitoring

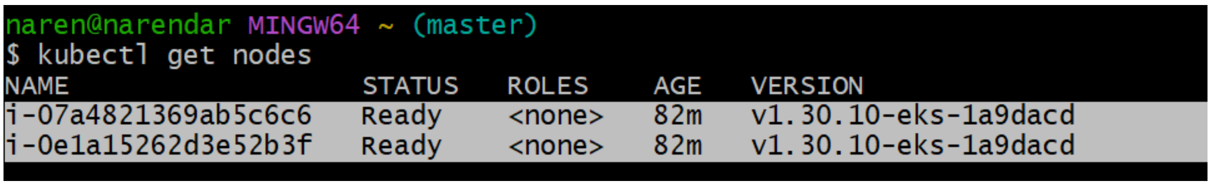
CPU utilization (%) — Percent; Network in (bytes) — Bytes; Network out (bytes) — Bytes; Network packets in (c... — Count

--now connect to the cluster in local using below command

aws eks update-kubeconfig --region us-east-2 --name eks-cluster-console


```
naren@narendar MINGW64 ~ (master)
$ aws eks update-kubeconfig --region us-east-2 --name eks-cluster-console
Added new context arn:aws:eks:us-east-2:971422718404:cluster/eks-cluster-console to C:\Users\naren\.kube\config
```

--check nodes two are created

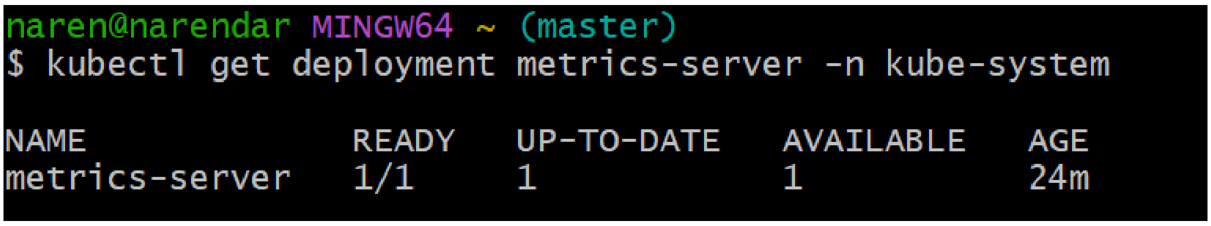

```
naren@narendar MINGW64 ~ (master)
$ kubectl get nodes
NAME                  STATUS   ROLES    AGE   VERSION
i-07a4821369ab5c6c6   Ready    <none>   82m   v1.30.10-eks-1a9dacd
i-0e1a15262d3e52b3f   Ready    <none>   82m   v1.30.10-eks-1a9dacd
```

3) Setup HPA.

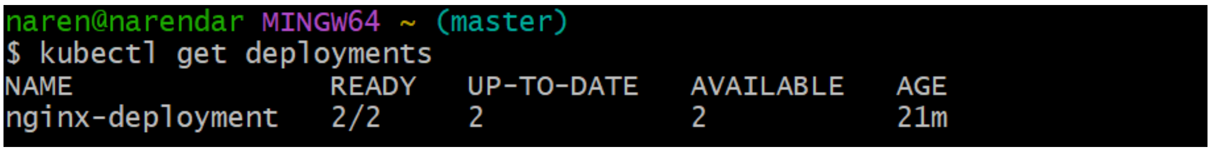
--install metrics server

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

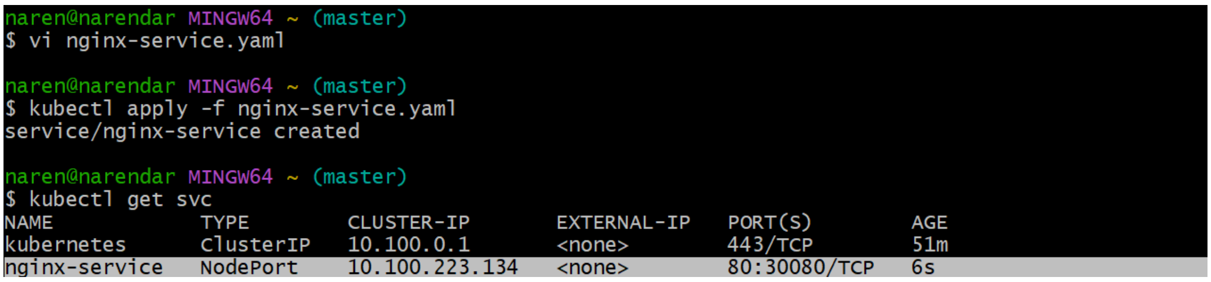
-check it


```
naren@narendar MINGW64 ~ (master)
$ kubectl get deployment metrics-server -n kube-system

NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           24m
```

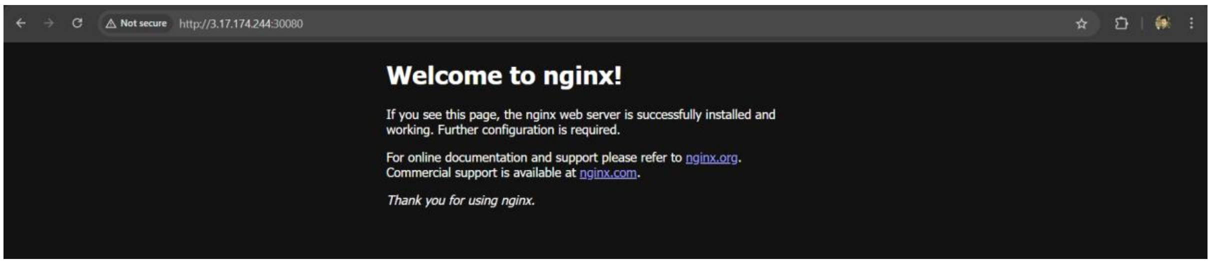
-- Create a Deployment


```
naren@narendar MINGW64 ~ (master)
$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           21m
```

--expose nginx using nodeport



```
naren@narendar MINGW64 ~ (master)
$ vi nginx-service.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f nginx-service.yaml
service/nginx-service created

naren@narendar MINGW64 ~ (master)
$ kubectl get svc
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.100.0.1       <none>        443/TCP        51m
nginx-service   NodePort    10.100.223.134   <none>        80:30080/TCP   6s
```



Not secure http://3.17.174.244:30080

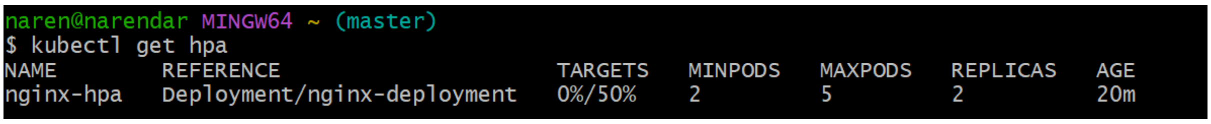
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

-- Create HPA Object

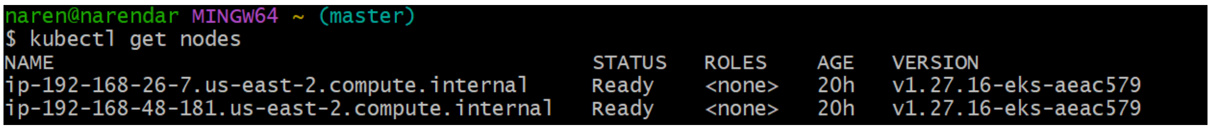


```
naren@narendar MINGW64 ~ (master)
$ kubectl get hpa
NAME        REFERENCE                     TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-deployment   0%/50%    2         5         2          20m
```

If it crosses 50%, HPA will scale the deployment!

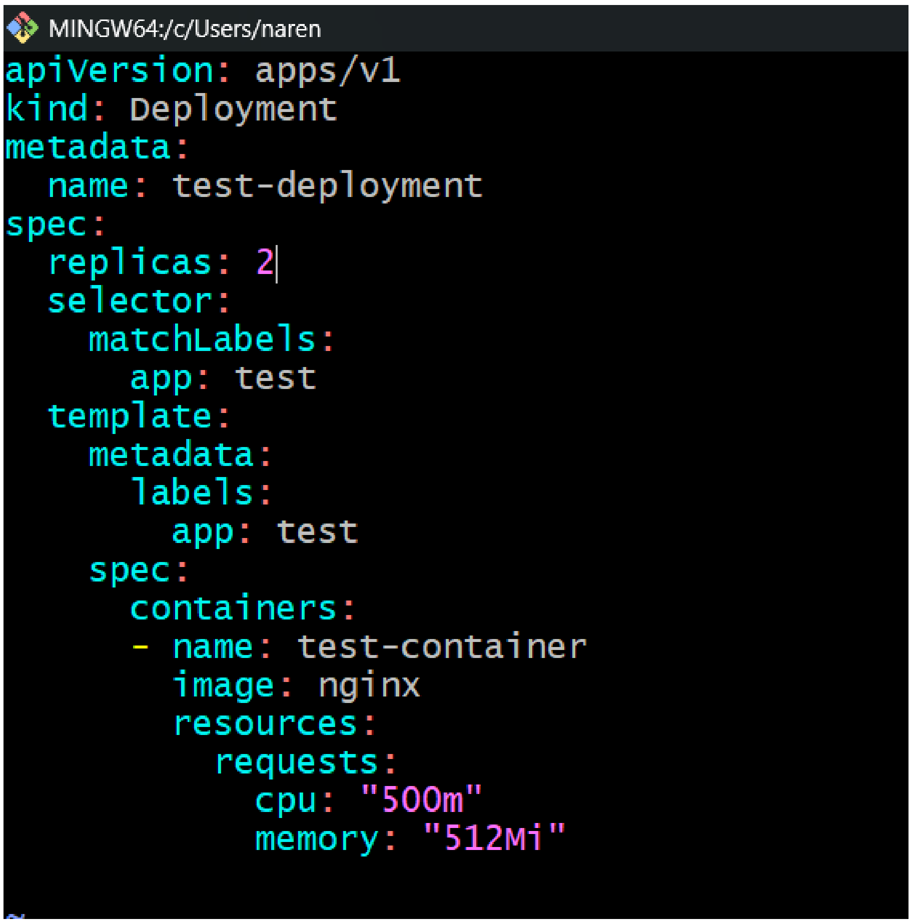
4) Setup cluster autoscale.

