

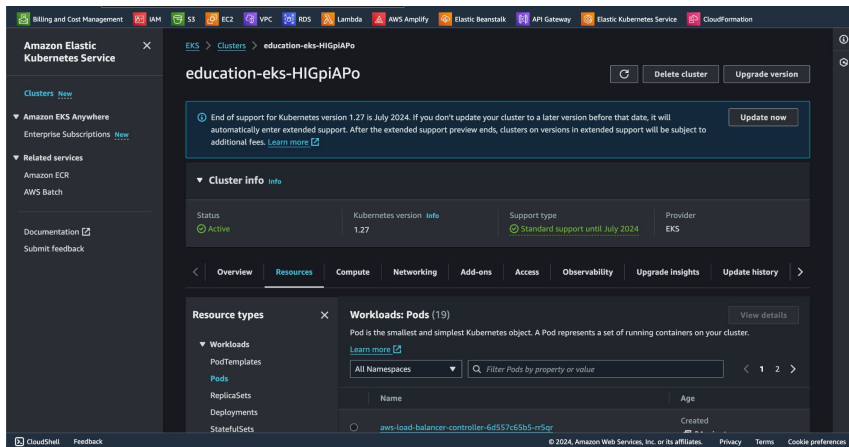
# 세미프로젝트

2조

팀장: 송동훈

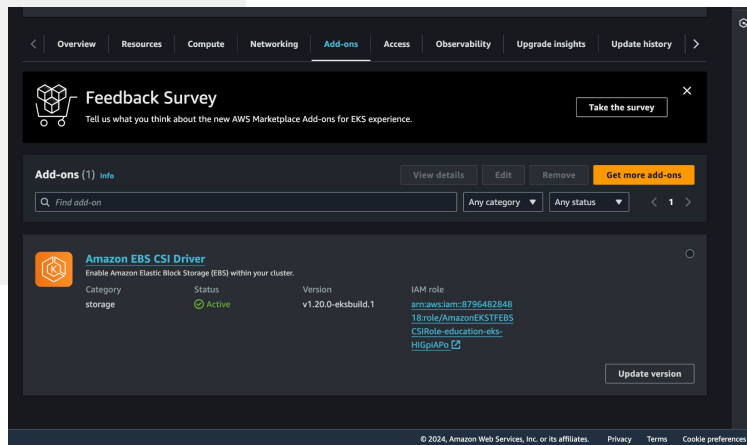
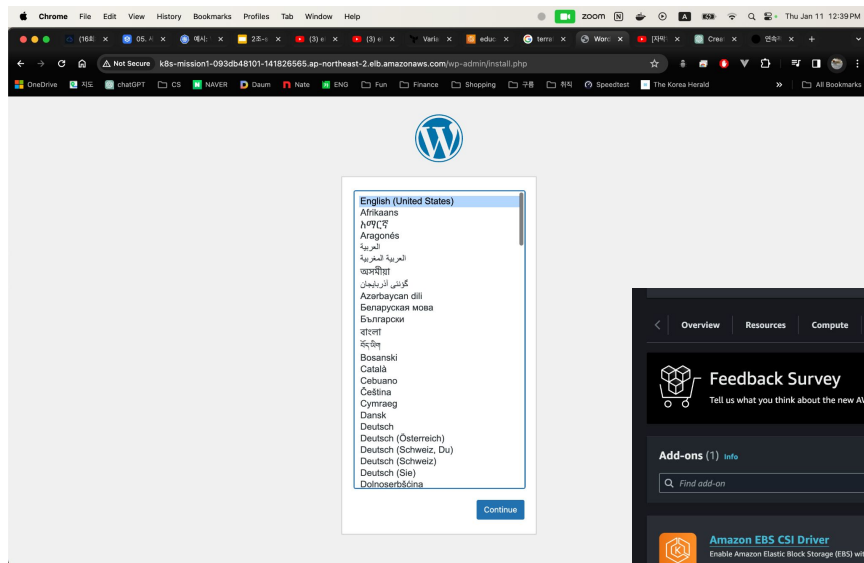
팀원: 박지원, 임종훈,  
최민규

# Step 0: EKS 클러스터 구성



출처: <https://developer.hashicorp.com/terraform/tutorials/kubernetes/eks>  
<https://docs.aws.amazon.com/eks/latest/userguide/aws-load-balancer-controller.html>

# Step 1: Stateful 애플리케이션 배포



## Step 2-1: Wordpress app 구성 (Liveness probe)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  namespace: assignment
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: frontend
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
      containers:
        - image: wordpress:6.2.1-apache
          name: wordpress
          env:
```

