# AWS 클라우드 프로젝트 워크샵

# [스스로 구축하는 AWS 클라우드 인프라 - 기본편]

# 워크샵 개요

# 워크샵 개요

## 1. 대상

AWS에 입문하신 분들

AWS 클라우드 인프라 구성과 관련 서비스에 익숙하지 않으신 분들

## 2. 난이도

기초~기본 (주요 인프라 관련 서비스들의 기초적인 개념과 컨셉 그리고 기본적인 사용 방법 수준을 학습합니다.)

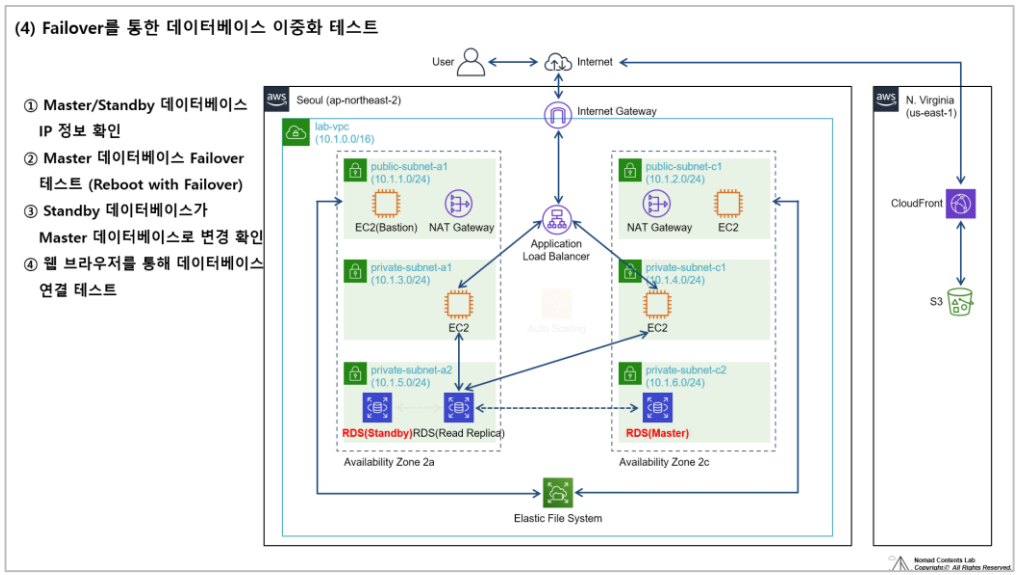
## 3. 워크샵을 통해 배우는 것

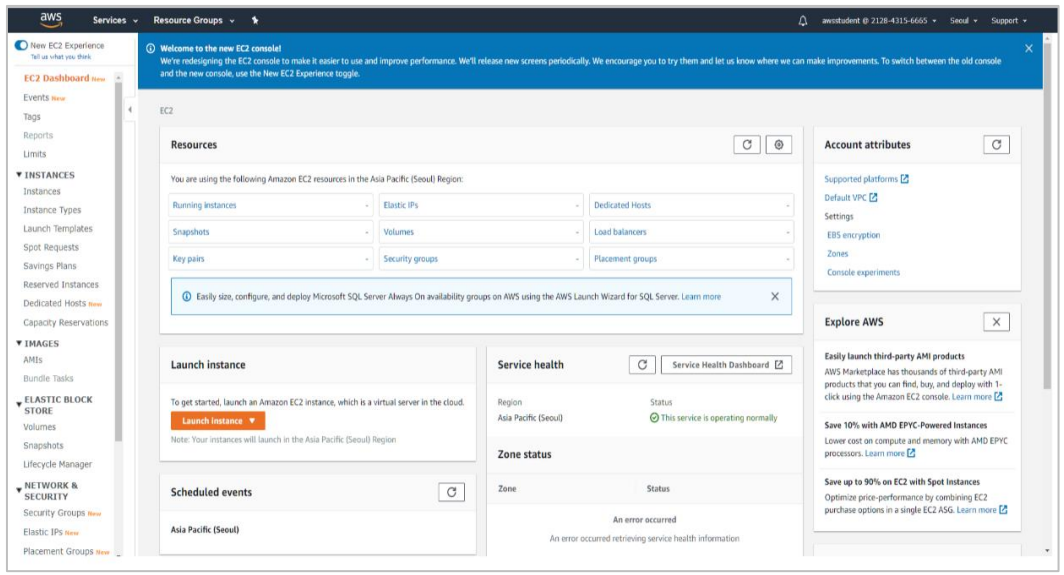
AWS의 핵심적인 주요 서비스들의 실제 사용 방법 (매니지먼트콘솔 기준)

