Tao Chen
Project 1

I decided to use AWS EKS for creating the Kubernetes.
What's needed is aws cli, kubectl, Required IAM permissions, CloudFormation (VPC)
**Task 0 Setup for Amazon EKS:**

For CLI:
Need the access and secret key from My Security Credentials at console.aws.amazon.com/
Creating profile file and credentials

```
> aws configure
> access key
> secret key
> default region us-east-1                    #   for north virginia
> Default output json                         #   this will return output in json format
```

Command setting the environment, can manually input into environment

```
> setx AWS_ACCESS_KEY_ID access key
> setx AWS_SECRET_ACCESS_KEY secret key
> setx AWS_DEFAULT_REGION region        #For n.virginia it is ‘us-east-1’
```

For kubectl  :   on Windows powershell

```
> curl -o kubectl.exe
https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/windows/amd64/kubectl.exe
> kubectl                                  # to execute the installation
> kubectl version --short --client
```

For IAM roles (what the node are e.g. cluster role, or set permission on what the node can do):
Following the doc
docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html
You need to create a role with policy from a json file (For the Cluster)

```
>aws iam create-role --role-name myAmazonEKSClusterRole
--assume-role-policy-document file://"cluster-role-trust-policy.json"
```

Returns a json set from before

```
{  "Role": {
  "Path": "/",
  "RoleName": "myAmazonEKSClusterRole",
"RoleId": "AROASUFTCNB5WI7WT4WLR",
"Arn": "arn:aws:iam::180764174459:role/myAmazonEKSClusterRole",
 "CreateDate": "2021-03-16T16:02:43+00:00",
   "AssumeRolePolicyDocument": {
     "Version": "2012-10-17",
    "Statement": [
     { "Effect": "Allow",
```

"Principal": { "Service": "eks.amazonaws.com" },
"Action": "sts:AssumeRole"} ] } }}

> aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy --role-name myAmazonEKSClusterRole

Creating the VPC: (asw cloudformation)

> aws cloudformation create-stack --stack-name my-eks-vpc-stack --template-url https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml

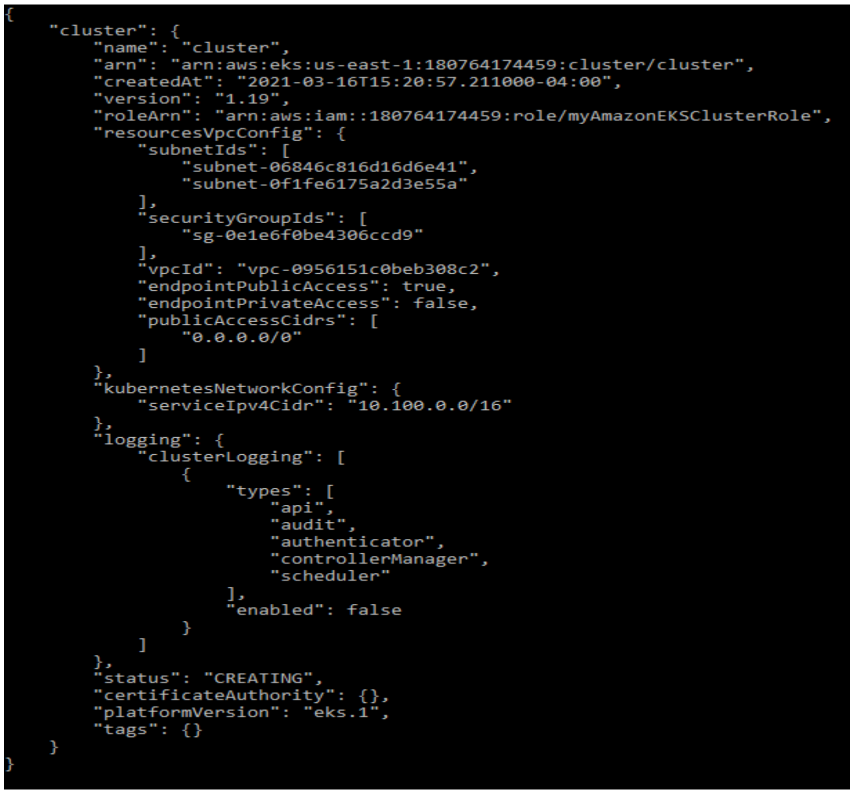
returns :
{"StackId":"arn:aws:cloudformation:us-east-1:180764174459:stack/my-eks-vpc-stack/cb93afe0-8688-11eb-a011-0e72707fff19" }

#At the end when deleting the VPC, I realized Amazon aws gave you one as default#

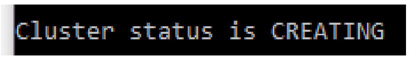
**Task 1 Master Node + Worker Node:**

(Master Node)

>aws eks create-cluster --region us-east-1 --name cluster --kubernetes-version 1.19 --role-arn arn:aws:iam::180764174459:role/myAmazonEKSClusterRole --resources-vpc-config subnetIds=subnet-06846c816d16d6e41,subnet-0f1fe6175a2d3e55a,securityGroupIds=sg-0e1e6f0be4306ccd9



