

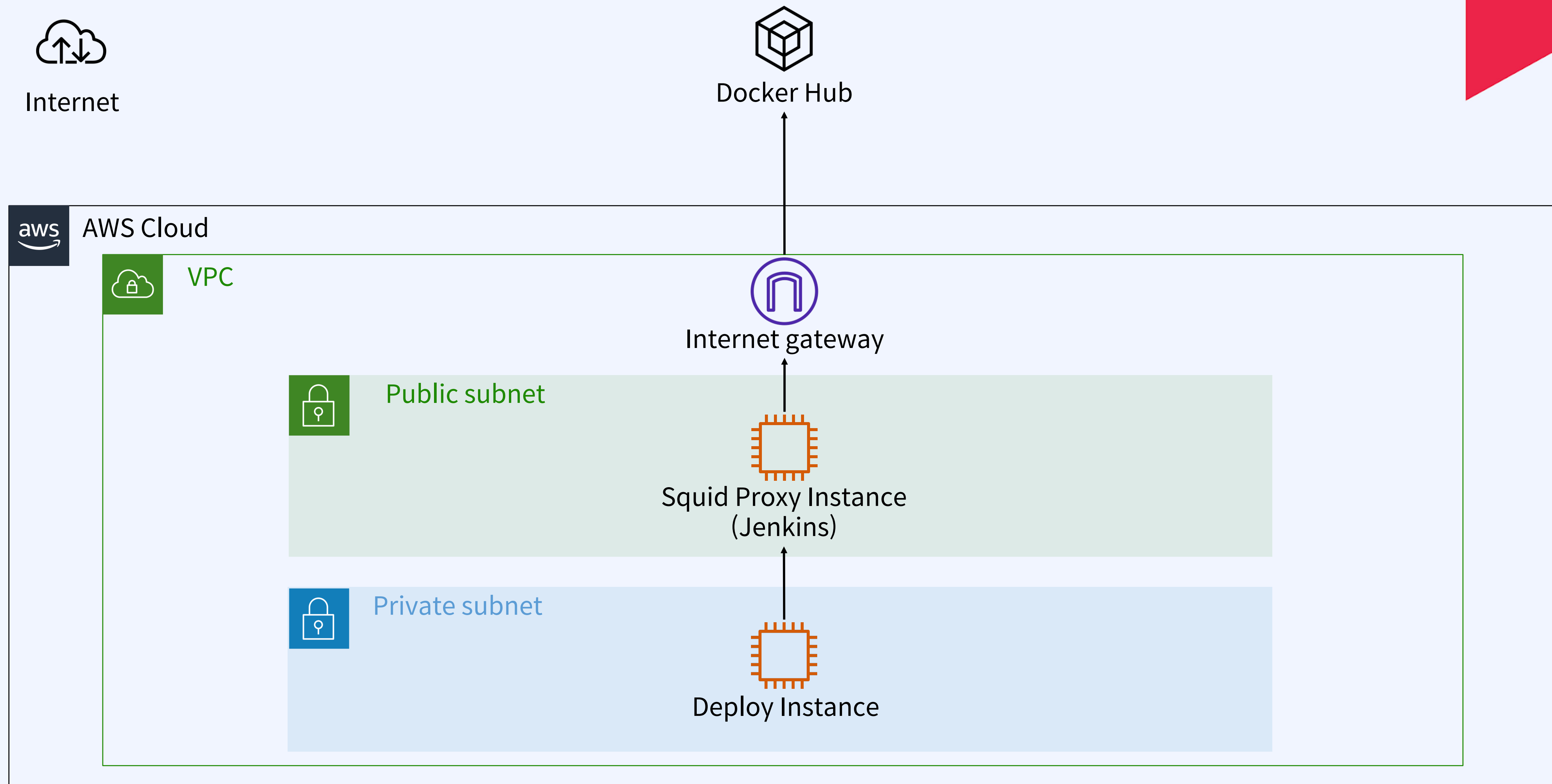
8 Docker 트러블 슈팅 방법

05 Squid를 활용한 Private 환경 Docker 빌드

Private 환경에서 Proxy 활용

05

Squid를 활용한
Private 환경
Docker 빌드



실습 진행사항

05

Squid를 활용한
Private 환경
Docker 빌드

Squid를 활용한 Private 환경 Docker 빌드 실습

1. Private 환경 구성 (AWS VPC, Subnet)
2. Squid 설정
3. Docker Image 빌드/Push/실행

사전 준비사항

- AWS EC2 VM Instance 2개
 - 이전 강의 때 생성한 Jenkins/Deploy VM 사용
 - Deploy VM의 경우, Private Subnet에 있기 때문에 Jenkins VM을 경유해서 접속해야함 (직접 접속 x)
- AWS VPC내 2개 Subnet 및 Internet Gateway 구성
 - Private Subnet 1개, Public Subnet 1개, Internet Gateway 1개
- Dockerhub 계정 준비 및 로그인