--check nodes

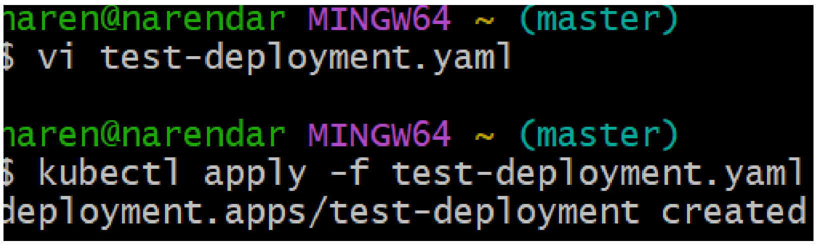


```
naren@narendar MINGW64 ~ (master)
$ kubectl get nodes
NAME                                          STATUS   ROLES    AGE   VERSION
ip-192-168-26-7.us-east-2.compute.internal    Ready    <none>   20h   v1.27.16-eks-aeac579
ip-192-168-48-181.us-east-2.compute.internal  Ready    <none>   20h   v1.27.16-eks-aeac579
```

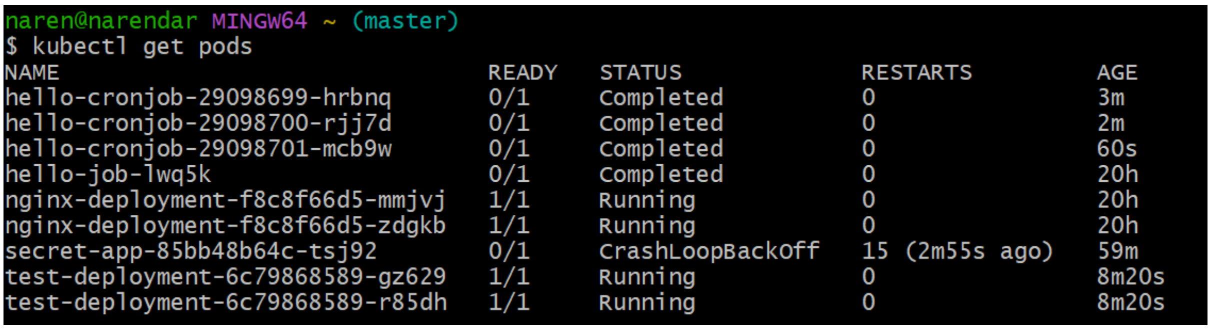
--create test-deployment


MINGW64:/c/Users/naren
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
      - name: test-container
        image: nginx
        resources:
          requests:
            cpu: "500m"
            memory: "512Mi"
```

--run yaml

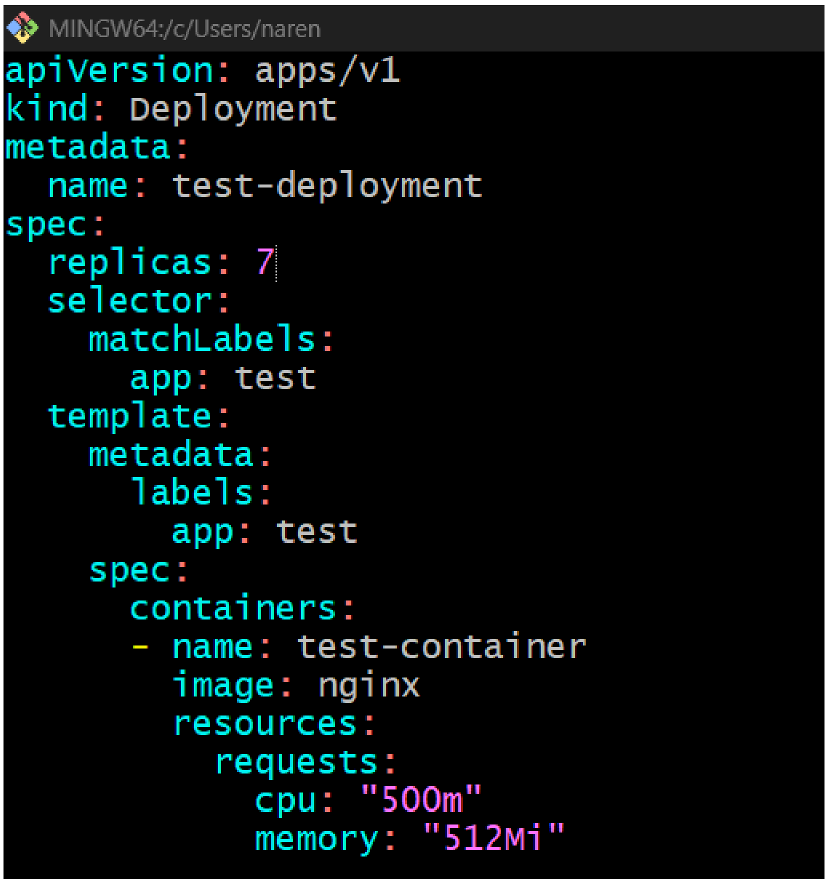

```
aren@narendar MINGW64 ~ (master)
 vi test-deployment.yaml

aren@narendar MINGW64 ~ (master)
 kubectl apply -f test-deployment.yaml
eployment.apps/test-deployment created
```

--check pods


```
naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                                  READY   STATUS             RESTARTS         AGE
hello-cronjob-29098699-hrbnq          0/1     Completed          0                3m
hello-cronjob-29098700-rjj7d          0/1     Completed          0                2m
hello-cronjob-29098701-mcb9w          0/1     Completed          0                60s
hello-job-lwq5k                       0/1     Completed          0                20h
nginx-deployment-f8c8f66d5-mmjvj      1/1     Running            0                20h
nginx-deployment-f8c8f66d5-zdgkb      1/1     Running            0                20h
secret-app-85bb48b64c-tsj92           0/1     CrashLoopBackOff   15 (2m55s ago)   59m
test-deployment-6c79868589-gz629      1/1     Running            0                8m20s
test-deployment-6c79868589-r85dh      1/1     Running            0                8m20s
```

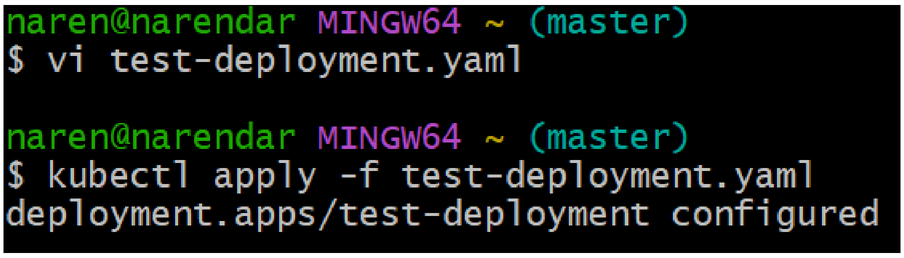
We can see two pods are running

--now edit yaml file pods from 2 to 7



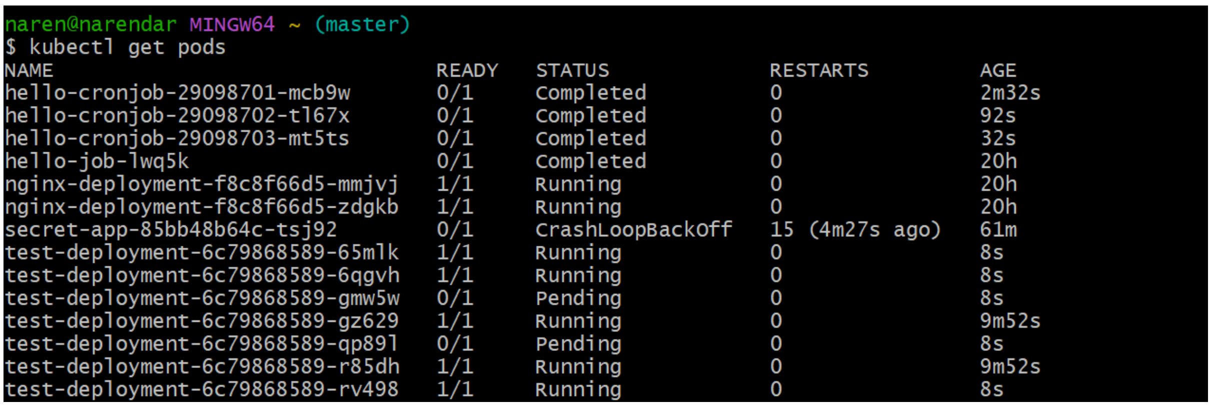
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-deployment
spec:
  replicas: 7
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
      - name: test-container
        image: nginx
        resources:
          requests:
            cpu: "500m"
            memory: "512Mi"
```

