

4-3. EFS 볼륨 생성 및 PV, PVC 생성

클라우드 네이티브 운영체제

Exported on 09/22/2022

Table of Contents

1	사전준비	3
1.1	실습 파일 다운로드	3
1.2	EFS용 보안 그룹 생성	3
2	EFS 생성	4
2.1	Bastion에 EFS Mount	6
3	PV, PVC 생성	7
4	Deployment 배포	8
4.1	디렉토리 로깅 확인	8
5	ReplicaSet 조정	10
6	Service 및 Ingress 배포	11

1 사전준비

1.1 실습 파일 다운로드

```
cd source
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/pv-pvc.yaml
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/deployment.yaml
wget https://eks-fargate-test.s3.ap-northeast-2.amazonaws.com/ingress-deployment.yaml
```

1.2 EFS용 보안 그룹 생성

VPC IP 범위를 소스로 NFS 를 허용하는 inbound 규칙을 만든다.

인바운드 규칙 편집 [정보](#)

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 [정보](#)

보안 그룹 규칙 ID

-

유형 [정보](#)

NFS

프로토콜 [정보](#)

TCP

포트 범위 [정보](#)

2049

소스 [정보](#)

사용자 지정

Q

10.0.0.0/16 X

설명 - 선택 사항 [정보](#)