사전 준비1. AWS VPC내 2개 Subnet 및 Internet Gateway 구성

Internet Gateway 기본 구성 확인

- 보통은 VPC 만들어질 때 기본 구성됨

Public Subnet 구성

- Internet Gateway 라우팅 추가
- 임의의 분리된 가용 영역으로 지정(예 : apne2-az3, ap-northeast-2c)

Private Subnet 구성

- Internet Gateway 라우팅 제거
- 임의의 분리된 가용 영역으로 지정(예 : apne2-az2, ap-northeast-2b)

사전 준비2. 예제 코드 준비 및 Dockerhub 로그인

05

Squid를 활용한
Private 환경
Docker 빌드

예제 코드 준비

- Deploy VM 대상
- Chapter08 > 5_pri-squid-docker
- 개인 Repository에서 미리 예제 코드를 VM에 git clone

Dockerhub 개인 Repository에 로그인

- Deploy VM 대상
- docker login -u <dockerhub 개인 계정 ID>
- Password는 dockerhub 개인 계정 Password 입력

실습1. Private 환경 구성 (AWS VPC, Subnet)

AWS EC2 VM Instance의 Subnet 확인

- Jenkins VM (Squid Proxy용) : Public Subnet
- Deploy VM (Docker빌드용) : Private Subnet

Public Subnet 라우팅 생성

- Internet Routing
 - 라우팅 설정 : Internet Gateway로 0.0.0.0/0 CIDR 모든 트래픽이 향하도록 라우팅 설정
 - 서브넷 연결 : Public Subnet만 설정

Private Subnet 라우팅 생성

- Squid Routing
 - 라우팅 설정 : Squid Proxy(Jenkins) VM 인스턴스로 0.0.0.0/0 CIDR 모든 트래픽이 향하도록 라우팅 설정
 - 서브넷 연결 : Private Subnet만 설정

실습2. Squid 설정 #1

05

Squid를 활용한
Private 환경
Docker 빌드

Squid Proxy(Jenkins) VM 내에서 수행

Squid 설정파일 복사

- 예제코드 Chapter08 > 5_squid-docker/squid/squid.conf 사용
- Squid VM 내 \$home 아래 squid 디렉토리 생성 뒤 설정파일 복사

Squid Proxy Docker기동

- `docker run --name squid -d --restart=always --publish 3128:3128 --volume <Home 디렉토리>/squid/squid.conf:/etc/squid/squid.conf --volume <Home 디렉토리>/squid/cache:/var/spool/squid sameersbn/squid:3.5.27-2`

실습2. Squid 설정 #2

05

Squid를 활용한
Private 환경
Docker 빌드

Deploy VM 내에서 수행

환경변수로 Squid Proxy 설정 (curl로 외부 연결 확인 시)

- `export HTTP_PROXY=http://<Squid Proxy서버의 Private IP>:3128`
- `export HTTPS_PROXY=$HTTP_PROXY`

Docker 데몬에 Squid Proxy 설정

- `sudo vi /etc/systemd/system/multi-user.target.wants/docker.service`
[Service] # 탭아래 다음의 환경변수를 추가
Environment="`HTTP_PROXY=http://<Squid Proxy서버의 Private IP>:3128`"
Environment="`HTTPS_PROXY=http://<Squid Proxy서버의 Private IP>:3128`"
Environment="`NO_PROXY=localhost,127.0.0.1`"

Docker 데몬 reload 및 restart

- `sudo systemctl daemon-reload`
- `sudo systemctl restart docker`

실습3. Docker Image 빌드/Push/실행

05

Squid를 활용한
Private 환경
Docker 빌드

Docker Image 빌드 및 Push 명령어

- `docker build -t <Dockerhub 개인 Repository명>:<이미지 TAG> .`

Docker 이미지 Pull 및 실행 명령어

- `docker pull <Dockerhub 개인 Repository명>:<이미지 TAG>`
- `docker run -t <Dockerhub 개인 Repository명>:<이미지 TAG>`