--run yaml



```
naren@narendar MINGW64 ~ (master)
$ vi test-deployment.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f test-deployment.yaml
deployment.apps/test-deployment configured
```

--check pods

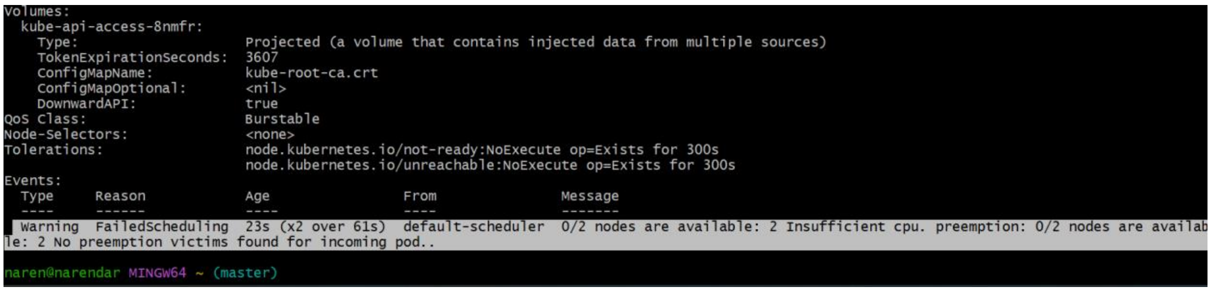


```
naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                               READY   STATUS             RESTARTS        AGE
hello-cronjob-29098701-mcb9w       0/1     Completed          0               2m32s
hello-cronjob-29098702-tl67x       0/1     Completed          0               92s
hello-cronjob-29098703-mt5ts       0/1     Completed          0               32s
hello-job-lwq5k                    0/1     Completed          0               20h
nginx-deployment-f8c8f66d5-mmjvj   1/1     Running            0               20h
nginx-deployment-f8c8f66d5-zdgkb   1/1     Running            0               20h
secret-app-85bb48b64c-tsj92        0/1     CrashLoopBackOff   15 (4m27s ago)  61m
test-deployment-6c79868589-65mlk   1/1     Running            0               8s
test-deployment-6c79868589-6qgvh   1/1     Running            0               8s
test-deployment-6c79868589-gmw5w   0/1     Pending            0               8s
test-deployment-6c79868589-gz629   1/1     Running            0               9m52s
test-deployment-6c79868589-qp89l   0/1     Pending            0               8s
test-deployment-6c79868589-r85dh   1/1     Running            0               9m52s
test-deployment-6c79868589-rv498   1/1     Running            0               8s
```

-we can see some pods are not running here

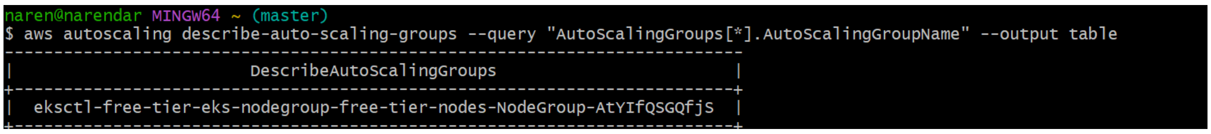
So there is no space to schedule pods

We can see describe pending pod



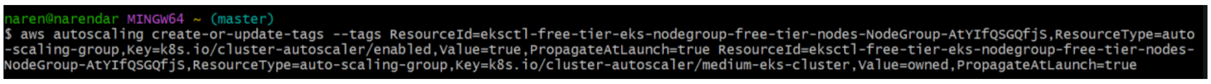
```
Volumes:
  kube-api-access-8nmfr:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age                From               Message
  ----     ------            ----               ----               -------
  Warning  FailedScheduling  23s (x2 over 61s)  default-scheduler  0/2 nodes are available: 2 Insufficient cpu. preemption: 0/2 nodes are availab
le: 2 No preemption victims found for incoming pod..

naren@narendar MINGW64 ~ (master)
```

--now get the ASG group name by using below command



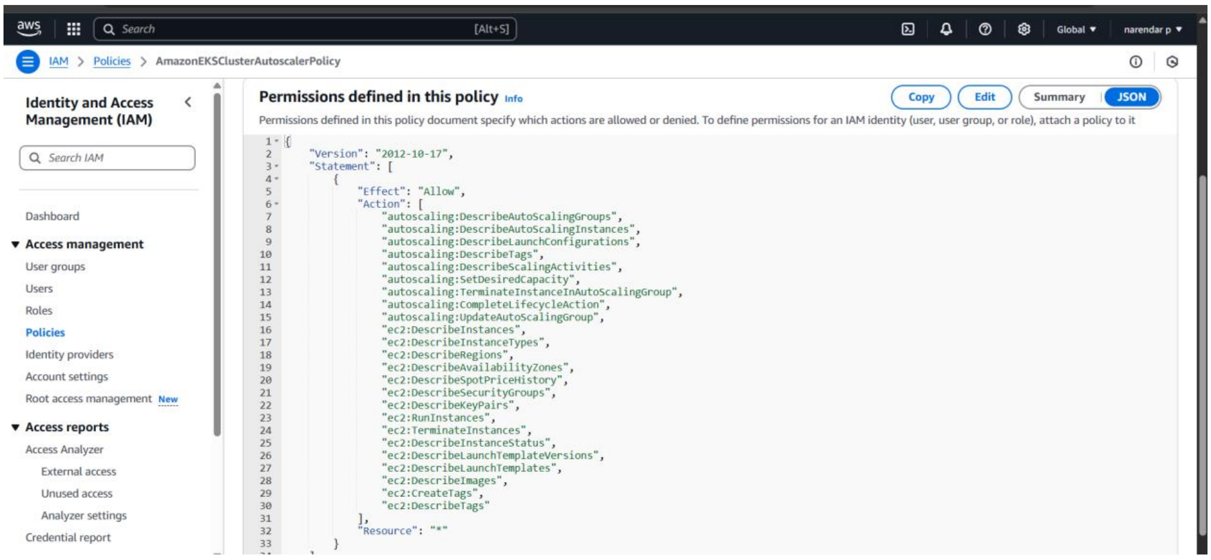
```
naren@narendar MINGW64 ~ (master)
$ aws autoscaling describe-auto-scaling-groups --query "AutoScalingGroups[*].AutoScalingGroupName" --output table
-------------------------------------------------------------------------
|                       DescribeAutoScalingGroups                       |
+-----------------------------------------------------------------------+
|  eksctl-free-tier-eks-nodegroup-free-tier-nodes-NodeGroup-AtYIfQSGQfjS  |
+-----------------------------------------------------------------------+
```

--update the tags now using below command



```
naren@narendar MINGW64 ~ (master)
$ aws autoscaling create-or-update-tags --tags ResourceId=eksctl-free-tier-eks-nodegroup-free-tier-nodes-NodeGroup-AtYIfQSGQfjS,ResourceType=auto
-scaling-group,Key=k8s.io/cluster-autoscaler/enabled,Value=true,PropagateAtLaunch=true ResourceId=eksctl-free-tier-eks-nodegroup-free-tier-nodes-
NodeGroup-AtYIfQSGQfjS,ResourceType=auto-scaling-group,Key=k8s.io/cluster-autoscaler/medium-eks-cluster,Value=owned,PropagateAtLaunch=true
```

