

구름 <b>KDT</b> 쿠버네티스 전문가 양성과정_6기 프로젝트 - 구축			
팀 명	1조	일 자	2022년 11 월 23 일
주 제 명	도서정보 인프라 구축		

# CONTENTS

<b>1. 네트워크 구축</b>	<b>4</b>
1.1. VPC 구축	4
1.2. 인스턴스	4
1.3. Auto Scaling 그룹	4
1.4. 로드밸런서	4
1.5. 서브넷	4
1.6. 게이트 웨이	5
<b>2. 서버 구축</b>	<b>5</b>
2.1. IAM 사용자 추가	5
2.2. Cloud9 환경 구축	7
<b>3. IAM Role 생성</b>	<b>9</b>
3.1. IAM 역할	9
3.3. IDE(AWS Cloud9 인스턴스)에 IAM Role 부여	10
3.4. Cloud9 환경 설정 & EKS 클러스터 생성	10
<b>4. DB 구축</b>	<b>14</b>
4.1. DB 인스턴스	14
4.2. Table 생성 script	21
<b>5. 보안 구축</b>	<b>24</b>
5.1. IAM 보안	24
5.2. Port 규칙	24
<b>6. 모니터링 구축</b>	<b>25</b>
6.1. helm install을 위한 values 파일 생성	25
6.2. Prometheus, Grafana를 Helm chart로 설치	26
6.3. Terminal로 배포된 모니터링 서비스 확인	26
6.4. Grafana	27
6.5. 왼쪽 사이드바 - Dashboards - Import	27
6.6. Dashboard 생성 및 수정	28
<b>7. CI/CD 구축</b>	<b>29</b>
7.1. 프로젝트 레포지토리 생성	29
7.2. GitHub token 생성	30
7.3. GitHub secret 설정	30
7.4. 깃허브 액션을 위한 build 스크립트 작성	32
7.5. Dockerfile 작성	33
7.6. 개발한 프로젝트를 깃허브에 push해 깃허브 액션을 실행	33
7.7. Docker로 빌드하고 ECR에 push된 이미지 확인	34
7.8. GitOps 레포지토리 생성	34
7.9. Kubernetes에 ArgoCD 설치	35
7.10. ArgoCD 로그인	36
7.11. ArgoCD에 GitOps 레포지토리 연결	36

7.12. 애플리케이션 배포를 위한 <b>manifest</b> 를 작성해 <b>GitOps</b> 레포지토리에 <b>push</b>	37
7.13. <b>GitOps</b> 레포지토리의 변경 사항을 <b>ArgoCD</b> 가 확인하고 <b>Kubernetes</b> 에 <b>Sync</b>	37

## 1. 네트워크 구축

### 1.1. VPC 구축

#### VPC (2/4) 정보

Q VPC 필터링

<div><div></div></div>	Name	VPC ID	상태	IPv4 CIDR	IPv6 CIDR
<div><div></div></div>	eksctl-aladinEKS-cluster/VPC	vpc-0e3c59ead8678b9bd	<div><div></div>Available</div>	192.168.0.0/16	-
<div><div></div></div>	eksctl-eks-demo-cluster/VPC	vpc-0efd9063e1148af9a	<div><div></div>Available</div>	192.168.0.0/16	-
<div><div></div></div>	-	vpc-00154e85cd24a2126	<div><div></div>Available</div>	172.31.0.0/16	-
<div><div></div></div>	goorm-vpc	vpc-07a6378f0d19b8d30	<div><div></div>Available</div>	10.0.0.0/16	-

### 1.2. 인스턴스

#### 인스턴스 (5) 정보

Find 인스턴스 by attribute or tag (case-sensitive)

demo

필터 지우기

<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...
<input type="checkbox"/>	eks-demo-node-group-Node	i-0b699555b925b9260	<span>실행 중</span>	m5.large	2/2개 검사 통과	경보 없음	ap-northeast-2b	ec2-3-38-160-191.ap-n...	3.38.160.191
<input type="checkbox"/>	eks-demo-node-group-Node	i-0bcaf32d902cb7ce	<span>실행 중</span>	m5.large	2/2개 검사 통과	경보 없음	ap-northeast-2b	ec2-3-35-228-34.ap-no...	3.35.228.34
<input type="checkbox"/>	eks-demo-node-group-Node	i-00128b485a196ec8f	<span>실행 중</span>	m5.large	2/2개 검사 통과	경보 없음	ap-northeast-2c	ec2-15-164-163-73.ap-...	15.164.163.73
<input type="checkbox"/>	eks-demo-node-group-Node	i-0d22118570c7550d2	<span>실행 중</span>	m5.large	2/2개 검사 통과	경보 없음	ap-northeast-2c	ec2-52-79-251-18.ap-n...	52.79.251.18
<input type="checkbox"/>	eks-demo-node-group-Node	i-032e60d7050275409	<span>실행 중</span>	m5.large	2/2개 검사 통과	경보 없음	ap-northeast-2a	ec2-13-125-231-48.ap-...	13.125.231.48
<input type="checkbox"/>	aws-cloud9-eks-workspace-472d14d977c041...	i-0613f61f5ae07138f	<span>실행 중</span>	t3.medium	2/2개 검사 통과	경보 없음	ap-northeast-2d	ec2-3-37-175-196.ap-n...	3.37.175.196

### 1.3. Auto Scaling 그룹

Auto Scaling 그룹 (1/2) Info

Q Auto Scaling 그룹 검색

<input type="checkbox"/>	이름	시작 템플릿/구성	인스턴스	상태	원하는 용량	최소	최대	가용 영역
<input checked="" type="checkbox"/>	eks-node-group-7ec249a8-32fc-7e...	eks-7ec249a8-32fc-7ecd-a705-d93659b5	5	-	5	3	6	ap-northeast-2a, ap-northeast-2b, ap-northeast-2c
<input type="checkbox"/>	eks-default-ng-bcc23f46-2c3d-3e...	eks-bcc23f46-2c3d-3e12-b684-dad082c	3	-	3	3	6	ap-northeast-2a, ap-northeast-2b, ap-northeast-2c

### 1.4. 로드밸런서

로드 밸런서 생성

작업

Q 태그 및 속성별 필터 또는 키워드별 검색

<input type="checkbox"/>	이름	DNS 이름	상태	VPC ID	가용 영역	알림	생성 날짜	모니터링
<input checked="" type="checkbox"/>	a491e790da2e480a7f13365da1b85	a491e790da2e480a7f13365da1b85-1199409123.ap-northeast-2.elb.amazonaws.com	-	vpc-0efd9063e1148af9a	ap-northeast-2c, ap-northeast-2b, ap-northeast-2a	classic	2022년 11월 20일 오후 3시 44	
<input checked="" type="checkbox"/>	a489bc9202084f0a089871402d6d4	a489bc9202084f0a089871402d6d4-1014715528.ap-northeast-2.elb.amazonaws.com	-	vpc-0efd9063e1148af9a	ap-northeast-2c, ap-northeast-2b, ap-northeast-2a	classic	2022년 11월 20일 오후 4시 54	
<input checked="" type="checkbox"/>	aaa9330067a46171a10552c0b6d0a	aaa9330067a46171a10552c0b6d0a-1460717518.ap-northeast-2.elb.amazonaws.com	-	vpc-0efd9063e1148af9a	ap-northeast-2c, ap-northeast-2b, ap-northeast-2a	classic	2022년 11월 20일 오후 7시 43	

