

WEEK 7 ASSIGNMENT 7 - AWS EKS

Create Cluster using eksctl

```
C:\Users\sudeev.divakar>eksctl create cluster --name sudeev-cluster --version 1.28 --nodes=1 --node-type=t2.small --region ap-south-1
2025-02-28 15:27:43 [I] eksctl version 0.204.0
2025-02-28 15:27:43 [I] using region ap-south-1
2025-02-28 15:27:43 [I] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.small
2025-02-28 15:27:43 [I] setting availability zones to [ap-south-1a ap-south-1b]
2025-02-28 15:27:43 [I] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2025-02-28 15:27:43 [I] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2025-02-28 15:27:43 [I] nodegroup "ng-883c7509" will use "" [AmazonLinux2/1.28]
2025-02-28 15:27:43 [I] using Kubernetes version 1.28
2025-02-28 15:27:43 [I] creating EKS cluster "sudeev-cluster" in "ap-south-1" region with managed nodes
2025-02-28 15:27:43 [I] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-02-28 15:27:43 [I] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=sudeev-cluster'
2025-02-28 15:27:43 [I] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "sudeev-cluster" in "ap-south-1"
2025-02-28 15:27:43 [I] CloudWatch logging will not be enabled for cluster "sudeev-cluster" in "ap-south-1"
2025-02-28 15:27:43 [I] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=sudeev-cluster'
2025-02-28 15:27:43 [I] default add-ons metrics-server, vpc-cni, kube-proxy, coreDNS were not specified, will install them as EKS add-ons
2025-02-28 15:27:43 [I]
2 sequential tasks: { create cluster control plane "sudeev-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons }
```

```
2025-02-28 15:36:48 [I] creating addon: kube-proxy
2025-02-28 15:36:48 [I] successfully created addon: kube-proxy
2025-02-28 15:36:49 [I] creating addon: coreDNS
2025-02-28 15:36:49 [I] successfully created addon: coreDNS
2025-02-28 15:38:50 [I] building managed nodegroup stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:38:50 [I] deploying stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:38:50 [I] waiting for CloudFormation stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:39:21 [I] waiting for CloudFormation stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:40:16 [I] waiting for CloudFormation stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:41:56 [I] waiting for CloudFormation stack "eksctl-sudeev-cluster-nodegroup-ng-883c7509"
2025-02-28 15:41:56 [I] waiting for the control plane to become ready
2025-02-28 15:41:58 [I] saved kubeconfig as "C:\\Users\\sudeev.divakar\\.kube\\config"
2025-02-28 15:41:58 [I] no tasks
2025-02-28 15:41:58 [I] all EKS cluster resources for "sudeev-cluster" have been created
2025-02-28 15:41:58 [I] nodegroup "ng-883c7509" has 1 node(s)
2025-02-28 15:41:58 [I] node "ip-192-168-29-93.ap-south-1.compute.internal" is ready
2025-02-28 15:41:58 [I] waiting for at least 1 node(s) to become ready in "ng-883c7509"
2025-02-28 15:41:58 [I] nodegroup "ng-883c7509" has 1 node(s)
2025-02-28 15:41:58 [I] node "ip-192-168-29-93.ap-south-1.compute.internal" is ready
2025-02-28 15:41:58 [I] created 1 managed nodegroup(s) in cluster "sudeev-cluster"
2025-02-28 15:42:02 [I] kubectrl command should work with "C:\\Users\\sudeev.divakar\\.kube\\config", try 'kubectrl get nodes'
2025-02-28 15:42:02 [I] EKS cluster "sudeev-cluster" in "ap-south-1" region is ready
```

```
C:\Users\sudeev.divakar>kubectrl get nodes
NAME                                     STATUS  ROLES  AGE   VERSION
ip-192-168-29-93.ap-south-1.compute.internal Ready  <none> 2m45s v1.28.15-eks-aec579
```

```
C:\Users\sudeev.divakar>
```

Simple Hello World Spring Boot Application

The screenshot shows an IDE with a project named 'helloworld'. The project structure is as follows:

- helloworld (C:\Users\sudeev.divakar\Downloads\helloworld)
 - idea
 - mvn
 - src
 - main
 - java
 - com.helloworld
 - controller
 - Controller.java