```

Name: wordpress-54975f49c7-gf9wj
Namespace: assignment
Priority: 0
Service Account: default
Node: ip-10-0-1-253.ap-northeast-2.compute.internal/10.0.1.253
Start Time: Thu, 11 Jan 2024 12:10:37 +0900
Labels: app=wordpress
        pod-template-hash=54975f49c7
        tier=frontend
Annotations: <none>
Status: Running
IP: 10.0.1.64
IPs:
  IP: 10.0.1.64
Controlled By: ReplicaSet/wordpress-54975f49c7
Containers:
  wordpress:
    Container ID: containerd://3af6535b7c695f348e17ee8a5c89d0ce62c720f31872fcb681ece2b03b02ec5
    Image: wordpress:6.2.1-apache
    Image ID: docker.io/library/wordpress@sha256:f3b8d54fe9b80e88255121a46933ca1961fb9f3f39e69
    cdb7f019a5bbd
    Port: 80/TCP
    Host Port: 0/TCP
    State: Running
      Started: Thu, 11 Jan 2024 12:11:14 +0900
    Last State: Terminated
      Reason: Completed
      Exit Code: 0
      Started: Thu, 11 Jan 2024 12:11:03 +0900
      Finished: Thu, 11 Jan 2024 12:11:14 +0900
    Ready: True
    Restart Count: 1
    Liveness: http-get http://:80/ delay=3s timeout=1s period=3s #success=1 #failure=3

```

## Step 2-1: Wordpress app 구성 (HPA)

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: wp-hpa
  namespace: assignment
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: StatefulSet
    name: wordpress
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 30
```

```
x dhs@mchoon ~/k_digital/assignment/05/src main ± kubectl run -i --tty load-generator --rm --image=busybox:1.28 --restart=Never -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://k8s-mission1-093db48101-1204825192.ap-northeast-2.elb.amazonaws.com; done"
```

```
metrics-server 1/1 1 1 142m
dhs@mchoon ~/k_digital/assignment/05/src main ± kubectl
NAME READY STATUS RESTARTS AGE
wordpress-0 1/1 Running 0 5m3s
wordpress-1 1/1 Running 0 4m50s
wordpress-mysql-0 1/1 Running 0 5m3s
wordpress-mysql-1 1/1 Running 0 4m52s
wordpress-2 0/1 Pending 0 0s
wordpress-2 0/1 Pending 0 4s
wordpress-2 0/1 ContainerCreating 0 4s
wordpress-2 1/1 Running 0 9s
wordpress-2 0/1 Pending 0 0s
wordpress-3 0/1 Pending 0 4s
wordpress-3 0/1 ContainerCreating 0 4s
wordpress-3 1/1 Running 0 11s
wordpress-4 0/1 Pending 0 0s
wordpress-4 0/1 Pending 0 4s
wordpress-4 0/1 ContainerCreating 0 4s
wordpress-4 1/1 Running 0 10s
wordpress-5 0/1 Pending 0 0s
wordpress-5 0/1 Pending 0 4s
wordpress-5 0/1 ContainerCreating 0 4s
```

```
dhs@mchoon ~/k_digital/assignment/05/src main ± kubectl ge
NAME READY STATUS RESTARTS AGE
wordpress-0 1/1 Running 0 5m3s
wordpress-1 1/1 Running 0 4m50s
wordpress-mysql-0 1/1 Running 0 5m3s
wordpress-mysql-1 1/1 Running 0 4m52s
wordpress-2 0/1 Pending 0 0s
wordpress-2 0/1 Pending 0 4s
wordpress-2 0/1 ContainerCreating 0 4s
wordpress-2 1/1 Running 0 9s
wordpress-3 0/1 Pending 0 0s
wordpress-3 0/1 Pending 0 4s
wordpress-3 0/1 ContainerCreating 0 4s
wordpress-3 1/1 Running 0 11s
wordpress-4 0/1 Pending 0 0s
wordpress-4 0/1 Pending 0 4s
wordpress-4 0/1 ContainerCreating 0 4s
wordpress-4 1/1 Running 0 10s
wordpress-5 0/1 Pending 0 0s
wordpress-5 0/1 Pending 0 4s
wordpress-5 0/1 ContainerCreating 0 4s
wordpress-5 1/1 Running 0 10s
wordpress-6 0/1 Pending 0 1s
wordpress-6 0/1 Pending 0 5s
wordpress-6 0/1 ContainerCreating 0 5s
wordpress-6 1/1 Running 0 9s
wordpress-7 0/1 Pending 0 0s
wordpress-7 0/1 Pending 0 4s
wordpress-7 0/1 ContainerCreating 0 4s
wordpress-7 1/1 Running 0 12s
wordpress-8 0/1 Pending 0 0s
wordpress-8 0/1 Pending 0 4s
wordpress-8 0/1 ContainerCreating 0 4s
wordpress-8 1/1 Running 0 10s
wordpress-9 0/1 Pending 0 0s
wordpress-9 0/1 Pending 0 0s
wordpress-9 0/1 Terminating 0 6m5s
wordpress-9 0/1 Terminating 0 6m5s
wordpress-8 1/1 Terminating 0 6m15s
wordpress-8 0/1 Terminating 0 6m17s
wordpress-8 0/1 Terminating 0 6m17s
wordpress-8 0/1 Terminating 0 6m17s
wordpress-8 0/1 Terminating 0 6m17s
wordpress-8 0/1 Terminating 0 6m17s
```