--create iam policy

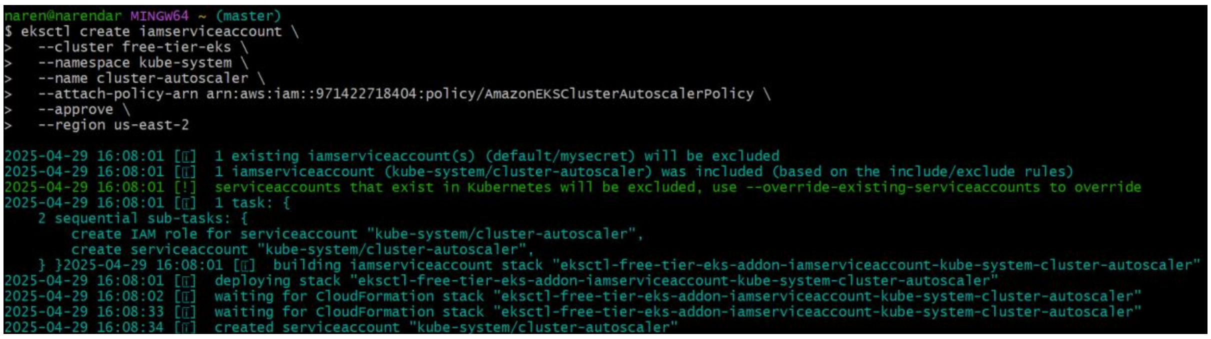


Permissions defined in this policy Info

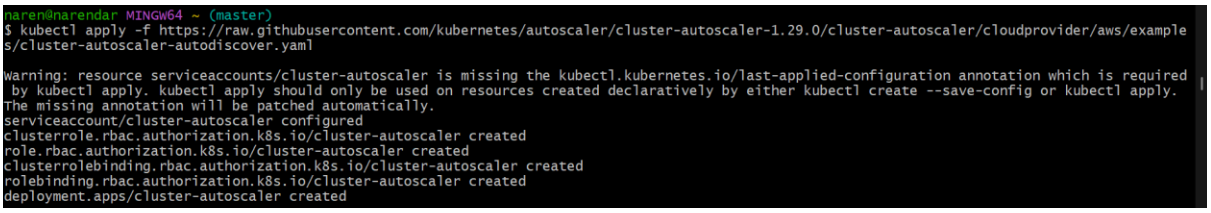
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeTags",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "autoscaling:CompleteLifecycleAction",
                "autoscaling:UpdateAutoScalingGroup",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeRegions",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeImages",
                "ec2:CreateTags",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

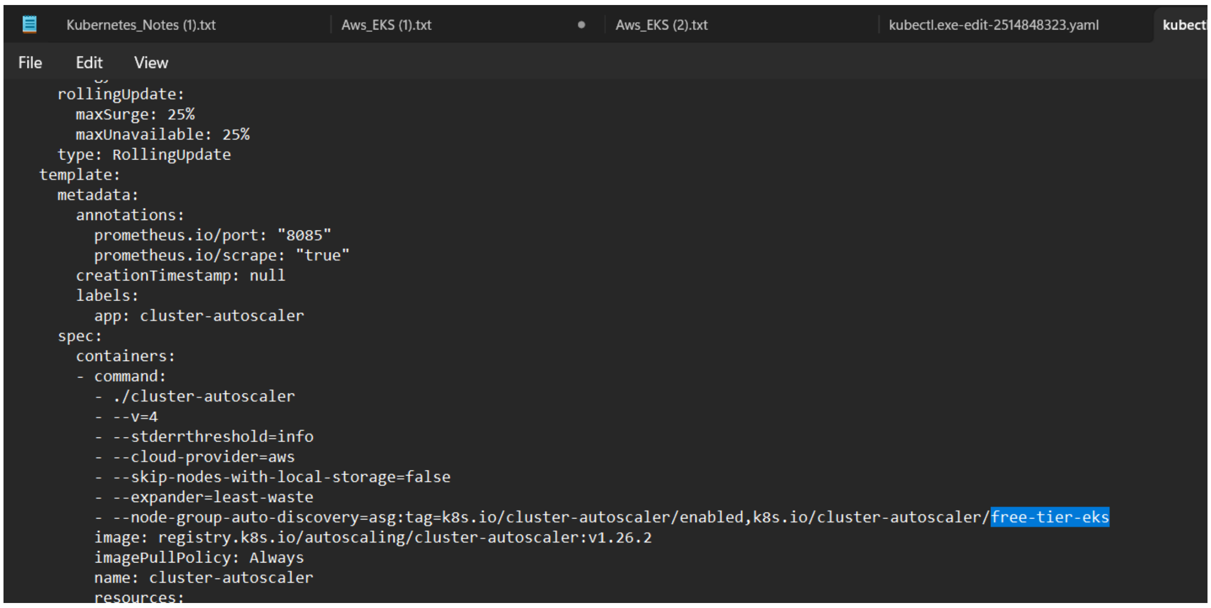
-attach policy to the role



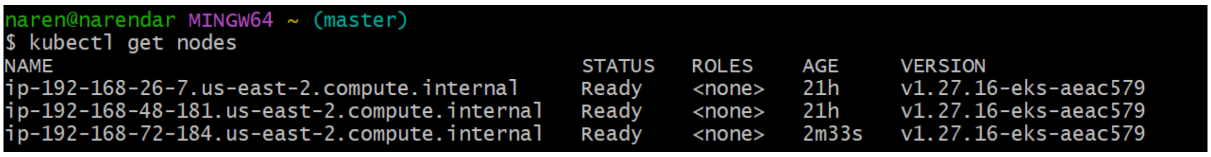
```
naren@narendar MINGW64 ~ (master)
$ eksctl create iamserviceaccount \
>   --cluster free-tier-eks \
>   --namespace kube-system \
>   --name cluster-autoscaler \
>   --attach-policy-arn arn:aws:iam::971422718404:policy/AmazonEKSClusterAutoscalerPolicy \
>   --approve \
>   --region us-east-2

2025-04-29 16:08:01 [ℹ]  1 existing iamserviceaccount(s) (default/mysecret) will be excluded
2025-04-29 16:08:01 [ℹ]  1 iamserviceaccount (kube-system/cluster-autoscaler) was included (based on the include/exclude rules)
2025-04-29 16:08:01 [!]  serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2025-04-29 16:08:01 [ℹ]  1 task: {
    2 sequential sub-tasks: {
        create IAM role for serviceaccount "kube-system/cluster-autoscaler",
        create serviceaccount "kube-system/cluster-autoscaler",
    } }2025-04-29 16:08:01 [ℹ]  building iamserviceaccount stack "eksctl-free-tier-eks-addon-iamserviceaccount-kube-system-cluster-autoscaler"
2025-04-29 16:08:01 [ℹ]  deploying stack "eksctl-free-tier-eks-addon-iamserviceaccount-kube-system-cluster-autoscaler"
2025-04-29 16:08:02 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-addon-iamserviceaccount-kube-system-cluster-autoscaler"
2025-04-29 16:08:33 [ℹ]  waiting for CloudFormation stack "eksctl-free-tier-eks-addon-iamserviceaccount-kube-system-cluster-autoscaler"
2025-04-29 16:08:34 [ℹ]  created serviceaccount "kube-system/cluster-autoscaler"
```



```
naren@narendar MINGW64 ~ (master)
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/autoscaler/cluster-autoscaler-1.29.0/cluster-autoscaler/cloudprovider/aws/example
s/cluster-autoscaler-autodiscover.yaml

Warning: resource serviceaccounts/cluster-autoscaler is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required
 by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply.
The missing annotation will be patched automatically.
serviceaccount/cluster-autoscaler configured
clusterrole.rbac.authorization.k8s.io/cluster-autoscaler created
role.rbac.authorization.k8s.io/cluster-autoscaler created
clusterrolebinding.rbac.authorization.k8s.io/cluster-autoscaler created
rolebinding.rbac.authorization.k8s.io/cluster-autoscaler created
deployment.apps/cluster-autoscaler created
```

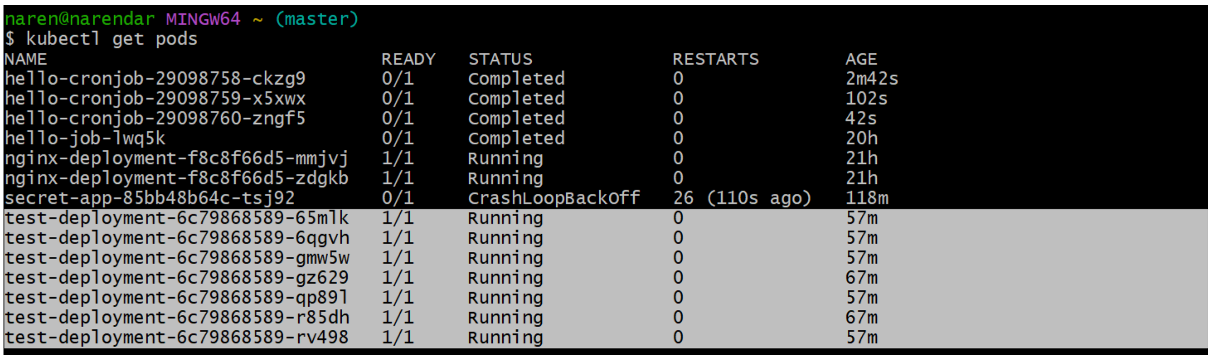


```
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      annotations:
        prometheus.io/port: "8085"
        prometheus.io/scrape: "true"
      creationTimestamp: null
      labels:
        app: cluster-autoscaler
    spec:
      containers:
      - command:
        - ./cluster-autoscaler
        - --v=4
        - --stderrthreshold=info
        - --cloud-provider=aws
        - --skip-nodes-with-local-storage=false
        - --expander=least-waste
        - --node-group-auto-discovery=asg:tag=k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/free-tier-eks
        image: registry.k8s.io/autoscaling/cluster-autoscaler:v1.26.2
        imagePullPolicy: Always
        name: cluster-autoscaler
        resources:
```



```
naren@narendar MINGW64 ~ (master)
$ kubectl get nodes
NAME                                          STATUS   ROLES    AGE    VERSION
ip-192-168-26-7.us-east-2.compute.internal    Ready    <none>   21h    v1.27.16-eks-aeac579
ip-192-168-48-181.us-east-2.compute.internal  Ready    <none>   21h    v1.27.16-eks-aeac579
ip-192-168-72-184.us-east-2.compute.internal  Ready    <none>   2m33s  v1.27.16-eks-aeac579
```

