

# 3 Artifact 관리를 위한 Nexus 활용

## 03 Nexus Artifact 활용한 Docker 빌드

## 실습 진행사항 목록

03

Nexus Artifact  
활용한 Docker 빌드

### Nexus Artifact 활용한 Docker 빌드

- Gradle에서 Repository를 Nexus로 설정
- Nexus Artifact 활용한 Docker 빌드

### 사전 준비사항

- Nexus 서버용 AWS EC2 VM 1개 (이전 실습 때 생성)
- Nexus Repository용 AWS S3 Bucket 1개 (이전 실습 때 생성)
- AWS ECR Repository 1개 (Docker 이미지 저장용)

# 실습 진행사항1

03

Nexus Artifact  
활용한 Docker 빌드

## Gradle에서 Repository를 Nexus로 설정 실습

### 기존 build.gradle 설정(AS-IS)

```
repositories {  
    mavenCentral()  
}
```

### 변경된 build.gradle 설정(TO-BE)

```
repositories {  
    maven {  
        url "https://<Nexus VM 퍼블릭 주소>:8081/repository/<Nexus에 생성한 Proxy Repository명>"  
    }  
}
```

## 실습 진행사항2

03

Nexus Artifact  
활용한 Docker 빌드

### Nexus Artifact 활용한 Docker 빌드 수행 실습

- Gradle 빌드시 Nexus Artifact 활용 및 Jar 정상 생성 확인
- Jib을 이용해 Docker 빌드 및 이미지 Push (AWS ECR)
- 로컬에 Docker 이미지 배포 및 컨테이너 실행

## 실습 관련 명령어

03

Nexus Artifact  
활용한 Docker 빌드

- Gradle 빌드 명령어 :
  - `gradle clean build --info`
- AWS ECR 로그인 명령어 :
  - `aws ecr get-login-password --region <리전명> | docker login --username AWS --password-stdin <AWS ECR Repository URL>`
- Jib을 이용한 Docker 빌드 명령어 :
  - `gradle jib --console=plain`
- 컨테이너 실행 명령어 :
  - `docker run -d -p 8080:8080 -t <AWS ECR Repository URL>:태그명`