## Step 2-2: (NodeAffinity)

```
> kubectl get node
```

```
NAME
```

```
ip-10-0-2-181.ap-northeast-2.compute.internal  
ip-10-0-3-253.ap-northeast-2.compute.internal
```



goorm-mission-5 - deploy.yaml

```
1  affinity:  
2      nodeAffinity:  
3          requiredDuringSchedulingIgnoredDuringExecution:  
4              nodeSelectorTerms:  
5                  - matchExpressions:  
6                      - key: topology.kubernetes.io/zone  
7                        operator: In  
8                        values:  
9                            - ap-northeast-2b
```

```
> kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
nginx-96f656d8-6w6wh	1/1	Running	0	25m	10.0.2.164	ip-10-0-2-181.ap-northeast-2.compute.internal
nginx-96f656d8-j759h	1/1	Running	0	25m	10.0.2.212	ip-10-0-2-181.ap-northeast-2.compute.internal
nginx-96f656d8-wr99p	1/1	Running	0	25m	10.0.2.222	ip-10-0-2-181.ap-northeast-2.compute.internal

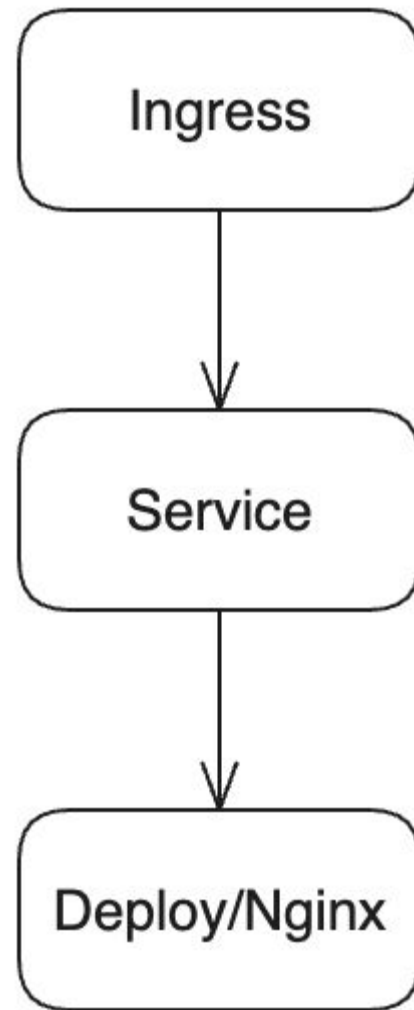
## Step 2-2: (Blue-Green Deploy)

goorm-mission-5 - ingress.yaml

```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: nginx
5    labels:
6      name: nginx
7    annotations:
8      alb.ingress.kubernetes.io/scheme: internet-facing
9      alb.ingress.kubernetes.io/target-type: ip
10     alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}]'
11  spec:
12    ingressClassName: alb
13    rules:
14      - http:
15          paths:
16            - path: /
17              pathType: Prefix
18              backend:
19                service:
20                  name: nginx
21                  port:
22                    number: 80
23
```

goorm-mission-5 - service.yaml