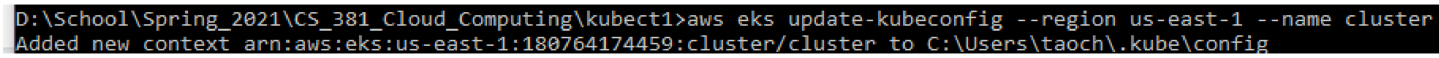
```
{
    "cluster": {
        "name": "cluster",
        "arn": "arn:aws:eks:us-east-1:180764174459:cluster/cluster",
        "createdAt": "2021-03-16T15:20:57.211000-04:00",
        "version": "1.19",
        "roleArn": "arn:aws:iam::180764174459:role/myAmazonEKSClusterRole",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-06846c816d16d6e41",
                "subnet-0f1fe6175a2d3e55a"
            ],
            "securityGroupIds": [
                "sg-0e1e6f0be4306ccd9"
            ],
            "vpcId": "vpc-0956151c0beb308c2",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": false,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": false
                }
            ]
        },
        "status": "CREATING",
        "certificateAuthority": {},
        "platformVersion": "eks.1",
        "tags": {}
    }
}
```

Need to make sure computer can communicate with the cluster (need to wait till finish creating it will take a while otherwise get 

```
Cluster status is CREATING
```

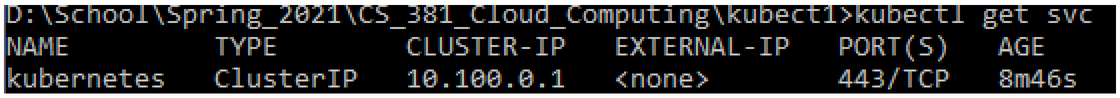
)

>aws eks update-kubeconfig --region us-east-1 --name cluster



```
D:\School\Spring_2021\CS_381_Cloud_Computing\kubect1>aws eks update-kubeconfig --region us-east-1 --name cluster
Added new context arn:aws:eks:us-east-1:180764174459:cluster/cluster to C:\Users\taoch\.kube\config
```

>kubectl get svc



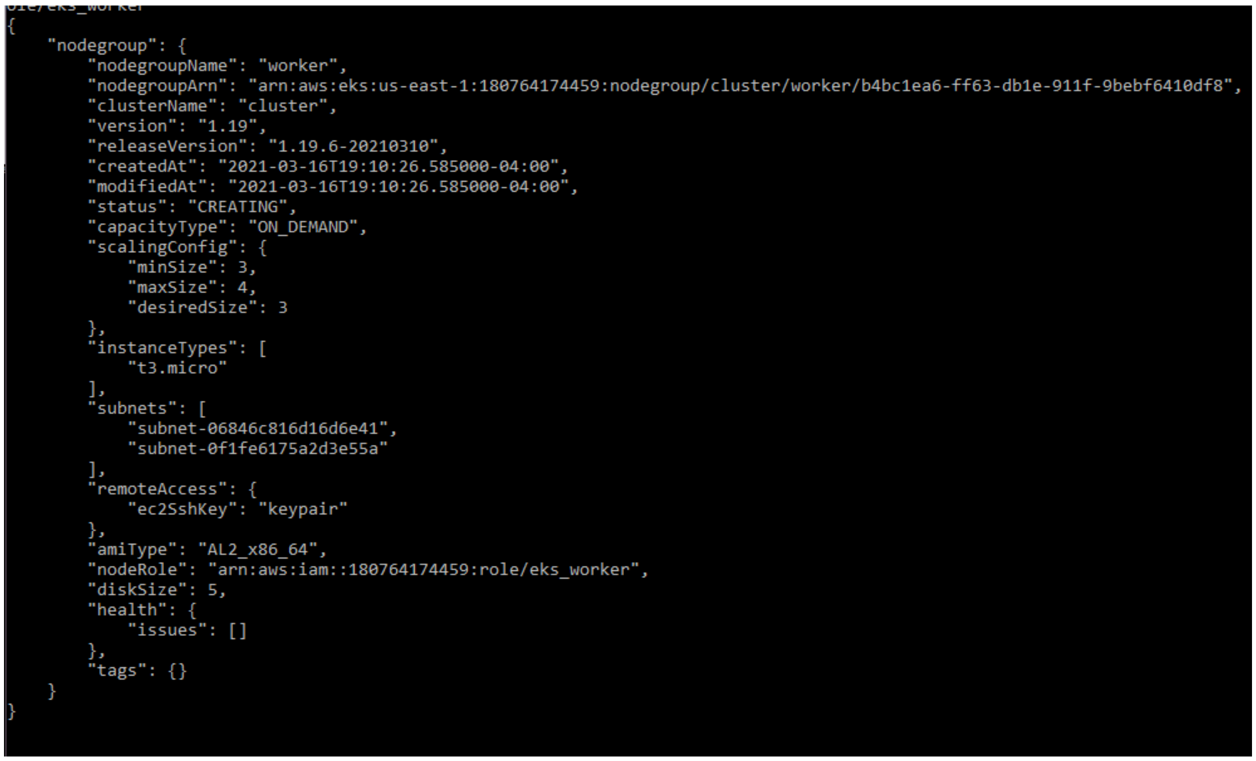
```
D:\School\Spring_2021\CS_381_Cloud_Computing\kubect1>kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   8m46s
```

Creating the worker nodes

Need a role for the worker node too. The step is the same just in the service part of json needing to switch eks.amazonaws.com with ec2.amazonaws.com each worker node is an EC2 instance.

