

WAS 설정, ELB 생성, 설정 후 연결하기

WAS AMI 빌드

PM2 부팅(재기동) 자동 실행 등록

AMI 빌드시 자동으로 PM2로 node.js 프로젝트(WAS)를 실행하기 위해 등록하는 과정입니다.

```
pm2 startup
```

해당 명령어 실행 시

```
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=root
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/root/.pm2
PIDFile=/root/.pm2/pm2.pid

ExecStart=/usr/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

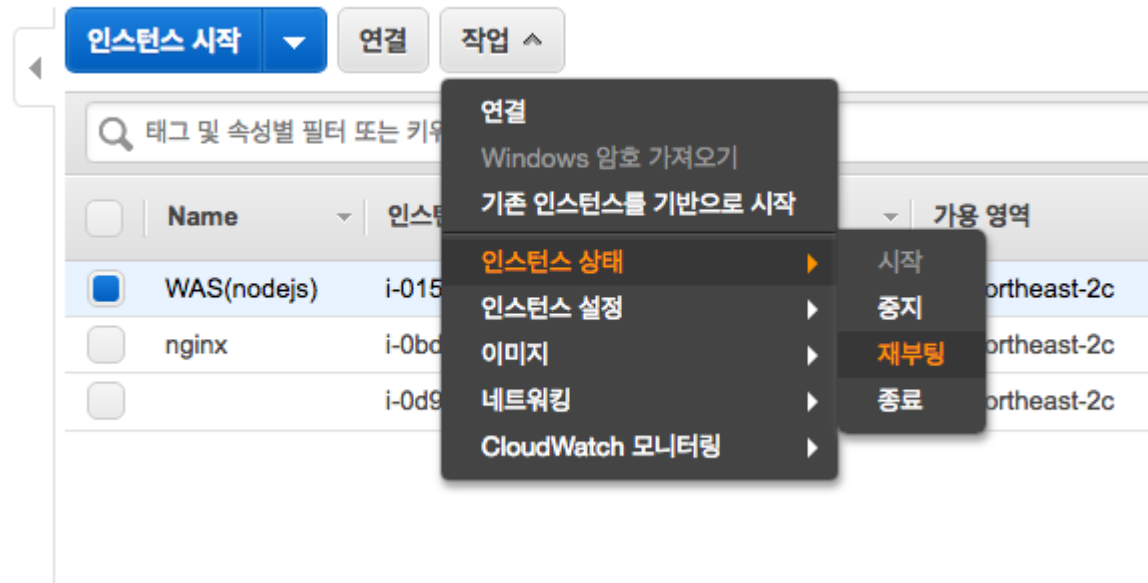
Target path
/etc/systemd/system/pm2-root.service
Command list
[ 'systemctl enable pm2-root' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-root.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-root...
[PM2] [v] Command successfully executed.
+-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save
```

위와 같이 자동으로 시스템 부팅(재기동) 시 현재 구동되고 있는 pm2 프로세스들을 기록하여 기동하는 스크립트가 작성됩니다. 이를 저장하기 위해

```
pm2 save
```

명령어를 실행하면 아래와 같이 위에서 자동으로 작성된 스크립트가 저장됩니다.

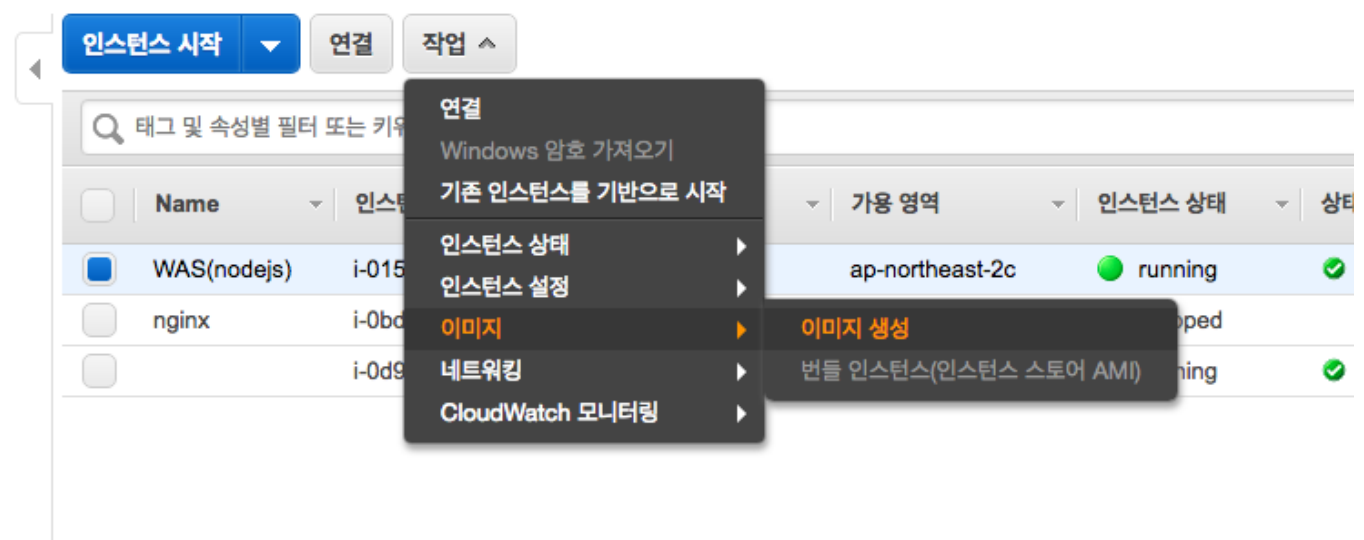
```
[PM2] Saving current process list...
[PM2] Successfully saved in /root/.pm2/dump.pm2
```