The 'Controller.java' file contains the following code:

```
1 package com.helloworld.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class Controller {
8     @GetMapping("hello")
9     public String hello() {
10         return "Hello World!";
11     }
12 }
13
```

The 'Run' configuration is set to 'HelloworldApplication'. The console output shows the following logs:

```
2025-02-28T15:49:44.419+05:30 INFO 22968 --- [helloworld] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext
2025-02-28T15:49:45.119+05:30 INFO 22968 --- [helloworld] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080
2025-02-28T15:49:45.133+05:30 INFO 22968 --- [helloworld] [main] com.helloworld.HelloworldApplication : Started HelloworldApplication
2025-02-28T15:50:31.473+05:30 INFO 22968 --- [helloworld] [nio-8080-exec-1] o.a.c.c.f.Tomcat.[localhost].[/] : Initializing Spring DispatcherServlet
2025-02-28T15:50:31.473+05:30 INFO 22968 --- [helloworld] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-02-28T15:50:31.475+05:30 INFO 22968 --- [helloworld] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization
2025-02-28T15:50:31.517+05:30 WARN 22968 --- [helloworld] [nio-8080-exec-1] w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerException: Method hello() on class com.helloworld.controller.Controller does not exist or no arguments were passed]
```

Create AWS ECR repository

```
C:\Users\sudeev.divakar>aws ecr create-repository --repository-name hello_world_application

CreateRepository

repository

  createdAt      2025-02-28T15:54:19.253000+05:30
  imageTagMutability  MUTABLE
  registryId     692859932788
  repositoryArn  arn:aws:ecr:ap-south-1:692859932788:repository/hello_world_application
  repositoryName hello_world_application
  repositoryUri  692859932788.dkr.ecr.ap-south-1.amazonaws.com/hello_world_application

encryptionConfiguration

  encryptionType  AES256

imageScanningConfiguration

  scanOnPush  False

C:\Users\sudeev.divakar>aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 692859932788.dkr.ecr.ap-south-1.amazonaws.com
Login Succeeded
```

Dockerize application and push onto AWS ECR

The screenshot shows an IDE with a project named 'helloworld' and a file 'Controller.java'. The terminal output shows the following commands and results:

```
PS C:\Users\sudeev.divakar\Downloads\helloworld> docker tag hello_world_application:latest 692859932788.dkr.ecr.ap-south-1.amazonaws.com/hello_world_application:latest
PS C:\Users\sudeev.divakar\Downloads\helloworld> docker push 692859932788.dkr.ecr.ap-south-1.amazonaws.com/hello_world_application:latest
The push refers to repository [692859932788.dkr.ecr.ap-south-1.amazonaws.com/hello_world_application]
38a980f2cc8a: Pushed
c0b5d50ff6d8: Pushed
c7db8ad712fb: Pushed
de849f1cfbe6: Pushed
a2ea9f69ac8a: Pushed
a7203ca35e75: Pushing [=====] 127.9MB/187.5MB
```

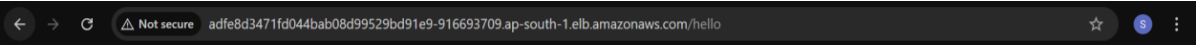
```
PS C:\Users\sudeev.divakar\Downloads\helloworld> kubectl apply -f deployment.yaml
deployment.apps/hello-world-application created
PS C:\Users\sudeev.divakar\Downloads\helloworld> kubectl apply -f service.yaml
service/hello-world-service created
```

```
PS C:\Users\sudeev.divakar\Downloads\helloworld> kubectl get pods

NAME                                READY   STATUS    RESTARTS   AGE
hello-world-application-67db74dddb-cpwwb  1/1     Running   0           14s
hello-world-application-67db74dddb-vflbb  1/1     Running   8 (6m8s ago)  17m
```

```
PS C:\Users\sudeev.divakar\Downloads\helloworld> kubectl get svc

NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
hello-world-service  LoadBalancer  10.100.50.179   adfe8d3471fd044bab08d99529bd91e9-916693709.ap-south-1.elb.amazonaws.com  80:30099/TCP    18m
kubernetes           ClusterIP     10.100.0.1      <none>           443/TCP          65m
```



Hello World