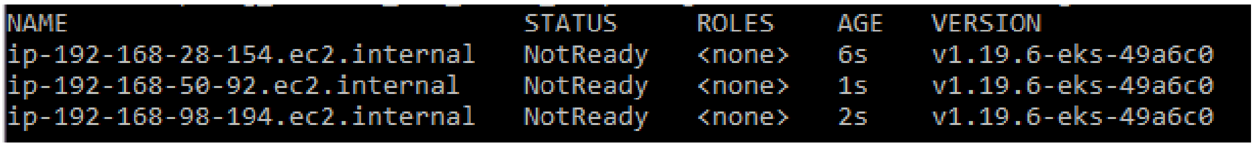
(Worker Node)

>aws eks create-nodegroup --cluster-name cluster --nodegroup-name worker --scaling-config minSize=3,maxSize=4,desiredSize=3 --disk-size 5 --subnets “subnet-06846c816d16d6e41” “subnet-0f1fe6175a2d3e55a” --instance-types t3.micro --ami-type AL2_x86_64 --remote-access ec2SshKey=keypair --node-role arn:aws:iam::180764174459:role/eks_worker



```
ole/eks_worker
{
    "nodegroup": {
        "nodegroupName": "worker",
        "nodegroupArn": "arn:aws:eks:us-east-1:180764174459:nodegroup/cluster/worker/b4bc1ea6-ff63-db1e-911f-9bebf6410df8",
        "clusterName": "cluster",
        "version": "1.19",
        "releaseVersion": "1.19.6-20210310",
        "createdAt": "2021-03-16T19:10:26.585000-04:00",
        "modifiedAt": "2021-03-16T19:10:26.585000-04:00",
        "status": "CREATING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 3,
            "maxSize": 4,
            "desiredSize": 3
        },
        "instanceTypes": [
            "t3.micro"
        ],
        "subnets": [
            "subnet-06846c816d16d6e41",
            "subnet-0f1fe6175a2d3e55a"
        ],
        "remoteAccess": {
            "ec2SshKey": "keypair"
        },
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::180764174459:role/eks_worker",
        "diskSize": 5,
        "health": {
            "issues": []
        },
        "tags": {}
    }
}
```

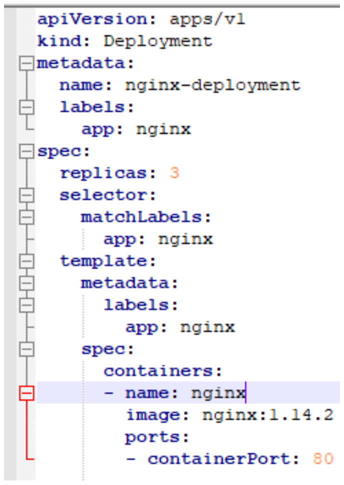
>kubectl get nodes



```
NAME                              STATUS     ROLES    AGE   VERSION
ip-192-168-28-154.ec2.internal    NotReady   <none>   6s    v1.19.6-eks-49a6c0
ip-192-168-50-92.ec2.internal     NotReady   <none>   1s    v1.19.6-eks-49a6c0
ip-192-168-98-194.ec2.internal    NotReady   <none>   2s    v1.19.6-eks-49a6c0
```

**Task 2 Deploy application:**

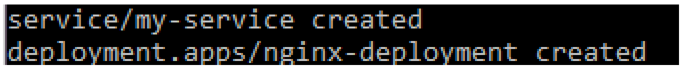
For the application, I use the sample from https://kubernetes.io/docs/concepts/workloads/controllers/deployment/ where it makes three copies.   (need to make sure the terminal is in the correct directory of the file).



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

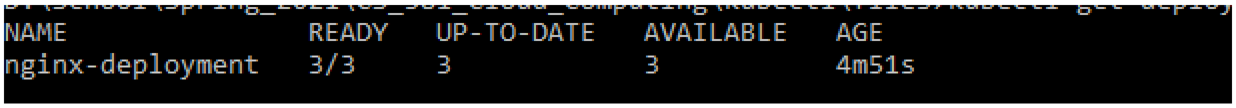
(Deployment)

>kubectl apply -f nginx_deploy.yaml



```
service/my-service created
deployment.apps/nginx-deployment created
```

>kubectl get deployment



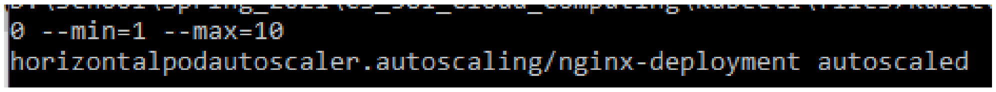
```
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           4m51s
```