--all pods are running



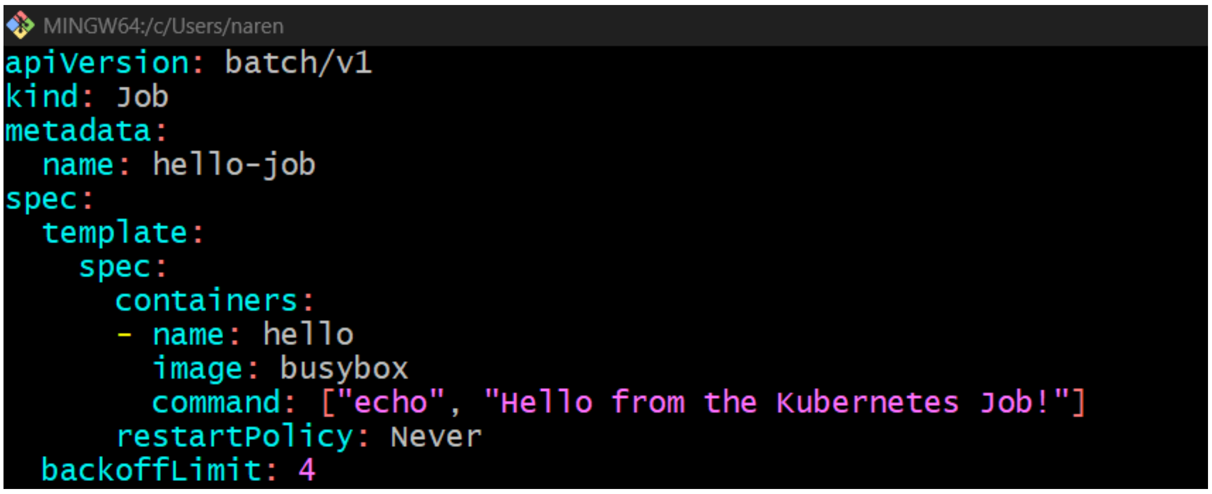
```
naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                               READY   STATUS             RESTARTS        AGE
hello-cronjob-29098758-ckzg9       0/1     Completed          0               2m42s
hello-cronjob-29098759-x5xwx       0/1     Completed          0               102s
hello-cronjob-29098760-zngf5       0/1     Completed          0               42s
hello-job-lwq5k                    0/1     Completed          0               20h
nginx-deployment-f8c8f66d5-mmjvj   1/1     Running            0               21h
nginx-deployment-f8c8f66d5-zdgkb   1/1     Running            0               21h
secret-app-85bb48b64c-tsj92        0/1     CrashLoopBackOff   26 (110s ago)   118m
test-deployment-6c79868589-65mlk   1/1     Running            0               57m
test-deployment-6c79868589-6qgvh   1/1     Running            0               57m
test-deployment-6c79868589-gmw5w   1/1     Running            0               57m
test-deployment-6c79868589-gz629   1/1     Running            0               67m
test-deployment-6c79868589-qp89l   1/1     Running            0               57m
test-deployment-6c79868589-r85dh   1/1     Running            0               67m
test-deployment-6c79868589-rv498   1/1     Running            0               57m
```

5) Setup job and cronjob.

--job

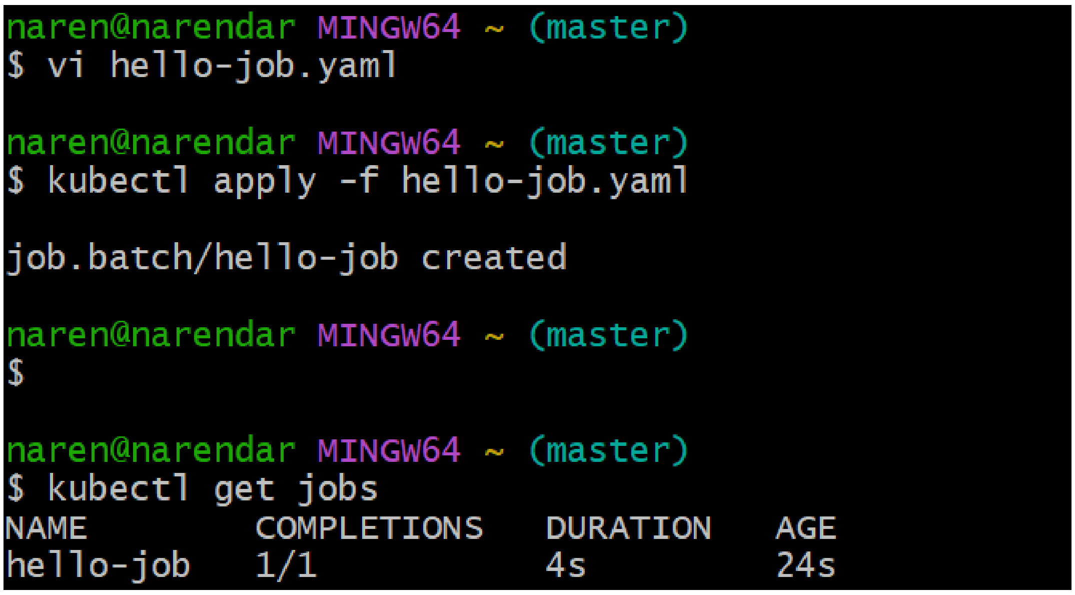
Job runs a one-time task (e.g., backup database, process a file).

--yaml file



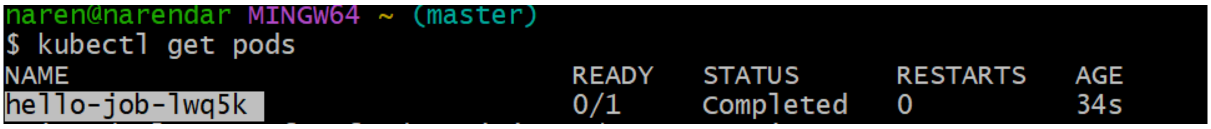
```
MINGW64:/c/Users/naren
apiVersion: batch/v1
kind: Job
metadata:
  name: hello-job
spec:
  template:
    spec:
      containers:
      - name: hello
        image: busybox
        command: ["echo", "Hello from the Kubernetes Job!"]
      restartPolicy: Never
  backoffLimit: 4
```

-run yaml job got created



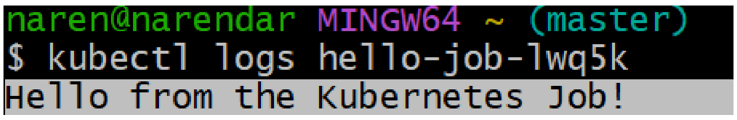
```
naren@narendar MINGW64 ~ (master)
$ vi hello-job.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f hello-job.yaml

job.batch/hello-job created

naren@narendar MINGW64 ~ (master)
$

naren@narendar MINGW64 ~ (master)
$ kubectl get jobs
NAME         COMPLETIONS   DURATION   AGE
hello-job    1/1           4s         24s
```

--check pods



```
naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                                        READY   STATUS      RESTARTS   AGE
hello-job-lwq5k                             0/1     Completed   0          34s
```

--check logs of a hello-job

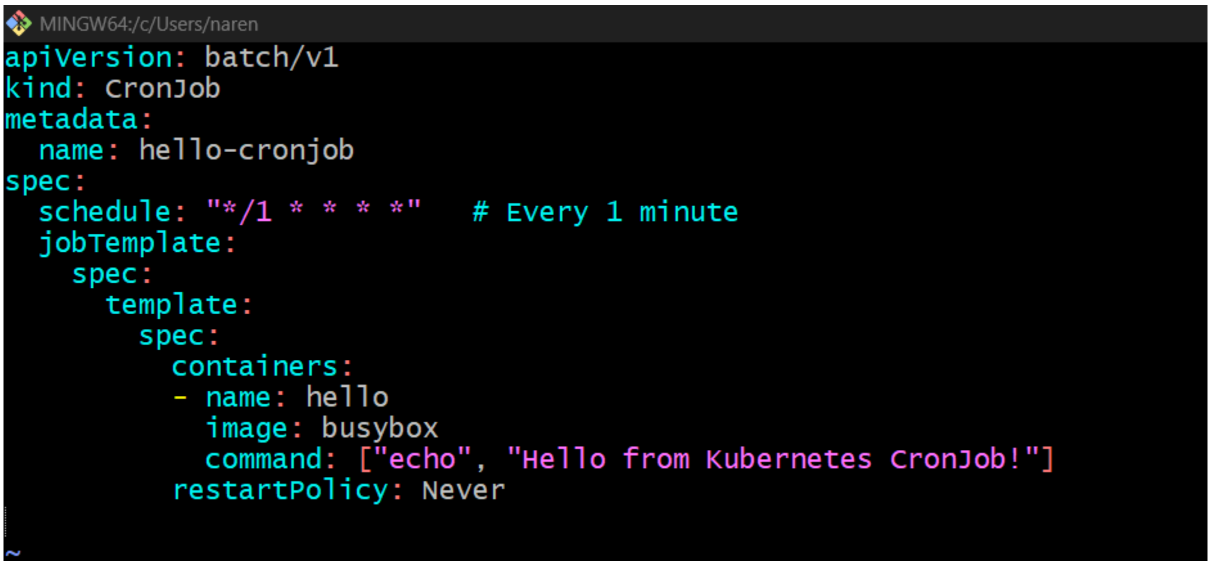


```
naren@narendar MINGW64 ~ (master)
$ kubectl logs hello-job-lwq5k
Hello from the Kubernetes Job!
```

