4-2. Pod 및 Service 배포

클라우드 네이티브 운영체계

Exported on 09/15/2022

Table of Contents

1 사전 준비

1.1 docker 설치

bastion에서 하기 명령어 수행

```
sudo yum install docker -y
sudo service docker start

##sudo를 사용하지 않고도 Docker 명령을 실행할 수 있도록 docker 그룹에 ec2-user를 추가
sudo usermod -a -G docker ec2-user
```

1.2 eksctl 설치

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/
download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin

eksctl version
```

1.3 helm 설치

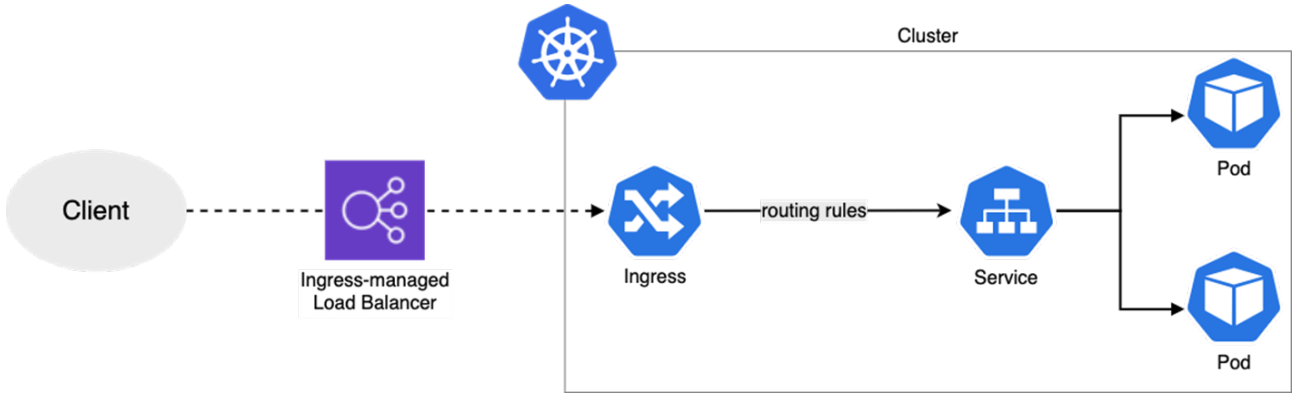
```
curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2

helm version
```

1.4 실습 파일 다운로드

```
mkdir source
cd source
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/Dockerfile
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/pod.yaml
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/web-ingress.yaml
```

2 구성


Cluster
Client
Ingress-managed
Load Balancer
Ingress
routing rules
Service
Pod
Pod

3 ECR 이미지 생성

docker 이미지 생성

```
cd source
docker build -t image-sts-httpd .

docker images --filter reference=image-sts-httpd
```

```
[ec2-user@ip-10-0-2-156 source]$ docker images --filter reference=image-sts-httpd
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
image-sts-httpd     latest    3ac7330c6f0b   2 hours ago     202MB
```

ECR Repository 생성

```
aws ecr create-repository --repository-name repository-sts-httpd --region ap-
northeast-2
```

리포지토리 이름	URI	생성 날짜	태그 변경 불가능	스캔 빈도	암호화 유형	풀스루 캐시
repository-sts-httpd	694833324522.dkr.ecr.ap-northeast-2.amazonaws.com/repository-sts-httpd	2022년 9월 14일, 21:11:36 (UTC+09)	비활성화됨	수동	AES-256	비활성

생성된 이미지 태깅

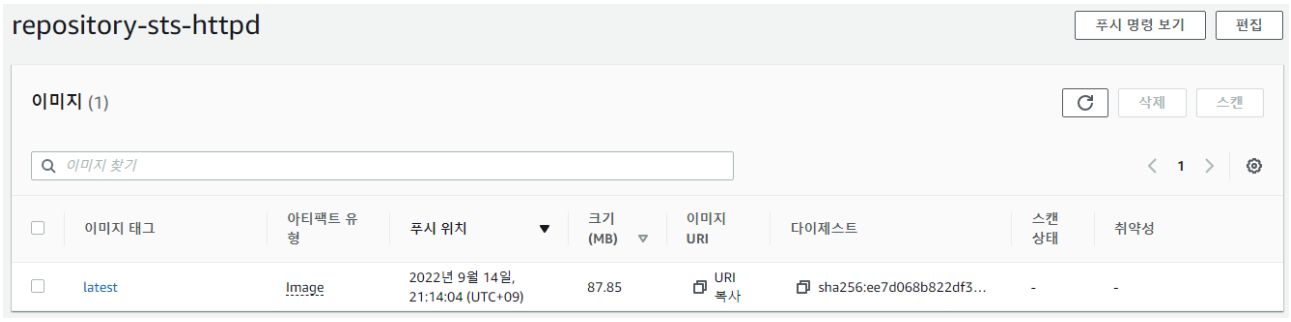
```
docker tag image-sts-httpd [aws_account_id].dkr.ecr.ap-northeast-2.amazonaws.com/
repository-sts-httpd
```

로그인