3 instance is created

**Step 3 scale the pods (horizontally) :**

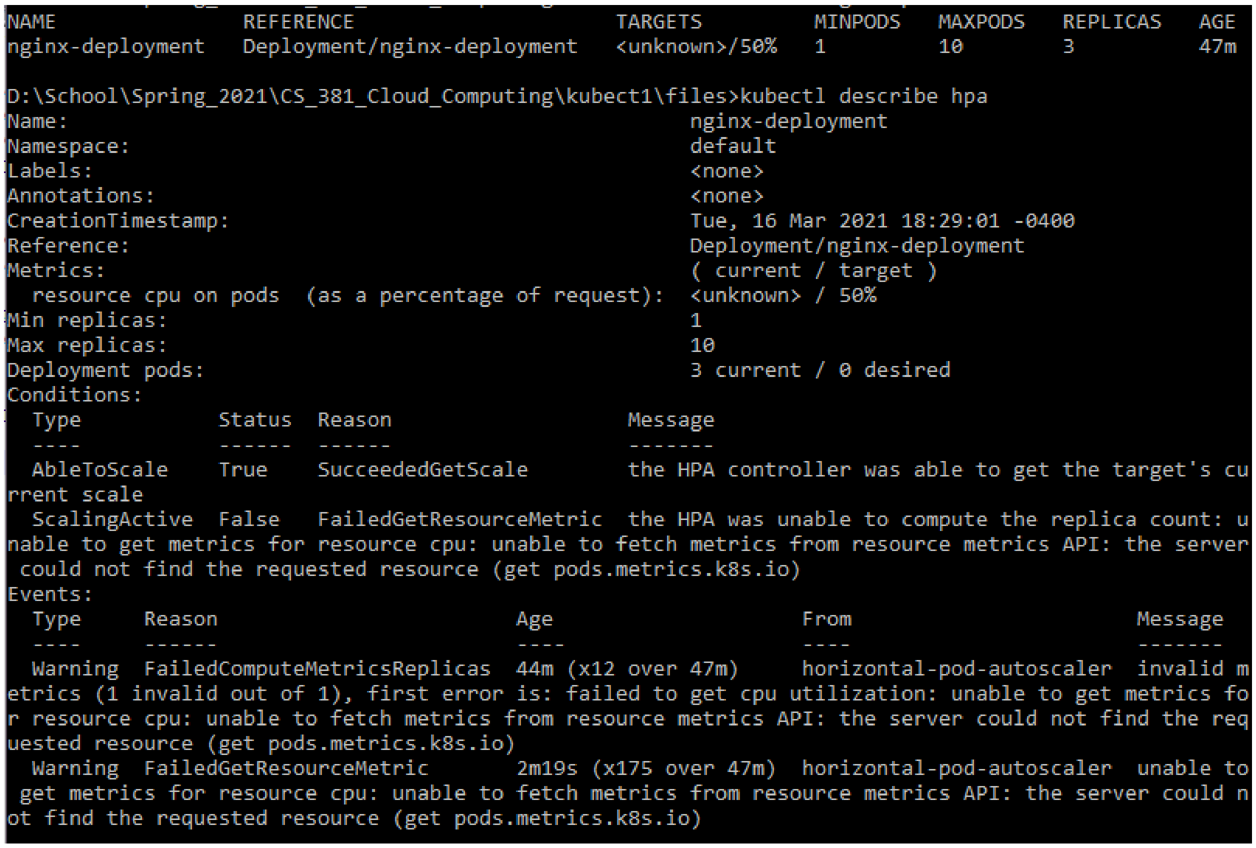
(Scaling up)

>kubectl autoscale deployment nginx-deployment --cpu-percent=50 --min=1 --max=10



```
0 --min=1 --max=10
horizontalpodautoscaler.autoscaling/nginx-deployment autoscaled
```

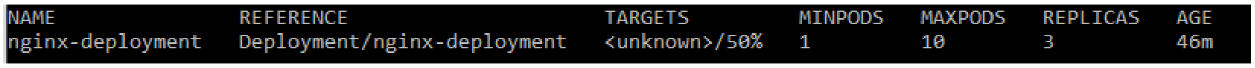
>kubectl describe hpa



```
NAME               REFERENCE                     TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
nginx-deployment   Deployment/nginx-deployment   <unknown>/50%   1         10        3          47m

D:\School\Spring_2021\CS_381_Cloud_Computing\kubectl\files>kubectl describe hpa
Name:                                                  nginx-deployment
Namespace:                                             default
Labels:                                                <none>
Annotations:                                           <none>
CreationTimestamp:                                     Tue, 16 Mar 2021 18:29:01 -0400
Reference:                                             Deployment/nginx-deployment
Metrics:                                               ( current / target )
  resource cpu on pods  (as a percentage of request):  <unknown> / 50%
Min replicas:                                          1
Max replicas:                                          10
Deployment pods:                                       3 current / 0 desired
Conditions:
  Type           Status  Reason                   Message
  ----           ------  ------                   -------
  AbleToScale    True    SucceededGetScale        the HPA controller was able to get the target's cu
rrent scale
  ScalingActive  False   FailedGetResourceMetric  the HPA was unable to compute the replica count: u
nable to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server
 could not find the requested resource (get pods.metrics.k8s.io)
Events:
  Type     Reason                         Age                   From                       Message
  ----     ------                         ----                  ----                       -------
  Warning  FailedComputeMetricsReplicas  44m (x12 over 47m)    horizontal-pod-autoscaler  invalid m
etrics (1 invalid out of 1), first error is: failed to get cpu utilization: unable to get metrics fo
r resource cpu: unable to fetch metrics from resource metrics API: the server could not find the req
uested resource (get pods.metrics.k8s.io)
  Warning  FailedGetResourceMetric       2m19s (x175 over 47m)  horizontal-pod-autoscaler  unable to
 get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server could n
ot find the requested resource (get pods.metrics.k8s.io)
```

>kubectl get hpa



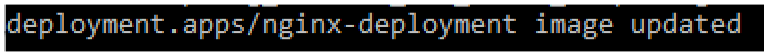
```
NAME               REFERENCE                     TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
nginx-deployment   Deployment/nginx-deployment   <unknown>/50%   1         10        3          46m
```