### 1.5. 서버넷

#### 서버넷 (6) 정보

Q 서버넷 필터링					
<input type="checkbox"/>	Name	서버넷 ID	상태	VPC	IPv4 CIDR
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPublicAPNORTHEAST2C	subnet-09f7e5e9f40892eb6	Available	vpc-0efd9063e1148af9a   eks...	192.168.0.0/19
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPublicAPNORTHEAST2B	subnet-05c6b1b17575c92b	Available	vpc-0efd9063e1148af9a   eks...	192.168.32.0/19
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPublicAPNORTHEAST2A	subnet-0a0d688a71dca5340	Available	vpc-0efd9063e1148af9a   eks...	192.168.64.0/19
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPrivateAPNORTHEAST2C	subnet-055e6507429b44175	Available	vpc-0efd9063e1148af9a   eks...	192.168.96.0/19
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPrivateAPNORTHEAST2B	subnet-0257eb01d01c350a6	Available	vpc-0efd9063e1148af9a   eks...	192.168.128.0/19
<input type="checkbox"/>	eksctl-eks-demo-cluster/SubnetPrivateAPNORTHEAST2A	subnet-0fa743b091986bd6c	Available	vpc-0efd9063e1148af9a   eks...	192.168.160.0/19

## 1.6. 게이트 웨이

인터넷 게이트웨이 (1/4) 정보					
<input type="text" value="인터넷 게이트웨이 필터링"/>					
<input type="checkbox"/>	Name	인터넷 게이트웨이 ID	상태	VPC ID	소유자
<input checked="" type="checkbox"/>	eksctl-eks-demo-cluster/InternetGateway	igw-033b4479a14c21b96	Attached	vpc-0efd9063e1148af9a   eksctl-eks-d...	535954374321
<input type="checkbox"/>	eksctl-eks-demo-cluster/InternetGateway	igw-033b4479a14c21b96	Attached	vpc-0efd9063e1148af9a   eksctl-eks-d...	535954374321

## 2. 서버 구축

### 2.1. IAM 사용자 추가

- AWS 로그인
- IAM 콘솔 화면 사이드 바 - Users(사용자) - Add user(사용자 추가)
- 사용자 정보 입력
  - 사용자 이름: Administrator
  - Access type: 암호 - AWS 관리 콘솔 액세스
  - 콘솔 비밀번호: 사용자 지정
  - 비밀번호 재설정 필요: 해제
- 다음: 권한 클릭

#### 사용자 추가

1 2 3 4 5

#### 사용자 세부 정보 설정

동일한 액세스 유형 및 권한을 사용하여 한 번에 여러 사용자를 추가할 수 있습니다. [자세히 알아보기](#)

사용자 이름\* eks-Administrator

[다른 사용자 추가](#)

#### AWS 액세스 유형 선택

이러한 사용자가 주로 AWS에 액세스하는 방법을 선택합니다. 프로그래밍 방식의 액세스만 선택하면 사용자가 위임된 역할을 사용하여 콘솔에 액세스하는 것을 방지할 수 없습니다. 액세스 키와 자동 생성된 암호가 마지막 단계에서 제공됩니다. [자세히 알아보기](#)

- AWS 자격 증명 유형 선택\* ☐ 액세스 키 - 프로그래밍 방식 액세스  
AWS API, CLI, SDK 및 기타 개발 도구에 대해 액세스 키 ID 및 비밀 액세스 키(를) 활성화합니다.
- ☒ 암호 - AWS 관리 콘솔 액세스  
사용자가 AWS Management Console에 로그인할 수 있도록 허용하는 비밀번호(를) 활성화합니다.

콘솔 비밀번호\* ☐ 자동 생성된 비밀번호  
☒ 사용자 지정 비밀번호

.....  
☐ 비밀번호 표시

비밀번호 재설정 필요 ☐ 사용자가 다음에 로그인할 때 새 비밀번호 생성 요청  
사용자가 비밀번호를 변경할 수 있도록 허용하는 IAMUserChangePassword 정책을 자동으로 가져옵니다.

\* 필수

[취소](#)

[다음: 권한](#)

- 기존 정책 직접 연결 - AdministratorAccess 선택
- 다음: 태그 클릭

## 사용자 추가

1 2 3 4 5

### ▼ 권한 설정

그룹에 사용자 추가

기존 사용자에서 권한 복사

기존 정책 직접 연결

정책 생성

↺

정책 필터 ▼

Q 검색

788 결과 표시

	정책 이름 ▼	유형	사용 용도
<input checked="" type="checkbox"/>	AdministratorAccess	직무 기반	Permissions policy (7)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS 관리형	없음
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS 관리형	없음
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS 관리형	없음
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS 관리형	없음
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS 관리형	없음

취소

이전

다음: 태그

- 다음: 검토 클릭
- Administrator 사용자에게 AdministratorAccess 관리형 정책이 추가된 것을 확인하고 사용자 만들기 클릭
- 로그아웃 - Administrator으로 로그인

## 2.2. Cloud9 환경 구축

- 환경 구축 순서
  - AWS Cloud9으로 IDE 구성
  - IAM Role 생성
  - IDE(AWS Cloud9 인스턴스)에 IAM Role 부여
  - IDE에서 IAM 설정 업데이트
- AWS Cloud9으로 IDE 구성
  - Cloud9 콘솔창 - Create environment 클릭
  - IDE 정보 입력
    - Name: eks-admin
    - Environment type: New EC2 instance
    - EC2 Instance
      - Instance type: t3.medium

- Platform: Amazon Linux 2
  - Connection: SSM
  - VPC: default
- Create 클릭

AWS Cloud9 > Environments > Create environment

## Create environment Info

### Details

**Name**

Limit of 60 characters, alphanumeric, and unique per user.

**Description - optional**

Limit 200 characters.

**Environment type** Info  
Determines what the Cloud9 IDE will run on.

☒ **New EC2 instance**  
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

☐ **Existing compute**  
You have an existing instance or server that you'd like to use.

### New EC2 instance

**Instance type** Info  
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

☐ **t2.micro (1 GiB RAM + 1 vCPU)**  
Free-tier eligible. Ideal for educational users and exploration.

☐ **t3.small (2 GiB RAM + 2 vCPU)**  
Recommended for small web projects.

☐ **m5.large (2 GiB RAM + 2 vCPU)**  
Recommended for production and most general-purpose development.

☒ **Additional instance types**  
Explore additional instances to fit your need.

**Additional instance types**

t3.medium

**Platform** Info  
This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Amazon Linux 2

**Timeout**  
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

### Network settings Info

**Connection**  
How your environment is accessed.

☒ **AWS Systems Manager (SSM)**  
Accesses environment via SSM without opening inbound ports (no Ingress).

☐ **Secure Shell (SSH)**  
Accesses environment directly via SSH, opens inbound ports.