AWS 클라우드 인프라를 구축하는 프로세스와 방법

AWS 클라우드 인프라를 구성하는 서비스들의 상관 관계와 클라우드 인프라의 작동 원리

## 4. 워크샵의 구성



## 5. 다루는 내용

서버리스 정적 웹사이트 호스팅 및 성능 가속화

LAMP 웹 서버 및 Application Load Balancer 구성

관계형 데이터베이스 서비스 구성

Auto Scaling을 통한 확장성 및 탄력성 구현

# 워크샵 개요

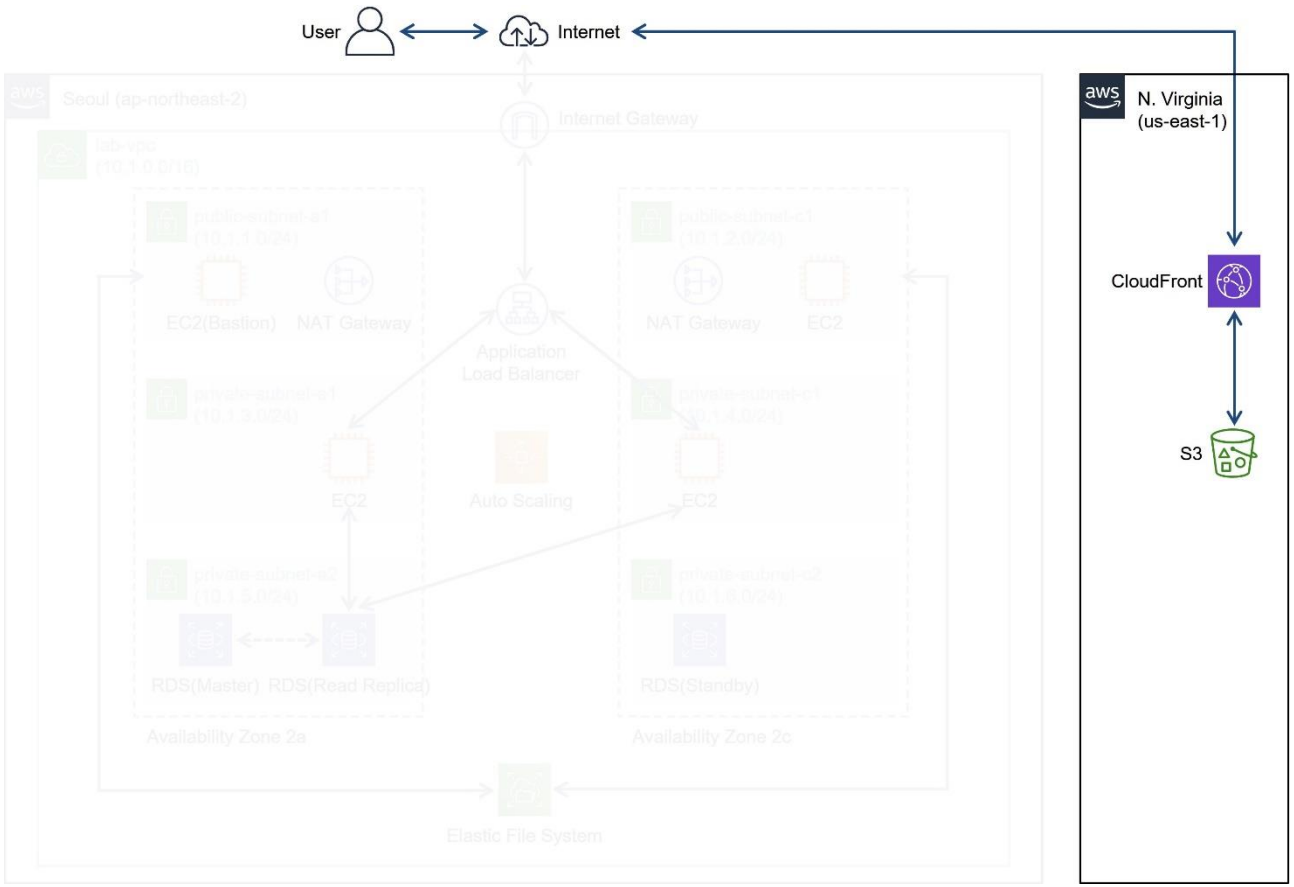
## 5. 다루는 내용

### 서버리스 정적 웹사이트 호스팅 및 성능 가속화

LAMP 웹 서버 및 Application Load Balancer 구성

관계형 데이터베이스 서비스 구성

Auto Scaling을 통한 확장성 및 탄력성 구현



# 워크샵 개요

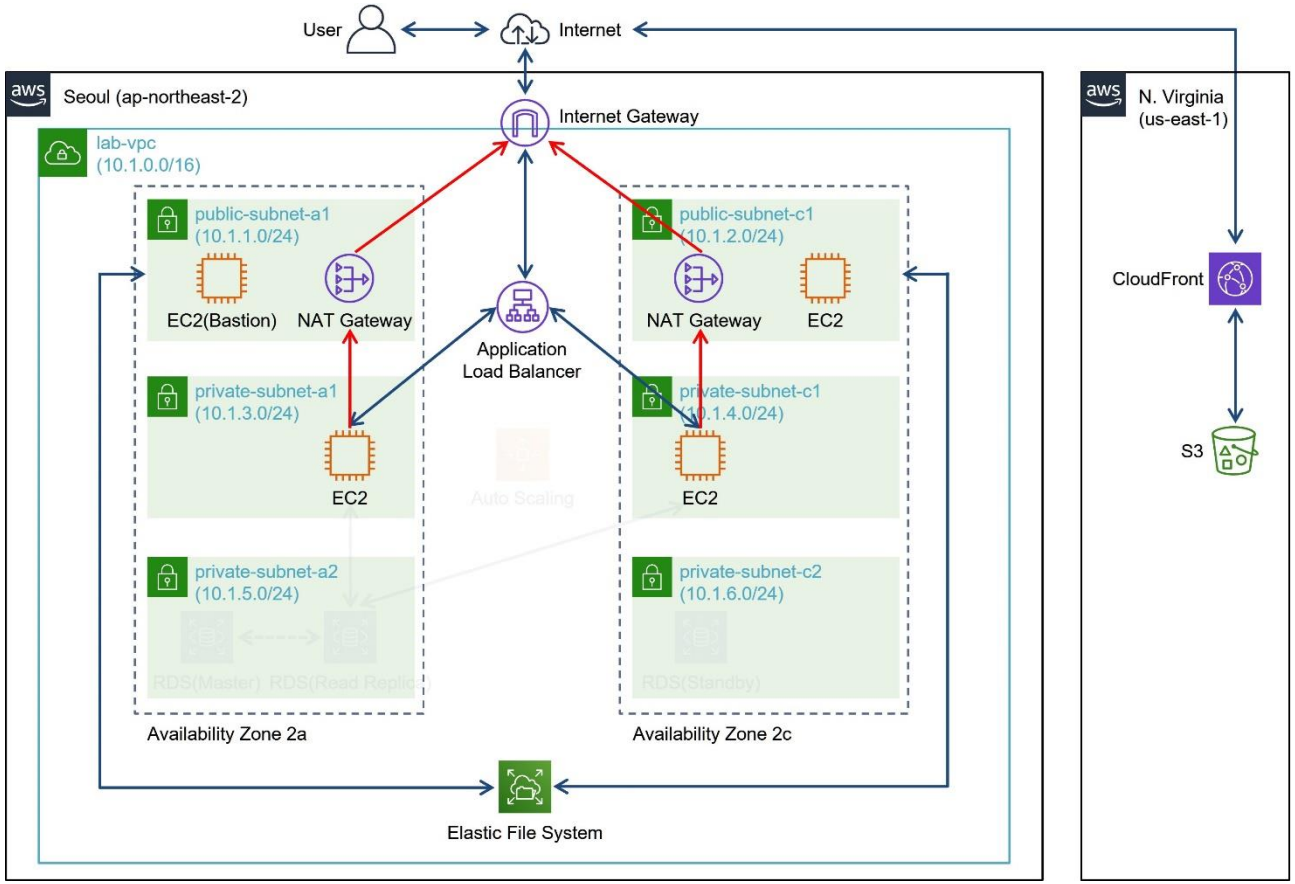
## 5. 다루는 내용

서버리스 정적 웹사이트 호스팅 및 성능 가속화

**LAMP 웹 서버 및 Application Load Balancer 구성**

관계형 데이터베이스 서비스 구성

Auto Scaling을 통한 확장성 및 탄력성 구현



# 워크샵 개요

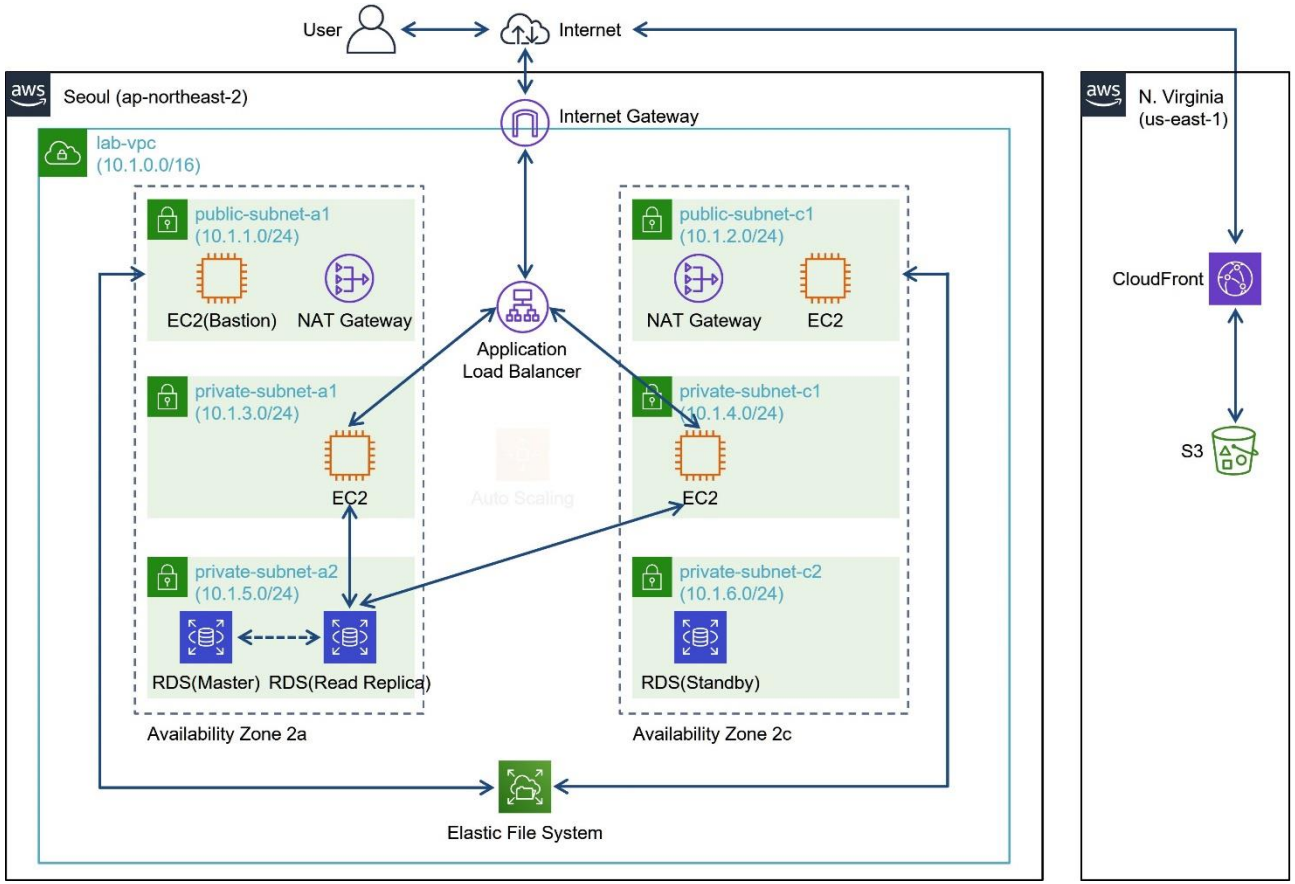
## 5. 다루는 내용

서버리스 정적 웹사이트 호스팅 및 성능 가속화

LAMP 웹 서버 및 Application Load Balancer 구성

**관계형 데이터베이스 서비스 구성**

Auto Scaling을 통한 확장성 및 탄력성 구현



# 워크샵 개요

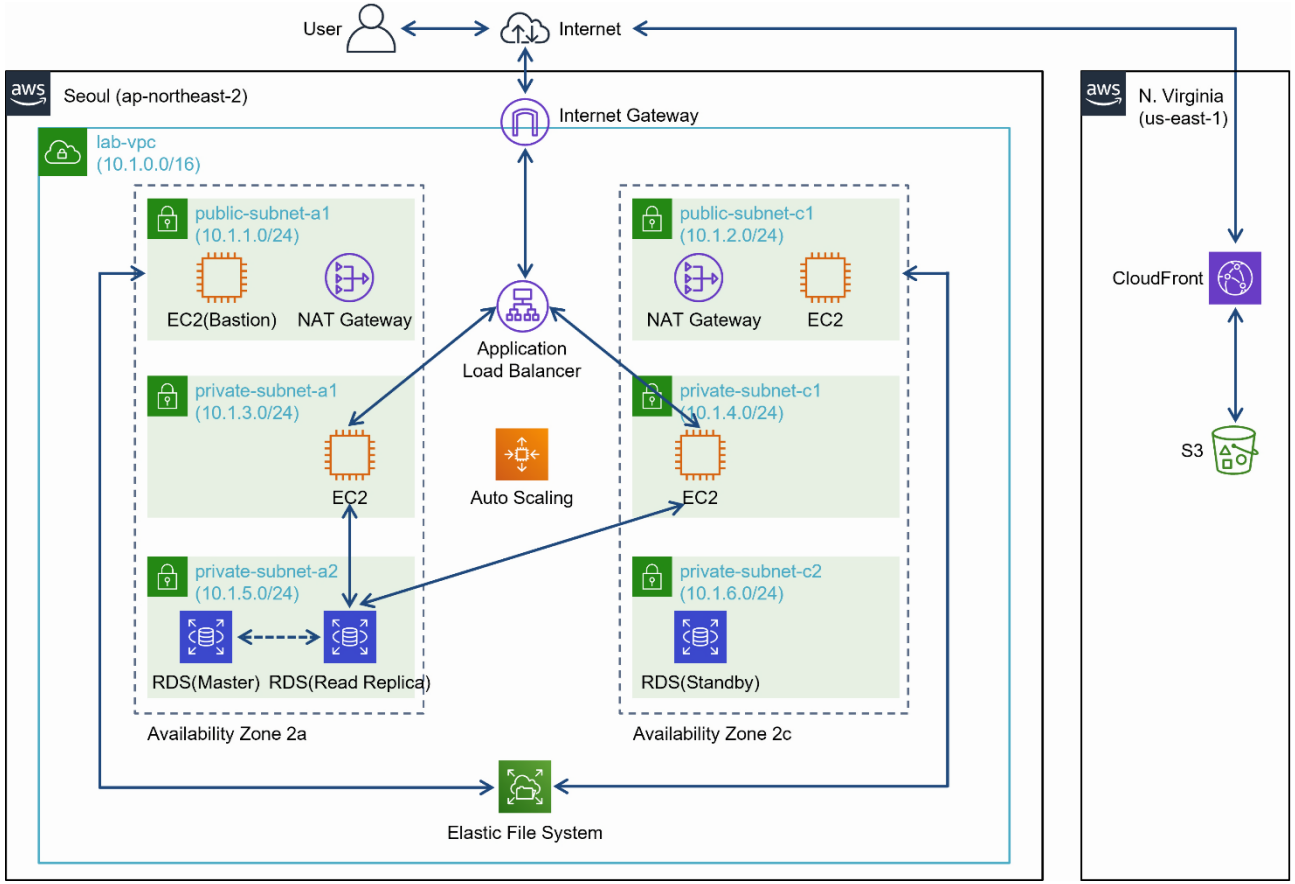
## 5. 다루는 내용

서버리스 정적 웹사이트 호스팅 및 성능 가속화

LAMP 웹 서버 및 Application Load Balancer 구성

관계형 데이터베이스 서비스 구성

**Auto Scaling을 통한 확장성 및 탄력성 구현**



# 워크샵 시작 전 준비사항

# 워크샵 시작 전 준비해야 할 사항

## 1. (권장) 주요 AWS 서비스의 개념

Amazon S3
Amazon CloudFront
Amazon EC2
Amazon VPC (VPC, Subnet, Internet Gateway, Route Table, Security Group, NAT Gateway)
Elastic Load Balancer (Application Load Balancer)
Amazon RDS
Auto Scaling

## 2. (권장) AWS 신규 Account (프리 티어 혜택으로 실습 비용 최소화)

## 3. (옵션) Linux O/S, TCP/IP, 네트워크

## 4. (옵션) IAM User 생성 및 설정

# 워크샵 시작 전 참고사항

# 워크샵 시작 전 참고해야 할 사항

## 1. 실습 환경

컴퓨터 O/S : 윈도우
브라우저 : 크롬
SSH 클라이언트 : PuTTY

## 2. 프리 티어에 해당되지 않는 서비스 / 범위

Elastic IP (단, 특정 리소스에 associate 되어 있으면 비용 발생하지 않음)
NAT Gateway
RDS (Multi AZ)

# Chapter 01. 서버리스 정적 웹사이트 호스팅 및 성능 가속화

# 서버리스 정적 웹사이트 호스팅 및 성능 가속화

## 1. Chapter 개요

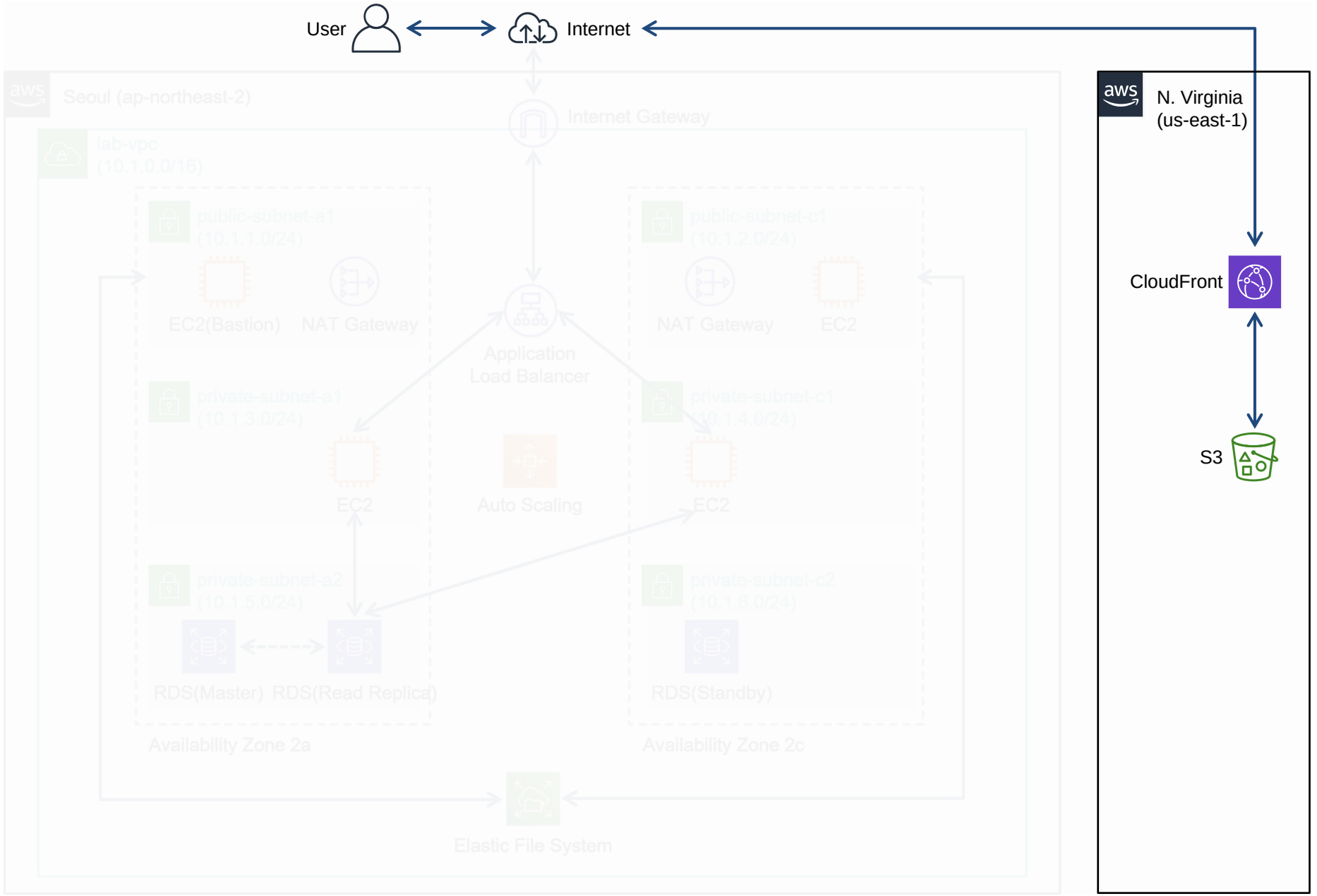
서버가 없어도 구성이 가능한 정적 웹 사이트를 호스팅하고, 콘텐츠 전송 네트워크(CDN) 서비스를 이용하여 웹 사이트의 성능을 향상시킵니다.

## 2. 사용하는 AWS 서비스

- Amazon S3 (Simple Storage Service)