```
aws ecr get-login-password | docker login --username AWS --password-stdin
[aws_account_id].dkr.ecr.ap-northeast-2.amazonaws.com
```

이미지 푸시

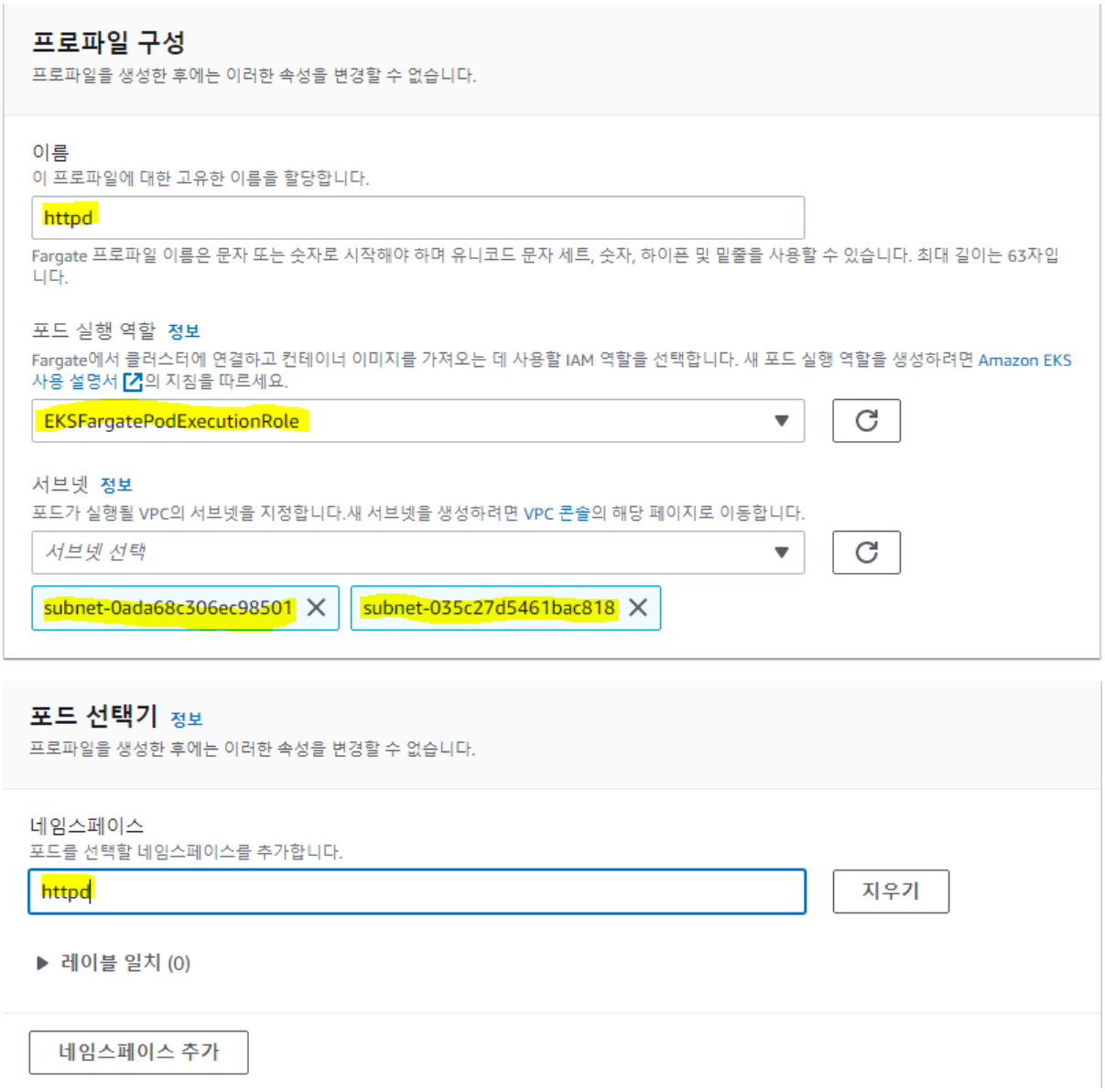
```
docker push [aws_account_id].dkr.ecr.ap-northeast-2.amazonaws.com/repository-sts-
httpd
```


repository-sts-httpd
푸시 명령 보기
편집
이미지 (1)
삭제
스캔
이미지 찾기
1
이미지 태그
아티팩트 유형
푸시 위치
크기 (MB)
이미지 URI
다이제스트
스캔 상태
취약성
latest
Image
2022년 9월 14일, 21:14:04 (UTC+09)
87.85
URI 복사
sha256:ee7d068b822df3...
-
-

4 Pod 및 Service 배포

4.1 Fargate Profile 추가

httpd 네임스페이스에 대한 profile 추가



4.2 NameSpace 생성

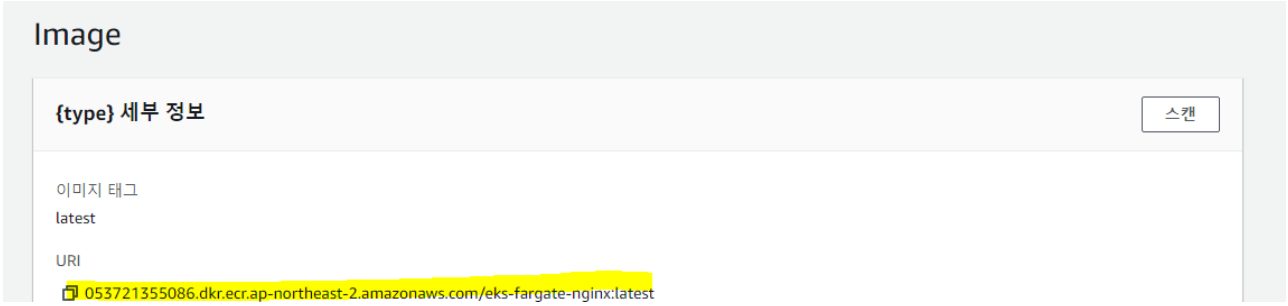
```
kubectl create namespace httpd
kubectl get ns
```

4.3 Pod 배포

이미지 수정