▼ **VPC settings** Info

**Amazon Virtual Private Cloud (VPC)**  
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#)

vpc-00154e85cd24a2126

**Subnet**  
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#)

No preference  
Uses default subnet in any Availability Zone

### Tags - optional Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

**The following IAM resources will be created in your account**

- AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

Cancel Create

### 3. IAM Role 생성

#### 3.1. IAM 역할

- IAM 역할은 IAM 사용자 및 AWS 서비스에 사용할 수 있으며, 서비스에 역할을 부여할 경우, 서비스가 사용자를 대신해 부여받은 역할을 수행

#### 3.2. IAM 역할 부여

- AdministratorAccess 정책을 가진 역할을 Cloud9에 부여
- 역할 부여 과정
  - IAM 역할 페이지에 접속
  - 역할 만들기
    - AWS 서비스: EC2
  - 다음 클릭
  - 권한 추가
    - AdministratorAccess 선택
  - 다음 클릭
  - 이름 지정, 쿼터 및 생성
    - 역할 이름: eksworkspace-admin
  - 역할 생성 클릭

Create role

1234

Review

Provide the required information below and review this role before you create it.

Role name\*

eksworkspace-admin

Enter Role name

Use alphanumeric and '+=, @-\_' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

Trusted entities

AWS service: ec2.amazonaws.com

Policies

AdministratorAccess

Check IAM Policies

Permissions boundary

Permissions boundary is not set

No tags were added.

\* Required

Cancel

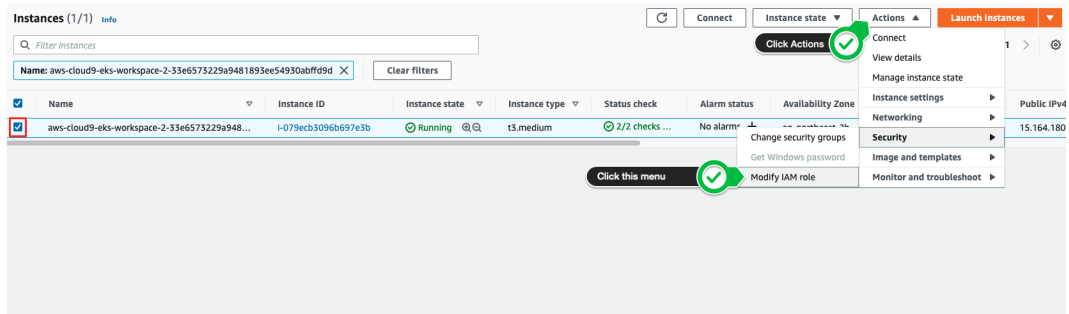
Previous

Create role

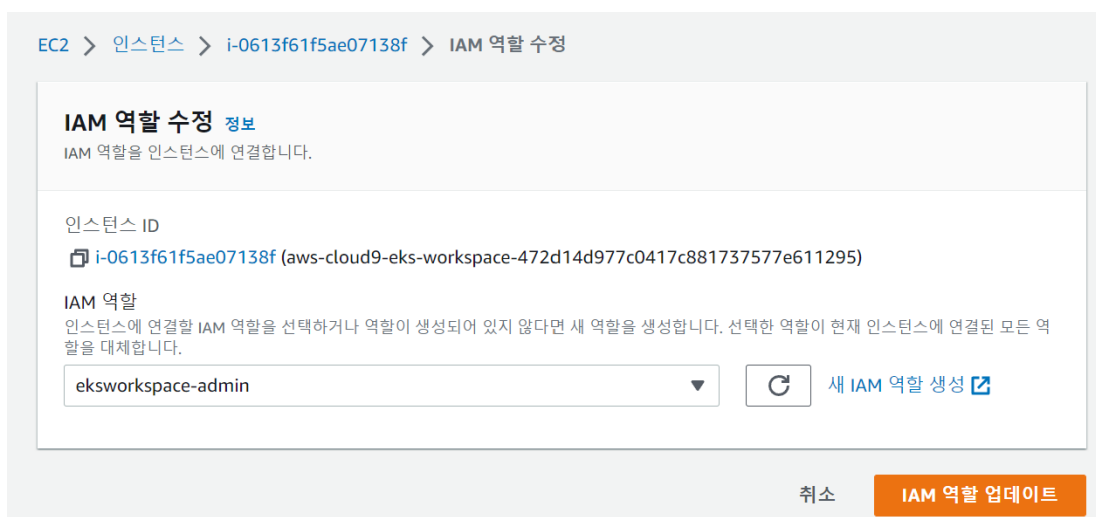


### 3.3. IDE(AWS Cloud9 인스턴스)에 IAM Role 부여

- EC2 인스턴스 페이지 접속
- Cloud9 인스턴스 선택 > 작업 > 보안 > IAM 역할 수정



- eksworkspace-admin 선택
- IAM 역할 업데이트 클릭



### 3.4. Cloud9 환경 설정 & EKS 클러스터 생성

- 환경 설정

```
# eks-cluster.yaml

---

apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: eks-aladin # 생성할 EKS 클러스터명
  region: ap-northeast-2 # 클러스터를 생성할 리전
```

```

version: "1.23"

iam:
  vpcResourceControllerPolicy: true
  withOIDC: true

vpc:
  cidr: "192.168.0.0/16" # 클러스터에서 사용할 VPC의 CIDR

managedNodeGroups:
  - name: node-group # 클러스터의 노드 그룹명
    instanceType: m5.large # 클러스터 워커 노드의 인스턴스 타입
    minSize: 3
    desiredCapacity: 3 # 클러스터 워커 노드의 갯수
    MaxSize: 6
    volumeSize: 10 # 클러스터 워커 노드의 EBS 용량 (단위: GiB)
    iam:
      withAddonPolicies:
        imageBuilder: true # Amazon ECR에 대한 권한 추가
        # albIngress: true # albIngress에 대한 권한 추가
        cloudWatch: true # cloudWatch에 대한 권한 추가
        autoScaler: true # auto scaling에 대한 권한 추가

cloudWatch:
  clusterLogging:
    enableTypes: ["*"]

```

## ■ 클러스터 생성

```

# create-cluster.sh

#!/bin/bash

export CLUSTER_NAME=$1
envsubst < eks-cluster.yaml > $CLUSTER_NAME.yaml
# EKS 클러스터 생성
eksctl create cluster -f $CLUSTER_NAME.yaml

```

```
# 콘솔 Credential을 클러스터에 추가
rolearn=$(aws cloud9 describe-environment-memberships
--environment-id=$C9_PID | jq -r '.memberships[].userArn')
if [[ "$rolearn" =~ "assumed-role" ]]; then
    assumedrolename=$(echo ${rolearn} | awk -F/ '{print $(NF-1)}')
    rolearn=$(aws iam get-role --role-name ${assumedrolename}
--query Role.Arn --output text)
fi

# identity 맵핑
eksctl create iamidentitymapping --cluster $CLUSTER_NAME --arn
${rolearn} --group system:masters --username admin
# 관련 정보 확인
kubectl describe configmap -n kube-system aws-auth
```