- Amazon CloudFront

# 서버리스 정적 웹사이트 호스팅 및 성능 가속화



# (1) S3 Bucket 생성 및 정적 웹사이트 호스팅

① S3 Bucket 생성

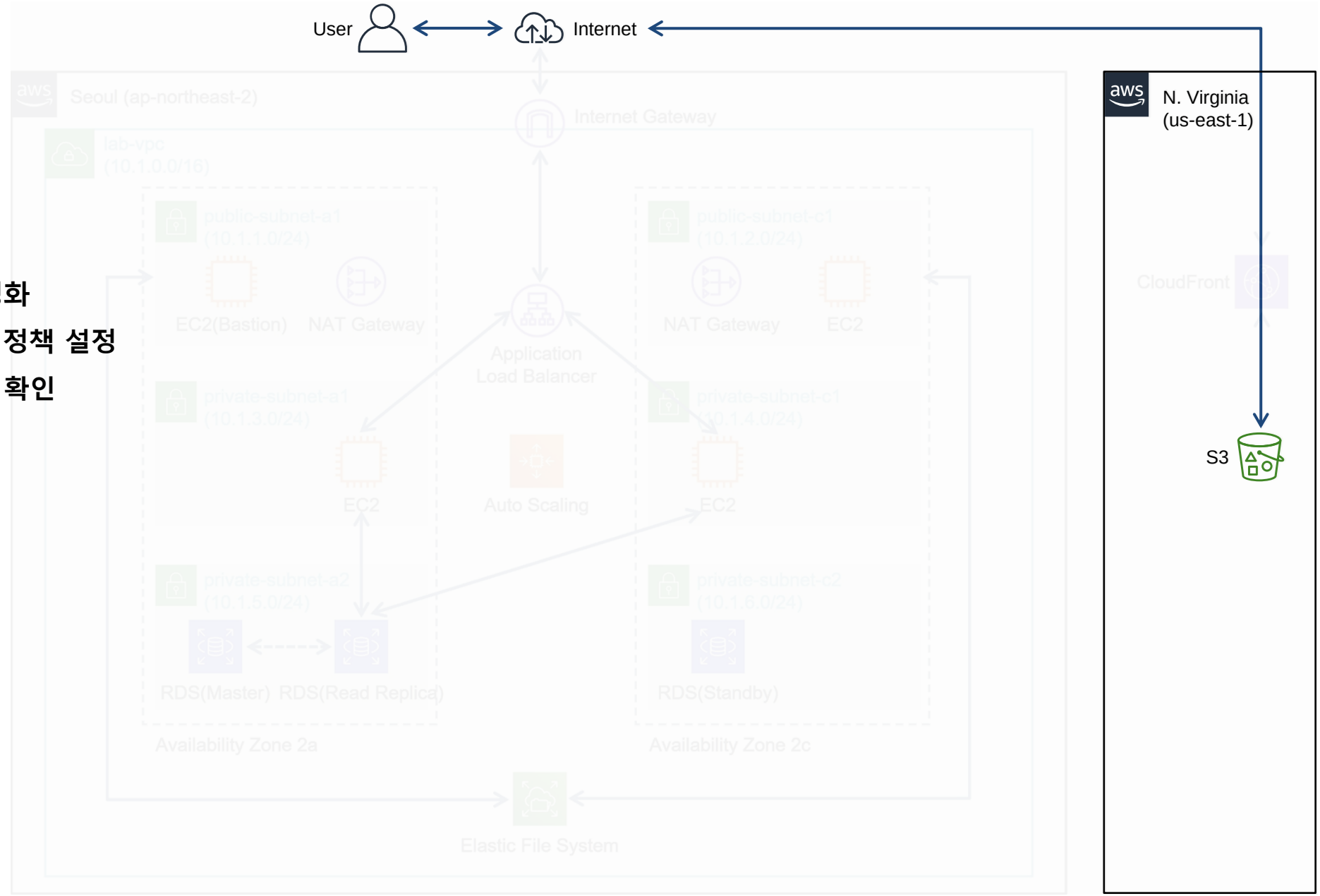
② Object(File) 업로드

- mycar.html
- car.jpg

③ 정적 웹 사이트 호스팅 기능 활성화

④ Bucket과 Object에 대한 액세스 정책 설정

⑤ 웹 브라우저에서 웹 사이트 작동 확인

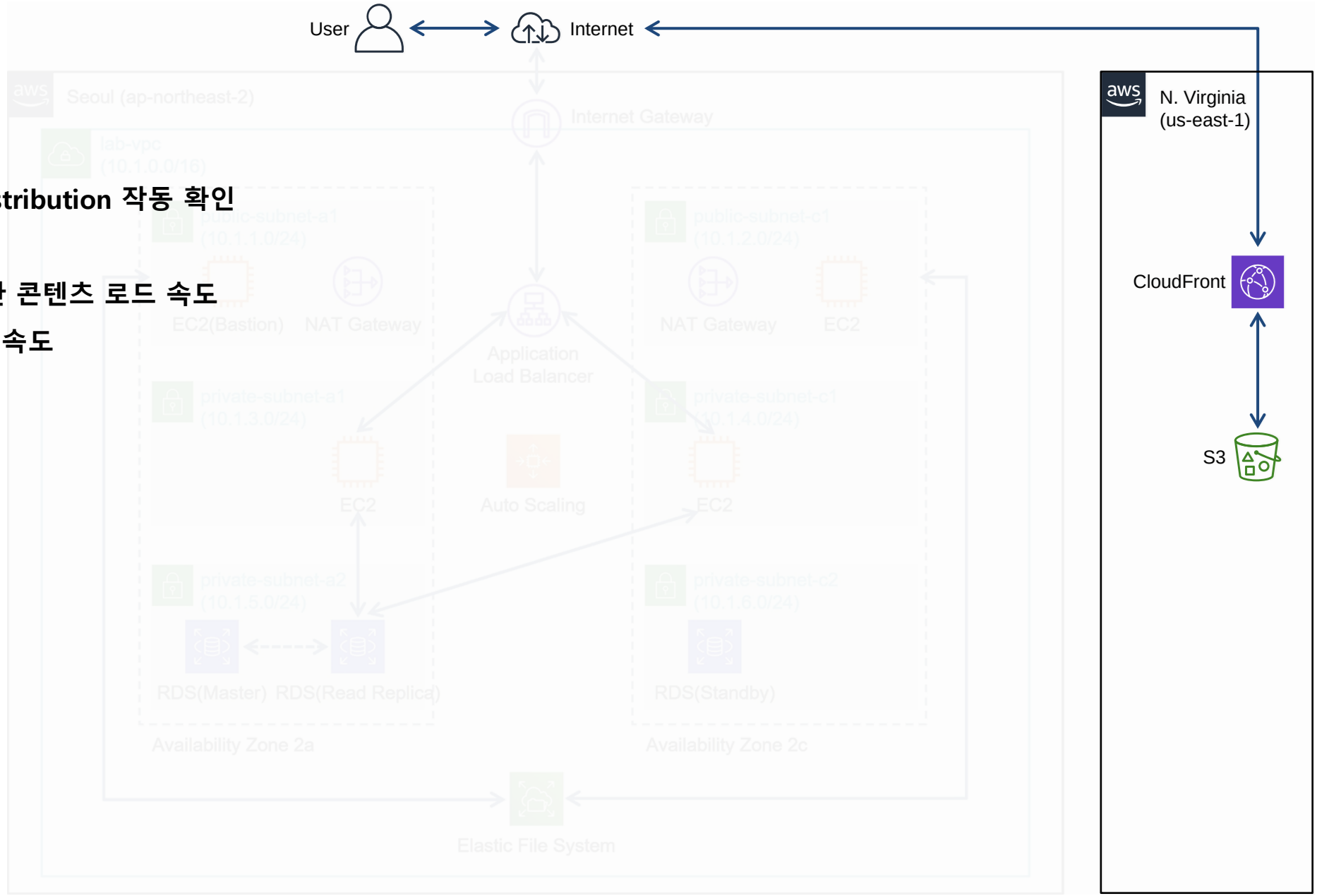


# (2) CloudFront를 통한 웹사이트 성능 가속화

① CloudFront Distribution 생성
- Origin, Cache behavior 등 설정

② 웹 브라우저에서 CloudFront Distribution 작동 확인

③ 웹 사이트 성능 테스트
- S3 정적 웹 사이트 호스팅을 통한 콘텐츠 로드 속도
- CloudFront를 통한 콘텐츠 로드 속도



# Chapter 02. LAMP 웹 서버 및 Application Load Balancer 구성

# LAMP 웹 서버 및 Application Load Balancer 구성

## 1. Chapter 개요

클라우드 네트워크 환경에 Linux 기반의 가상 서버에 Apache 웹 서버, MySQL 데이터베이스, PHP 어플리케이션을 구성하고 Application Load Balancer를 이용하여 이중화 된 네트워크를 구성합니다.

## 2. 사용하는 AWS 서비스

· Amazon VPC (VPC, Subnet, Internet Gateway, Route Table, NAT Gateway 등)

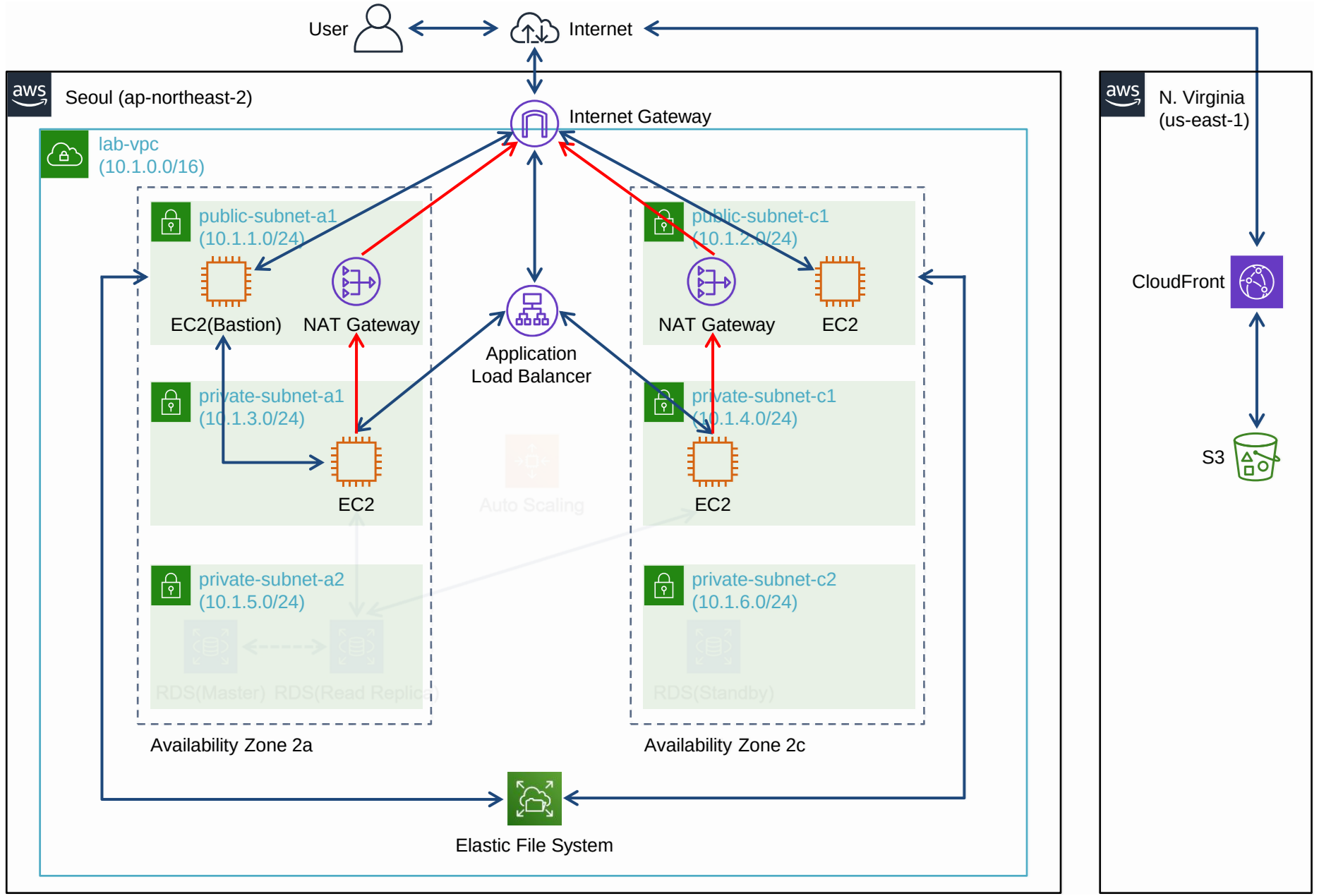
· Amazon EC2

· Amazon EBS

· Amazon EFS

· Elastic Load Balancer - Application Load Balancer

# LAMP 웹 서버 및 Application Load Balancer 구성



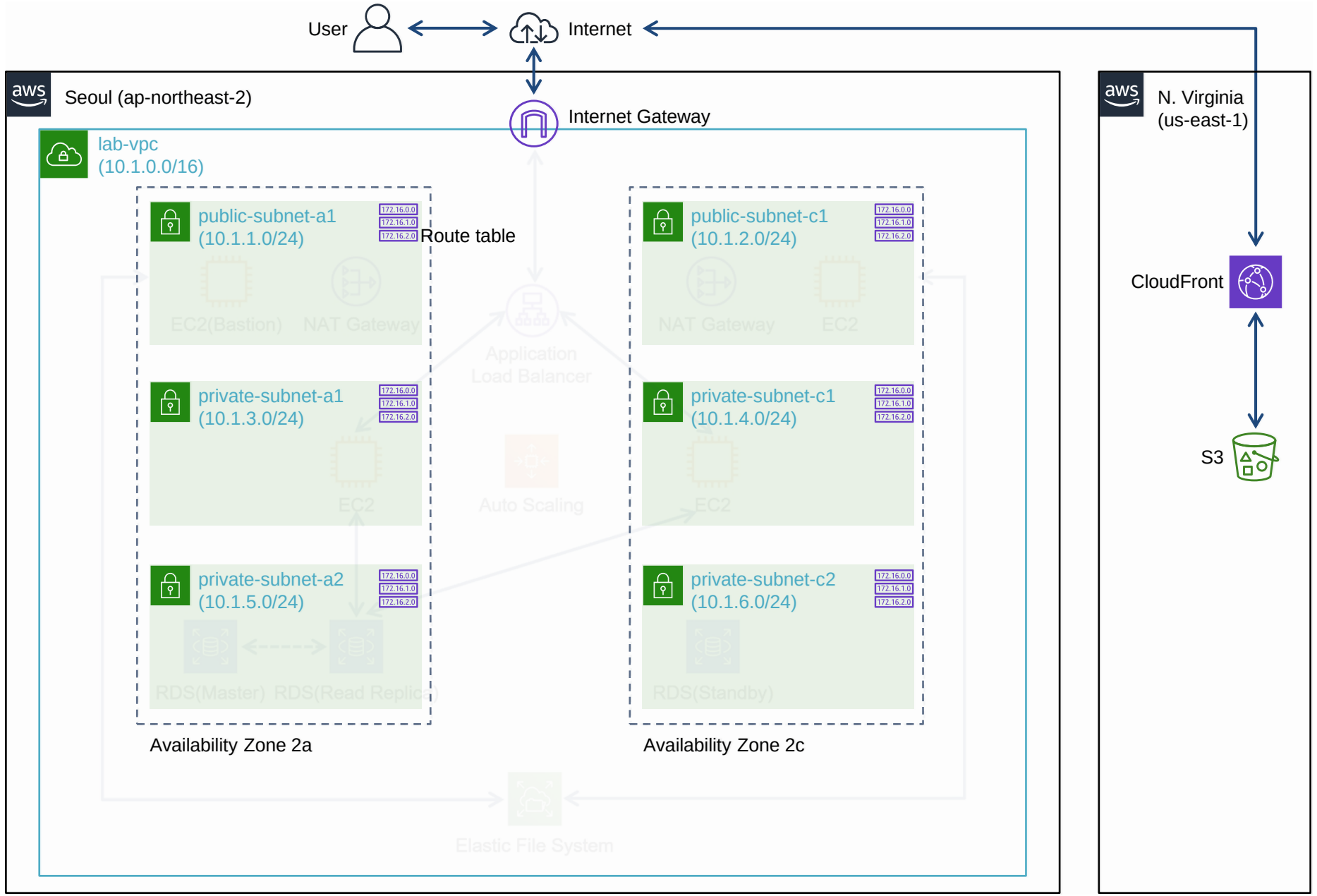
# (1) 기본 네트워크 환경 구성 (VPC/Subnet/Internet Gateway/Route Table)

① VPC 생성

② Subnet 생성

③ Internet Gateway 생성

④ Route Table 생성 및 Route 설정

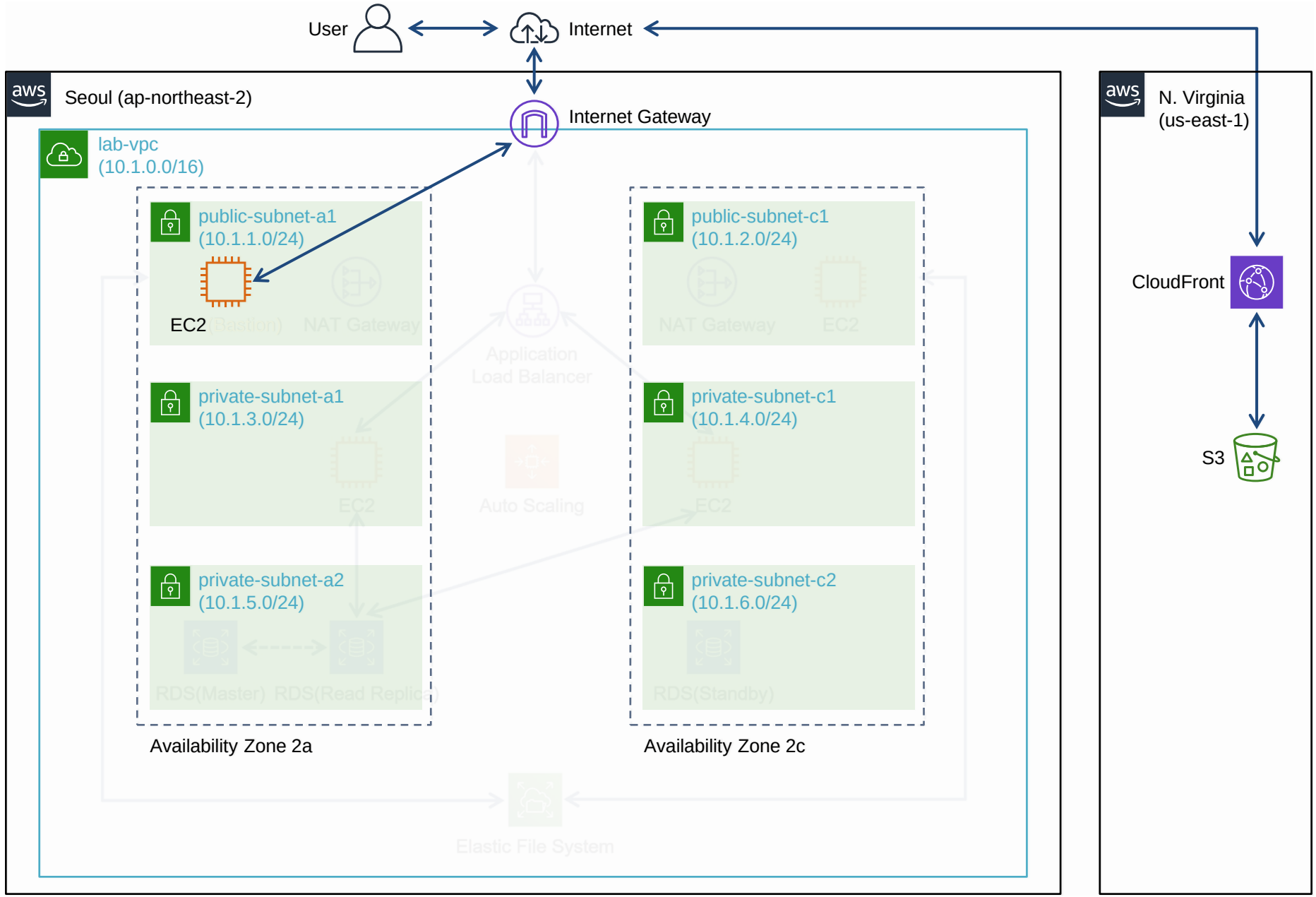


# (2) Public EC2 인스턴스 생성 및 LAMP 웹서버 구성

① EC2 생성
- AMI
- LAMP 웹 서버 구성
  (Linux, Apache, MySQL, PHP)
- Security group
- Storage(EBS)
- Key pair