AWS 콘솔 -> WAS가 구동되는 인스턴스 선택 -> 작업 -> 재부팅

방금 설정한 WAS가 구동되는 인스턴스를 재기동한 뒤 POSTMAN으로 재기동해도 API 요청/응답이 정상적으로 작동하는 지 확인합니다.

WAS 설정 기반 AMI 빌드



AMI 빌드를 위해 기반으로 할 인스턴스 선택(윗 단계에서 설정한 WAS) -> 작업 -> 이미지 -> 이미지 생성

이미지 생성

인스턴스 ID

i-0155ca5e85831f342

이미지 이름

was-nodejs-ami

이미지 설명

nodejs + pm2 + fastcampus-api-deploy

재부팅 안 함

☐

인스턴스 볼륨

볼륨 유형	디바이스	스냅샷	크기(GiB)	볼륨 유형	IOPS	처리량(MB/초)	종료 시 삭제	암호화
	/dev/xvda	snap-0e3cd46cca462068c	8	범용 SSD(GP2)	100/3000	해당 사항 없음	<input checked="" type="checkbox"/>	암호화되지 않음

새 볼륨 추가

EBS 볼륨의 전체 크기: 8 GiB

EBS 이미지를 생성할 때 위의 각 볼륨에 대해 EBS 스냅샷이 생성됩니다.

취소

이미지 생성

1. 이미지 이름 입력(ex. was-nodejs-ami)
2. 이미지 설명(ex. nodejs + pm2 + fastcampus-api-deploy)
3. 이미지 생성(3~7분 대기)

AMI(WAS) 기반 EC2 인스턴스 생성

1. AMI 선택

2. 인스턴스 유형 선택

3. 인스턴스 구성

4. 스토리지 추가

5. 태그 추가

6. 보안 그룹 구성

7. 검토

빠른 시작

나의 AMI

AWS Marketplace

커뮤니티 AMI

소유권

☒ 내 소유

☐ 나와 공유됨

아키텍처

☐ 32비트

Q 나의 AMI 검색

ec2-nginx-ami - ami-0120fdd8943f2549f

amazon linux2 + nginx1.12 + Fastcampus-web-deploy

루트 디바이스 유형: ebs 가상화 유형: hvm 소유자: 322749112518

선택

64비트

was-nodejs-ami - ami-0d6d1cb4a3bf7afeb

nodejs + npm + pm2 + fastcampus-api-deploy

루트 디바이스 유형: ebs 가상화 유형: hvm 소유자: 322749112518

선택

64비트

1. EC2 인스턴스 생성하기
2. AMI 선택 탭에서 왼쪽 탭 두번째에 위치한 나의 AMI 선택
3. 방금 생성한 was-nodejs-ami 선택

기존 EC2 인스턴스 생성 방법과 동일합니다. 6. 보안 그룹 구성까지 기본 값을 사용합니다.