#### ■ ALB Ingress controller 설치

```
# create-loadBalancerController

#!/bin/bash

OIDC_provider_URL=$(aws eks describe-cluster --name $CLUSTER_NAME
--query "cluster.identity.oidc.issuer" --output text | awk -F "/" '{
print $5 }')
CHECK_OIDC=$(aws iam list-open-id-connect-providers | grep
$OIDC_provider_URL)
if [[ -z ${CHECK_OIDC} ]]; then
    eksctl utils associate-iam-oidc-provider \
        --region ${AWS_REGION} \
        --cluster $CLUSTER_NAME \
        --approve
fi

aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://iam-policy.json

# AWS Load Balancer Controller를 위한 ServiceAccount를 생성
```

```
eksctl create iamserviceaccount \
  --cluster $CLUSTER_NAME \
  --namespace kube-system \
  --name aws-load-balancer-controller \
  --attach-policy-arn
arn:aws:iam::$ACCOUNT_ID:policy/AWSLoadBalancerControllerIAMPolicy \
  --override-existing-serviceaccounts \
  --approve

## 클러스터에 컨트롤러 추가
# 인증서 구성을 웹훅에 삽입할 수 있도록 cert-manager를 설치
# cert-manager: 쿠버네티스 내에서 TLS 인증서를 자동으로 프로비저닝 및
# 관리하는 오픈소스
kubectl apply --validate=false -f
https://github.com/jetstack/cert-manager/releases/download/v1.5.4/cert-manager.yaml

# 로드밸런서 YAML 파일 다운로드
wget
https://github.com/kubernetes-sigs/aws-load-balancer-controller/releases/download/v2.4.4/v2_4_4_full.yaml
# YAML 파일 편집
sed -i 's/your-cluster-name/$CLUSTER_NAME/g' v2_4_4_full.yaml
sed '480,488d' v2_4_4_full.yaml

kubectl apply -f v2_4_4_full.yaml
```

## ■ AWS EBS CSI Driver 설치

```
# aws-ebs-csi-driver.sh

#!/bin/bash

eksctl create iamserviceaccount \
  --name ebs-csi-controller-sa \
  --namespace kube-system \
  --cluster $CLUSTER_NAME \
  --attach-policy-arn
```

```

arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
--approve \
--role-only \
--role-name AmazonEKS_EBS_CSI_DriverRole

IS_EXIST=$(aws eks describe-addon-versions --addon-name
aws-ebs-csi-driver)

if [[ -n $IS_EXIST ]]; then
    echo "EBS CSI addon already exist."
    exit 1
fi

eksctl create addon --name aws-ebs-csi-driver \
    --cluster $CLUSTER_NAME \
    --service-account-role-arn
arn:aws:iam::$ACCOUNT_ID:role/AmazonEKS_EBS_CSI_DriverRole \
    --force

```

## 4. DB 구축

### 4.1. DB 인스턴스

- RDS 스크립트 - 기능(생성, 삭제, 상태확인)

```

#!/bin/bash

up() {
    echo 'rds create-db-instance'

    # describe-db-instances check
    db_instance_check
    echo count $is_db_existence
    if [ $is_db_existence -ne 0 ]
    then
        echo 'is db-instance(book-db-mysql)'
        unset db_existence
        return
    fi
    unset db_existence
}

```

```

# aws rds create-db
aws rds create-db-instance \
  --engine mysql \
  --engine-version 8.0.28 \
  --multi-az \
  --db-instance-identifier book-db-mysql \
  --master-username admin \
  --master-user-password mysqladmin \
  --db-instance-class db.t3.micro \
  --storage-type gp2 \
  --allocated-storage 20 \
  --max-allocated-storage 1000 \
  --db-subnet-group-name eks-demo-rds-subnet-group \
  --no-publicly-accessible \
  --vpc-security-group-ids "sg-0bc87f6799b8dd098" \
  --backup-retention-period 7 \
  --preferred-backup-window 02:00-03:00 \
  --no-auto-minor-version-upgrade \
  --port 3306 \
  --deletion-protection \
  --db-name book \
> ~/rds-info.txt

cat ~/rds-info.txt
}

down() {
  echo 'rds delete-db-instance'

  # is db-instance?
  db_instance_check
  echo count $is_db_existence
  if [ $is_db_existence -eq 0 ]
  then
    echo 'no db-instance(book-db-mysql)'
    unset db_existence
    return
  fi
}

```

```

unset db_existence

# really deleted?
echo 'are you sure you want to delete?? [y/n]'
read input

if [ $input == 'y' ]
then
    # cancel deletion protection
    aws rds modify-db-instance \
        --db-instance-identifier book-db-mysql \
        --no-deletion-protection \
        --apply-immediately
elif [ $input == 'n' ]
then
    echo 'rds delete-db-instance cancel'
    return
else
    echo 'command does not exist'
    echo 'rds delete-db-instance fail'
    return
fi

# rds delete-db-instance
aws rds delete-db-instance \
    --db-instance-identifier book-db-mysql \
    --skip-final-snapshot \
    --no-delete-automated-backups
}

status() {
    echo 'rds db-instance describe'
    aws rds describe-db-instances \
        --query
    "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUser
name]"
}

db_instance_check() {

```

```

echo 'book-db-check'

# db_existence

local db_existence=$(aws rds describe-db-instances \
    --query
    "*"[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUser
name]" \
    | grep 'book-db-mysql')

is_db_existence=${#db_existence}
}

rds_select() {
    input_str=${1}

    # input check
    if [ -z ${input_str} ]
    then
        echo 'please enter the command'
        return
    fi

    echo input : ${input_str}

    # command check
    if [ ${input_str} == 'up' ]
    then
        up
    elif [ ${input_str} == 'down' ]
    then
        down
    elif [ ${input_str} == 'status' ]
    then
        status
    elif [ ${input_str} == 'check' ]
    then
        db_instance_check
    else

```



```

        echo 'command does not exist'
    fi
}

rds_select $1

```

- 고려사항 : db-subnet-group-name, vpc-security-group-ids

■ bastion server instance 스크립트 - 기능(생성, 삭제, 상태확인)

```

#!/bin/bash

up() {
    echo 'ec2-instance create - tag(name : rds-bastion-server)'

    # describe-instances check
    instanceId_extraction
    if [ $instance_existence -gt 2 ]
    then
        echo 'is instance(rds-bastion-server)'
        unset instance_existence
        unset instanceId
        return
    fi
    unset instance_existence
    unset instanceId

    # aws instance create
    aws ec2 run-instances \
        --image-id ami-07d16c043aa8e5153 \
        --instance-type t2.micro \
        --block-device-mappings
'DeviceName=/dev/sda1,Ebs={DeleteOnTermination=true,VolumeSize=8,VolumeType=gp2}' \
        --associate-public-ip-address \
        --key-name goorm-aicore0909-20221111 \
        --security-group-ids sg-0b823e0ebbc068ef3 \
        --subnet-id subnet-01973eeead6577a9d \

```

```

        --tag-specifications
        'ResourceType=instance,Tags=[{Key=Name,Value=rds-bastion-server}]' \
        > ~/instance_info.txt

        cat ~/instance_info.txt
    }

    down() {
        echo 'rds delete-instance'

        # is instance?
        instanceId_extraction
        if [ $instance_existence -le 2 ]
        then
            echo 'no instance(rds-bastion-server)'
            unset instance_existence
            unset instanceId
            return
        fi
        unset instance_existence

        # really deleted?
        echo ${instanceId} - 'are you sure you want to delete?? [y/n]'
        read input

        if [ $input == 'y' ]
        then
            # file(instance_info.txt) delete
            if [ -f ~/instance_info.txt ]
            then
                echo 'file(instance_info.txt) delete success!'
                rm -f ~/instance_info.txt
            fi

            # delete-instance
            aws ec2 terminate-instances --instance-ids $instanceId
            unset instanceId
        elif [ $input == 'n' ]
        then