```
1  kind: Service
2  metadata:
3    name: nginx
4  spec:
5    type: ClusterIP
6    selector:
7      app: nginx
```



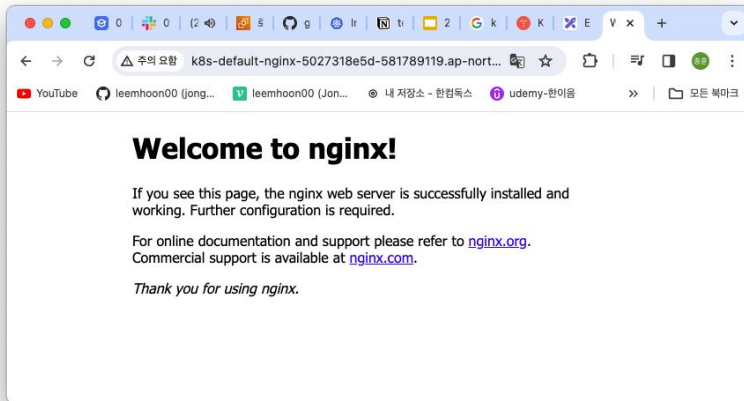
## Step 2-2: (Blue-Green Deploy)

**로드 밸런서 (1)**

Elastic Load Balancing은 수신 트래픽의 변화에 따라 자동으로 로드 밸런서 용량을 확장합니다.

로드 밸런서 필터링

<input type="checkbox"/>	이름	DNS 이름	상태
<input type="checkbox"/>	<a href="#">k8s-default-nginx-502...</a>	k8s-default-nginx-502731...	✓ <b>활성</b>





## Step 2-2: (Blue-Green Deploy)

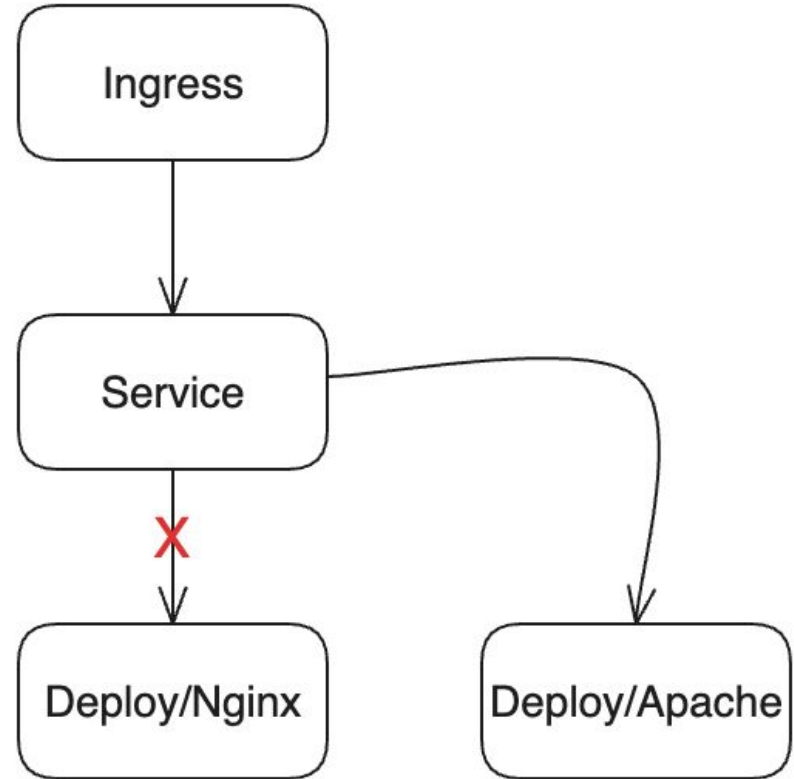
```
> kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
apache	1/1	1	1	12s
nginx	3/3	3	3	32m

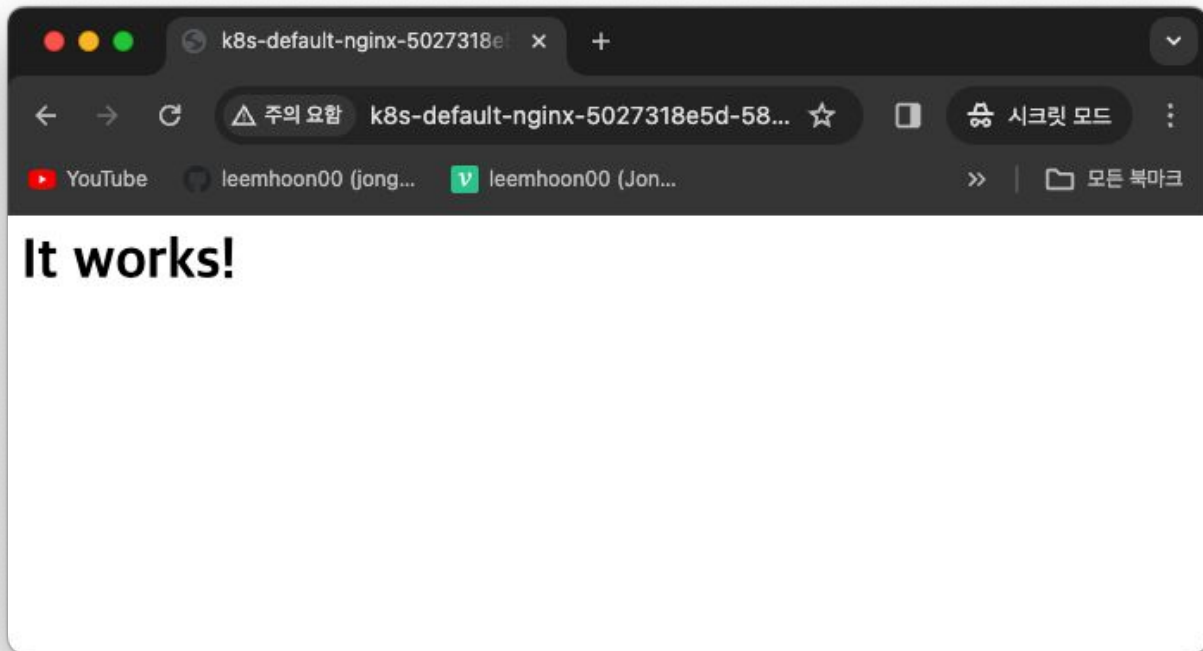
● ● ● goorm-mission-5 - service.yaml

```
1 kind: Service
2 metadata:
3   name: nginx
4 spec:
5   type: ClusterIP
6   selector:
7     app: apache
```

```
> kubectl apply -f mission-2/service.yaml
service/nginx configured
```