② Index.php 파일 생성

③ 웹 브라우저에서 LAMP 웹 서버 작동 테스트

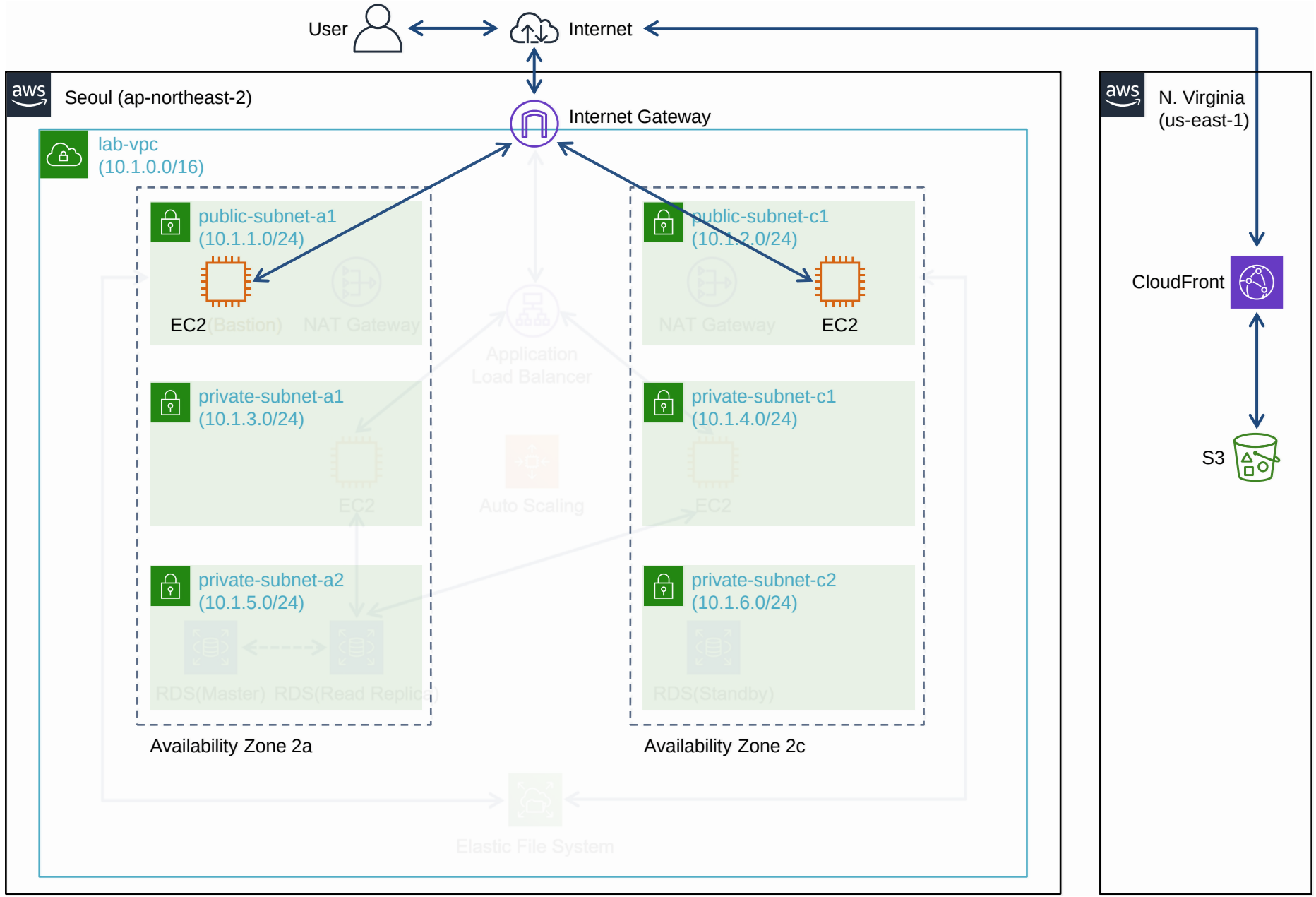


# (3) Custom AMI를 통한 Public EC2 인스턴스 생성

① Custom AMI 생성

② Custom AMI를 통해 EC2 추가 생성

② 웹 브라우저에서 LAMP 웹 서버 작동 테스트



# (4) EFS를 통한 네트워크 파일 시스템 구성

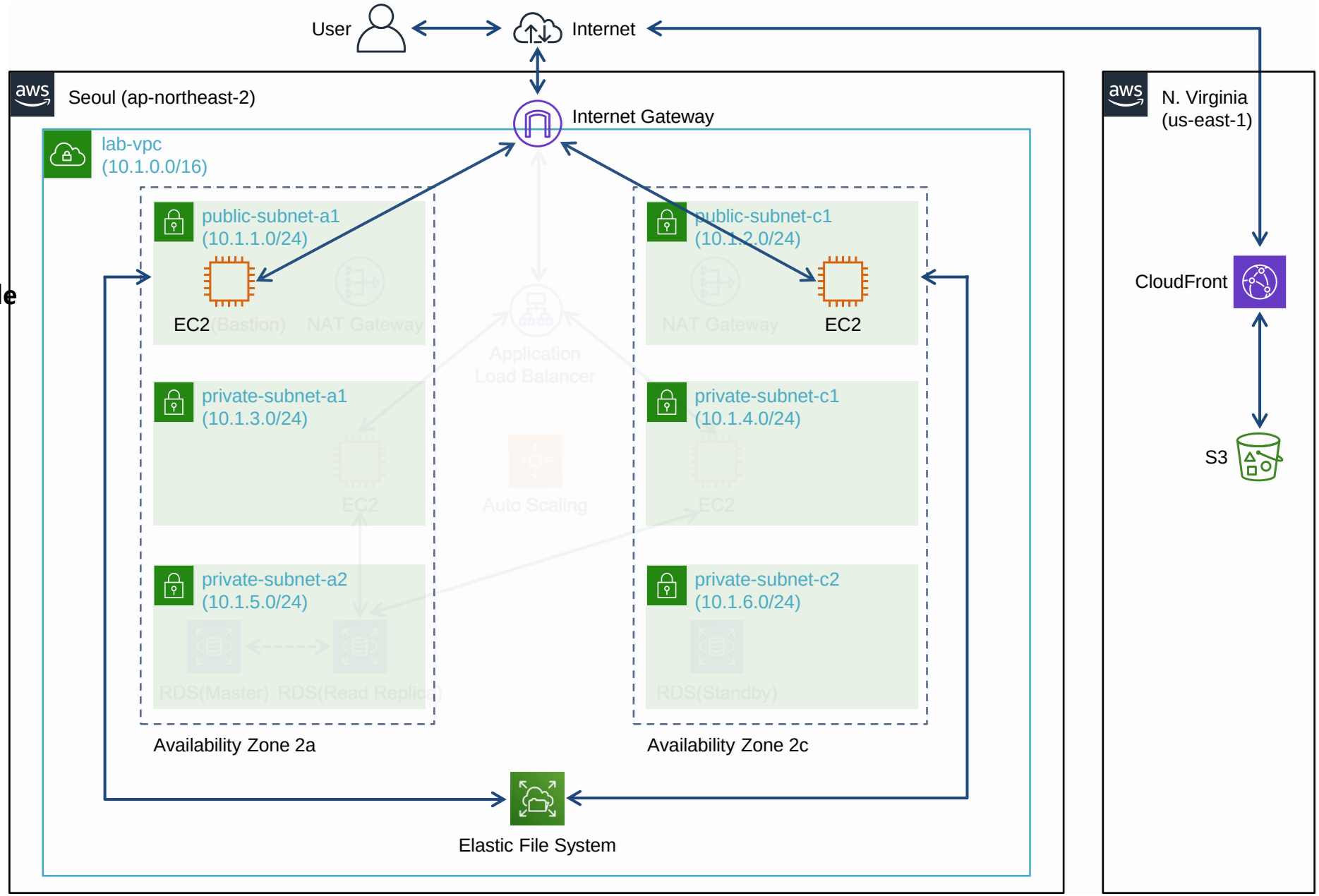
① EFS용 Security group 생성

② EFS 생성

- Availability
- Lifecycle
- Performance/Throughput mode
- Network 등

③ EFS-EC2 마운트

④ 웹 브라우저를 통한 EFS 마운트 테스트

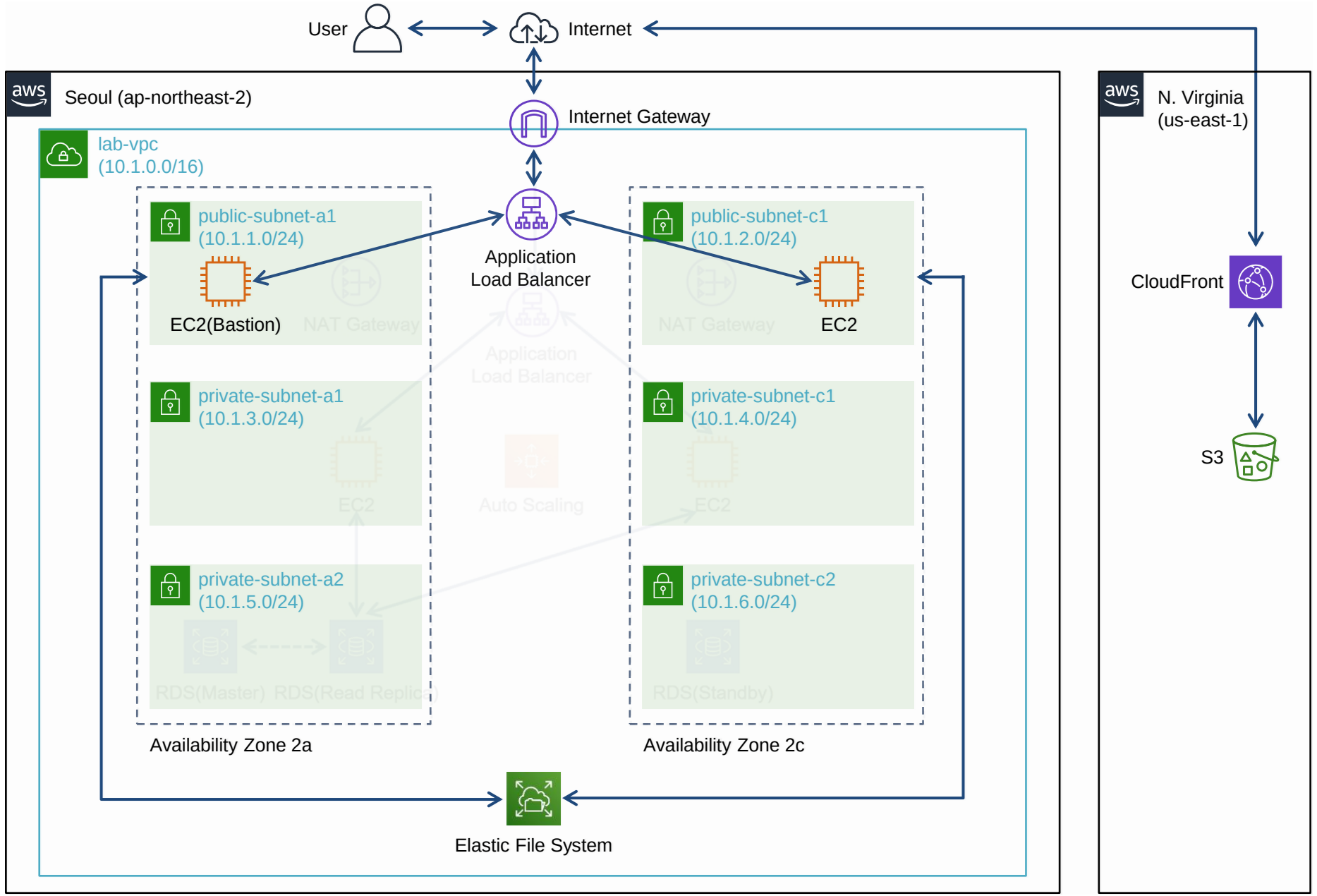


# (5) Application Load Balancer를 통한 이중화 네트워크 구성 (1)

① Target group 생성
- Target type
- Protocol/Port
- Target registration 등

② Application Load Balancer 구성
- Scheme
- Network
- Security group
- Listener/Rule 등

③ 웹 브라우저를 통한 Application Load Balancer 작동 테스트



# (6) Bastion host와 NAT Gateway를 통한 Private EC2 인스턴스의 외부 통신 구성

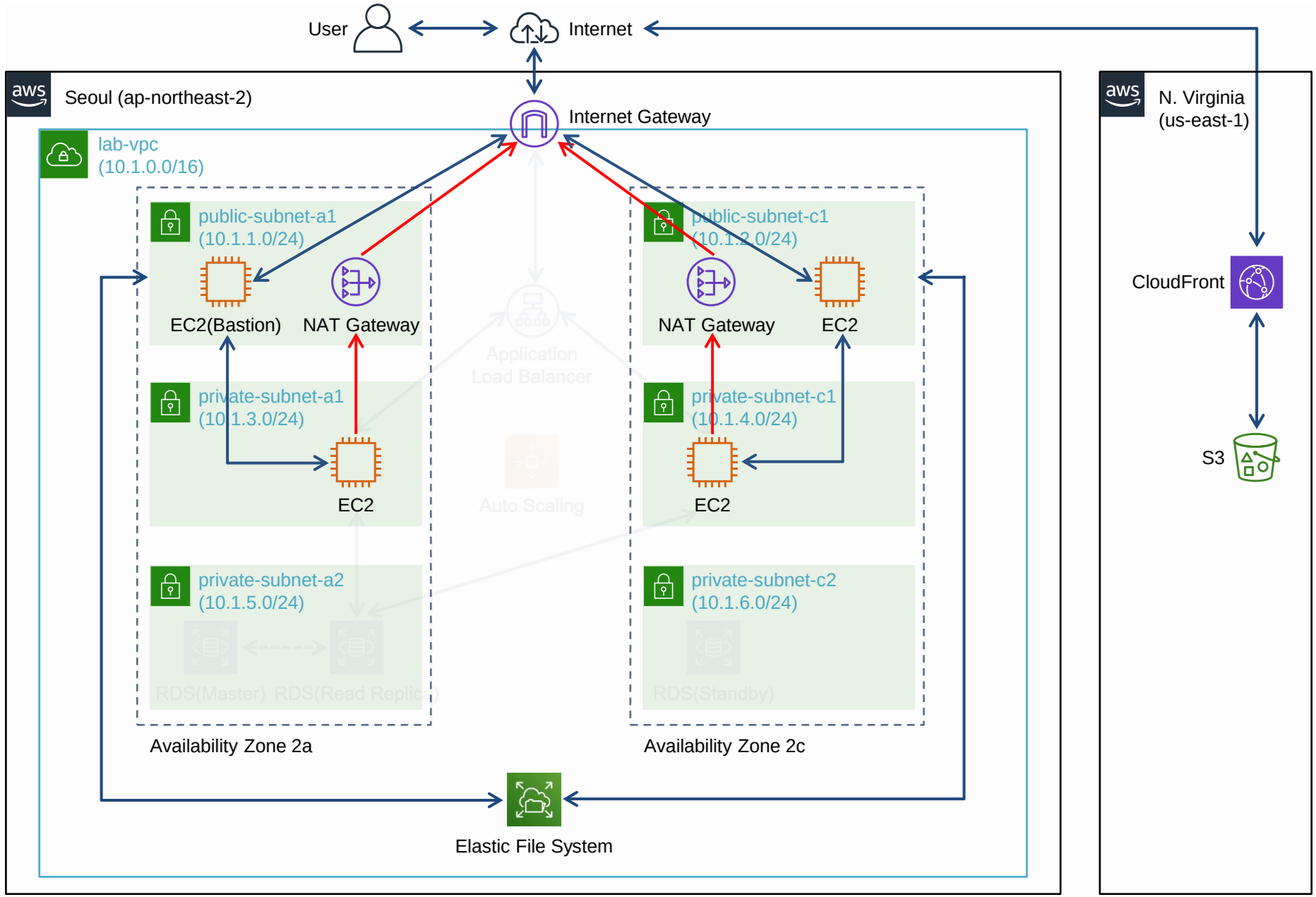
① Private subnet에 EC2 생성

② Public subnet의 EC2를 통해 Private subnet의 EC2에 접속 (+ key pair 생성)

