

Phase 1 - 기본 인프라 구축

포함 내용

Phase 1에서 생성되는 AWS 리소스:

-  VPC & Subnets (또는 기본 VPC 사용)
-  Security Group (RTMP 포트 허용)
-  IAM Roles (MediaLive, Lambda)
-  S3 Bucket (Archive 저장소)
-  Lifecycle Policy (7일 후 자동 삭제)

빠른 시작

```
bash
```

```
# 1. 폴더 이동
```

```
cd environments/dev
```

```
# 2. terraform.tfvars 수정 (선택사항)
```

```
nano terraform.tfvars
```

```
# 3. Terraform 초기화
```

```
terraform init
```

```
# 4. 실행 계획 확인
```

```
terraform plan
```

```
# 5. 리소스 생성
```

```
terraform apply
```

```
# 'yes' 입력
```

```
# 6. 결과 확인
```

```
terraform output
```

소요 시간: 약 2~3분 **비용:** 거의 \$0

파일 구조

```
phase1-code/
```

```
├── modules/
```

```
│   ├── network/      # VPC, Security Group
```

```
│   ├── iam/          # IAM Roles & Policies
```

```
│   └── storage/       # S3 Bucket
```

```
└── environments/
```

```
|   └── dev/
|       ├── main.tf      # 모듈 통합
|       ├── variables.tf  # 변수 정의
|       ├── terraform.tfvars # 실제 값 설정 ← 여기 수정
|       └── outputs.tf    # 출력 정의
|── PHASE1_EXECUTION_GUIDE.md # 상세 가이드
|── README.md              # 이 파일
```

⚙️ 설정 변경

terraform.tfvars 파일

```
hcl

# 프로젝트 이름 (원하는 대로 변경)
project_name = "my-live-stream"

# AWS 리전
region = "ap-northeast-2" # 서울

# RTMP 접속 허용 IP (보안 중요!)
allowed_rtmp_cidrs = ["0.0.0.0/0"] # 모든 IP 허용 (테스트용)
# allowed_rtmp_cidrs = ["123.456.789.0/24"] # 특정 IP만 (권장)

# 태그 (Owner만 변경)
tags = {
  Owner = "YourName" # ← 여기에 본인 이름
}
```

📦 생성되는 리소스

1. Network 모듈

- **VPC:** 기본 VPC 사용 (create_vpc = false)
- **Security Group:**
 - RTMP (TCP 1935) 허용
 - RTMPS (TCP 443) 허용

2. IAM 모듈

- **MediaLive Role:**
 - MediaPackage 접근 권한
 - S3 Archive 쓰기 권한
- **Lambda Role:**
 - MediaLive 시작/중지 권한

- CloudWatch Logs 권한

3. Storage 모듈

- **S3 Bucket:**
 - Archive 저장소
 - 7일 후 자동 삭제
 - 암호화 활성화
 - Public 접근 차단

리소스 확인

Terraform 출력으로 확인

```
bash

# 모든 출력 보기
terraform output

# 특정 값만 보기
terraform output vpc_id
terraform output archive_bucket_name
terraform output medialive_role_arn
```

AWS Console에서 확인

1. S3 Bucket

- AWS Console → S3
- 버킷 이름:

2. Security Group

- AWS Console → EC2 → Security Groups
- 이름:
- Inbound Rules 확인

3. IAM Roles

- AWS Console → IAM → Roles
-
-

AWS CLI로 확인

```
bash
```

```
# S3 Bucket
```

```
aws s3 ls | grep my-live-stream
```

```
# Security Group
```

```
aws ec2 describe-security-groups \
--filters "Name=group-name,Values=*medialive-input*" \
--query 'SecurityGroups[0].[GroupId,GroupName]' \
--output table
```

```
# IAM Roles
```

```
aws iam list-roles \
--query 'Roles[?contains(RoleName, `my-live-stream`)].RoleName' \
--output table
```

비용

Phase 1 생성 비용:

- VPC: 무료
- Security Group: 무료
- IAM Roles: 무료
- S3 Bucket: **\$0.023/GB** (저장된 데이터만)

예상 월 비용: 거의 \$0

아직 MediaLive를 시작하지 않았으므로 추가 비용 없음

리소스 삭제

```
bash
```

```
# 전체 삭제
```

```
terraform destroy
```

```
# 'yes' 입력
```

```
# 특정 모듈만 삭제
```

```
terraform destroy -target=module.storage
```

주의: S3 Bucket에 데이터가 있으면 삭제 실패

```
bash
```

```
# S3 Bucket 비우기
```

```
aws s3 rm s3://버킷이름 --recursive
```

트러블슈팅

문제 1: Bucket 이름 충돌

Error: bucket already exists

해결: terraform.tfvars에서 project_name 변경

문제 2: 권한 부족

Error: AccessDenied

필요한 IAM 권한:

- AmazonS3FullAccess
- IAMFullAccess
- AmazonVPCFullAccess

문제 3: 리전 오류

Error: operation not supported in region

해결: terraform.tfvars에서 region 확인

```
hcl
region = "ap-northeast-2" # 서울
```

학습 자료

상세 가이드:

- [PHASE1 EXECUTION GUIDE.md](#) - 단계별 실행 및 코드 설명

Terraform 문서:

- [Terraform 기초](#)
- [AWS Provider](#)

AWS 문서:

- [VPC](#)
- [IAM](#)
- [S3](#)

완료 체크리스트

☐ AWS 인증 설정 완료

- ☐ Terraform 설치 확인
- ☐ terraform.tfvars 수정
- ☐ terraform init 성공
- ☐ terraform plan 확인
- ☐ terraform apply 성공
- ☐ AWS Console에서 리소스 확인
- ☐ terraform output 확인

다음 단계

Phase 1 완료 후:

1. 코드 세부 분석 (PHASE1_EXECUTION_GUIDE.md 참고)
2. AWS Console에서 생성된 리소스 확인
3. Phase 3 (MediaPackage) 준비

Phase 3로 진행하려면:

- Phase 3 코드 요청
- MediaPackage 모듈 학습
- HLS Endpoint 생성

팁

학습 방법:

1. 각 모듈의 `main.tf` 파일 읽기
2. AWS Console에서 실제 리소스 확인
3. 설정 변경 후 `terraform apply` 재실행
4. 변경 사항 관찰

보안 권장사항:

- `allowed_rtmp_cidrs`를 특정 IP로 제한
- IAM Role 최소 권한 원칙 적용
- S3 Public 접근 차단 유지

궁금한 점이나 오류가 발생하면 언제든지 질문하세요! 😊