

BUDDYVERSE AWS CLOUD MIGRATION

2023 클라우드 인프라 구축 제안서



CONTENTS

01 사업 개요

- 사업 배경
- 고객 요구 사항

02 분석 및 솔루션

- 고객 요구 사항 분석
- 제공 서비스

03 구축 및 검증

- 아키텍처 설계
- 아키텍처 분석
- 구축 및 검증

04 기대 효과

- 요구사항 확인 및 기대 효과

01 사업 개요

■ 사업 배경

■ 고객사 PLAY BUDDY 의 AWS 클라우드 마이그레이션 ■



고객사 PLAY BUDDY는 공통의 관심사와 취향이 맞는 사람들이 취미를 공유하고 소통하는 소셜 미디어 플랫폼이다. 최근 비대면 모임 제한이 해제되고 모임에 대한 자유가 안정화되면서 이용자가 수가 급격하게 늘어나며 매출이 42% 증가하였다. PLAY BUDDY는 시공간의 제약 없이 많은 사람들이 소통할 수 있는 메타버스 기반 소셜 미디어 「BUDDYVERSE」로 사업 규모를 확장하기로 했다. 메타버스 상에서 게임, 쇼핑, 헬스 등 서비스를 확대할 계획이고, 2025년 글로벌 시장 진출을 통해 전 세계적으로 인프라 확장을 추진할 예정이다.



고객사는 급격한 사용자 증가로 인한 트래픽 폭주로 기존의 온프레미스 환경에서 메타버스 플랫폼을 출시할 경우 서버의 안정성 저하, 서비스 장애, 속도 저하, 운영 복잡성 증가 등의 문제를 지적하였다. 또한 메타버스를 구성하는 기술적 요소와 인력, 서버 운영 비용 등의 초기 투자 비용이 많이 들고, 추후 사용자 수나 서비스 활성화 정도에 대한 예측이 어려워 대처 방안이 필요한 상황이다.

따라서 PLAY BUDDY는 발생할 수 있는 문제에 대해 분석하고 그에 대응할 수 있는 인프라를 구축하여 이용자들이 불편함을 느끼지 않고 서비스를 사용할 수 있도록 클라우드로 마이그레이션을 의뢰하였다.

■ 고객 요구 사항

인프라의 유연한 확장성

초기 인프라 구축 및 유지 비용 절감

장애를 대비한 가용성 보장으로 지속적인 서비스 제공

서버 과부화로 인한 속도 저하 문제 해결

인프라 및 애플리케이션의 성능, 가용성 및 오류 감시

사용자 개인 정보 및 민감한 정보에 대한 보안

02 분석 및 솔루션

▣ 고객 요구 사항 분석

CSP 업체 선정

기존 인프라 환경을 클라우드로 마이그레이션하기 위해 클라우드 컴퓨팅 분야에서 압도적 점유율을 차지하는 **AWS (AWS Web Services) 클라우드 서비스**를 이용하였다. AWS는 가장 큰 클라우드 인프라 서비스를 제공하여 전 세계에 분산된 데이터 센터 네트워크를 보유하고 있으며, 컴퓨팅, 스토리지, 데이터베이스와 같은 인프라 기술부터 기계 학습 및 인공 지능, 사물인터넷 등의 신기술까지 다양한 서비스를 이용하여 클라우드 인프라를 쉽게 구축 및 운영할 수 있다.

고객사는 현재 메타버스로 서비스를 확장하여 다양한 플랫폼 런칭을 계획하고 있으며, 메타버스의 대규모 가상 세계를 지원하기 위해서는 안정적인 인프라 및 확장 가능한 컴퓨팅 리소스가 필요하다. AWS는 이러한 요구 사항을 충족하기 위한 다양한 제품과 서비스를 제공하고 있기 때문에 이번 프로젝트에서는 메타버스 환경을 안정적으로 운영하고, 사용자들이 원활하게 서비스를 이용할 수 있도록 AWS 업체의 서비스를 선정하여 클라우드 마이그레이션을 진행하였다.

