

ec2-ddd

1. ECS 클러스터 생성 부터 차근차근 시작할게 이 콘솔을 nextjs 프로젝트에서 실행

아니요! AWS CLI 명령어는 로컬 개발 환경에서 실행하는 것입니다. NextJS 프로젝트 내에서 실행하는 것이 아닙니다.

## ECS 클러스터 생성 방법 2가지

### 방법 1: AWS 콘솔에서 생성 (추천 - 초보자용)

1. AWS 콘솔 접속
  - [AWS 콘솔](#) 에 로그인
  - 리전을 `ap-northeast-2` (서울)로 설정
2. ECS 서비스로 이동
  - 서비스 검색에서 "ECS" 입력
  - "Elastic Container Service" 클릭
3. 클러스터 생성
  - 좌측 메뉴에서 "클러스터" 클릭
  - "클러스터 생성" 버튼 클릭
  - 클러스터 이름: `nextjs-cluster`
  - 인프라: `AWS Fargate` (서버리스) 선택
  - "생성" 버튼 클릭

훌륭합니다! 클러스터 생성이 완료되었네요. 이제 다음 단계로 진행하겠습니다.

## 2단계: Task Definition 생성

Task Definition은 컨테이너가 어떻게 실행될지 정의하는 청사진입니다.

### AWS 콘솔에서 Task Definition 생성

1. ECS 콘솔에서 Task Definition으로 이동
  - 좌측 메뉴에서 "태스크 정의" 클릭
  - "새 태스크 정의 생성" 버튼 클릭
2. 기본 구성
  - 태스크 정의 패밀리: `pet-app-router-test-task`
  - 시작 유형: `AWS Fargate` 선택
  - 운영 체제/아키텍처: `Linux/X86_64`
  - CPU: `0.5 vCPU` (512 단위)
  - 메모리: `1 GB` (1024 MiB)
3. 컨테이너 정의
  - 컨테이너 이름: `nextjs-container`
  - 이미지 URI: `YOUR_ACCOUNT_ID.dkr.ecr.ap-northeast-2.amazonaws.com/YOUR_ECR_REPO_NAME:latest`

**? ECR 이미지 URI 확인 방법:**

```
aws ecr describe-repositories --region ap-northeast-2
```
4. 포트 매핑
  - 컨테이너 포트: `3000`
  - 프로토콜: `TCP`
  - 포트 이름: `nextjs-port` (선택사항)
5. 환경 변수 (선택사항)
  - `NODE_ENV` = `production`
  - `PORT` = `3000`
6. 로그 구성
  - 로그 드라이버: `awslogs`
  - `awslogs-group`: `/ecs/pet-app-router-test`
  - `awslogs-region`: `ap-northeast-2`
  - `awslogs-stream-prefix`: `ecs`

### 필요한 정보 확인

## Task Definition에서 포트 설정

### 컨테이너 포트 매핑

- 컨테이너 포트: 8080 ✓
- 프로토콜: TCP
- 포트 이름: nextjs-port (선택사항)

### 수정된 Task Definition 설정

#### 4. 포트 매핑 (수정됨)

- 컨테이너 포트: 8080
- 프로토콜: TCP

#### 5. 환경 변수 (수정됨)

- NODE\_ENV = production
- PORT = 8080
- HOSTNAME = 0.0.0.0

#### 6. 헬스체크 (나중에 설정할 때)

- 명령어: CMD-SHELL, `curl -f http://localhost:8080/api/health || exit 1`

### 다음 단계 진행

이제 정확한 포트 정보(8080)를 가지고 Task Definition을 생성해보겠습니다.

#### AWS ECS 콘솔에서:

1. "태스크 정의" → "새 태스크 정의 생성"
2. 위에서 안내한 설정들 입력
3. 포트는 8080으로 설정 ✓

Task Definition 생성을 계속 진행하시겠어요? 다른 설정값들은 이전에 안내한 대로 하시면 됩니다!



환영합니다! Task Definition 생성이 완료되었네요. 이제 다음 단계로 진행하겠습니다.

### 3단계: Application Load Balancer (ALB) 생성

ECS 서비스에 트래픽을 분산시키기 위해 로드 밸런서가 필요합니다.