## Step 2-2: (Blue-Green Deploy)



## Step 3: (Liveness Probe)

```
● ● ● goorm-mission-5 - statefulset.yaml  
1  livenessProbe:  
2      exec:  
3          command: ['mysqladmin', 'ping']  
4      initialDelaySeconds: 30  
5      periodSeconds: 10
```

```
> mysqladmin -u root -p shutdown  
Enter password:  
command terminated with exit code 137
```

```
> kubectl get pod --watch  
NAME      READY   STATUS    RESTARTS   AGE  
mysql-0   1/1     Running   0           6m49s  
mysql-1   1/1     Running   0           6m40s  
mysql-0   0/1     Completed 0           10m  
mysql-0   1/1     Running   1 (1s ago) 10m
```

## Step 3: (Secret)

● ● ● goorm-mission-5 - secret.yaml

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: mysql
5  type: Opaque
6  stringData:
7    password: 'password'
8
```

● ● ● goorm-mission-5 - statefulset.yaml

```
1  env:
2    - name: MYSQL_ROOT_PASSWORD
3      valueFrom:
4        secretKeyRef:
5          name: mysql
6          key: password
```

## Step 3: (PVC)

goorm-mission-5 - statefulset.yaml

```
1 volumeClaimTemplates:
2   - metadata:
3       name: mysql-persistent-storage
4     spec:
5       accessModes: ['ReadWriteOnce']
6       resources:
7         requests:
8           storage: 1Gi
```

볼륨 (2/5) 정보

검색

<input type="checkbox"/>	Name ▾	볼... ▾	유형 ▾	크기 ▲	IOPS ▾
<input checked="" type="checkbox"/>	mission-5-dynamic-pvc-...	vol-...	gp2	1 GiB	100
<input checked="" type="checkbox"/>	mission-5-dynamic-pvc-...	vol-...	gp2	1 GiB	100
<input type="checkbox"/>	-	vol-...	gp3	8 GiB	3000
<input type="checkbox"/>	node-group-1	vol-...	gp2	20 GiB	100
<input type="checkbox"/>	node-group-2	vol-...	gp2	20 GiB	100

> kubectl get pvc

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
mysql-persistent-storage-mysql-0	Bound	pvc-a764c39e-fb6b-4467-be6c-0dea1eadf835	1Gi	RWO	gp2	86m
mysql-persistent-storage-mysql-1	Bound	pvc-de4a6a99-2543-4878-9276-943fb86f8bc0	1Gi	RWO	gp2	86m

## Step 3: (HeadLess Service)

goorm-mission-5 - service.yaml

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql
5  spec:
6    ports:
7      - port: 3306
8        targetPort: 3306
9      name: mysql
10   selector:
11     app: mysql
12   clusterIP: None
13
```

```
> kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	172.20.0.1	<none>	443/TCP	10h
<u>mysql</u>	<u>ClusterIP</u>	<u>None</u>	<u>&lt;none&gt;</u>	3306/TCP	32m

```
> kubectl get endpoints
```

NAME	ENDPOINTS	AGE
kubernetes	10.0.2.75:443,10.0.3.15:443	10h
<u>mysql</u>	<u>10.0.2.212:3306,10.0.3.189:3306</u>	31m

## Step 3: (HeadLess Service)

```
> mysql -u root -h mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 118
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```