--cron job

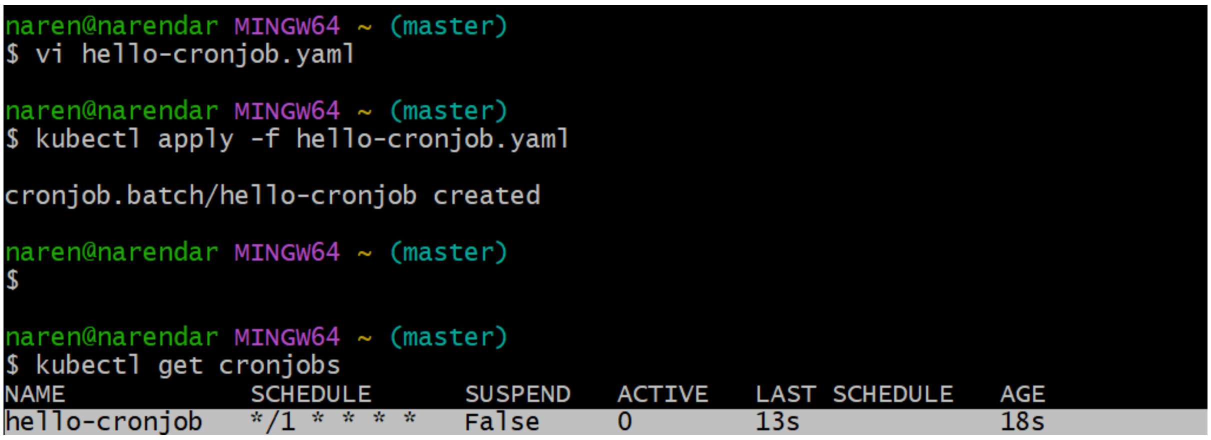
CronJob runs a job on a schedule (like Linux cron)

--yaml file



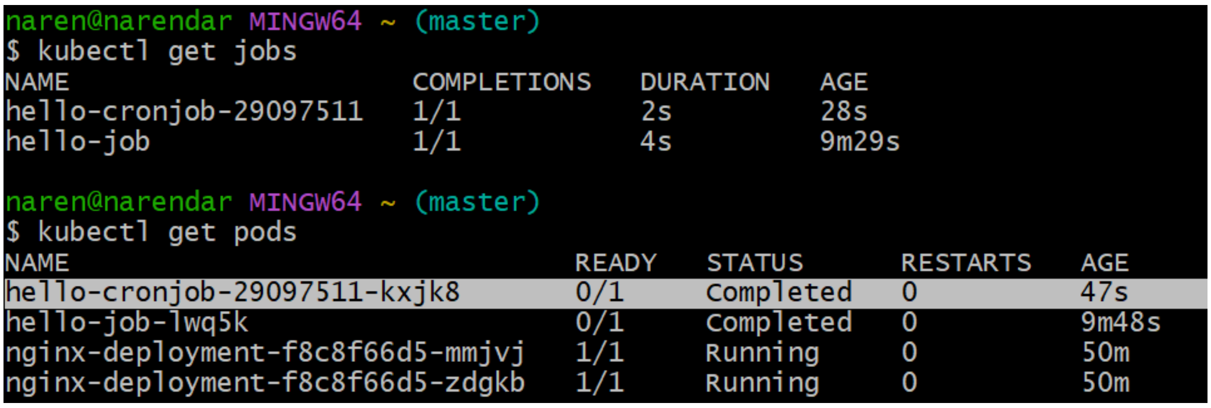
```
MINGW64:/c/Users/naren
apiVersion: batch/v1
kind: CronJob
metadata:
  name: hello-cronjob
spec:
  schedule: "*/1 * * * *"   # Every 1 minute
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox
            command: ["echo", "Hello from Kubernetes CronJob!"]
          restartPolicy: Never

~
```

-run yaml cronjob got created



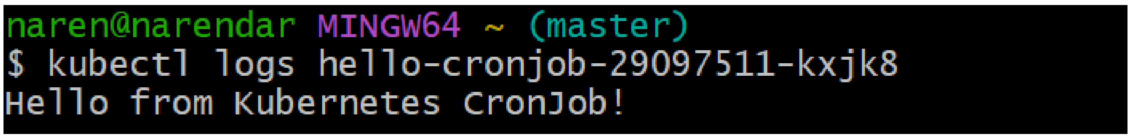
```
naren@narendar MINGW64 ~ (master)
$ vi hello-cronjob.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f hello-cronjob.yaml

cronjob.batch/hello-cronjob created

naren@narendar MINGW64 ~ (master)
$

naren@narendar MINGW64 ~ (master)
$ kubectl get cronjobs
NAME            SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello-cronjob   */1 * * * *   False     0        13s             18s
```

--check jobs pods



```
naren@narendar MINGW64 ~ (master)
$ kubectl get jobs
NAME                     COMPLETIONS   DURATION   AGE
hello-cronjob-29097511   1/1           2s         28s
hello-job                1/1           4s         9m29s

naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                                READY   STATUS      RESTARTS   AGE
hello-cronjob-29097511-kxjk8        0/1     Completed   0          47s
hello-job-lwq5k                     0/1     Completed   0          9m48s
nginx-deployment-f8c8f66d5-mmjvj    1/1     Running     0          50m
nginx-deployment-f8c8f66d5-zdgkb    1/1     Running     0          50m
```

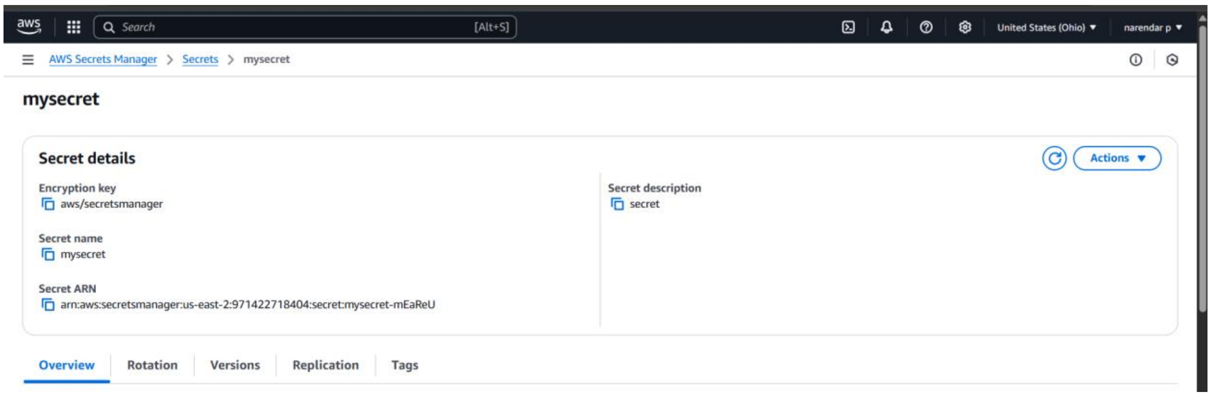
--check logs of a hello-job



```
naren@narendar MINGW64 ~ (master)
$ kubectl logs hello-cronjob-29097511-kxjk8
Hello from Kubernetes CronJob!
```

6) Create secret and inject inside pod.