③ NAT Gateway 생성

④ Route table 설정

⑤ Private subnet의 EC2의 외부 통신 테스트

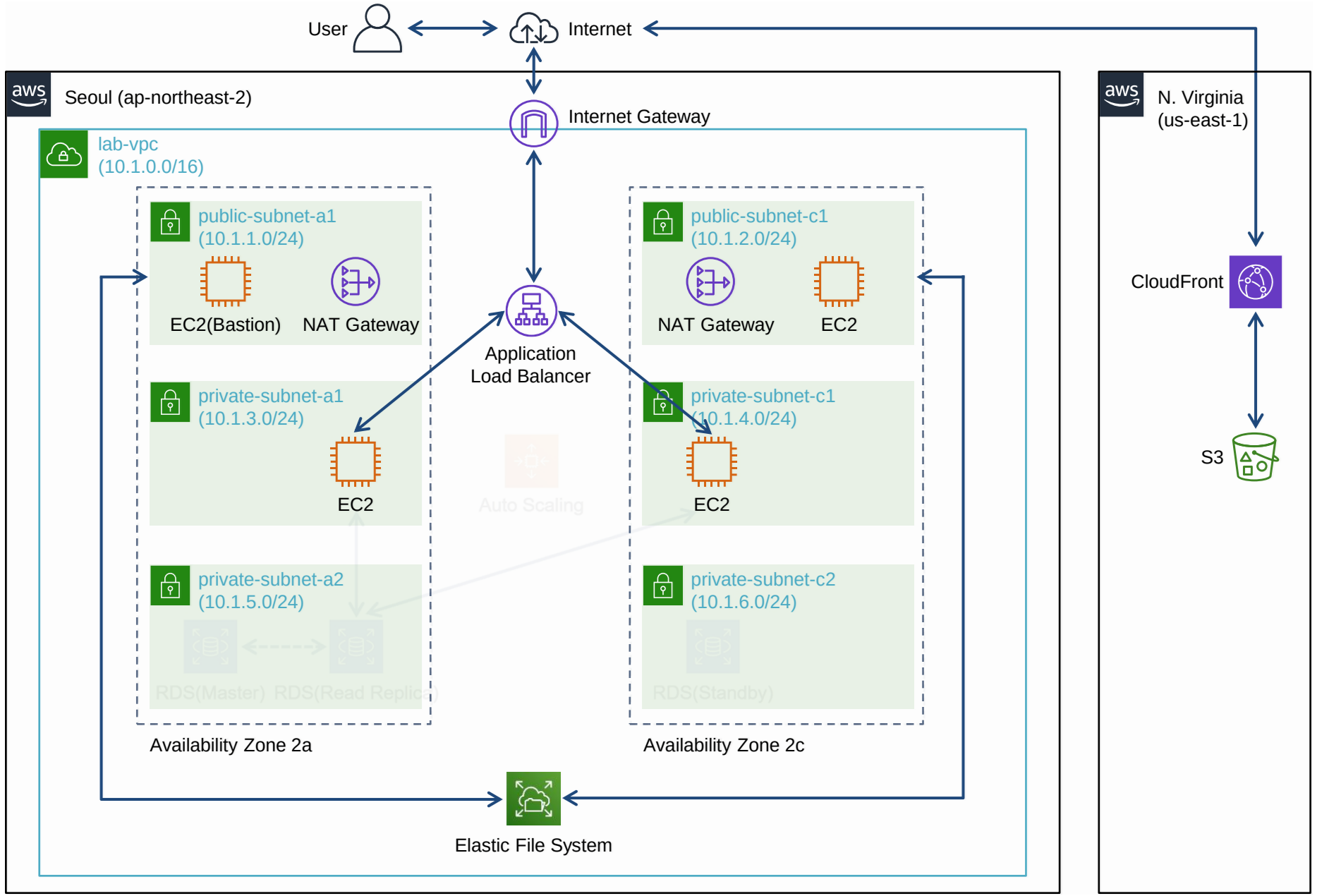


# (7) Application Load Balancer를 통한 이중화 네트워크 구성 (2)

① Target group 생성

② Application Load Balancer 구성

③ 웹 브라우저를 통한 Application Load Balancer 작동 테스트



# Chapter 03. 관계형 데이터베이스 서비스 구성

# 관계형 데이터베이스 서비스 구성

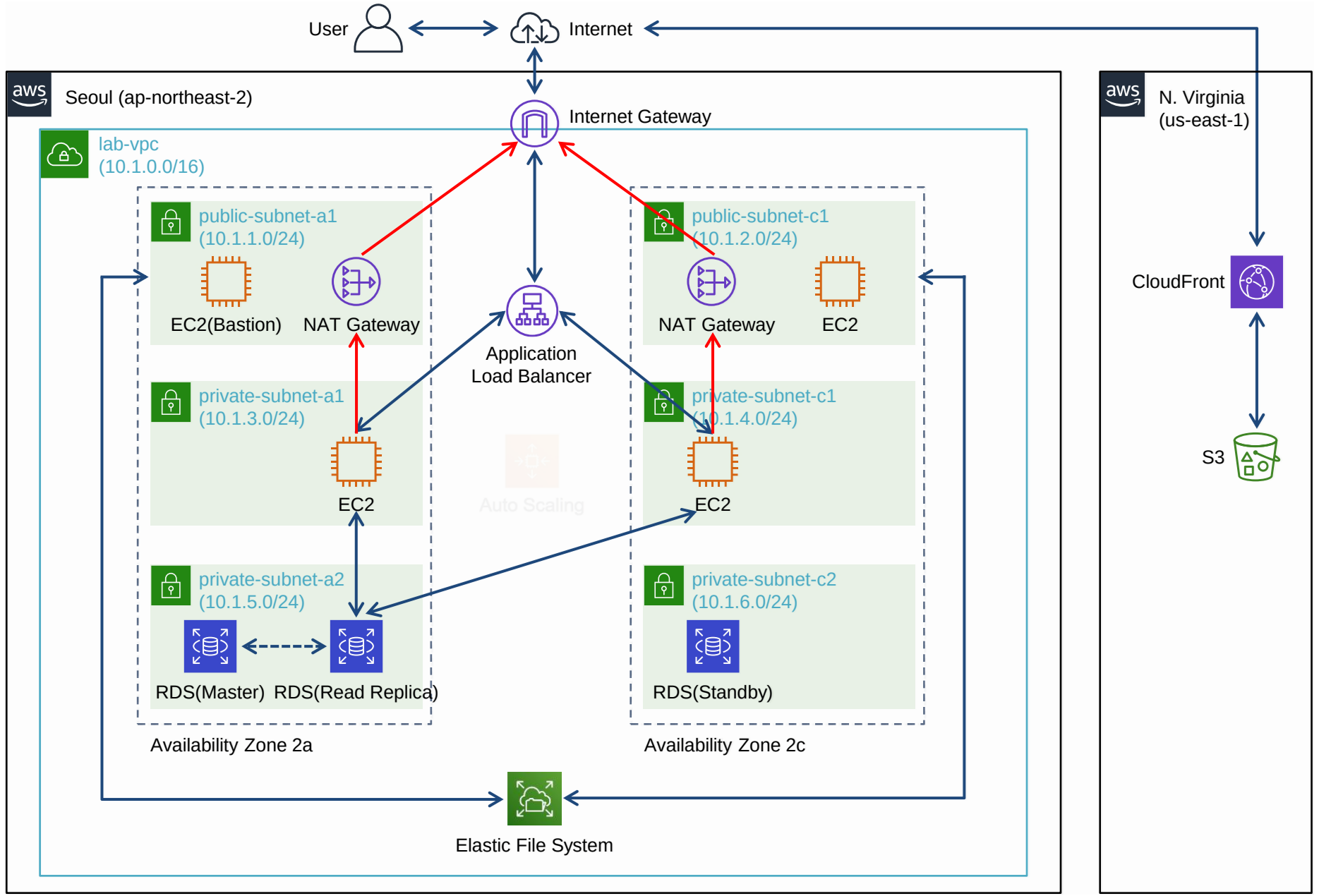
## 1. Chapter 개요

MySQL 데이터베이스를 복수의 가용영역에 이중화로 구성하고 Linux 기반의 가상 서버와 MySQL 데이터베이스를 연결합니다.

## 2. 사용하는 AWS 서비스

· Amazon RDS

# 관계형 데이터베이스 서비스 구성



# (1) Amazon RDS를 통한 MySQL 데이터베이스 이중화(Multi-AZ) 구성

① RDS용 Security group 생성

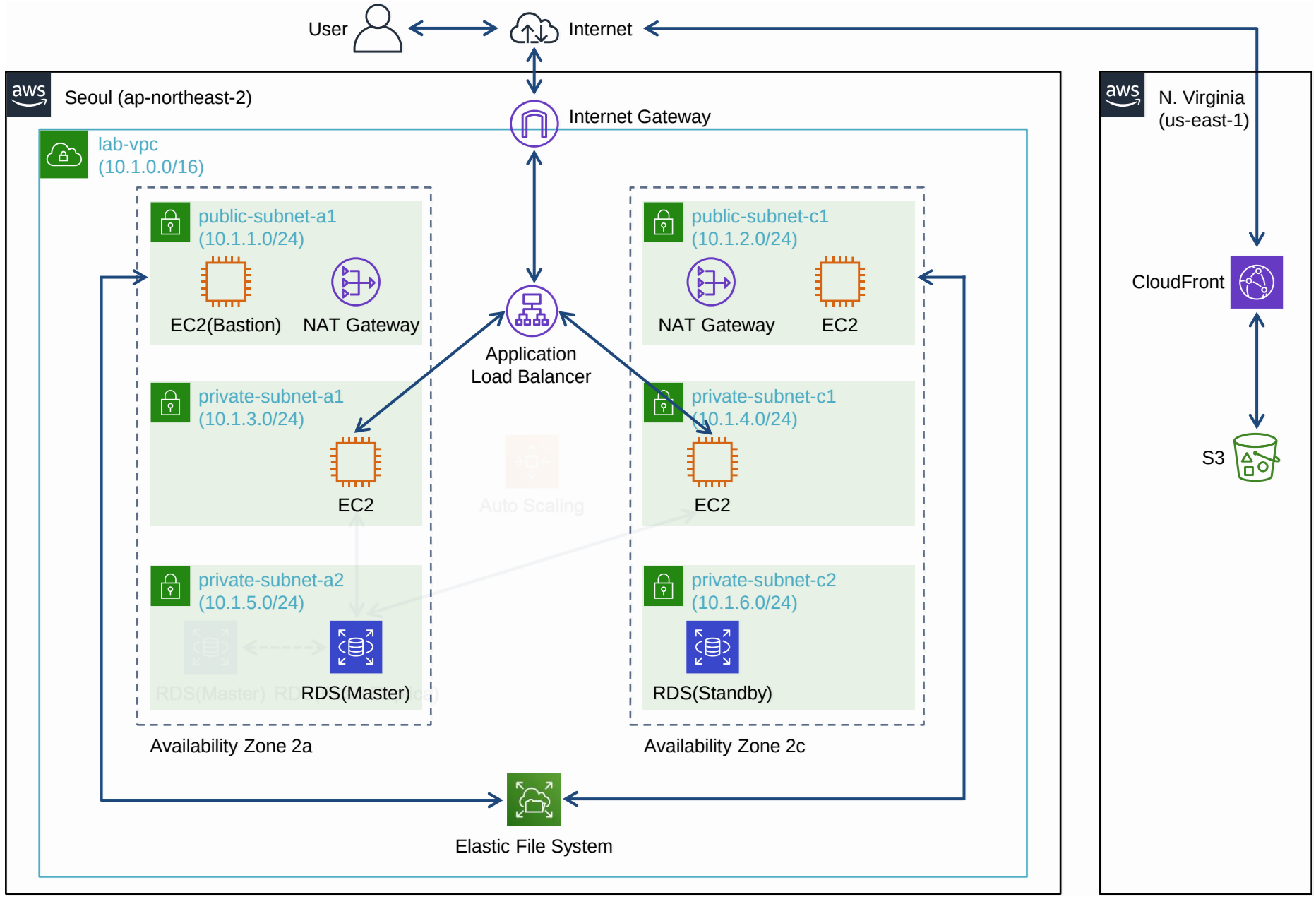
② Subnet group 생성

③ 복수의 가용영역에 MySQL 데이터베이스 생성

(Multi-AZ Deployment)

- DB Engine
- DB Instance
- Storage
- Availability & Durability
- Connectivity
- Authentication
- Backup 등