This shows that there are more instances available. What is the min and max replicas allowed

**Step 4 Update the deployment:**

(Update deployment)

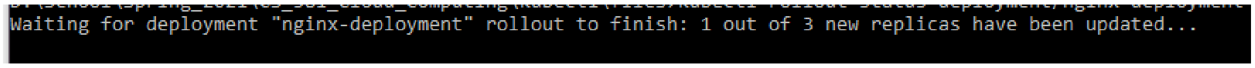
>kubectl set image deployment/nginx-deployment nginx=nginx:1.16.1 --record



```
deployment.apps/nginx-deployment image updated
```

In the deployment file the container image is nginx:1.14.2 show that and now it will update the image to nginx:1.16.1

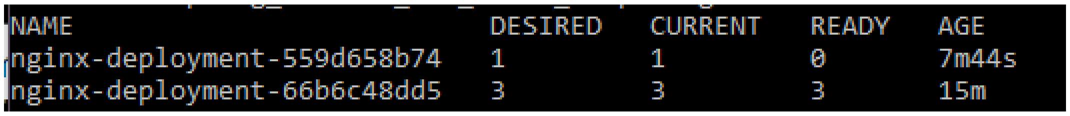
>kubectl rollout status deployment/nginx-deployment



```
Waiting for deployment "nginx-deployment" rollout to finish: 1 out of 3 new replicas have been updated...
```

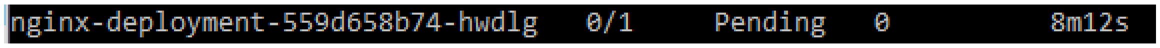
According the the documentation this is fine, it show the status of the update

>kubectl get rs



```
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-559d658b74   1         1         0       7m44s
nginx-deployment-66b6c48dd5   3         3         3       15m
```

>kubectl get pods



```
nginx-deployment-559d658b74-hwdlg   0/1     Pending   0          8m12s
```

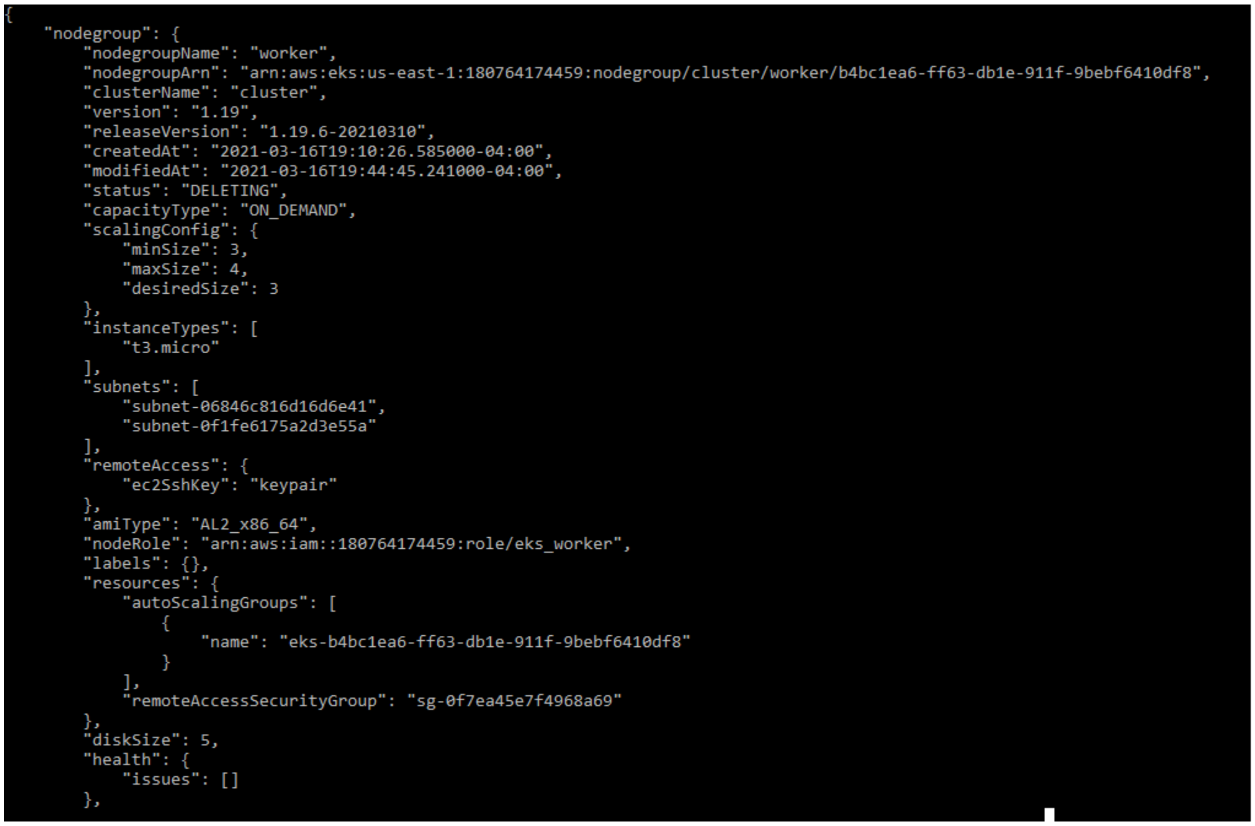
This one got stuck. As you can see when using >"kubectl get rs" there is another deployment. In order to troubleshoot I created multiple which took the resource and deleting them took a while. That is why it is pending, since the max already reached.