vpc

삭제

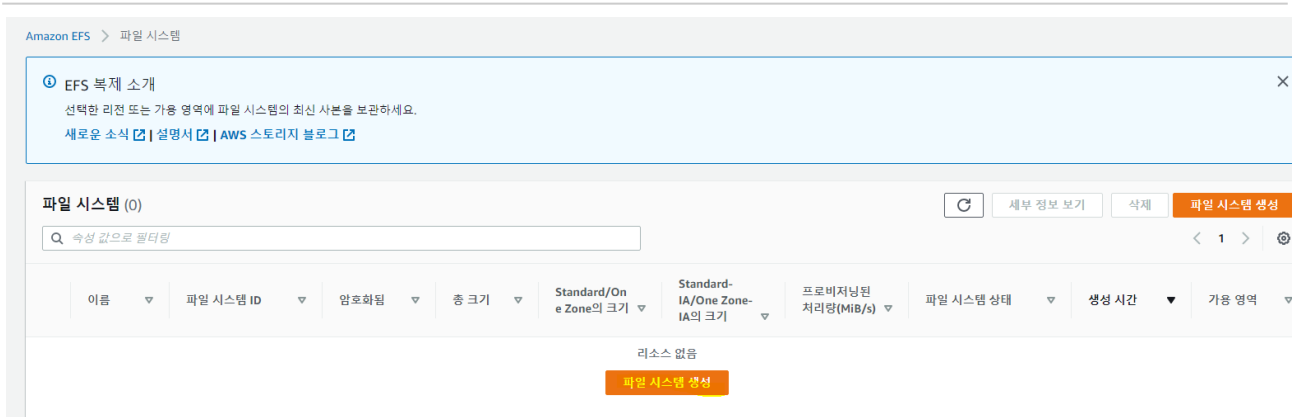
규칙 추가

취소

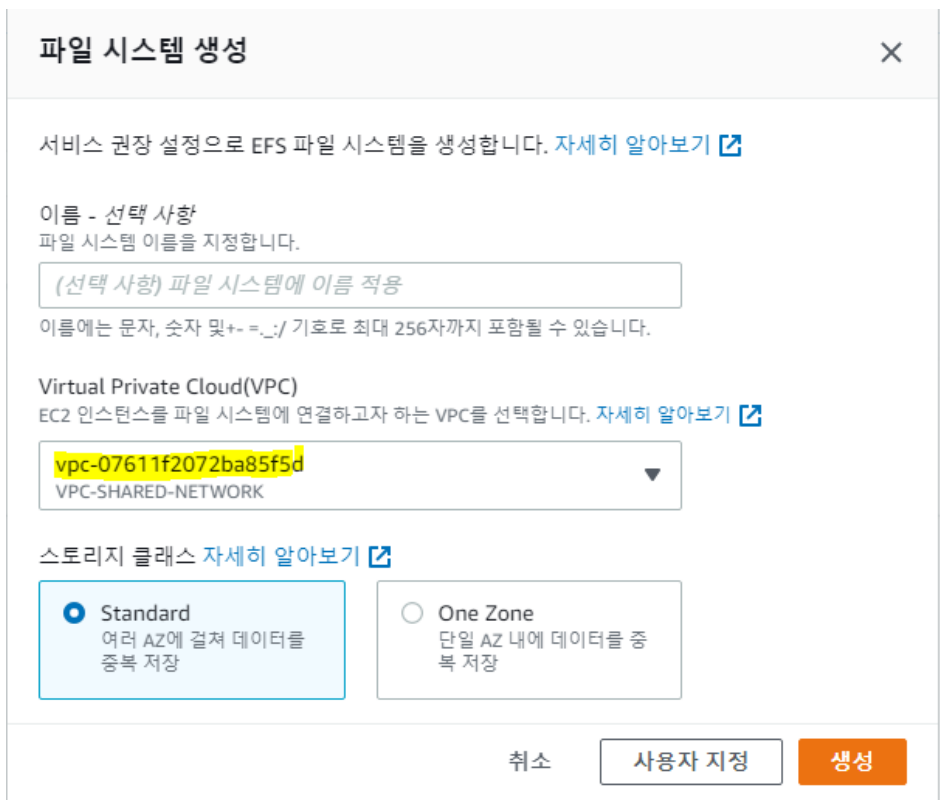
변경 사항 미리 보기

규칙 저장

2 EFS 생성



VPC 선택



사용자 지정 선택

기본 설정 유지 후 다음

Amazon EFS > 파일 시스템 > 생성

1단계
파일 시스템 설정

2단계
네트워크 액세스

3단계 - 선택 사항
파일 시스템 정책

4단계
검토 및 생성

파일 시스템 설정

일반

이름 - 선택 사항
파일 시스템 이름을 지정합니다.
(선택 사항) 파일 시스템에 이름 적용
이름에는 문자, 숫자 및 +, -, _/ 기호로 최대 256자까지 포함될 수 있습니다.

스토리지 클래스 자세히 알아보기

☒ Standard
여러 AZ에 걸쳐 데이터를 중복 저장

☐ One Zone
단일 AZ 내에 데이터를 중복 저장

자동 백업
권장 설정을 사용하여 AWS Backup으로 파일 시스템 데이터를 자동으로 백업합니다. 추가 요금이 적용됩니다. 자세히 알아보기

☒ 자동 백업 활성화

수명 주기 관리
EFS Intelligent-Tiering은 수명 주기 관리를 사용하여 파일을 Standard와 Standard-Infrequent Access 스토리지 클래스 사이에서 이동함으로써 애플리케이션에 적합한 가격 및 성능 볼륨드를 자동으로 실현합니다. 자세히 알아보기

IA로 전환
파일을 Standard에서 Standard-Infrequent Access로 전환합니다.

마지막 액세스 이후 30일 경과

IA 외부로 전환
파일을 Standard-Infrequent Access에서 Standard로 전환합니다.

저음 액세스할 때

성능 모드

private subnet 선택

사전 생성한 보안그룹 선택

Amazon EFS > 파일 시스템 > 생성

1단계
파일 시스템 설정

2단계
네트워크 액세스

3단계 - 선택 사항
파일 시스템 정책

4단계
검토 및 생성

네트워크 액세스

네트워크

Virtual Private Cloud(VPC)
EC2 인스턴스를 파일 시스템에 연결하고자 하는 VPC를 선택합니다. 자세히 알아보기

vpc-07611f2072ba85f5d
VPC-SHARED-NETWORK

타겟 대상
타겟 대상은 Amazon EFS 파일 시스템을 탑재할 수 있는 NFSv4 엔드포인트를 제공합니다. 가용 영역마다 타겟 대상 한 개씩 생성하는 것이 좋습니다. 자세히 알아보기

가용 영역	서브넷 ID	IP 주소	보안 그룹
ap-northeast-2a	subnet-039f50f7ec60ba83b	자동	보안 그룹 선택 sg-0b97fc9f627a558ff default
ap-northeast-2b	subnet-075e502461d7e1b5c	자동	보안 그룹 선택 sg-0b97fc9f627a558ff default

타겟 대상 추가

취소

기본설정 그대로 둔 후 생성

Amazon EFS > 파일 시스템 > 생성

1단계
파일 시스템 설정

2단계
네트워크 액세스

3단계 - 선택 사항
파일 시스템 정책

4단계
검토 및 생성

파일 시스템 정책 - 선택 사항

정책 옵션

이러한 공통 정책 옵션 중 하나 이상을 선택하거나 편집기를 사용하여 사용자 지정 정책을 하나 더 생성합니다. 자세히 알아보기

- ☐ 기본적으로 루트 액세스 금지*
- ☐ 기본적으로 읽기 전용 액세스 적용*
- ☐ 익명 액세스 금지
- ☐ 모든 클라이언트에 대해 전송 중 암호화 적용

*각각 중앙 기본 정책은 이러한 기본 권한을 대체할 수 있습니다.

▶ 추가 권한 부여

정책 편집기 (JSON)

지우기

1

수동으로 변경하면 편집기를 버릴 때까지 왼쪽의 정책 옵션을 사용할 수 없습니다.