④ 데이터베이스 정보 확인



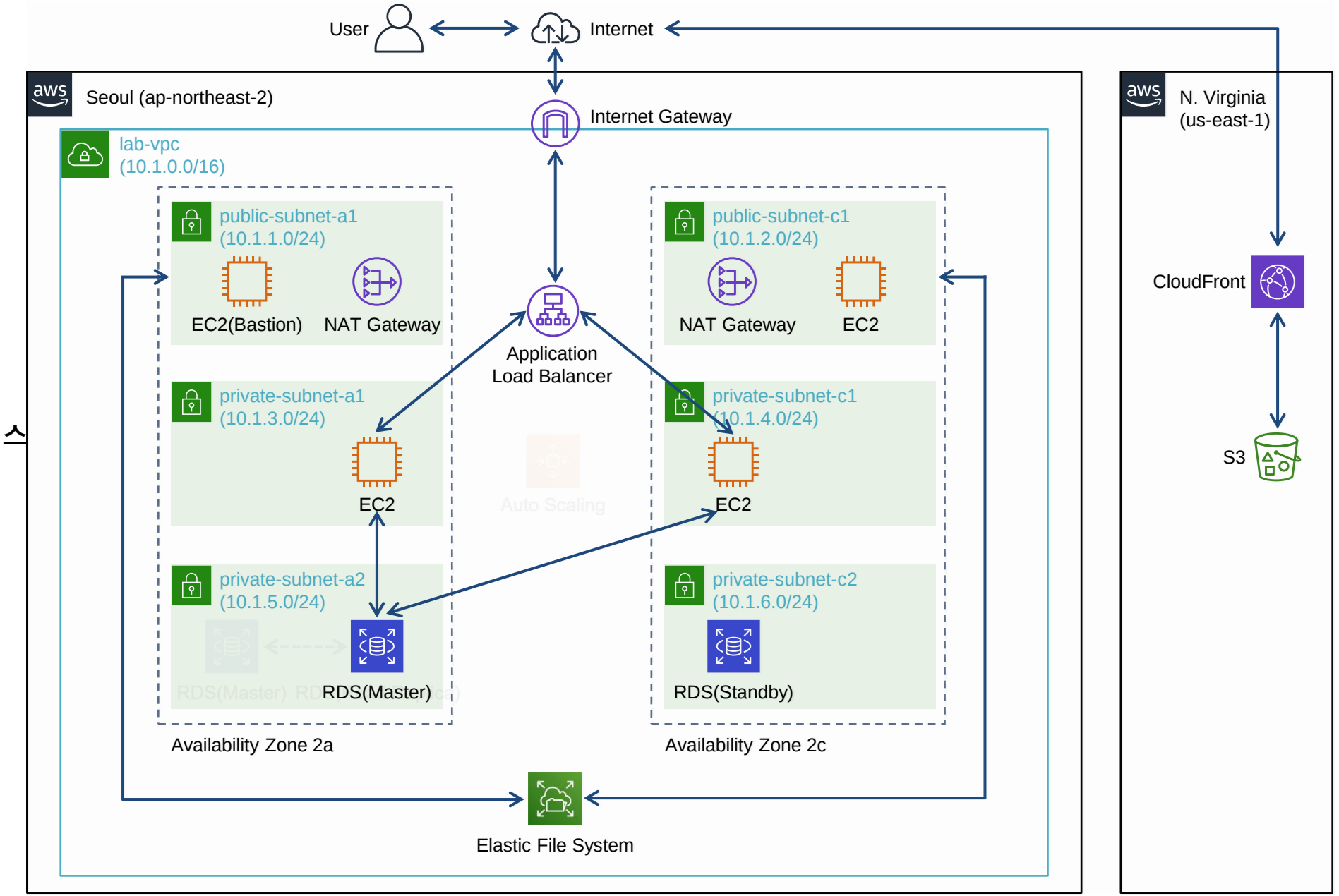
# (2) 웹 서버와 데이터베이스 인스턴스 연결

① EC2-데이터베이스 연결을 위한 정보 구성
- Endpoint
- Master user
- Master password 등

② 데이터베이스 접속

③ 테이블 생성 및 데이터 입력

④ 웹 브라우저를 통해 데이터베이스 연결 테스트



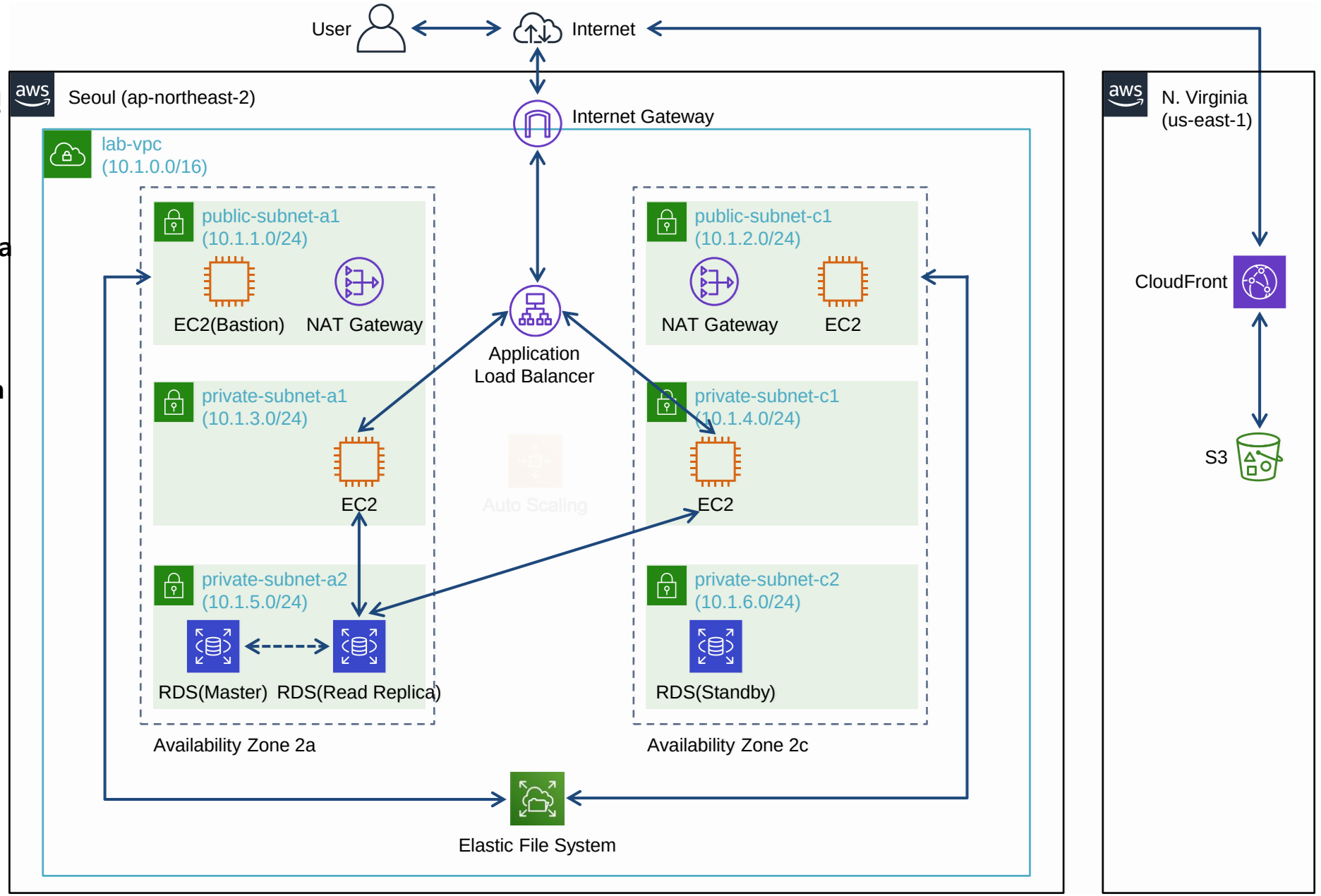
# (3) 데이터베이스의 Read replica 생성 및 웹 서버 연결

① 데이터베이스 Read Replica 생성

② EC2-Read Replica 데이터베이스 연결

③ 웹 브라우저를 통해 Read Replica 데이터베이스 연결 테스트

④ 원본 데이터베이스 변경 후 웹 브라우저를 통해 Read Replica 반영 테스트



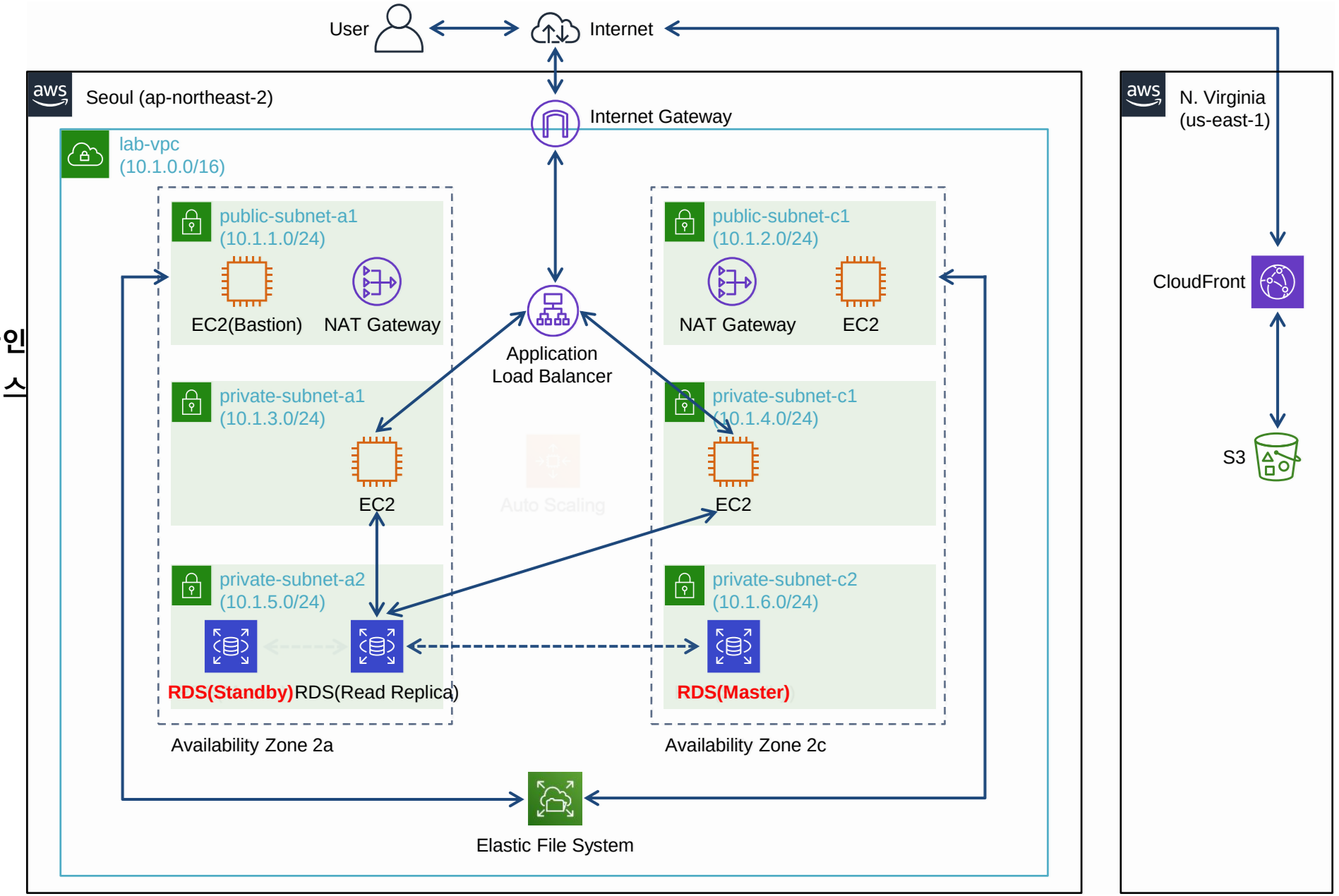
# (4) Failover를 통한 데이터베이스 이중화 테스트

① Master/Standby 데이터베이스 IP 정보 확인

② Master 데이터베이스 Failover 테스트 (Reboot with Failover)

③ Standby 데이터베이스가 Master 데이터베이스로 변경 확인

④ 웹 브라우저를 통해 데이터베이스 연결 테스트



# Chapter 04. Auto Scaling을 통한 확장성 및 탄력성 구현

# Auto Scaling을 통한 확장성 및 탄력성 구현

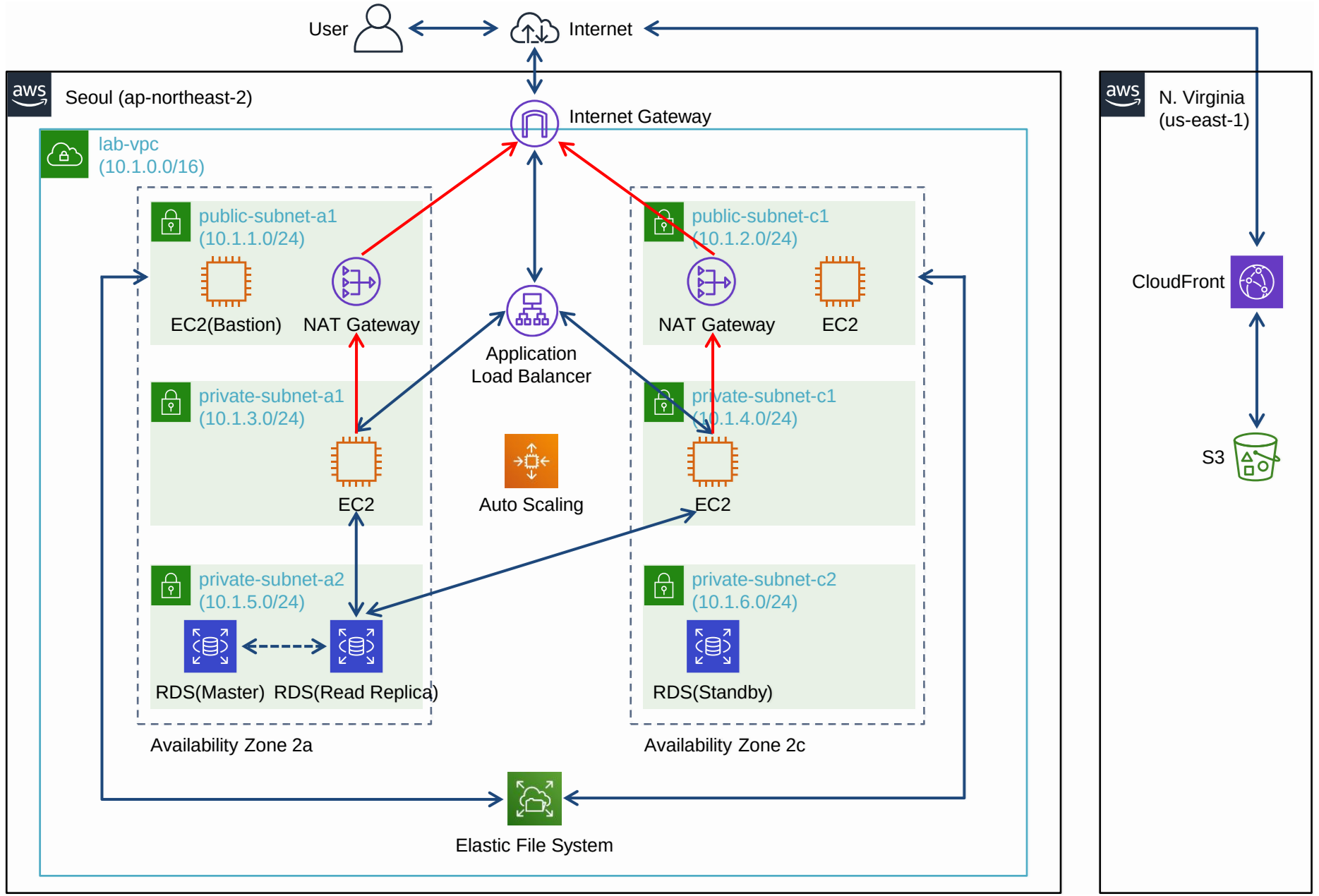
## 1. Chapter 개요

가상 서버의 용량을 자동으로 증가 또는 축소하는 기능을 사용하여 인프라에 확장성과 탄력성을 구현합니다.