--create a secret in aws secret manager



mysecret

Secret details

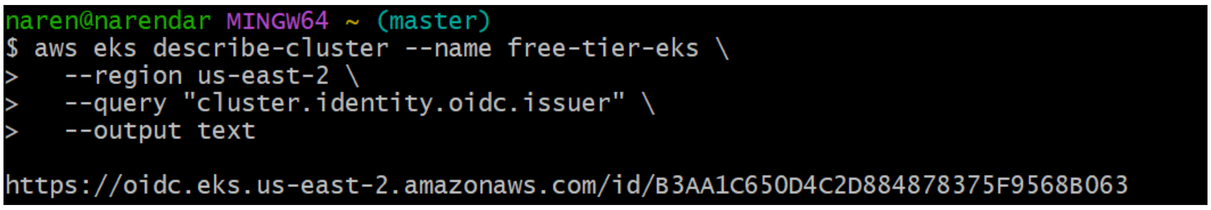
Encryption key
aws/secretsmanager

Secret name
mysecret

Secret ARN
arn:aws:secretsmanager:us-east-2:971422718404:secret:mysecret-mEaReU

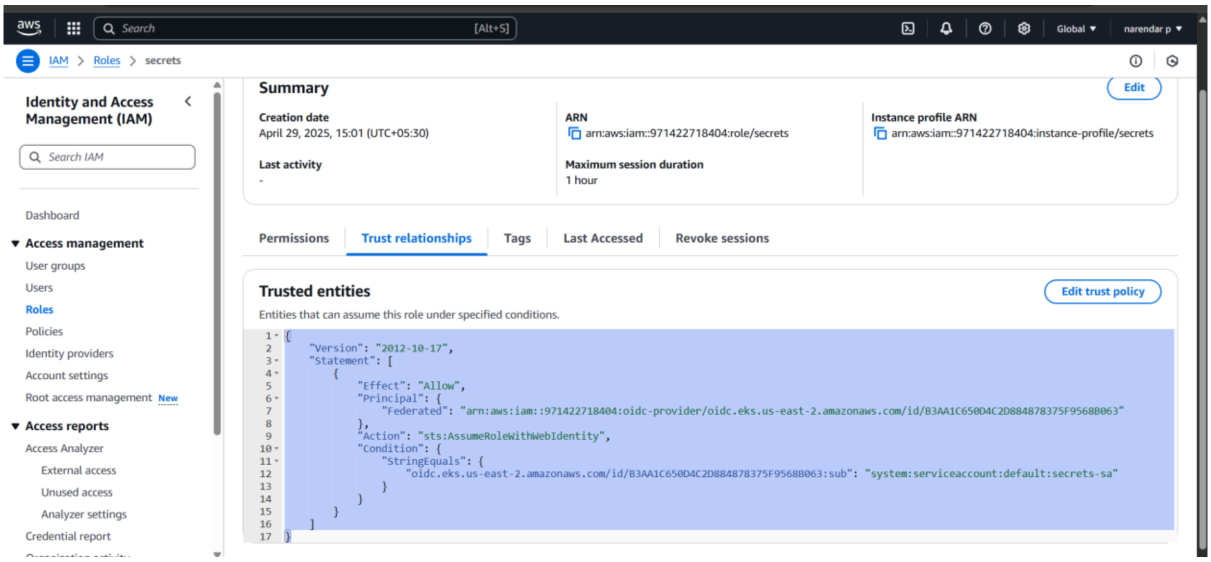
Secret description
secret

--set oidc



```
naren@narendar MINGW64 ~ (master)
$ aws eks describe-cluster --name free-tier-eks \
>   --region us-east-2 \
>   --query "cluster.identity.oidc.issuer" \
>   --output text

https://oidc.eks.us-east-2.amazonaws.com/id/B3AA1C650D4C2D884878375F9568B063
```

--create iam policy and attach it to the role



Summary

Creation date
April 29, 2025, 15:01 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::971422718404:role/secrets

Maximum session duration
1 hour

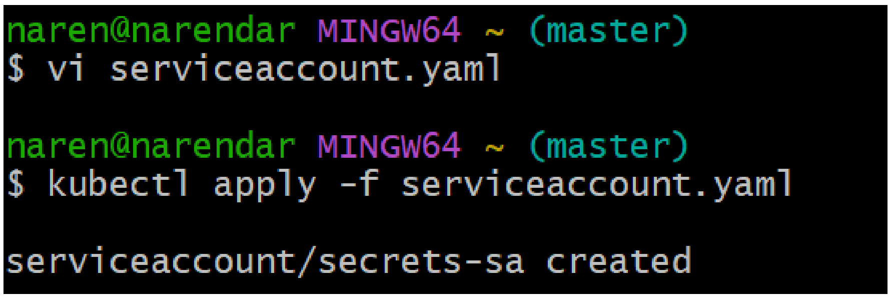
Instance profile ARN
arn:aws:iam::971422718404:instance-profile/secrets

Trusted entities

Entities that can assume this role under specified conditions.

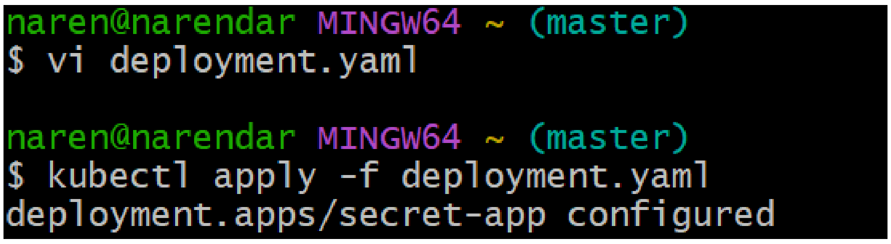
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::971422718404:oidc-provider/oidc.eks.us-east-2.amazonaws.com/id/B3AA1C650D4C2D884878375F9568B063"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.us-east-2.amazonaws.com/id/B3AA1C650D4C2D884878375F9568B063:sub": "system:serviceaccount:default:secrets-sa"
                }
            }
        }
    ]
}
```

--create service account

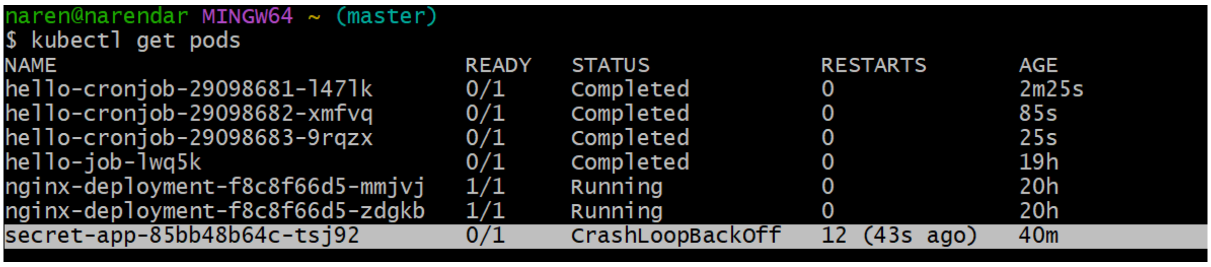


```
naren@narendar MINGW64 ~ (master)
$ vi serviceaccount.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f serviceaccount.yaml

serviceaccount/secrets-sa created
```

--create deployment yaml



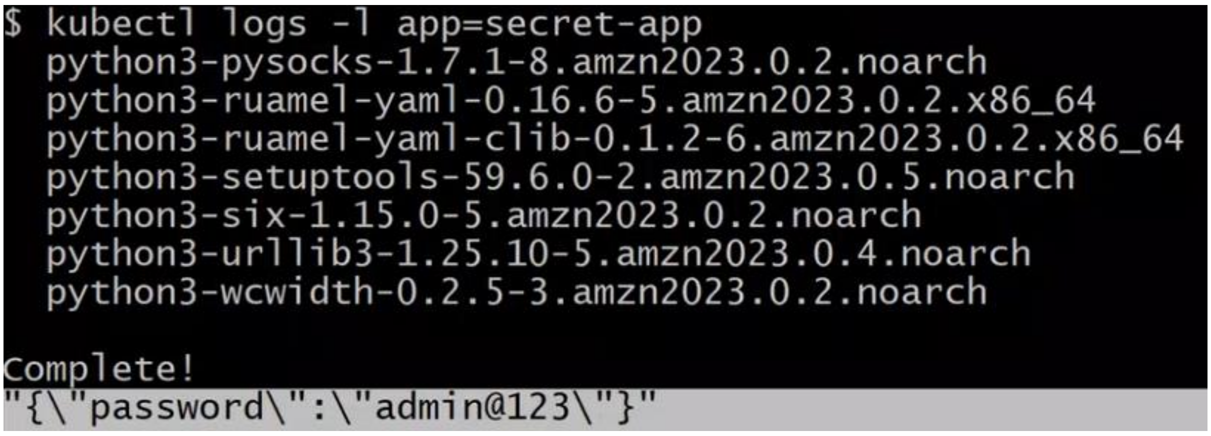
```
naren@narendar MINGW64 ~ (master)
$ vi deployment.yaml

naren@narendar MINGW64 ~ (master)
$ kubectl apply -f deployment.yaml
deployment.apps/secret-app configured
```