보안 강화

데이터베이스와 웹 서버 등의 리소스에 대한 액세스 제어 및 보안을 제공하고, 최신 보안 기술을 적용하여 사용자 개인 정보 및 데이터를 보호

부하 분산

여러 대의 서버에 웹 서비스의 부하를 분산하여 웹 서비스의 가용성과 성능이 향상하고, 서버 폭주로 인한 웹 페이지 제공 지연 문제를 해결

유연한 확장성

필요에 따라 자동으로 확장할 수 있으므로, 예측할 수 없는 사용자 증가 및 서비스 활성화에 대처 용이



비용 절감

하드웨어를 구매하거나 유지보수를 위한 인력을 최소화하여 인프라 구축 및 유지 비용 절감

고가용성

여러 지역에 걸쳐 가용 영역을 제공하고, 각 가용 영역은 독립적으로 운영되기 때문에 장애 대비를 통한 고가용성 제공

업그레이드 및 유지보수 용이

자체적으로 하드웨어 및 소프트웨어를 업그레이드하여 비용과 시간을 절약할 수 있으며 지속적인 업데이트 및 보안 패치를 제공

■ 고객 요구 사항 분석 ■

1. 수요에 따른 인프라의 유연한 확장성

메타버스 플랫폼은 사용자 수 예측 및 서비스의 활성화 여부에 대한 예측이 불안정하므로 수요에 따른 유연한 확장성이 필수적이다. 클라우드는 수요에 따라 인스턴스의 크기를 즉시 조정하거나 데이터베이스의 크기와 처리량을 쉽게 조정할 수 있으므로 탄력적인 인프라 구축이 가능하다.

2. 최적의 리소스 사용을 통한 비용 절감

클라우드로 마이그레이션 시 물리 인프라를 구축할 필요가 없어 초기 구축 비용을 절감할 수 있으며 유연한 인프라 사용으로 최적의 리소스를 사용하여 비용을 절감할 수 있다. AWS에서는 예약 인스턴스, Spot 인스턴스 등 여러 가지 인스턴스 유형을 제공하여 상황 맞는 적합한 가격과 성능을 선택할 수 있어 효율적으로 자원을 사용할 수 있다.

3. 장애 대비 고가용성

대규모의 이용자들이 동시에 접속하고 서비스를 이용하기 때문에 장애가 발생할 경우 서비스 이용에 큰 영향을 미칠 수 있다. 따라서 장애를 대비한 가용성 보장으로 지속적인 서비스를 제공하여 장애 시간을 최소화하고 이용자들의 만족도를 높일 수 있다.

4. 읽기 요청에 대한 부하 분산

사용자의 읽기 요청이 증가하면서 서버가 다운되거나 성능이 저하되어 웹 페이지 제공에 문제가 발생하고 있다. AWS RDS 읽기 전용 복제본을 생성하면 단일 DB 인스턴스 용량을 한도 이상으로 탄력적 스케일 아웃이 가능하여 읽기 중심의 데이터베이스 워크로드를 처리할 수 있다.

5. 실시간 모니터링을 통한 리소스 관리

AWS CloudWatch를 이용하면 실시간 모니터링을 통해 다양한 AWS 서비스 및 리소스에 대한 데이터를 수집하고, 대시보드를 사용하여 시스템의 상태와 성능 정보를 시각화하여 각종 지표를 한눈에 파악하고 빠르게 대응할 수 있다.

6. SSL/TLS 설정을 통한 보안 강화

커뮤니티 기반의 메타버스 플랫폼은 가상 공간에서 여러 사용자들이 상호작용하는 환경으로 사용자 콘텐츠, 개인정보 등의 모든 데이터가 암호화되어 저장되어야 한다. AWS 보안그룹의 리소스에 대한 인바운드 및 아웃바운드 트래픽 제어와 AWS ACM 서비스를 이용하여 SSL/TLS 인증서 발급 관리 및 배포를 통하여 보안성을 강화할 수 있다.

제공 서비스

COMPUTING

[AWS EC2]

AWS Elastic Compute Cloud



- 클라우드 컴퓨팅 인프라 서비스로 가상 서버를 제공하여 사용자가 필요에 따라 쉽게 서버를 생성, 시작, 중지, 종료 등의 인프라 관리 가능
- 수평적 확장이 가능하여 사용자의 요구에 따라 필요한 만큼의 컴퓨팅 리소스 확장이 가능하며 다양한 보안 그룹, 네트워크 설정, 로드 밸런싱 등의 기능을 제공하여 안정성을 높임
- Auto Scaling 기능을 통해 사용자의 서비스가 갑작스럽게 인기를 얻어도 서비스를 안정적으로 제공하며 사용한 만큼만 비용을 지불하므로 사용자는 필요한 만큼의 리소스만 사용하고 비용을 절감