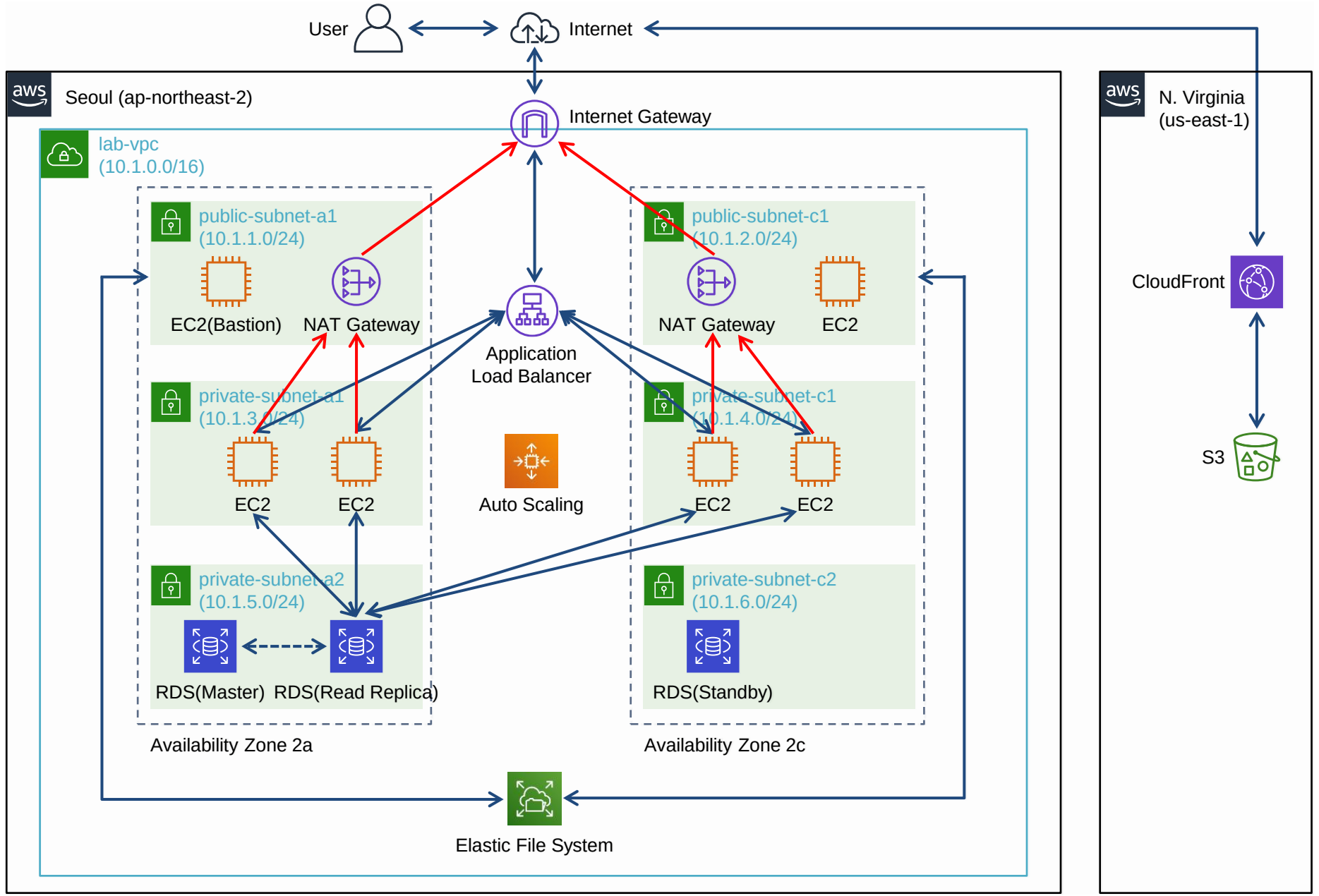
## 2. 사용하는 AWS 서비스

· Auto Scaling

# Auto Scaling을 통한 확장성 및 탄력성 구현



# Auto Scaling을 통한 확장성 및 탄력성 구현



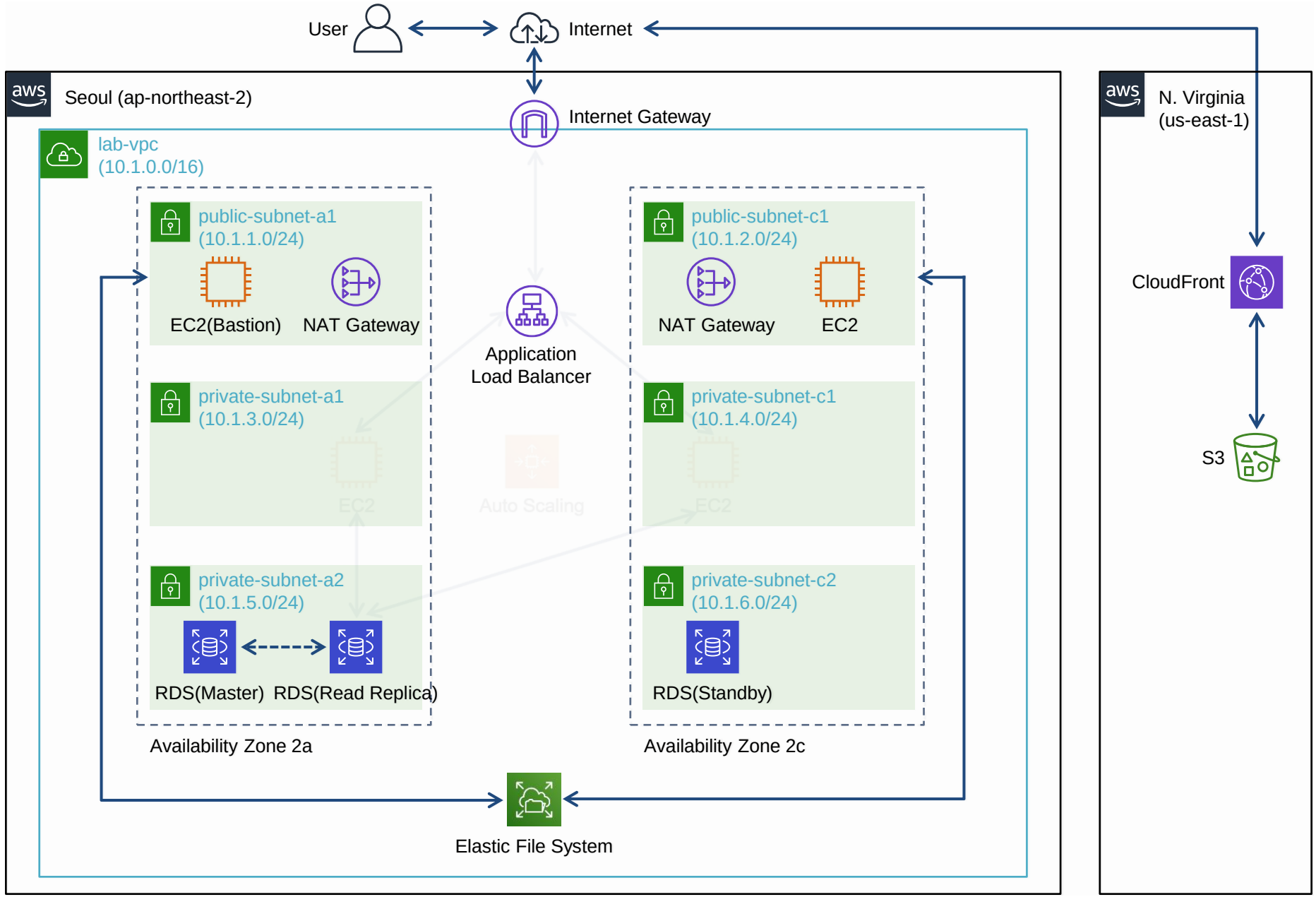
# (1) Auto Scaling을 위한 Launch Template 및 Application Load Balancer 구성

① Private subnet의 EC2에 대한 Custom AMI 생성

② Auto Scaling을 위한 Launch Template 생성

- AMI
- Key pair
- Network
- Storage 등

③ Auto Scaling용 Application Load Balancer 구성 (+ Target group)

