

Kubernetes Task-2

Task Description:

Create the K8s EKS,further you have to do the deployment of the Nginx application and access the application outside the cluster.

Techstacks needs to be used :

- AWS EKS
- EKSCtl
- Kubectl

```
Install the latest PowerShell for new features and improvements! https://aka.ms/Windows
PS C:\WINDOWS\system32> curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.34.2/2025-11-13/bin/windows/amd64/kubectl.exe.sha256
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload Total Spent   Left  Speed
100  77 100  77  0      0   29      0 0:00:02  0:00:02  --:--   29
PS C:\WINDOWS\system32> Get-FileHash kubectl.exe
Resolve-Path : Cannot find path 'C:\WINDOWS\system32\kubectl.exe' because it does not exist.
At C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\Microsoft.PowerShell.Utility\Microsoft.PowerShell.Utility.psm1:110 char:36
+             $pathsToProcess += Resolve-Path $path | Foreach-Objec ...
+             ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\WINDOWS\system32\kubectl.exe:String) [Resolve-Path], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.ResolvePathCommand

PS C:\WINDOWS\system32> curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.33.5/2025-11-13/bin/windows/amd64/kubectl.exe
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload Total Spent   Left  Speed
100 58.8M 100 58.8M  0      0 590k      0 0:01:42  0:01:42  --:-- 2430k
PS C:\WINDOWS\system32> Get-FileHash kubectl.exe
Algorithm      Hash                                         Path
-----          ----                                         -----
SHA256         16826333B05D0757F639EB3B4303951550D2B628F290E1722368D925116E9D62
PS C:\WINDOWS\system32>
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install eksctl
Chocolatey v2.6.0
Installing the following packages:
eksctl
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading eksctl 0.220.0... 100%
eksctl v0.220.0 [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 scripts/[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/ekscctl-io/ekscctl/releases/download/v0.220.0/eksctl_Windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\sheer\AppData\Local\Temp\chocolatey\eksctl\0.220.0\eksctl_Windows_amd64.zip (35.76 MB).
Download of eksctl_Windows_amd64.zip (35.76 MB) completed.
Hashes match.
Extracting C:\Users\sheer\AppData\Local\Temp\chocolatey\eksctl\0.220.0\eksctl_Windows_amd64.zip to C:\ProgramData\chocolatey\lib\eksctl\tools...
C:\ProgramData\chocolatey\lib\eksctl\tools
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
ShimGen has successfully created a shim for eksctl.exe
The install of eksctl was successful.
  Deployed to 'C:\ProgramData\chocolatey\lib\eksctl\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32> eksctl --version
Error: unknown flag: --version
PS C:\WINDOWS\system32> eksctl version
0.220.0
```

The screenshot shows the AWS IAM User Details page for a user named 'Harsh'. The top navigation bar includes 'Search' and 'Account ID: 5368-3256-1054'. The left sidebar has sections for 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Root access management', 'Temporary delegation requests'), and 'Access reports' (with 'Access Analyzer'). The main content area is titled 'Harsh Info' and contains a 'Summary' section with details like ARN (arn:aws:iam::336832361054:user/Harsh), creation date (October 17, 2025, 18:52 UTC+05:30), last console sign-in (1 month ago), and access key information (AKIAU43GF6JPJXEFCINU - Active, Never used. Created today). Below the summary is a 'Permissions' tab showing one policy attached: 'AdministratorAccess' (AWS managed - job function, Directly). There are buttons for 'Remove' and 'Add permissions'.