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 자세히 알아보기

보안 그룹 할당: ☐ 새 보안 그룹 생성
☒ 기존 보안 그룹 선택

보안 그룹 ID	이름	설명	작업
<input type="checkbox"/> sg-06d43ed8bd192964	aws-ec2-secure	secure test interface	새로 복사
<input type="checkbox"/> sg-2c8efe46	default	default VPC security group	새로 복사
<input type="checkbox"/> sg-0158a6749c6c1230a	launch-wizard-1	launch-wizard-1 created 2018-07-14T22:49:31.157+09:00	새로 복사
<input type="checkbox"/> sg-0ca09d09c7badcb91	nginx-lb-secure	elb nginx secure group	새로 복사
<input checked="" type="checkbox"/> sg-01c3ef7223e5fa0e4	nodejs-was-secure	nodejs was secure group	새로 복사

sg-01c3ef7223e5fa0e4에 대한 인바운드 규칙 (선택한 보안 그룹: sg-01c3ef7223e5fa0e4)

유형	프로토콜	포트 범위	소스	설명
사용자 지정 TCP 규칙	TCP	8080	0.0.0.0/0	
사용자 지정 TCP 규칙	TCP	8080	:::0	
SSH	TCP	22	0.0.0.0/0	

[취소](#) [이전](#) [검토 및 시작](#)

해당 WAS AMI는 8080포트로 통신합니다. 기존 생성된 보안 그룹인 nodejs-was-secure을 사용합니다. (8080포트 / 22포트 - SSH를 확인합니다)

생성이 끝나시면 생성이 완료된 DNS 주소로 8080포트를 통해 POSTMAN으로 API 요청/응답이 제대로 작동하는 지 확인합니다.

ELB를 생성하여 두 대의 WAS 인스턴스 연결

- 로드밸런서 탭 -> 로드밸런서 생성
- 맨 오른쪽 Classic load balancer 생성 버튼 클릭

aws 서비스 리소스 그룹

1. 로드 밸런서 정의 2. 보안 그룹 할당 3. 보안 설정 구성 4. 상태 검사 구성 5. EC2 인스턴스 추가 6. 태그 추가 7. 검토

단계 1: 로드 밸런서 정의

기본 구성

이 마법사는 새 로드 밸런서를 설정하는 방법을 안내합니다. 먼저 새 로드 밸런서를 다른 로드 밸런서와 구별할 수 있도록 고유한 이름을 지정하는 것부터 시작합니다. 또한 로드 밸런서에 포트 및 프로토콜도 구성해야 합니다. 클라이언트의 트래픽은 로드 밸런서 포트부터 EC2 인스턴스의 포트까지 라우팅됩니다. 기본적으로 로드 밸런서는 포트 80에서 표준 웹 서버로 구성되어 있습니다.

로드 밸런서 이름: was-lb

LB 생성할 VPC: 내 기본 VPC (172.31.0.0/16)

내부 로드 밸런서 생성: ☐ (자세히 알아보기)

고급 VPC 구성 활성화: ☐

리스너 구성:

로드 밸런서 프로토콜	로드 밸런서 포트	인스턴스 프로토콜	인스턴스 포트
HTTP	8080	HTTP	8080

추가

- 로드 밸런서 이름 입력 (ex. was-lb)
- 하단 리스너 구성
 - 로드밸런서 포트와 인스턴스 포트를 각각 8080포트로 입력해줍니다.

1. 로드 밸런서 정의 2. 보안 그룹 할당 3. 보안 설정 구성 4. 상태 검사 구성 5. EC2 인스턴스 추가 6. 태그 추가 7. 검토

단계 2: 보안 그룹 할당

VPC에서 탄력적 로드 밸런서를 사용하는 옵션을 선택하셨습니다. 그러므로 로드 밸런서에 보안 그룹을 할당할 수 있습니다. 이 로드 밸런서에 할당할 보안 그룹을 선택하십시오. 이 선택은 언제든지 변경할 수 있습니다.

보안 그룹 할당: ☐ 새 보안 그룹 생성
☒ 기존 보안 그룹 선택

필터 VPC 보안 그룹

보안 그룹 ID	이름	설명	작업
<input type="checkbox"/> sg-06d43ed8fbd192964	aws-ec2-secure	secure test interface	새로 복사
<input type="checkbox"/> sg-2c8efe46	default	default VPC security group	새로 복사
<input type="checkbox"/> sg-0158a6749c6c1230a	launch-wizard-1	launch-wizard-1 created 2018-07-14T22:49:31.157+09:00	새로 복사
<input type="checkbox"/> sg-0ca09d09c7badcb91	nginx-lb-secure	elb nginx secure group	새로 복사
<input checked="" type="checkbox"/> sg-01c3ef7223e5fa0e4	nodejs-was-secure	nodejs was secure group	새로 복사