```
naren@narendar MINGW64 ~ (master)
$ kubectl get pods
NAME                                READY   STATUS             RESTARTS      AGE
hello-cronjob-29098681-l47lk        0/1     Completed          0             2m25s
hello-cronjob-29098682-xmfvq        0/1     Completed          0             85s
hello-cronjob-29098683-9rqzx        0/1     Completed          0             25s
hello-job-lwq5k                     0/1     Completed          0             19h
nginx-deployment-f8c8f66d5-mmjvj    1/1     Running            0             20h
nginx-deployment-f8c8f66d5-zdgkb    1/1     Running            0             20h
secret-app-85bb48b64c-tsj92         0/1     CrashLoopBackOff   12 (43s ago)  40m
```

--Test the credentials by checking logs

kubectl logs -l app=secret-app

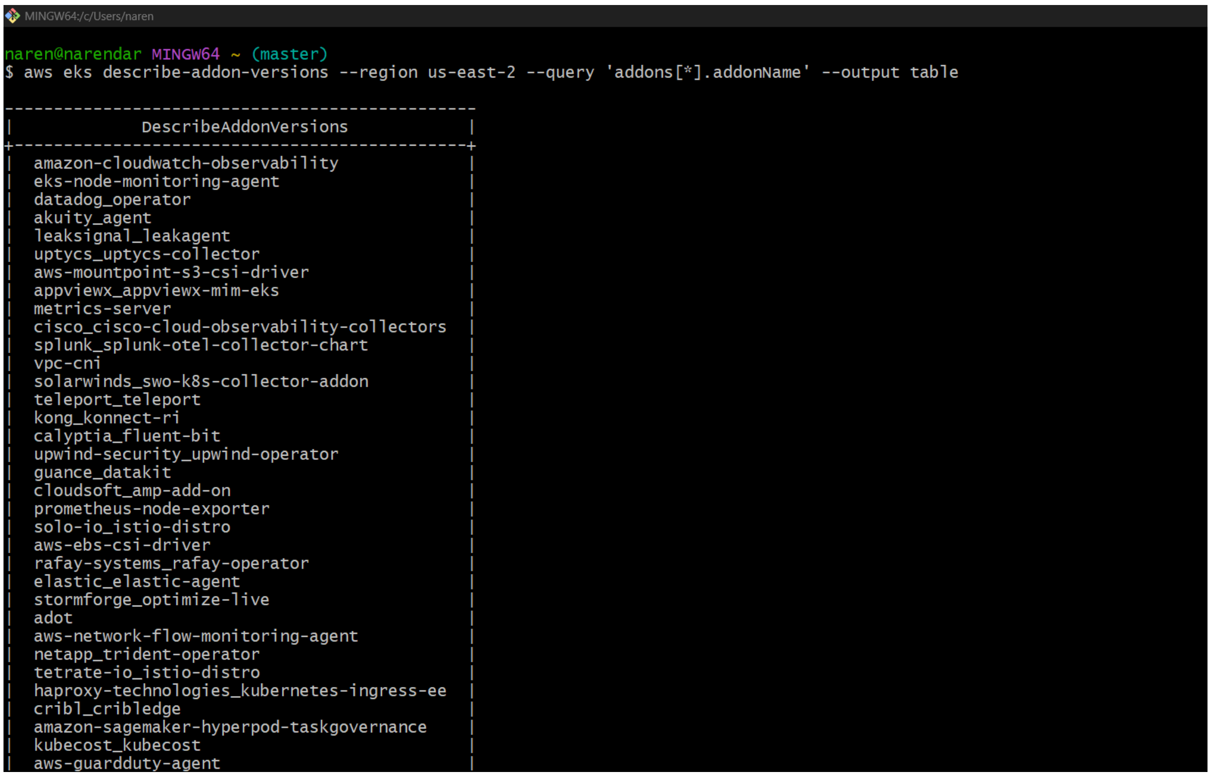


```
$ kubectl logs -l app=secret-app
  python3-pysocks-1.7.1-8.amzn2023.0.2.noarch
  python3-ruamel-yaml-0.16.6-5.amzn2023.0.2.x86_64
  python3-ruamel-yaml-clib-0.1.2-6.amzn2023.0.2.x86_64
  python3-setuptools-59.6.0-2.amzn2023.0.5.noarch
  python3-six-1.15.0-5.amzn2023.0.2.noarch
  python3-urllib3-1.25.10-5.amzn2023.0.4.noarch
  python3-wcwidth-0.2.5-3.amzn2023.0.2.noarch

Complete!
"{\"password\":\"admin@123\"}"
```

7) Check different add-ons available on eks.

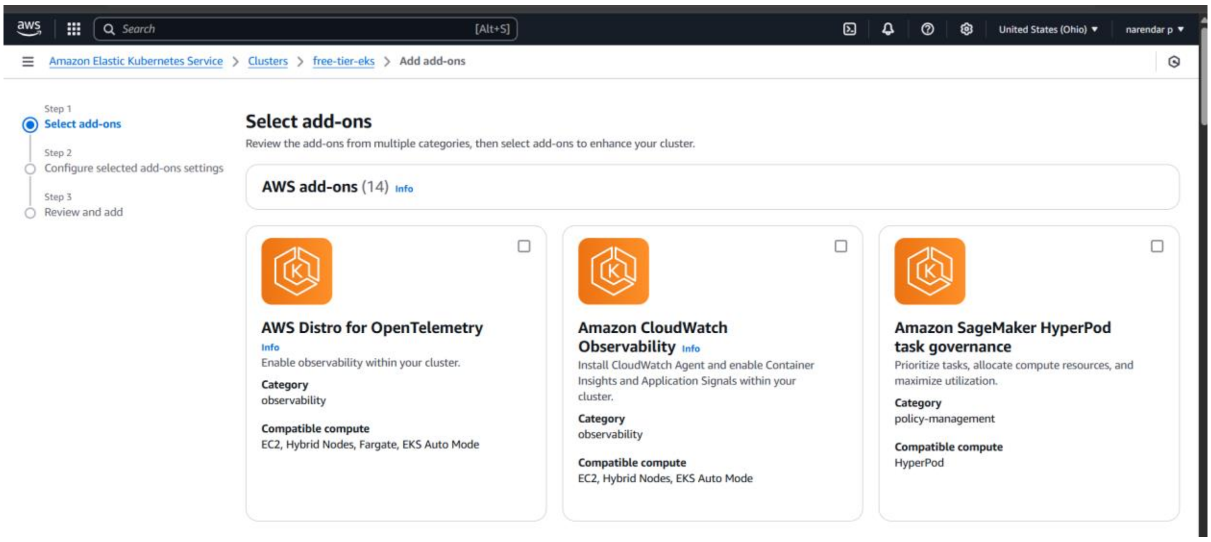
--check add-ons with below command

aws eks describe-addon-versions --region us-east-2 --query 'addons[*].addonName' --output table



```
MINGW64:/c/Users/naren

naren@narendar MINGW64 ~ (master)
$ aws eks describe-addon-versions --region us-east-2 --query 'addons[*].addonName' --output table

-----------------------------------------------
|            DescribeAddonVersions            |
+---------------------------------------------+
|  amazon-cloudwatch-observability            |
|  eks-node-monitoring-agent                  |
|  datadog_operator                           |
|  akuity_agent                               |
|  leaksignal_leakagent                       |
|  uptycs_uptycs-collector                    |
|  aws-mountpoint-s3-csi-driver               |
|  appviewx_appviewx-mim-eks                  |
|  metrics-server                             |
|  cisco_cisco-cloud-observability-collectors |
|  splunk_splunk-otel-collector-chart         |
|  vpc-cni                                    |
|  solarwinds_swo-k8s-collector-addon         |
|  teleport_teleport                          |
|  kong_konnect-ri                            |
|  calyptia_fluent-bit                        |
|  upwind-security_upwind-operator            |
|  guance_datakit                             |
|  cloudsoft_amp-add-on                       |
|  prometheus-node-exporter                   |
|  solo-io_istio-distro                       |
|  aws-ebs-csi-driver                         |
|  rafay-systems_rafay-operator               |
|  elastic_elastic-agent                      |
|  stormforge_optimize-live                   |
|  adot                                       |
|  aws-network-flow-monitoring-agent          |
|  netapp_trident-operator                    |
|  tetrate-io_istio-distro                    |
|  haproxy-technologies_kubernetes-ingress-ee |
|  cribl_cribledge                            |
|  amazon-sagemaker-hyperpod-taskgovernance   |
|  kubecost_kubecost                          |
|  aws-guardduty-agent                        |
```

-- we can also check this login AWS→EKS→select the cluster→select the tab add-ons



aws
Search [Alt+S]
United States (Ohio)
narendar p
Amazon Elastic Kubernetes Service > Clusters > free-tier-eks > Add add-ons
Step 1
Select add-ons
Step 2
Configure selected add-ons settings
Step 3
Review and add

Select add-ons

Review the add-ons from multiple categories, then select add-ons to enhance your cluster.

AWS add-ons (14) Info

AWS Distro for OpenTelemetry
Info
Enable observability within your cluster.
Category
observability
Compatible compute
EC2, Hybrid Nodes, Fargate, EKS Auto Mode

Amazon CloudWatch Observability Info
Install CloudWatch Agent and enable Container Insights and Application Signals within your cluster.
Category
observability
Compatible compute
EC2, Hybrid Nodes, EKS Auto Mode

Amazon SageMaker HyperPod task governance
Prioritize tasks, allocate compute resources, and maximize utilization.
Category
policy-management
Compatible compute
HyperPod