```
vi pod.yaml
```

ECR 이미지 URI 복사




```
apiVersion: v1
kind: Pod
metadata:
  name: web
  namespace: nginx
spec:
  containers:
  - image: 053721355086.dkr.ecr.ap-northeast-2.amazonaws.com/eks-fargate-nginx:latest
    name: web
    ports:
    - containerPort: 7080
```

```
kubectl apply -f pod.yaml
kubectl get pod -n httpd
```

4.4 Service 배포

```
kubectl expose pod web --name=web-svc -n httpd
```

```
kubectl get service -n httpd
```

5 LoadBalancer 배포

5.1 LoadBalancer Controller 설치

IAM 에서 Identity providers 에 oidc 생성

```
eksctl utils associate-iam-oidc-provider --region=ap-northeast-2 --
cluster=[clustername] --approve
```

policy 생성

```
curl -o iam_policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-
balancer-controller/v2.4.3/docs/install/iam_policy.json

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

aws-load-balancer에 대한 iam serviceaccount 권한 생성

```
eksctl create iamserviceaccount \
  --cluster=[clustername] \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name "AmazonEKSLoadBalancerControllerRole" \
  --attach-policy-arn=arn:aws:iam::[계정]:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

aws-load-balancer-controller 생성

```
helm repo add eks https://aws.github.io/eks-charts
helm repo update

helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
  -n kube-system \
  --set clusterName=[clustername] \
  --set serviceAccount.create=false \
  --set serviceAccount.name=aws-load-balancer-controller  \
  --set region=ap-northeast-2 \
  --set vpcId=[VPCID]
```