```

```

        echo 'delete-instance cancel'
    else
        echo 'command does not exist'
        echo 'delete-instance fail'
    fi
}

instance() {
    echo 'instance(rds-bastion-server) instanceId'
    aws ec2 describe-instances \
        --filters "Name=tag:Name,Values=rds-bastion-server"
    "Name=instance-state-name,Values=running" \
        --query "Reservations[].Instances[].InstanceId"
}

instanceId_extraction(){
    echo 'rds-bastion-server instanceId_extraction'

    # instance running & name is rds-bastion-server
    local instanceId_temp=$(aws ec2 describe-instances \
        --filters "Name=tag:Name,Values=rds-bastion-server"
    "Name=instance-state-name,Values=running" \
        --query "Reservations[].Instances[].InstanceId")

    instanceId=$(echo $instanceId_temp | cut -d '"' -f 2)

    instance_existence=${#instanceId}
}

instance_select() {
    input_str=${1}

    # input check
    if [ -z ${input_str} ]
    then
        echo 'please enter the command'
        return
    fi
}

```

```

echo input : ${input_str}

# command check
if [ ${input_str} == 'up' ]
then
    up
elif [ ${input_str} == 'down' ]
then
    down
elif [ ${input_str} == 'instance' ]
then
    instance
else
    echo 'command does not exist'
fi
}

instance_select $1

```

#### 4.2. Table 생성 script

- Table 생성 스크립트

```

-- MEMBER Table Create SQL
CREATE TABLE MEMBER
(
    `PID`                INT UNSIGNED    NOT NULL    AUTO_INCREMENT
COMMENT 'PID',
    `ID`                 VARCHAR(45)     NOT NULL    COMMENT
'아이디',
    `PASSWD`             VARCHAR(255)    NOT NULL    COMMENT
'비밀번호',
    `NIC_NAME`           VARCHAR(45)     NOT NULL    COMMENT
'닉네임',
    `REGISTRATION_DATE`  DATETIME        NOT NULL    COMMENT
'등록일',
    `MODIFICATION_DATE`  DATETIME        NOT NULL    COMMENT

```

```

'수정일',
    `AUTH`          INT          NOT NULL    COMMENT
'권한(일반, 관리자)',
    PRIMARY KEY (PID)
);

ALTER TABLE MEMBER COMMENT '회원';

-- BOOK_REVIEW Table Create SQL
CREATE TABLE BOOK_REVIEW
(
    `PID`            INT UNSIGNED NOT NULL    AUTO_INCREMENT
COMMENT 'PID',
    `TITLE`          VARCHAR(45)  NOT NULL    COMMENT '제목',
    `CONTENTS`       LONGTEXT     NOT NULL    COMMENT '내용',
    `WRITER`         VARCHAR(45)  NOT NULL    COMMENT
'작성자(닉네임)',
    `REGISTRATION_DATE` DATETIME   NOT NULL    COMMENT
'등록일',
    `MODIFICATION_DATE` DATETIME   NOT NULL    COMMENT
'수정일',
    `ISBN`           VARCHAR(15)  NOT NULL    COMMENT '도서
ISBN',
    `MEMBER_PID`     INT UNSIGNED NOT NULL    COMMENT '회원
PID',
    PRIMARY KEY (PID)
);

ALTER TABLE BOOK_REVIEW COMMENT '서평';

ALTER TABLE BOOK_REVIEW
    ADD CONSTRAINT FK_BOOK_REVIEW_MEMBER_PID_MEMBER_PID FOREIGN KEY
(MEMBER_PID)
    REFERENCES MEMBER (PID) ON DELETE RESTRICT ON UPDATE
RESTRICT;

-- BOOK_REGISTRATION Table Create SQL

```

```

CREATE TABLE BOOK_REGISTRATION
(
    `PID`                INT UNSIGNED    NOT NULL    AUTO_INCREMENT
COMMENT 'PID',
    `BOOK_TITLE`         VARCHAR(45)     NOT NULL    COMMENT
'도서명',
    `ISBN`               VARCHAR(15)     NOT NULL    COMMENT '도서
ISBN',
    `REVIEW_COUNT`       INT              NOT NULL    COMMENT '등록된
서평 개수',
    `REPLY_COUNT`        INT              NOT NULL    COMMENT '등록된
댓글 수',
    `TOTAL_STAR_RATING`  INT              NOT NULL    COMMENT '총
별점',
    `REGISTRATION_DATE`  DATETIME         NOT NULL    COMMENT
'등록일',
    `MODIFICATION_DATE`  DATETIME         NOT NULL    COMMENT
'수정일',
    PRIMARY KEY (PID)
);

ALTER TABLE BOOK_REGISTRATION COMMENT '등록 도서';

-- BOOK_REPLY Table Create SQL
CREATE TABLE BOOK_REPLY
(
    `PID`                INT UNSIGNED    NOT NULL    AUTO_INCREMENT
COMMENT 'PID',
    `WRITER`            VARCHAR(45)     NOT NULL    COMMENT
'작성자(닉네임)',
    `CONTENTS`          VARCHAR(45)     NOT NULL    COMMENT '내용',
    `STAR_RATING`       INT              NOT NULL    COMMENT '별점',
    `ISBN`              VARCHAR(15)     NOT NULL    COMMENT '도서
ISBN',
    `REGISTRATION_DATE`  DATETIME         NOT NULL    COMMENT
'등록일',
    `MODIFICATION_DATE`  DATETIME         NOT NULL    COMMENT
'수정일',

```

```

`MEMBER_PID`          INT UNSIGNED    NOT NULL    COMMENT '회원
PID',
PRIMARY KEY (PID)
);

ALTER TABLE BOOK_REPLY COMMENT '댓글+별점';

ALTER TABLE BOOK_REPLY
ADD CONSTRAINT FK_BOOK_REPLY_MEMBER_PID_MEMBER_PID FOREIGN KEY
(MEMBER_PID)
REFERENCES MEMBER (PID) ON DELETE RESTRICT ON UPDATE
RESTRICT;

```

## 5. 보안 구축

### 5.1. IAM 보안

<input checked="" type="checkbox"/>	rds-monitoring-role	AWS 서비스: monitoring.rds
<input checked="" type="checkbox"/>	eksctl-eks-demo-addon-iamserviceaccount-kube-Role1-1W77QHWSDN6I3	자격 증명 공급자: arn:aws:iam::535954374321:oidc-prov
<input checked="" type="checkbox"/>	AWSServiceRoleForElasticLoadBalancing	AWS 서비스: elasticloadbalancing (서비스 연결 역할)
<input checked="" type="checkbox"/>	AmazonEKS_EBS_CSI_DriverRole	자격 증명 공급자: arn:aws:iam::535954374321:oidc-prov
<input checked="" type="checkbox"/>	AWSServiceRoleForRDS	AWS 서비스: rds (서비스 연결 역할)
<input checked="" type="checkbox"/>	eksctl-aladinEKS-addon-iamserviceaccount-kub-Role1-IAZFZXFSJ2B7	자격 증명 공급자: arn:aws:iam::535954374321:oidc-prov
<input checked="" type="checkbox"/>	AWSServiceRoleForAWSCloud9	AWS 서비스: cloud9 (서비스 연결 역할)
<input checked="" type="checkbox"/>	AdministratorAccess	계정: 535954374321