**Step 5 Deleting application and Cluster:**

(Deleting Worker Node)

When deleting have to wait for them to completely terminate at each step, where it does not appear in the console.asw.amazon.com

>aws eks delete-nodegroup --cluster-name cluster --nodegroup-name worker



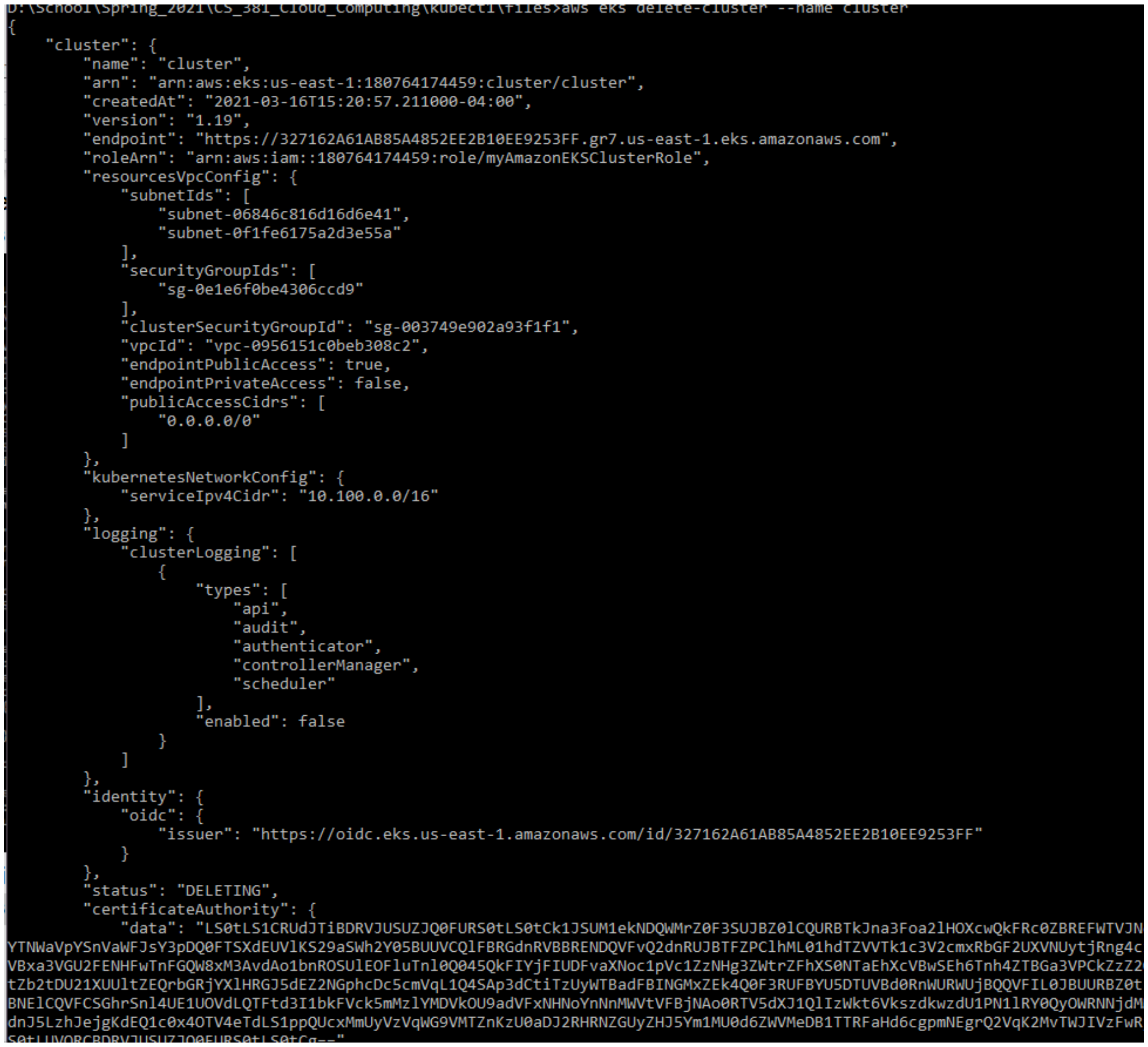
```
{
    "nodegroup": {
        "nodegroupName": "worker",
        "nodegroupArn": "arn:aws:eks:us-east-1:180764174459:nodegroup/cluster/worker/b4bc1ea6-ff63-db1e-911f-9bebf6410df8",
        "clusterName": "cluster",
        "version": "1.19",
        "releaseVersion": "1.19.6-20210310",
        "createdAt": "2021-03-16T19:10:26.585000-04:00",
        "modifiedAt": "2021-03-16T19:44:45.241000-04:00",
        "status": "DELETING",
        "capacityType": "ON_DEMAND",
        "scalingConfig": {
            "minSize": 3,
            "maxSize": 4,
            "desiredSize": 3
        },
        "instanceTypes": [
            "t3.micro"
        ],
        "subnets": [
            "subnet-06846c816d16d6e41",
            "subnet-0f1fe6175a2d3e55a"
        ],
        "remoteAccess": {
            "ec2SshKey": "keypair"
        },
        "amiType": "AL2_x86_64",
        "nodeRole": "arn:aws:iam::180764174459:role/eks_worker",
        "labels": {},
        "resources": {
            "autoScalingGroups": [
                {
                    "name": "eks-b4bc1ea6-ff63-db1e-911f-9bebf6410df8"
                }
            ],
            "remoteAccessSecurityGroup": "sg-0f7ea45e7f4968a69"
        },
        "diskSize": 5,
        "health": {
            "issues": []
        },
```