The screenshot shows the AWS IAM Access Keys page for the same user 'Harsh'. The top navigation bar and left sidebar are identical to the previous screenshot. The main content area is titled 'Access keys (1)' and shows a single access key entry: 'AKIAU43GF6JPJXEFCINU'. The key has a status of 'Active', was created 'Now', and has no last used information. There is a 'Create access key' button and an 'Actions' dropdown.

```
sheer@SheershPC MINGW64 ~ (main)
$ aws configure
AWS Access Key ID [*****BKN7]: AKIAU43GF6JPJXEFCINU
AWS Secret Access Key [*****LE9z]: RxWg50fwLpxfVATufbAqn2s4sB+9waeR6N
273F4Y
Default region name [us-east-2]: us-east-2
Default output format [json]:
```

```
sheer@SheershPC MINGW64 ~ (main)
$ |
```

```

sheer@SheersHPC MINGW64 ~ (main)
$ eksctl create cluster --name demo-cluster-eks --region us-east-2 --node-type t3.micro
2025-12-05 20:25:02 [0] eksctl version v220.0
2025-12-05 20:25:02 [0] eksctl managed-nodegroups version v2.2
2025-12-05 20:25:02 [0] setting availability zones to [us-east-2c - public:192.168.0.0/19 private:192.168.96.0/19
2025-12-05 20:25:02 [0] subnets for us-east-2c - public:192.168.0.0/19 private:192.168.96.0/19
2025-12-05 20:25:02 [0] subnets for us-east-2a - public:192.168.32.0/19 private:192.168.128.0/19
2025-12-05 20:25:02 [0] subnets for us-east-2b - public:192.168.64.0/19 private:192.168.160.0/19
2025-12-05 20:25:02 [0] nodegroup "ng-Sa0C83eb" will use "AmazonLinux2023.7.32"
2025-12-05 20:25:02 [1] Auto Mode will be enabled by default in an upcoming release of eksctl. This means managed node groups and managed networking add-ons will no longer be created by default. To maintain current behavior, explicitly set `autoModeConfig.enabled: false` in your cluster configuration. Learn more: https://eksctl.io/usage/auto-mode/
2025-12-05 20:25:02 [0] creating Kubernetes version 1.32
2025-12-05 20:25:02 [0] creating EKS cluster "demo-cluster-eks" in "us-east-2" region with managed nodes
2025-12-05 20:25:02 [0] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=demo-cluster-eks' in "us-east-2"
2025-12-05 20:25:02 [0] CloudWatch Metrics will be enabled by default for cluster "demo-cluster-eks". You can enable it with 'eksctl utils update-cluster-logging --enable-types=[SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)] --region=us-east-2 --cluster=demo-cluster-eks'
2025-12-05 20:25:02 [0] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-12-05 20:25:02 [0] sequential sub-tasks: [
    2 sequential sub-tasks: [
        1 task: { create addons },
        wait for control plane to become ready,
    ],
    create managed nodegroup "ng-Sa0C83eb",
]
}
2025-12-05 20:25:02 [0] building cluster stack "eksctl-demo-cluster-eks-cluster"
2025-12-05 20:25:04 [0] deploying stack "eksctl-demo-cluster-eks-cluster"
2025-12-05 20:25:34 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-cluster"
2025-12-05 20:25:05 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:06 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:07 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:08 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:09 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:10 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:11 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:12 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:13 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:15 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:16 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup"
2025-12-05 20:25:17 [0] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update a
daddon'
2025-12-05 20:25:24 [0] creating addon: vpc-cni
2025-12-05 20:25:24 [0] successfully created addon: vpc-cni
2025-12-05 20:25:25 [0] creating addon: kube-proxy
2025-12-05 20:25:27 [0] successfully created addon: kube-proxy
2025-12-05 20:25:28 [0] creating addon: coredns
2025-12-05 20:25:29 [0] successfully created addon: coredns
2025-12-05 20:25:30 [0] building nodegroup stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:37 [0] deploying stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:38 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:39 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:40 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:41 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:42 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:43 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:44 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:45 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:46 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:47 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:48 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:49 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:50 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:51 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:52 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:53 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:54 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:55 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:56 [0] waiting for CloudFormation stack "eksctl-demo-cluster-eks-nodegroup-ng-Sa0C83eb"
2025-12-05 20:25:57 [0] waiting for the control plane to become ready
2025-12-05 20:25:57 [0] saved kubeconfig as "C:\Users\sheer\kube\config"
2025-12-05 20:25:58 [0] no tasks
2025-12-05 20:25:58 [0] EKS Cluster resources for "demo-cluster-eks" have been created
2025-12-05 20:25:58 [0] nodegroup "ng-Sa0C83eb" has 2 node(s)
2025-12-05 20:25:58 [0] node "ip-192-168-19-45.us-east-2.compute.internal" is ready
2025-12-05 20:25:58 [0] node "ip-192-168-91-134.us-east-2.compute.internal" is ready
2025-12-05 20:25:58 [0] both nodes at https://ip-192-168-19-45.us-east-2.eksctl.io are ready in ng-Sa0C83eb
2025-12-05 20:25:59 [0] nodegroup "ng-Sa0C83eb" has 2 node(s)
2025-12-05 20:25:59 [0] node "ip-192-168-19-45.us-east-2.compute.internal" is ready
2025-12-05 20:25:59 [0] node "ip-192-168-91-134.us-east-2.compute.internal" is ready
2025-12-05 20:25:59 [0] created managed nodegroup(s) in cluster "demo-cluster-eks"
2025-12-05 20:25:59 [0] adding add-on services
2025-12-05 20:39:01 [0] successfully created addon: metrics-server
2025-12-05 20:39:08 [0] kubectl command should work with 'C:\Users\sheer\kube\config', try 'kubectl get nodes'
2025-12-05 20:39:08 [0] EKS cluster "demo-cluster-eks" in "us-east-2" region is ready
2025-12-05 20:39:08 [0] 