- RBAC(역할 기반 액세스 제어)을 기반으로 제어할 수 있는 서비스를 제한하는 방식으로 보안을 적용했다

### 5.2. Port 규칙

보안 그룹 (6) 정보									
<div> <input type="text" value="보안 그룹 필터링"/> <input type="button" value="검색: demo"/> <input type="button" value="필터 지우기"/> </div>									
<input type="checkbox"/>	Name	보안 그룹 ID	보안 그룹 이름	VPC ID	설명	소유자	인바운드 규칙 수	아웃바운드 규칙 수	
<input type="checkbox"/>	eksctl-eks-demo-cluster/...	sg-00b4061e792b6353f	eksctl-eks-demo-clust...	vpc-0efd9063e1148af9a	Communication between all nodes in the cluster	535954374321	2 권한 항목	1 권한 항목	
<input type="checkbox"/>	-	sg-03b416e092577d337	k8s-elb-a480bc69f2c8...	vpc-0efd9063e1148af9a	Security group for Kubernetes ELB a480bc69f2c864...	535954374321	2 권한 항목	1 권한 항목	
<input type="checkbox"/>	eks-cluster-sg-eks-demo...	sg-09a25fa7fdb9b54	eks-cluster-sg-eks-de...	vpc-0efd9063e1148af9a	EKS created security group applied to ENI that is att...	535954374321	5 권한 항목	1 권한 항목	
<input type="checkbox"/>	eksctl-eks-demo-cluster/...	sg-0bde5b4536530e707	eksctl-eks-demo-clust...	vpc-0efd9063e1148af9a	Communication between the control plane and wor...	535954374321	0 권한 항목	1 권한 항목	
<input type="checkbox"/>	-	sg-0ee15a67f5c3346f7	k8s-elb-aaa903808d7...	vpc-0efd9063e1148af9a	Security group for Kubernetes ELB aaa903808d7ef...	535954374321	2 권한 항목	1 권한 항목	
<input type="checkbox"/>	-	sg-0f95eebdc056de09a	k8s-elb-af491e790da2...	vpc-0efd9063e1148af9a	Security group for Kubernetes ELB af491e790da2e...	535954374321	3 권한 항목	1 권한 항목	

## 6. 모니터링 구축

### 6.1. helm install을 위한 values 파일 생성

```
# values-prometheus.yaml

server:
  enabled: true

persistentVolume:
  enabled: true
  accessModes:
    - ReadWriteOnce
  mountPath: /data
  size: 100Gi
  replicaCount: 1

  ## Prometheus data retention period (default if not specified is 15
  days)
  retention: "15d"
```

```
# values-grafana.yaml

replicas: 1

service:
  type: LoadBalancer #Local환경 이라면, NodePort로 설정한다.

persistence:
  type: pvc
  enabled: true
  # storageClassName: default
  accessModes:
    - ReadWriteOnce
  size: 10Gi
  # annotations: {}
  finalizers:
    - kubernetes.io/pvc-protection
```



```
# Administrator credentials when not using an existing secret (see below)
adminUser: admin
adminPassword: test1234
```

## 6.2. Prometheus, Grafana를 Helm chart로 설치

```
# monitoring.sh

#!/bin/bash

# https://may9noy.tistory.com/303
# Add helm repository
helm repo add prometheus-community
https://prometheus-community.github.io/helm-charts
helm repo add grafana https://grafana.github.io/helm-charts
helm repo update

# Deploy prometheus & grafana by helm
kubectl create ns monitoring
helm install prometheus prometheus-community/prometheus -f \
values-prometheus.yaml -n monitoring
helm install grafana grafana/grafana -f values-grafana.yaml -n monitoring

# 배포 확인
kubectl get pod,svc -n monitoring
```

## 6.3. Terminal로 배포된 모니터링 서비스 확인

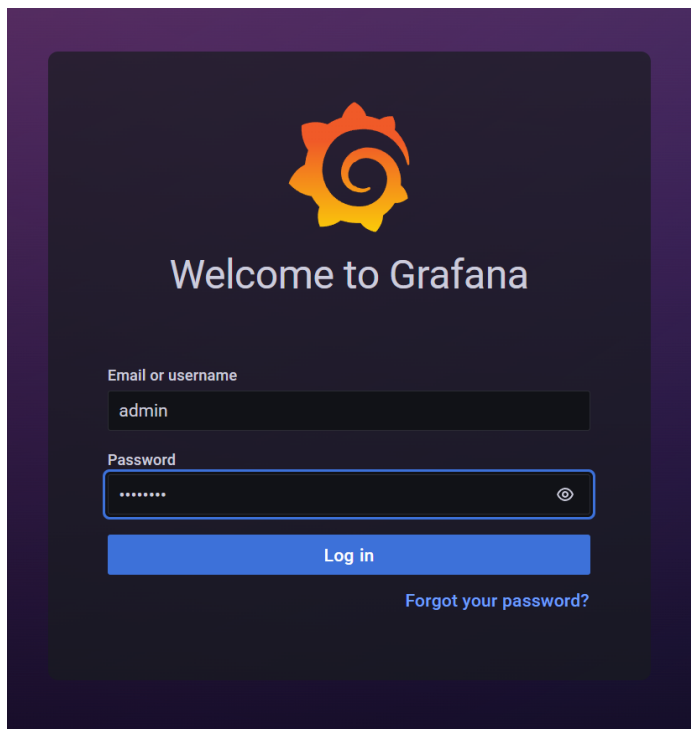
```
Administrator:~/environment $ helm list -A
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
grafana	monitoring	1	2022-11-20 10:39:12.153488377 +0000 UTC	deployed	grafana-6.44.6	9.2.5
prometheus	monitoring	1	2022-11-20 10:39:09.902837726 +0000 UTC	deployed	prometheus-16.0.0	2.39.1

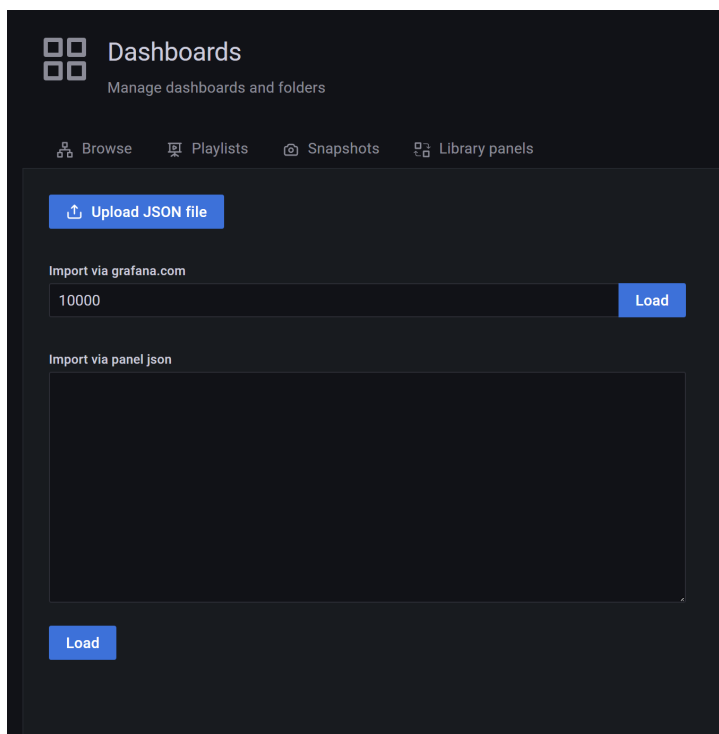
```
Administrator:~/environment $ k get svc -n monitoring
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	LoadBalancer	10.100.131.251	aaa903808d7ef4617aa16952c6f8e9da-1468717518.ap-northeast-2.elb.amazonaws.com	80:30321/TCP	42h
prometheus-alertmanager	ClusterIP	10.100.203.224	<none>	80/TCP	42h
prometheus-kube-state-metrics	ClusterIP	10.100.114.119	<none>	8080/TCP	42h
prometheus-prometheus-node-exporter	ClusterIP	10.100.216.232	<none>	9100/TCP	42h
prometheus-pushgateway	ClusterIP	10.100.182.175	<none>	9091/TCP	42h
prometheus-server	ClusterIP	10.100.114.14	<none>	80/TCP	42h