The worker group can be view by going to the EKS → Cluster → Click on the cluster used →  configuration → compute

When the node group is empty means next step can work else get an error that there is a node group attached

(Deleting Cluster)

>aws eks delete-cluster --name cluster



```
D:\School\Spring_2021\CS_381_Cloud_Computing\kubectl\files>aws eks delete-cluster --name cluster
{
    "cluster": {
        "name": "cluster",
        "arn": "arn:aws:eks:us-east-1:180764174459:cluster/cluster",
        "createdAt": "2021-03-16T15:20:57.211000-04:00",
        "version": "1.19",
        "endpoint": "https://327162A61AB85A4852EE2B10EE9253FF.gr7.us-east-1.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::180764174459:role/myAmazonEKSClusterRole",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-06846c816d16d6e41",
                "subnet-0f1fe6175a2d3e55a"
            ],
            "securityGroupIds": [
                "sg-0e1e6f0be4306ccd9"
            ],
            "clusterSecurityGroupId": "sg-003749e902a93f1f1",
            "vpcId": "vpc-0956151c0beb308c2",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": false,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
                        "api",
                        "audit",
                        "authenticator",
                        "controllerManager",
                        "scheduler"
                    ],
                    "enabled": false
                }
            ]
        },
        "identity": {
            "oidc": {
                "issuer": "https://oidc.eks.us-east-1.amazonaws.com/id/327162A61AB85A4852EE2B10EE9253FF"
            }
        },
        "status": "DELETING",
        "certificateAuthority": {
            "data": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJN
YTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTFZPClhML01hdTZVVTk1c3V2cmxRbGF2UXVNUytjRng4c
VBxa3VGU2FENHFwTnFGQW8xM3AvdAo1bnROSUlEOFluTnl0Q045QkFIYjFIUDFvaXNoc1pVc1ZzNHg3ZWtrZFhXS0NTaEhXcVBwSEh6Tnh4ZTBGa3VPCkZzZ2
tZb2tDU21XUUltZEQrbGRjYXlHRGJ5dEZ2NGphcDc5cmVqL1Q4SAp3dCtiTzUyWTBadFBINGMxZEk4Q0F3RUFBYU5DTUVBd0RnWURWUjBQQVFIL0JBUURBZ0t
BNElCQVFCSGhrSnl4UE1UOVdLQTFtd3I1bkFVck5mMzlYMDVkOU9adVFxNHNoYnNnMWVtVFBjNAo0RTV5dXJ1QlIzWkt6VkszdkwzdU1PN1lRY0QyOWRNNjdM
dnJ5LzhJejgKdEQ1c0x4OTV4eTdLS1ppQUcxMmUyVzVqWG9VMTZnKzU0aDJ2RHRNZGUyZHJ5Ym1MU0d6ZWVMeDB1TTRFaHd6cgpmNEgrQ2VqK2MvTWJIVzFwR
S0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg=="
```

This will delete the Cluster.

When deleting the node group also deleted the EC2 instance

(Optional for Deletion instead of deleting application and cluster these can be deleted too but can be kept for next time)

For the next step, have to delete the EC2 instance the NAT Gateway, network interfaces, internet gateway, than you can manually delete the VPC

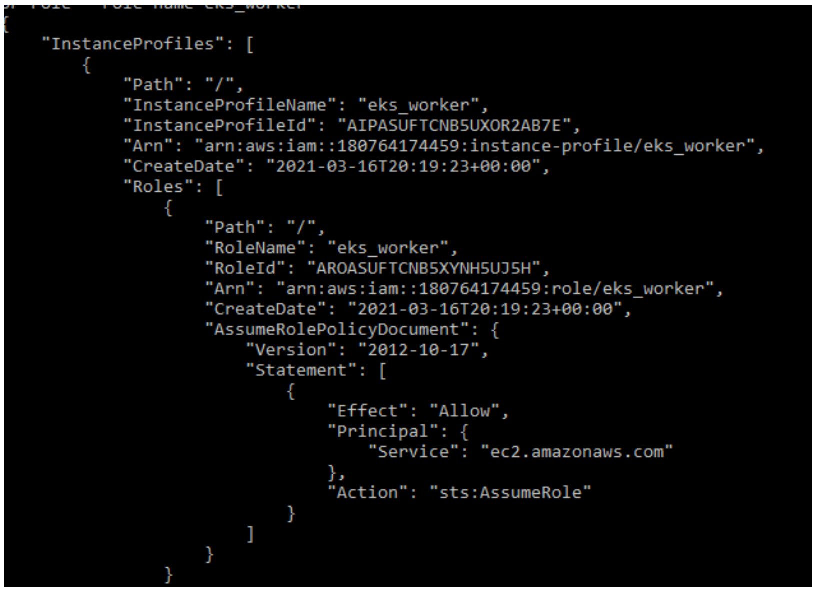
> aws ec2 describe-subnets

This will give you a list of all subnet

And using asw ec2 to delete them one by one, I used the amazon console as it was easier to view and time saving.