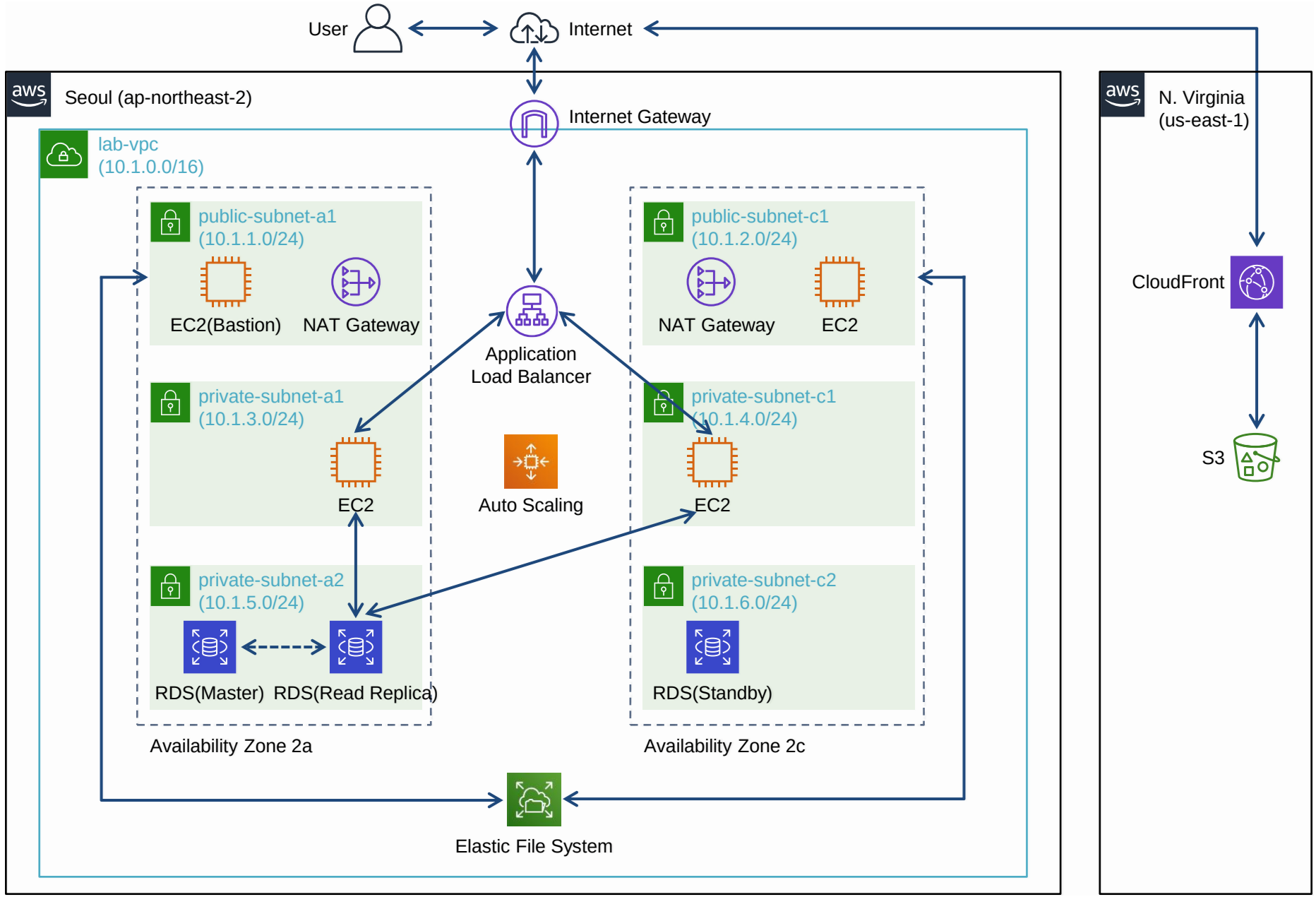


# (2) Auto Scaling Group 및 Scaling Policy 구성

① Auto Scaling Group 생성
- Launch options
- Group size
- Load balancer
- Tags 등

② Scaling Policy 구성

③ 콘솔 및 웹 브라우저를 통해
Auto Scaling Group 시작 확인



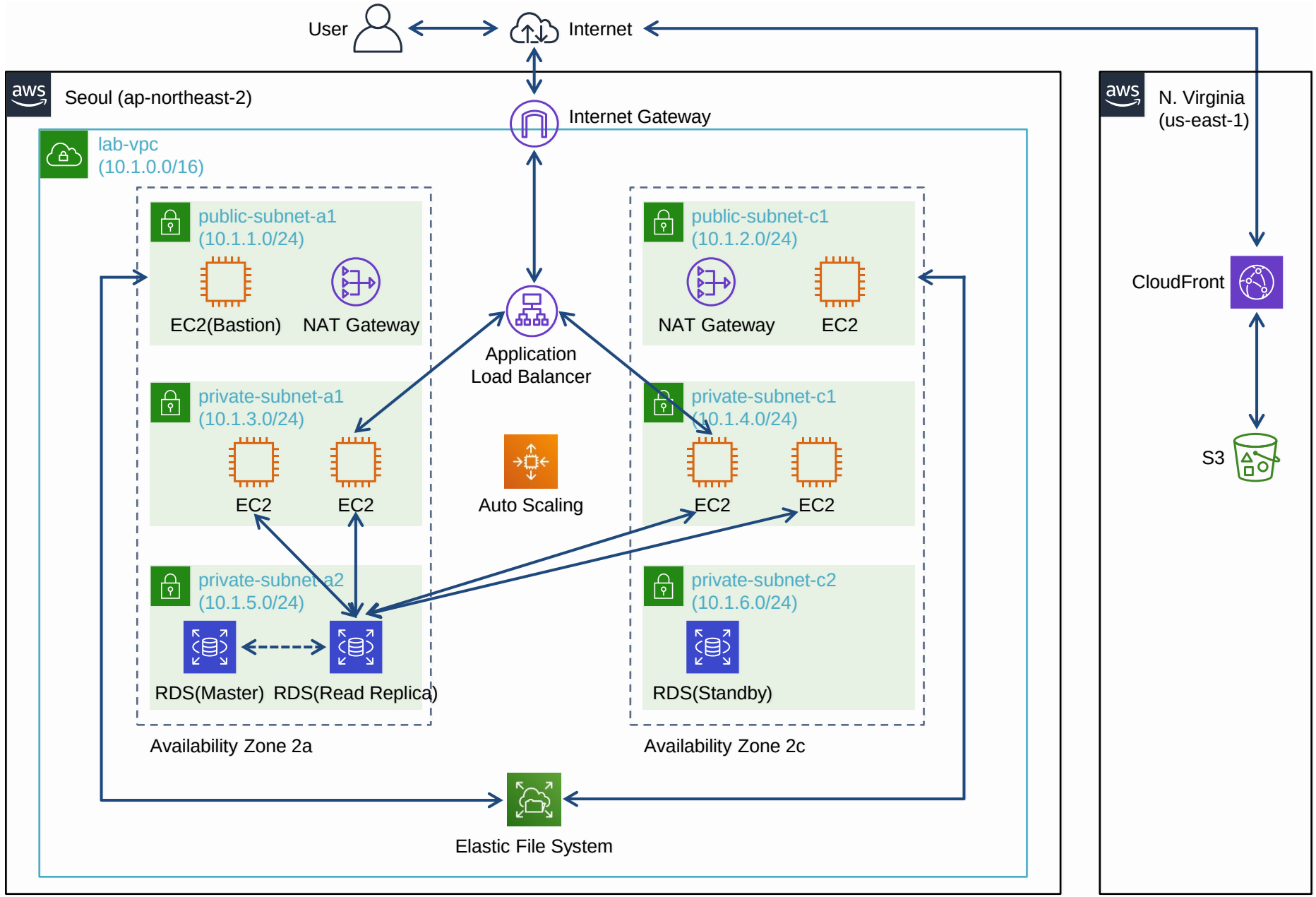
# (3) Auto Scaling Scale-Out 테스트

① Apache Bench Test를 위한 패키지(httpd-tools) 설치

② Application Load Balancer에 부하(로드) 테스트

③ CloudWatch를 통해 CPU Utilization rate 증가 확인

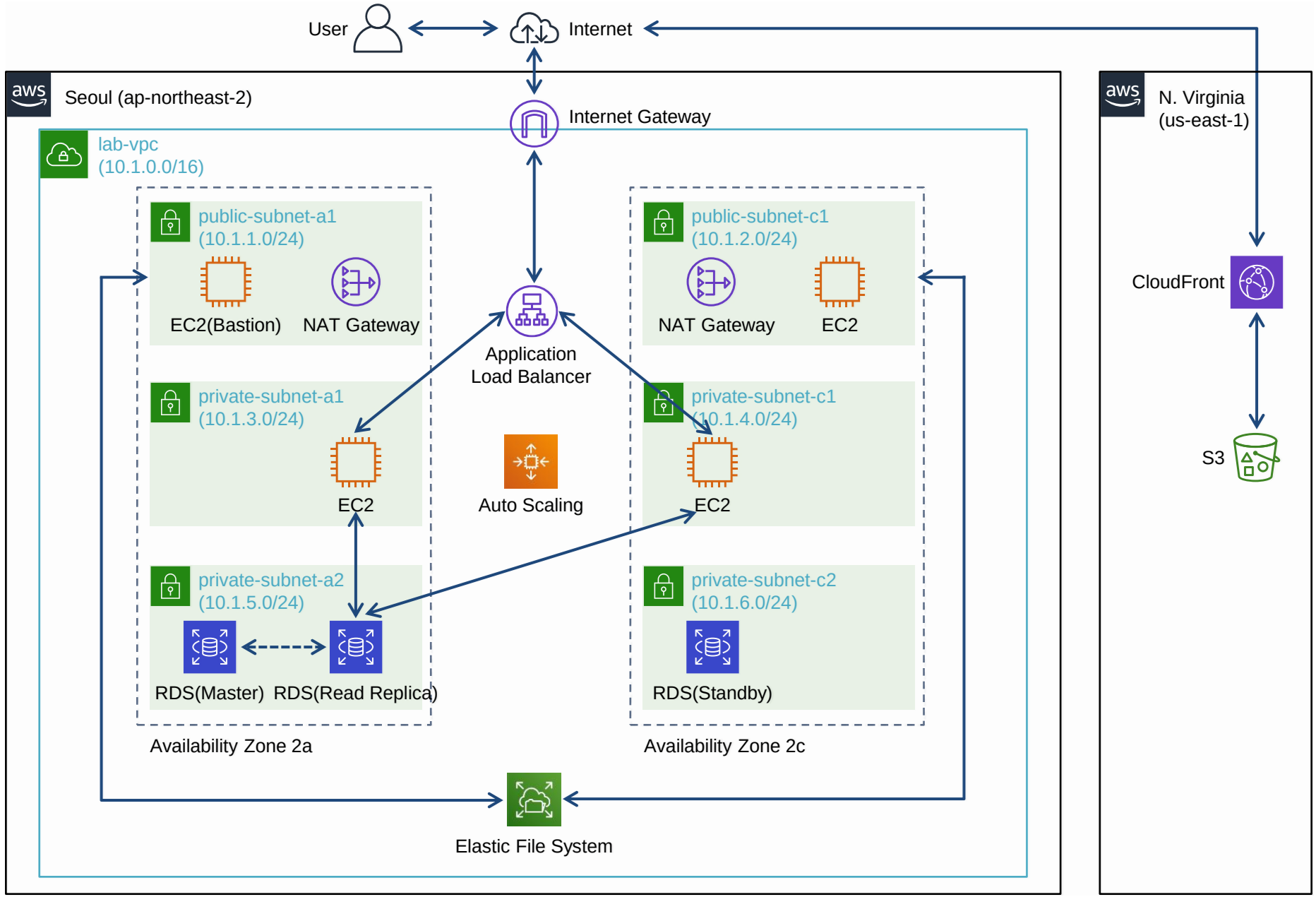
④ Scale-out으로 EC2 증가 확인



# (4) Auto Scaling Scale-In 및 Termination policy 살펴보기

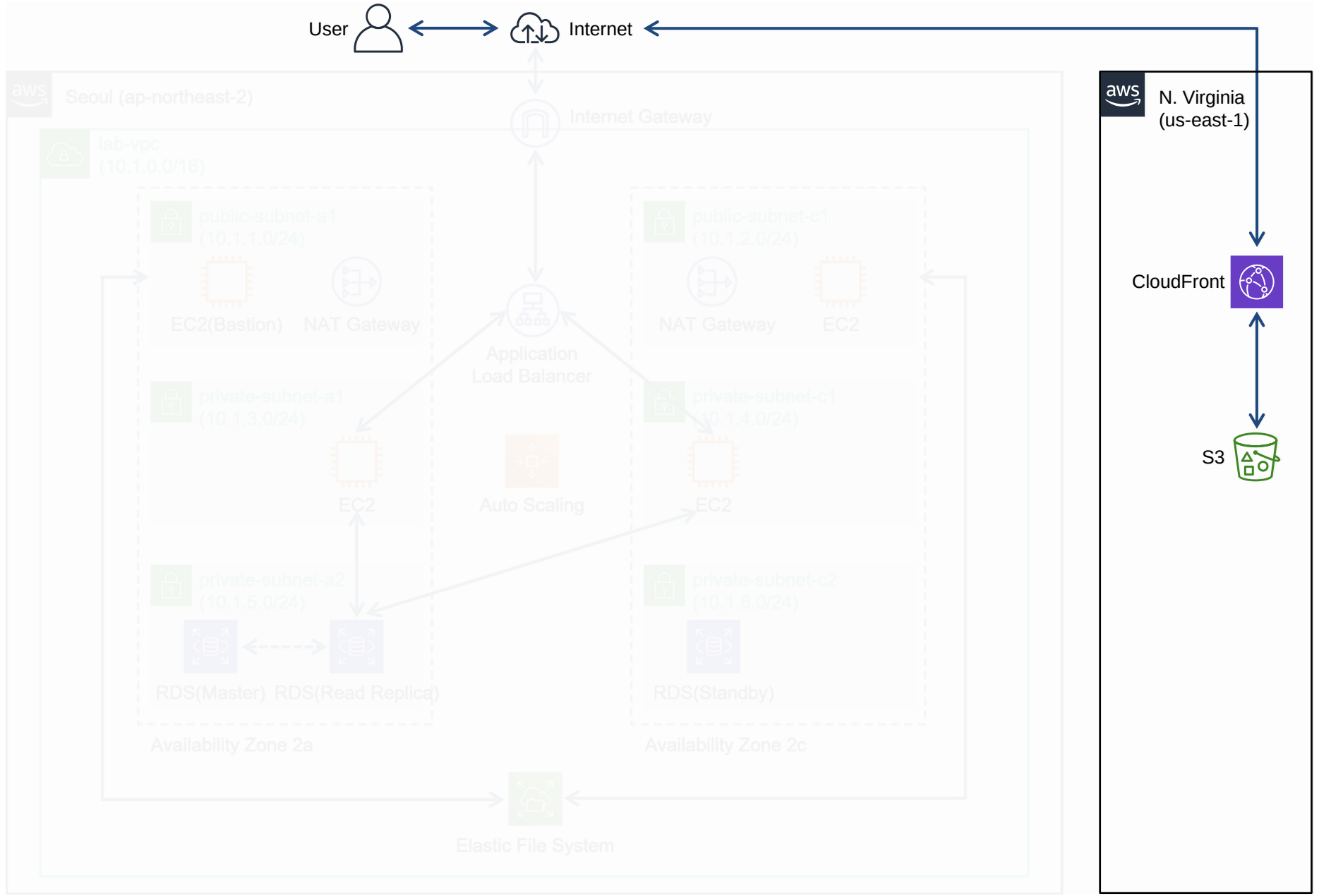
① Scale-in으로 EC2 감소 확인

② Auto Scaling Group의
Termination policy 확인

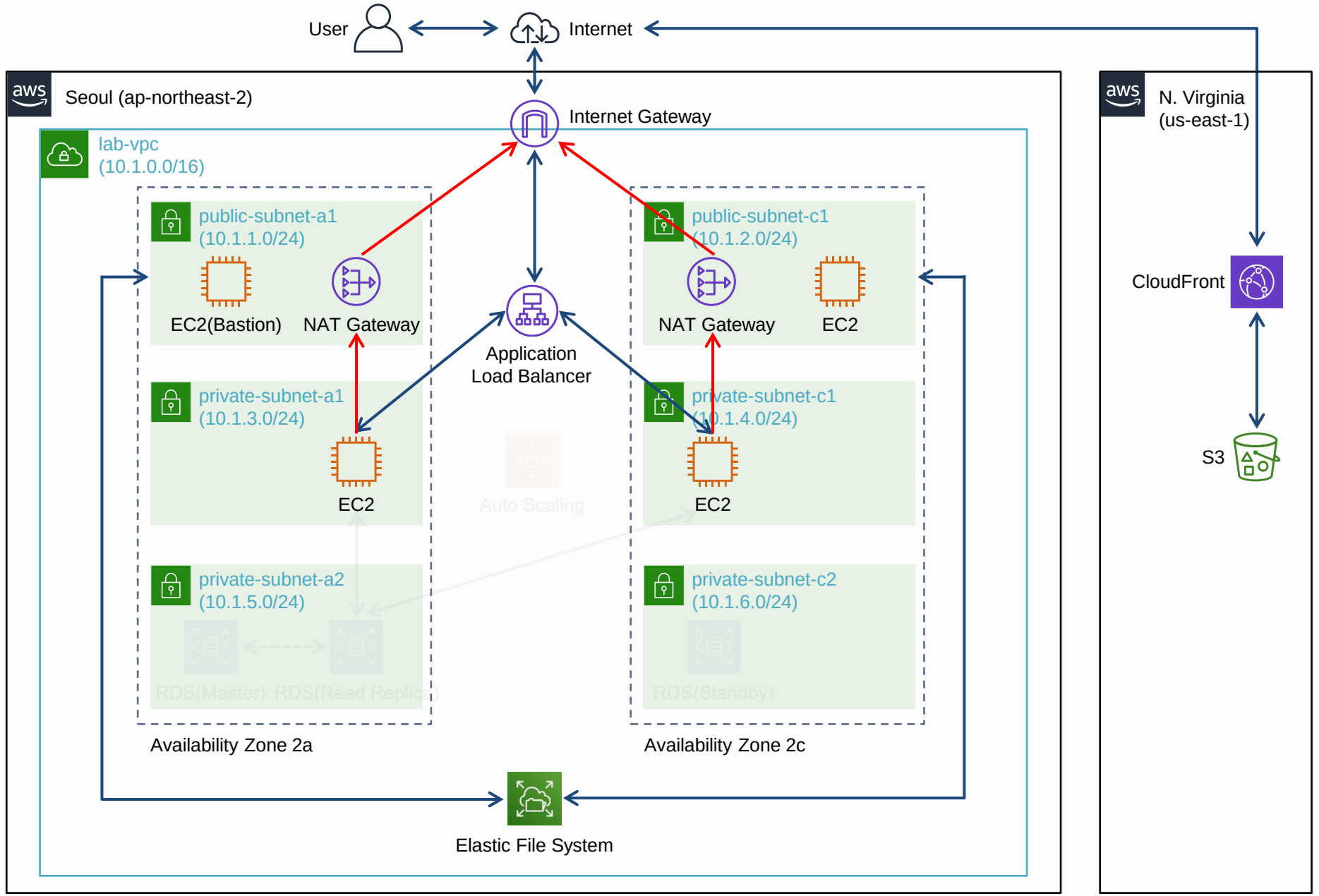


# 리뷰 및 마무리

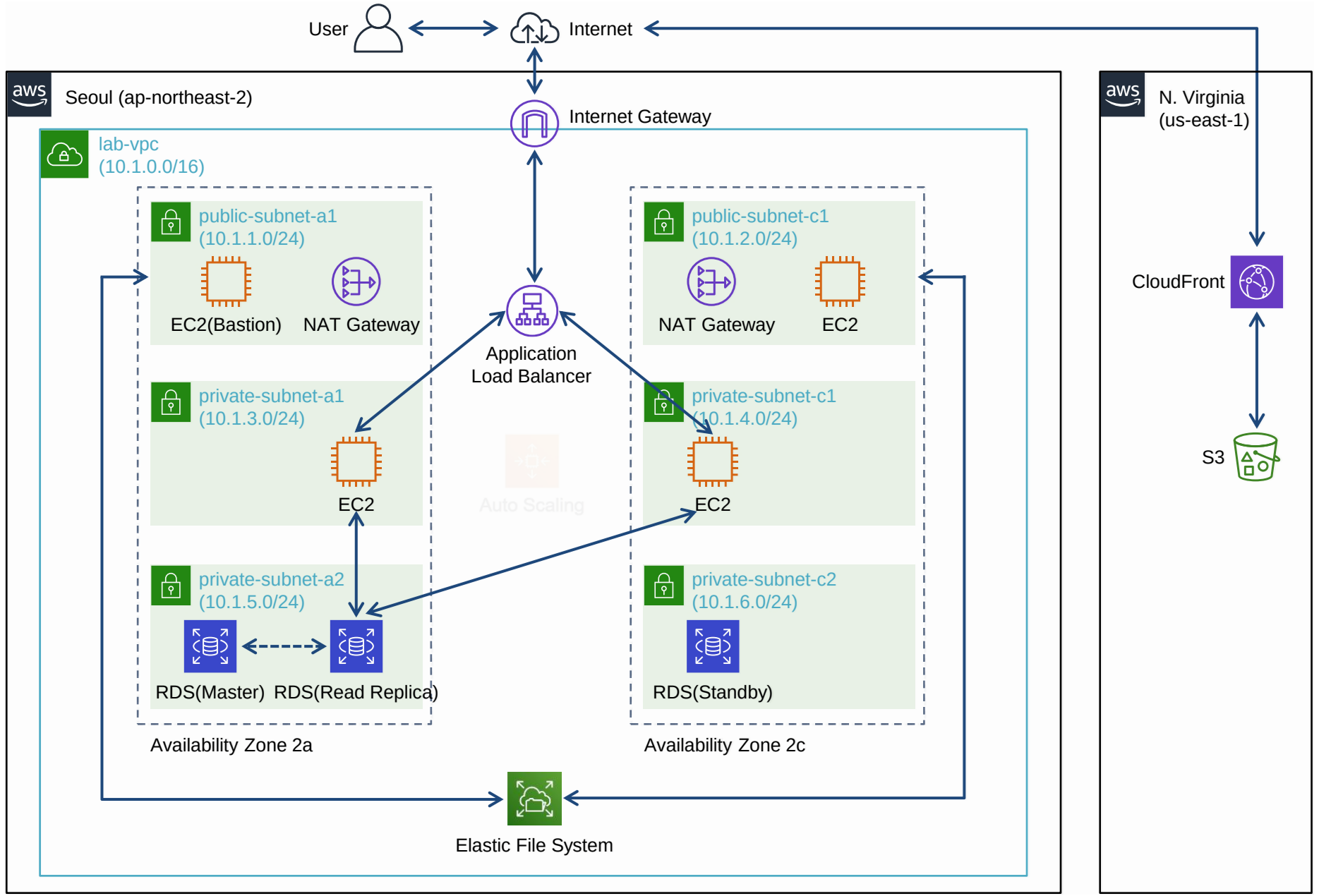
# 리뷰 : 서버리스 정적 웹사이트 호스팅 및 성능 가속화



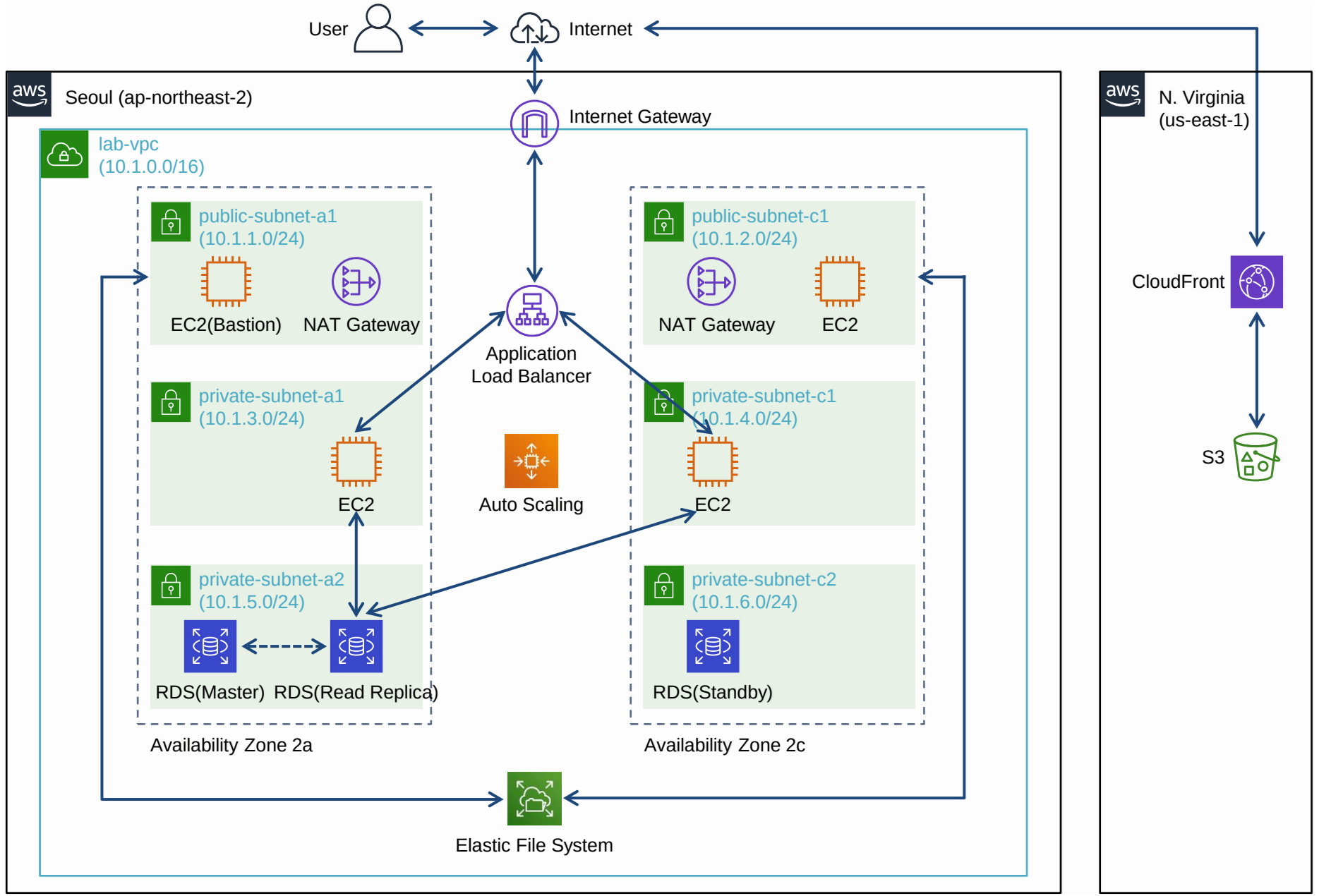
# 리뷰 : LAMP 웹 서버 및 Application Load Balancer 구성



# 리뷰 : 관계형 데이터베이스 서비스 구성



# 리뷰 : Auto Scaling을 통한 확장성 및 탄력성 구현



오랜 시간동안 고생 많으셨습니다!

다음 워크샵에서 다시 뵙겠습니다!