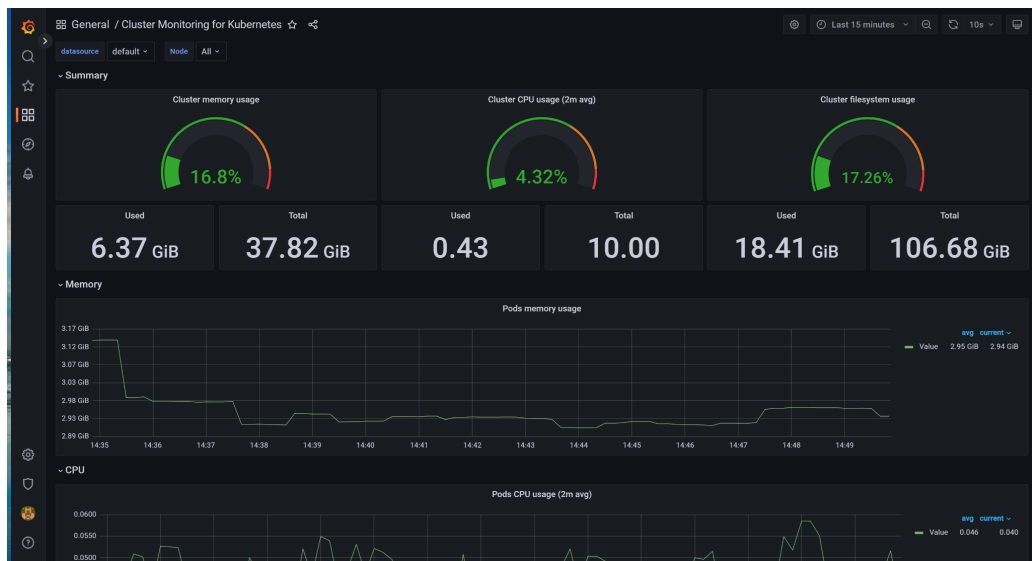
## 6.4. Grafana



## 6.5. 왼쪽 사이드바 - Dashboards - Import



## 6.6. Dashboard 생성 및 수정



## 7. CI/CD 구축

### 7.1. 프로젝트 레포지토리 생성

#### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* Goorm-Project-Aladin / Repository name \* web-app ✓

Great repository names are short and memorable. Need inspiration? How about [curly-octo-enigma](#)?

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None**

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None**

ⓘ You are creating a public repository in the Goorm-Project-Aladin organization.

**Create repository**

### 7.2. GitHub token 생성

GitHub Apps  
OAuth Apps  
**Personal access tokens**

#### New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

access token for github action

What's this token for?

**Expiration \***

30 days The token will expire on Sat, Aug 28 2021

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry

### 7.3. GitHub secret 설정

Actions secrets / New secret

---

Name

Value

PERSONAL ACCESS TOKEN HERE

Add secret

Actions secrets / New secret

---

Name

Value

AWS ACCESS KEY HERE

Add secret

Actions secrets / New secret

---

Name

Value

PERSONAL ACCESS TOKEN HERE

Add secret

## 7.4. 깃허브 액션을 위한 **build** 스크립트 작성

main web / .github / workflows / gradle.yml View runs Go to file ...

EC2 Default User Update github action build script ✓ Latest commit f274525 2 days ago History

1 contributor

53 lines (45 sloc) 1.69 KB Raw Blame

```

1 # This workflow uses actions that are not certified by GitHub.
2 # They are provided by a third-party and are governed by
3 # separate terms of service, privacy policy, and support
4 # documentation.
5 # This workflow will build a Java project with Gradle and cache/restore any dependencies to improve the workflow execution time
6 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-java-with-gradle
7
8 name: Java CI with Gradle
9
10 on:
11   push:
12     branches: [ "main" ]
13   pull_request:
14     branches: [ "main" ]
15
16 permissions:
17   contents: read
18
19 jobs:
20   build:
21     runs-on: ubuntu-latest
22     steps:
23       - uses: actions/checkout@v3
24
25       - name: Configure AWS credentials
26         uses: aws-actions/configure-aws-credentials@v1
27         with:
28           aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
29           aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
30           aws-region: ap-northeast-2
31
32       - name: Login to Amazon ECR
33         id: login-ecr
34         uses: aws-actions/amazon-ecr-login@v1
35
36       - name: Get image tag(version)
37         id: image
38         run: |
39           VERSION=$(echo ${ github.sha } | cut -c1-8)
40           echo VERSION=$VERSION
41           echo "::set-output name=version::$VERSION"
42
43       - name: Build, tag, and push image to Amazon ECR
44         id: image-info
45         env:
46           ECR_REGISTRY: ${ steps.login-ecr.outputs.registry }
47           ECR_REPOSITORY: aladin
48           IMAGE_TAG: ${ steps.image.outputs.version }
49         run: |
50           echo "::set-output name=ecr_repository::$ECR_REPOSITORY"
51           echo "::set-output name=image_tag::$IMAGE_TAG"
52           docker build -t $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG .
53           docker push $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG

```

## 7.5. Dockerfile 작성

```

1  ## Multi stage build
2  # Builder
3  FROM openjdk:11.0-jdk AS builder
4  LABEL description="Java Application builder"
5  RUN apt install git
6  RUN git clone https://github.com/Goorm-Project-Aladin/web.git
7  WORKDIR web
8  RUN chmod 700 gradlew
9  RUN ./gradlew clean build
10
11 # Runnig
12 FROM gcr.io/distroless/java:11
13 LABEL description="Java Application run image"
14 ARG JAR_FILE=/build/libs/web-0.0.1-SNAPSHOT.jar
15 COPY --from=builder web/${JAR_FILE} /app.jar
16 ENTRYPOINT ["java", "-jar", "/app.jar"]

```

## 7.6. 개발한 프로젝트를 깃허브에 **push**해 깃허브 액션을 실행

The screenshot shows the GitHub Actions interface for a workflow named 'Java CI with Gradle'. The specific run is '[+]HelloController #10', which is in a successful state. The left sidebar shows the 'Summary' tab selected, with a list of jobs including 'build'. The main area displays the details of the 'build' job, which succeeded yesterday in 1m 43s. The job steps are listed as follows:

- > Set up job
- > Run actions/checkout@v3
- > Configure AWS credentials
- > Login to Amazon ECR
- > Get image tag(version)
- > Build, tag, and push image to Amazon ECR
- > Post Login to Amazon ECR
- > Post Configure AWS credentials
- > Post Run actions/checkout@v3
- > Complete job

## 7.7. Docker로 빌드하고 ECR에 push된 이미지 확인

Amazon ECR > 리포지토리 > aladin

aladin 푸시 명령 보기 편집

이미지 (2) 🔄 삭제 스캔

🔍 이미지 찾기

<input type="checkbox"/>	이미지 태그 ▼	아티팩트 유형	푸시 위치 ▼	크기 (MB) ▼	이미지 URI	다이제스트	스캔 상태	취약성
<input type="checkbox"/>	84920182	Image	2022년 11월 21일, 17:33:29 (UTC+09)	113.19	URI 복사	sha256:774ae5a8cbd1ec...	완료	🟢 없음
<input type="checkbox"/>	f274525c	Image	2022년 11월 20일, 17:18:19 (UTC+09)	113.19	URI 복사	sha256:6a0477b21e2286...	완료	🟢 없음


## 7.8. GitOps 레포지토리 생성

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner \*


Repository name \*

 Goorm-Project-Aladin ▼ / GitOps-repo ✓

Great repository names are short and memorable. Need inspiration? How about [glowing-octo-lamp?](#)

Description (optional)

☐  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None ▼**

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None ▼**

 You are creating a private repository in the Goorm-Project-Aladin organization.