```
kubectl get pods -A |grep load
```

```
[ec2-user@ip-10-0-4-88 source]$ kubectl get pods -A |grep load
kube-system   aws-load-balancer-controller-5c54c5df45-s7rfq   1/1     Running   0          79s
kube-system   aws-load-balancer-controller-5c54c5df45-z78mj   1/1     Running   0          79s
```

5.2 Ingress 배포

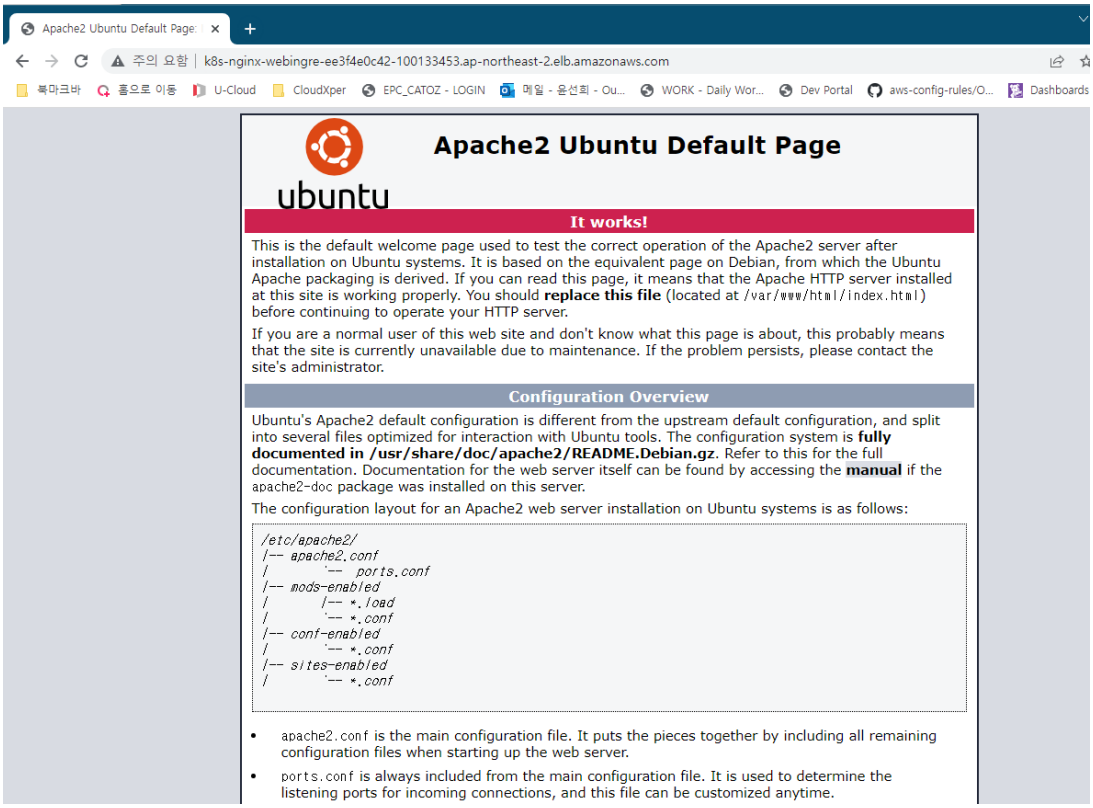
```
vi web-ingress.yaml
```

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: httpd
  name: web-ingress
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/target-type: ip
    alb.ingress.kubernetes.io/load-balencer-name: ingress-web-alb
    alb.ingress.kubernetes.io/subnets: subnet-098e1c8fb4e06a78b, subnet-02349c9370fbf6af5
spec:
  rules:
  - http:
      paths:
      - backend:
          service:
            name: web-svc
            port:
              number: 7080
        pathType: ImplementationSpecific
```

퍼블릭 서브넷 ID로 수정한다.

```
kubectl apply -f web-ingress.yaml
kubectl get ingress -n httpd
```

```
[ec2-user@ip-10-0-4-88 source]$ kubectl get ingress -n httpd
NAME          CLASS    HOSTS   ADDRESS                                                                 PORTS   AGE
web-ingress   <none>   *       k8s-nginx-webingre-ee3f4e0c42-100133453.ap-northeast-2.elb.amazonaws.com   80      72s
```


Apache2 Ubuntu Default Page:
주의 요함 | k8s-nginx-webingre-ee3f4e0c42-100133453.ap-northeast-2.elb.amazonaws.com
북마크바
홈으로 이동
U-Cloud
CloudXper
EPC_CATOZ - LOGIN
메일 - 윤선희 - Ou...
WORK - Daily Wor...
Dev Portal
aws-config-rules/O...
Dashboards
Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.