취소

생성 완료

파일 시스템 (1)									
<input type="text" value="Q 속성 값으로 필터링"/> < 1 >									
이름	파일 시스템 ID	암호화됨	총 크기	Standard/One Zone의 크기	Standard-IA/One Zone-IA의 크기	프로비저닝된 처리량(MiB/s)	파일 시스템 상태	생성 시간	가용 영역
-	fs-0562d5b421d609359	암호화됨	6.00 KiB	6.00 KiB	0바이트	-	사용 가능	Wed, 21 Sep 2022 06:35:43 GMT	Standard

2.1 Bastion에 EFS Mount

Amazon EFS > 파일 시스템 > fs-0562d5b421d609359

fs-0562d5b421d609359

일반

성능 모드
범용

처리량 모드
버스트

수명 주기 관리
IA로 전환: 마지막 액세스 이후 30일 경과
IA 외부로 전환: 처음 액세스할 때

가용 영역
Standard

자동 백업
활성화됨

암호화됨
06500dc0-85ee-4527-ad07-8821afae0ea6 (aws/elasticfilesystem)

파일 시스템 상태
사용 가능

DNS 이름
fs-0562d5b421d609359.efs.ap-northeast-2.amazonaws.com

bastion에 EFS Mount 하여 파일 확인 가능하도록

```
mkdir ~/efs-mount-point
sudo mount -t nfs -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport {mount-
target-DNS 입력}:/ ~/efs-mount-point
cd ~/efs-mount-point
ls -al

sudo chmod go+rw .

df -h
```

3 PV, PVC 생성

```
vi pv-pvc.yaml
```

파일 수정

```
metadata:
  namespace: sunny
  name: sts-efs-pv-sunny
spec:
  capacity:
    storage: 10Gi
  volumeMode: Filesystem
  accessModes:
    - ReadwriteMany
  storageClassName: ""
  persistentVolumeReclaimPolicy: Retain
  csi:
    driver: efs.csi.aws.com
    volumeHandle: fs-0562d5b421d609359
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  namespace: sunny
  name: sts-efs-pvc-sunny
spec:
  accessModes:
    - ReadwriteMany
  storageClassName: ""
  volumeName: sts-efs-pv-sunny
  resources:
    requests:
      storage: 10Gi
```

```
kubectl apply -f pv-pvc.yaml
```

```
kubectl get -n 네임스페이스 pv
```

```
kubectl get -n 네임스페이스 pvc
```

4 Deployment 배포

```
vi deployment.yaml
```

수정

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: sunny
  name: deploy-web
spec:
  replicas: 1
  selector:
    matchLabels:
      app: echo
  template:
    metadata:
      labels:
        app: echo
    spec:
      containers:
        - name: deploy-app
          image: 053721355086.dkr.ecr.ap-northeast-2.amazonaws.com/repository-httpd-sunny:latest
          ports:
            - containerPort: 7080
          command: ["/bin/sh"]
          args: ["-c", "while true; do echo $HOSTNAME $(date -u) >> /deploy-app/deploy-out.txt; sleep 5; done"]
          volumeMounts:
            - mountPath: /deploy-app
              name: efs-storage
      volumes:
        - name: efs-storage
          persistentVolumeClaim:
            claimName: sts-efs-pvc-sunny
```

```
kubectl apply -f deployment.yaml
```

```
kubectl get -n 네임스페이스 pod
```

```
kubectl get -n 네임스페이스 deployment
```

4.1 디렉토리 로깅 확인

```
cd ~/efs-mount-point
ls -al
```

파드내에 접속해서 로그가 쌓이는 경로에 들어가보자

```
kubectl exec -it pod명 -n 네임스페이스명 -- /bin/bash
```

```
cd /deploy-app
ls
```


아파치가 기동되어 있지 않기 때문에 기동시켜 주자 (pod 내에서 수행)

```
/usr/sbin/apache2ctl start
```

5 ReplicaSet 조정

```
kubectl get -n 네임스페이스 rs
kubectl get -n 네임스페이스 pod
```

replicaset 크기 조정

```
kubectl scale deployment deploy-web --replicas=3 -n 네임스페이스
```

변경 확인

```
kubectl get -n 네임스페이스 rs
kubectl get -n 네임스페이스 pod
```

6 Service 및 Ingress 배포

서비스 배포

```
kubectl expose deployment deploy-web --port=7080 --target-port=7080 --name=service-deploy-web -n 네임스페이스

kubectl get -n 네임스페이스 service
```

파일 수정

```
vi ingress-deployment.yaml
```

namespace, public subnets 입력

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: sunny
  name: deploy-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-balancer-name: ingress-web-alb-sunny
    alb.ingress.kubernetes.io/subnets: subnet-098e1c8fb4e06a78b, subnet-02349c9370fbf6aff5
spec:
  rules:
  - http:
      paths:
      - backend:
          service:
            name: service-deploy-web
            port:
              number: 7080
        pathType: ImplementationSpecific
```

```
kubectl apply -f ingress-deployment.yaml


kubectl get -n 네임스페이스 ingress
```

아래 페이지를 확인하자

확인되지 않는다면 파드내에 접속하여 아파치 기동

```
kubectl exec -it pod명 -n 네임스페이스명 -- /bin/bash

/usr/sbin/apache2ctl start
```



Apache2 Ubuntu Default Page

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.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `./usr/bin/apache2` directly will not work with the default configuration.**