- 8080포트가 뚫려있는 기존 생성한 보안 그룹인 nodejs-was-secure를 선택합니다.
4. 상태 검사 구성까지 이동합니다



1. 로드 밸런서 정의 2. 보안 그룹 할당 3. 보안 설정 구성 4. 상태 검사 구성 5. EC2 인스턴스 추가 6. 태그 추가 7. 검토

단계 4: 상태 검사 구성

로드 밸런서는 자동으로 EC2 인스턴스에서 상태 검사를 수행하며 상태 검사를 통과하는 인스턴스로만 트래픽을 라우팅합니다. 상태 검사에 실패

Ping 프로토콜 HTTP

Ping 포트 8080

Ping 경로 /

고급 세부 정보

응답 시간 초과 ⓘ 5 초

간격 ⓘ 30 초

비정상 임계값 ⓘ 2

정상 임계값 ⓘ 10

- 헬스 체크 설정 내역입니다.
- 앞서 설정한 포트인 8080을 Ping 포트 기본값으로 설정되었습니다.
- Ping 경로를 /index.html이 아닌 API url /로 변경합니다.
- 5. EC2 인스턴스 추가로 이동합니다.

서비스

리소스 그룹

★

1. 로드 밸런서 정의

2. 보안 그룹 할당

3. 보안 설정 구성

4. 상태 검사 구성

5. EC2 인스턴스 추가

6. 태그 추가

7. 검토

단계 5: EC2 인스턴스 추가

아래 표에는 모든 실행 중인 EC2 인스턴스 목록이 있습니다. 현재 로드 밸런서에 인스턴스를 추가하려면 선택 열에서 확인란을 선택하십시오.

VPC vpc-ec151b84 (172.31.0.0/16)

<input type="checkbox"/>	인스턴스	이름	상태	보안 그룹
<input checked="" type="checkbox"/>	i-0155ca5e85831f342	WAS(nodejs)	running	nodejs-was-secure
<input type="checkbox"/>	i-0bd9a5d98e49b444e	nginx	stopped	aws-ec2-secure
<input checked="" type="checkbox"/>	i-0d9a274d7fe06fa3a		running	nodejs-was-secure

- 저희는 WAS 인스턴스를 연결할 로드밸런서를 설정하고 있습니다. WAS가 구동되고 있는 두 개의 인스턴스를 선택합니다.
- 인스턴스의 이름을 미리 설정해두면 헛갈리지 않습니다.
- 혹시 이름을 설정해두지 않으셨다면 보안 그룹에 nodejs-was-secure가 설정된 인스턴스들만 선택합니다.
- 7. 검토 탭까지 이동해서 생성 버튼을 클릭합니다.

로드 밸런서 생성

작업

태그 및 속성별 필터 또는 키워드별 검색

<input type="checkbox"/>	이름	DNS 이름	상태	VPC ID	가용 영역
<input type="checkbox"/>	nginx-lb	nginx-lb-2007386994.ap-nort...		vpc-ec151b84	ap-northeast
<input checked="" type="checkbox"/>	was-lb	was-lb-1986321715.ap-north...		vpc-ec151b84	ap-northeast

로드 밸런서: was-lb

설명

인스턴스

상태 검사

리스너

모니터링

태그

마이그레이션

연결 드레이닝: 활성, 300 초 (편집)

인스턴스 편집

인스턴스 ID	이름	가용 영역	상태
i-0155ca5e85831f342	WAS(nodejs)	ap-northeast-2c	InService ⓘ
i-0d9a274d7fe06fa3a	WAS(nodejs2)	ap-northeast-2c	InService ⓘ

- 방금 생성된 로드밸런서(was-lb)를 체크하시고 하단 인스턴스 탭을 확인해서 물려있는 두 개의 인스턴스의 상태가(OutOfService -> InService)로 변경되기까지 기다립니다.
- 설명 탭에서 해당 로드밸런서의 DNS 주소를 POSTMAN으로 API 요청/응답이 정상적으로 작동되는 지 확인합니다.

GET /album

POST /user/login

+

...

No Environment

GET /album

Examples (0)

GET

was-lb-1986321715.ap-northeast-2.elb.amazonaws.com:8080/album

Params

Send

Save

Authorization

Headers

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 196 n

Pretty

Raw

Preview

JSON

Save Respons

1

[

2

{

3

"header": "api-album-test1",

4

"description": "api-album-description1",

5

"imgUrl": "https://s3.ap-northeast-2.amazonaws.com/fastcampus-s3-test-bucket/fastcampusImg.png"

6

},

7

{

8

"header": "api-album-test2",

9

"description": "api-album-description2",

10

"imgUrl": "https://s3.ap-northeast-2.amazonaws.com/fastcampus-s3-test-bucket/subwayImg.png"

11

}

12

]