You cannot delete the role until the EC2 instance that use it is gone

>aws iam list-instance-profiles-for-role --role-name myAmazonEKSClusterRole



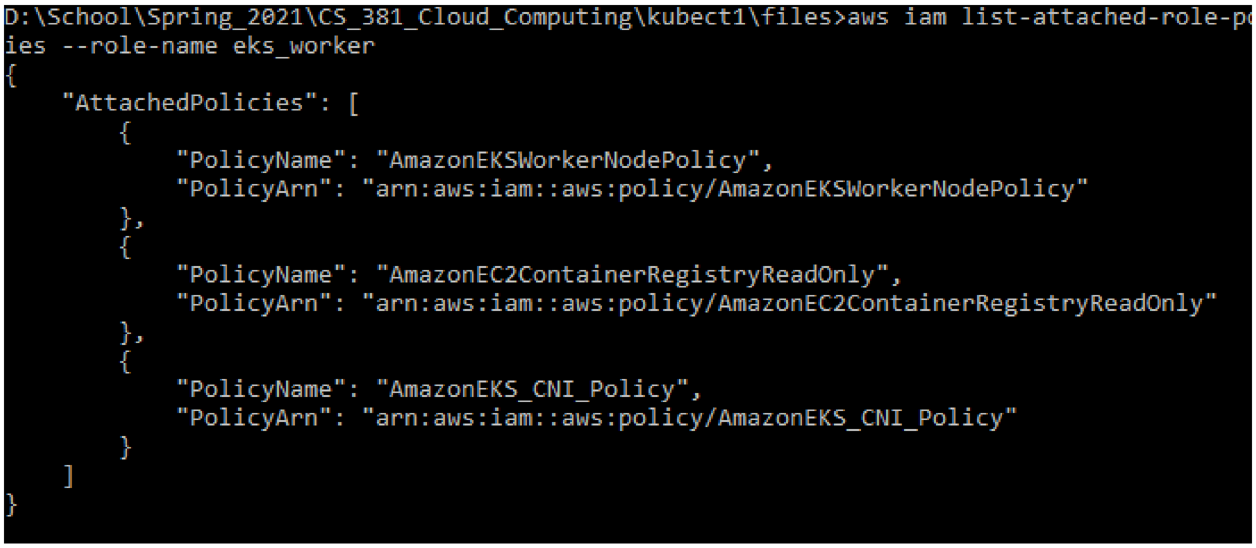
```
r-role --role-name eks_worker
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "eks_worker",
            "InstanceProfileId": "AIPASUFTCNB5UXOR2AB7E",
            "Arn": "arn:aws:iam::180764174459:instance-profile/eks_worker",
            "CreateDate": "2021-03-16T20:19:23+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "eks_worker",
                    "RoleId": "AROASUFTCNB5XYNH5UJ5H",
                    "Arn": "arn:aws:iam::180764174459:role/eks_worker",
                    "CreateDate": "2021-03-16T20:19:23+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
```

Will gave list of instance attached

>aws iam remove-role-from-instance-profile --instance-profile-name eks_worker --role-name eks_worker

Will not return anything

>aws iam list-attached-role-policies --role-name eks_worker

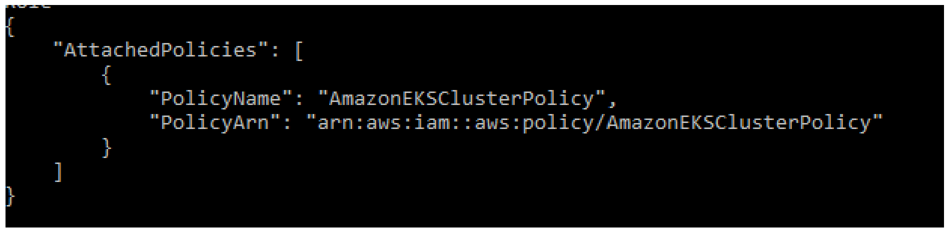


```
D:\School\Spring_2021\CS_381_Cloud_Computing\kubect1\files>aws iam list-attached-role-po
ies --role-name eks_worker
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEKSWorkerNodePolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
        },
        {
            "PolicyName": "AmazonEC2ContainerRegistryReadOnly",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
        },
        {
            "PolicyName": "AmazonEKS_CNI_Policy",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
        }
    ]
}
```

```
>aws iam detach-role-policy --role-name eks_worker --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
>aws iam detach-role-policy --role-name eks_worker --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
>aws iam detach-role-policy --role-name eks_worker --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
>aws iam delete-role --role-name eks_worker
```

All four will not return anything

```
>aws iam list-attached-role-policies --role-name myAmazonEKSClusterRole
```



```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEKSClusterPolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
        }
    ]
}
```

```
>aws iam detach-role-policy --role-name myAmazonEKSClusterRole --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
>aws iam delete-role --role-name myAmazonEKSClusterRole
```

**Variables created**

| | |
|---|---|
| Cluster Role name: | myAmazonEKSClusterRole |
| Worker role name: | eks_worker |
| SecurityGroups: | sg-0e1e6f0be4306ccd9 |
| SubnetIds: | subnet-06846c816d16d6e41, |
| | subnet-0f1fe6175a2d3e55a, |
| VpcId: | vpc-0956151c0beb308c2 |

Master node name (cluster name): cluster
Worker node name : worker

| | |
|---|---|
| Deployment: | nginx-deployment |