#### AWS 콘솔에서 ALB 생성

1. EC2 콘솔로 이동
  - AWS 콘솔에서 "EC2" 검색
  - 좌측 메뉴에서 "로드 밸런서" 클릭
  - "로드 밸런서 생성" 버튼 클릭
2. 로드 밸런서 유형 선택
  - Application Load Balancer 선택
  - "생성" 클릭
3. 기본 구성
  - 로드 밸런서 이름: pet-app-router-test-alb
  - 체계: 인터넷 경계 (Internet-facing)
  - IP 주소 유형: IPv4
4. 네트워크 매핑
  - VPC: 기본 VPC 선택
  - 매핑: 최소 2개 가용 영역 선택
    - ap-northeast-2a 체크
    - ap-northeast-2c 체크
  - 각 가용 영역의 퍼블릭 서브넷 선택
5. 보안 그룹
  - "새 보안 그룹 생성" 선택
  - 보안 그룹 이름: pet-app-alb-sg
  - 인바운드 규칙:
    - 유형: HTTP, 포트: 80, 소스: 0.0.0.0/0
    - 유형: HTTPS, 포트: 443, 소스: 0.0.0.0/0 (선택사항)
6. 리스너 및 라우팅
  - 프로토콜: HTTP
  - 포트: 80
  - 기본 작업: "대상 그룹 생성" 선택

#### Target Group 생성 (ALB 생성 과정에서)

7. 대상 그룹 세부 정보
  - 대상 유형: IP 주소
  - 대상 그룹 이름: pet-app-tg
  - 프로토콜: HTTP
  - 포트: 8080 (NextJS 컨테이너 포트)
  - VPC: 기본 VPC와 동일
8. 상태 확인
  - 상태 확인 프로토콜: HTTP
  - 상태 확인 경로: / (또는 /api/health 있다면)
  - 고급 상태 확인 설정:
    - 정상 임계값: 2
    - 비정상 임계값: 3
    - 제한 시간: 5초
    - 간격: 30초
9. 대상 등록
  - 지금은 건너뛰기 (ECS 서비스에서 자동 등록됨)
10. 검토 및 생성
  - 설정 확인 후 "로드 밸런서 생성" 클릭

#### 생성 완료 확인

ALB 생성이 완료되면 \*\*상태가 "활성"이 될 때까지 2-3분 기다려주세요.

다음 단계 준비사항:

- ALB ARN 복사해두기
- Target Group ARN 복사해두기
- ALB DNS 이름 확인하기

ALB 생성이 완료되면 다음 단계(ECS 서비스 생성)로 진행하겠습니다!

진행 상황을 알려주세요. 혹시 설정 중에 궁금한 점이 있으면 언제든지 물어보세요!

물음합니다! 이제 4단계 ECS 서비스 생성을 진행하겠습니다.

#### 4단계: ECS 서비스 생성

ECS 서비스는 Task Definition을 기반으로 컨테이너를 실행하고 ALB와 연결해줍니다.

#### AWS ECS 콘솔에서 서비스 생성

1. ECS 콘솔로 이동
  - AWS 콘솔에서 "ECS" 검색
  - 클러스터 → pet-app-test-cluster 클릭
2. 서비스 생성
  - 서비스 탭에서 "생성" 버튼 클릭
3. 환경 구성
  - 컴퓨팅 옵션: 시작 유형 선택
  - 시작 유형: FARGATE 선택
  - 플랫폼 버전: LATEST (기본값)
4. 배포 구성
  - 애플리케이션 유형: 서비스 선택
  - Task Definition:
    - 패밀리: pet-app-test-task 선택
    - 리버전: 1 (LATEST) 선택
5. 서비스 구성
  - 서비스 이름: pet-app-test-service
  - 원하는 작업 수: 2 (고가용성을 위해)
6. 네트워킹
  - VPC: 기본 VPC (자동 선택됨)
  - 서브넷:
    - ap-northeast-2a 서브넷 선택
    - ap-northeast-2b 서브넷 선택 (ALB와 동일한 AZ)
  - 보안 그룹: "새 보안 그룹 생성"
    - 보안 그룹 이름: pet-app-ecs-sg
    - 인바운드 규칙:
      - 유형: 사용자 지정 TCP
      - 포트: 8080
      - 소스: ALB 보안 그룹 (pet-app-alb-sg) 선택
  - 퍼블릭 IP: 활성화됨 (인터넷 접근용)
7. 로드 밸런싱
  - 로드 밸런서 유형: Application Load Balancer 선택
  - 기존 로드 밸런서 사용: 선택
  - 로드 밸런서: pet-app-test-alb 선택
  - 리스너: 80:HTTP 선택
  - 대상 그룹: pet-app-test-tg 선택
8. 서비스 자동 크기 조정 (선택사항)
  - 지금은 사용 안 함 선택 (나중에 설정 가능)
9. 태그 (선택사항)
  - 키: Name, 값: pet-app-test-service
  - 키: Environment, 값: test
10. 검토 및 생성
  - 모든 설정 확인
  - 생성 클릭

#### 서비스 배포 확인

서비스 생성 후 다음 사항들을 확인해야 합니다:

배포 상태 확인:

- 서비스 상태: ACTIVE