**Create repository**



## 7.9. Kubernetes에 ArgoCD 설치

```
# cicd.sh

#!/bin/bash

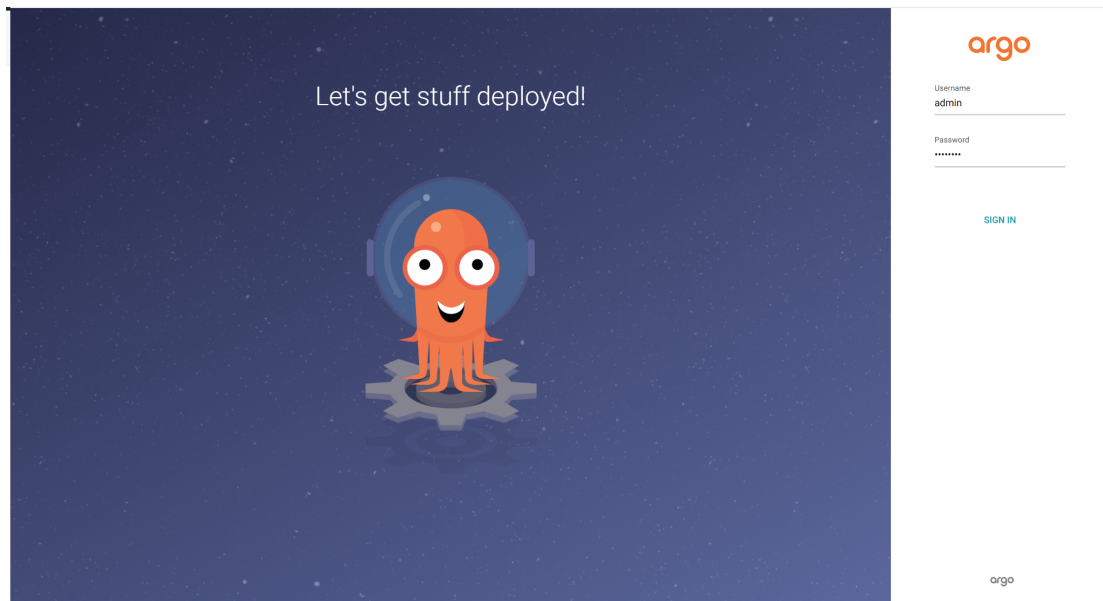
# EKS에 ArgoCD 설치
kubectl create namespace argocd
kubectl apply -n argocd -f
https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

# ELB 를 통해 접속 가능하도록 설정
kubectl patch svc argocd-server -n argocd -p '{"spec": {"type":
"LoadBalancer"}}'

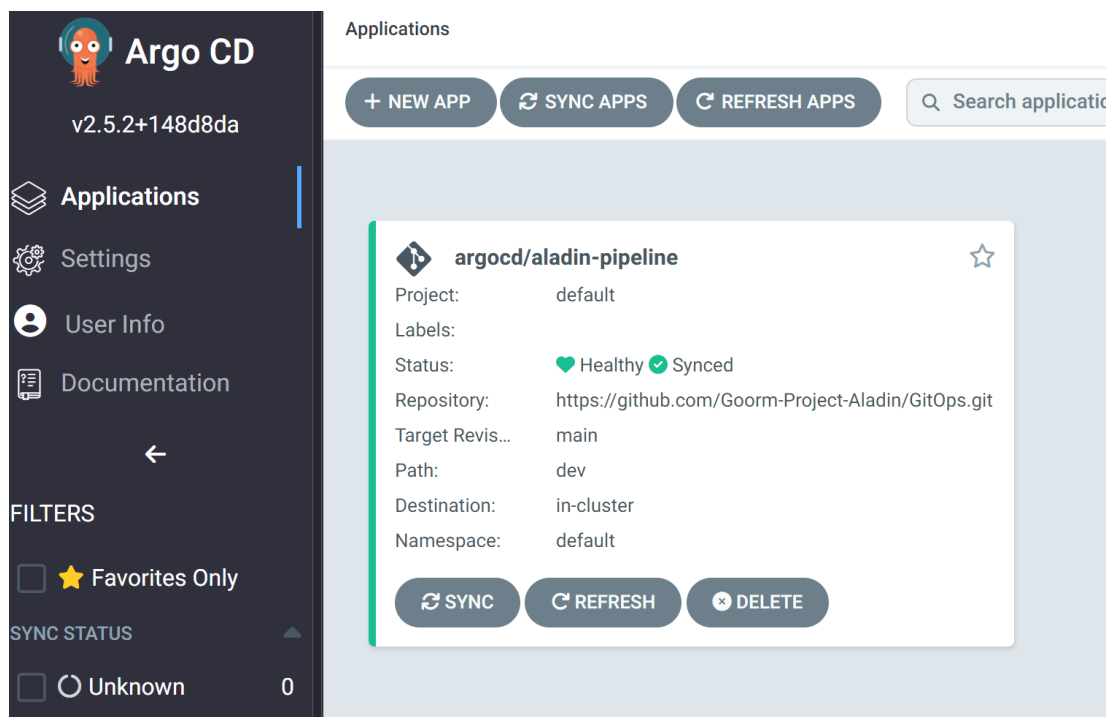
# ELB 주소 확인
export ARGOCD_SERVER=`kubectl get svc argocd-server -n argocd -o json |
jq --raw-output .status.loadBalancer.ingress[0].hostname`
echo $ARGOCD_SERVER

# ArgoCD password 확인
ARGO_PWD=`kubectl -n argocd get secret argocd-initial-admin-secret -o
jsonpath="{.data.password}" | base64 -d`
echo $ARGO_PWD
```

### 7.10. ArgoCD 로그인



### 7.11. ArgoCD에 GitOps 레포지토리 연결



## 7.12. 애플리케이션 배포를 위한 **manifest**를 작성해 **GitOps** 레포지토리에 **push**

main	GitOps / dev /	Go to file	Add file	...
britko Update port a78acdb 1 hour ago History				
..				
✓ deployment.yaml	Update port	1 hour ago		
✓ hpa.yaml	Update manifests	4 hours ago		
✓ service.yaml	Update port	1 hour ago		

## 7.13. **GitOps** 레포지토리의 변경 사항을 **ArgoCD**가 확인하고 **Kubernetes**에 Sync