```

```

sheer@SheersHPC MINGW64 ~ (main)
$ kubectl get pos
error: the server doesn't have a resource type "pos"

sheer@SheersHPC MINGW64 ~ (main)
$ kubectl get pods
No resources found in default namespace.

sheer@SheersHPC MINGW64 ~ (main)
$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-192-168-19-45.us-east-2.compute.internal   Ready        2m31s   v1.32.9-eks-ecaa3a6
ip-192-168-91-134.us-east-2.compute.internal   Ready        2m31s   v1.32.9-eks-ecaa3a6

```

```

Kubernetes > EKS > Deployment.yml > spec > template > spec
          io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: nginx-deployment
10   template:
11     metadata:
12       labels:
13         app: nginx-deployment
14   spec:
15     containers:
16       - name: nginx-deployment
17         image: nginx:alpine
18       ports:
19         - containerPort: 80
20

```

```

▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> eksctl get cluster
NAME      REGION      EKSCTL CREATED
demo-eks  us-east-2  True
▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-39-45.us-east-2.compute.internal  Ready   <none>  2m12s v1.32.9-eks-ecaa3a6
ip-192-168-82-79.us-east-2.compute.internal  Ready   <none>  2m12s v1.32.9-eks-ecaa3a6
▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> ls

Directory: C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS

Mode          LastWriteTime    Length Name
----          -----          ---- 
-a---  06-12-2025 15:13          368 Deployment.yml

▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl apply -f deployment.yml
deployment.apps/nginx-deployment created
▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get kube-system
error: the server doesn't have a resource type "kube-system"
▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get pods
NAME          READY  STATUS  RESTARTS  AGE
nginx-deployment-6d98745f75-26crp  1/1   Running   0   35s
nginx-deployment-6d98745f75-4hzfw  1/1   Running   0   35s
▶ PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get pods -n kube-system
NAME          READY  STATUS  RESTARTS  AGE
aws-node-pmdxn  2/2   Running   0   3m48s
aws-node-stvdz  2/2   Running   0   3m48s
coredns-64dfc67578-68kvr  1/1   Running   0   7m20s
coredns-64dfc67578-sj86h  1/1   Running   0   7m20s
kube-proxy-125cw  1/1   Running   0   3m48s
kube-proxy-zfdzf  1/1   Running   0   3m48s
metrics-server-5c9ffc7c9-86nbf  1/1   Running   0   2m51s
metrics-server-5c9ffc7c9-npx86  1/1   Running   0   2m51s

```

```

Kubernetes > EKS > service.yml > {} spec > {} ports > {} 0 > protocol
    io.k8s.api.core.v1.Service (v1@service.json)
1  apiVersion: v1
2  kind: Service
3  metadata:
4      name: nginx-service
5  spec:
6      type: LoadBalancer
7      selector:
8          app: nginx-deployment
9      ports:
10         - port: 90
11             targetPort: 80
12             protocol: TCP

```

```

PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl apply -f service.yml
service/nginx-service created
● PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP
kubernetes     ClusterIP  10.100.0.1   <none>
nginx-service  LoadBalancer 10.100.82.246 aa6a98cd6cbef41fba07ae0c913c5a57-667152870.us-east-2.elb.amazonaws.com
PORT(S)        AGE
443/TCP       83m
90:31203/TCP  14s
● PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> curl http://aa6a98cd6cbef41fba07ae0c913c5a57-667152870.us-east-2.elb.amazonaws.com:90
curl: (6) Could not resolve host: aa6a98cd6cbef41fba07ae0c913c5a57-667152870.us-east-2.elb.amazonaws.com
● PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> kubectl get endpoints
NAME           ENDPOINTS   AGE
kubernetes     192.168.114.155:443,192.168.172.26:443  84m
nginx-service  192.168.61.199:80,192.168.90.211:80  72s
● PS C:\Users\sheer\Documents\kubernetes\Kubernetes\EKS> curl http://aa6a98cd6cbef41fba07ae0c913c5a57-667152870.us-east-2.elb.amazonaws.com:90
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p><a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```