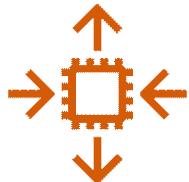
유연한
인프라 관리

확장성

안정성

고가용성

비용 절감



- 애플리케이션의 부하량에 따라 자동으로 컴퓨팅 리소스의 개수를 조정하는 기능으로 애플리케이션의 가용성과 성능을 향상시키고, 리소스를 효율적으로 사용 가능

확장성

고가용성

비용 절감

STORAGE

[AWS EBS]

AWS Elastic Block Store



- EC2 인스턴스에서 사용할 수 있는 영구 블록 스토리지로 안정적이고 안전한 데이터 스토리지 제공
- 스냅샷 기능을 제공하여 데이터를 백업하고, 이전 상태로 롤백하는 기능 제공
- EBS 볼륨은 다른 EC2 인스턴스에 연결하여 사용하여 볼륨을 사용하는 EC2 인스턴스가 중단되더라도 데이터 손실 없이 다른 인스턴스에서 계속 사용 가능

고성능 볼륨

고가용성

암호화

액세스 관리

스냅샷

[AWS EFS]

AWS Elastic Block Store



- 클라우드 기반의 완전 관리형 NFS(Network File System) 파일 스토리지 서비스
- 여러 EC2 인스턴스에서 공유할 수 있는 파일 시스템을 제공하며, 동시에 여러 가용 영역에서 데이터를 자동으로 복제하여 고가용성과 내구성을 보장
- 필요에 따라 파일 시스템을 쉽게 확장하거나 축소할 수 있으며, 사용한 용량과 시간 단위로 비용을 지불

확장성

고가용성

내구성

보안 강화

액세스 용이

DATABASE

[AWS RDS]

AWS Relational Database Service



- 클라우드 기반의 완전 관리형 관계형 데이터베이스 서비스로 MySQL, PostgreSQL, Oracle, SQL Server 등 다양한 데이터베이스 엔진을 지원하며, 데이터베이스 인스턴스를 쉽게 프로비저닝, 관리, 백업, 복원 가능
- 여러 가용 영역에서 자동으로 데이터베이스 인스턴스를 복제하여 고가용성과 내구성을 보장하며, 필요에 따라 인스턴스 크기를 확장하거나 축소

확장성

고가용성

내구성

보안 강화

NETWORKING & CONTENT DELIVERY

[AWS VPC]

AWS Virtual Private Cloud



- 사용자의 AWS 계정 내에서 가상의 프라이빗 클라우드 환경을 구성 할 수 있는 서비스로 전용 네트워크이기 때문에 고객사의 데이터 및 서비스를 외부로부터 격리시켜 보안성을 강화
- 고객사의 네트워크 구성에 맞게 필요한 수의 서브넷, 라우팅 테이블, 인터넷 게이트웨이 등을 생성하고 구성하여 확장성이 높은 시스템 구축
- 고객사가 필요한 수의 서브넷을 생성할 수 있으며, 필요한 경우에만 인터넷 연결을 추가하여 비용 절감

보안 강화

유연한 아키텍처 구성

효율적인 비용 관리

[AWS Route 53]

- 클라우드 기반의 DNS (Domain Name System) 웹 서비스
- 웹 사이트 또는 애플리케이션을 인터넷에서 호스팅할 때 필요한 도메인 이름과 IP 주소를 관리
- DNS 인프라를 관리하고, 고가용성 및 확장성을 갖춘 DNS 서비스를 제공하여 웹 사이트 및 애플리케이션의 가용성을 향상



고가용성

확장성

유연성

AWS 서비스와의 통합

[AWS ELB]

AWS Elastic Load Balancer

- 분산 웹 애플리케이션 로드 밸런서로 여러 EC2 인스턴스, 컨테이너, IP 주소 또는 Lambda 함수와 같은 여러 대상에 대한 트래픽을 자동으로 분산시켜주는 서비스
- 인스턴스 간에 부하 분산이 가능하며, 인스턴스 또는 컨테이너가 실패하는 경우 자동으로 다른 인스턴스 또는 컨테이너로 트래픽을 전달
- SSL 암호화와 같은 보안 요구 사항을 충족시키기 위한 기능을 제공하며, AWS Route 53과 통합하여 DNS 이름에 대한 로드 밸런싱을 지원



부하 분산

확장성

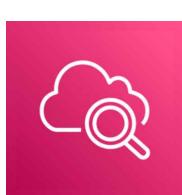
보안 강화

모니터링

Auto Scaling
그룹과의 통합

MANAGEMENT & GOVERNANCE

[AWS CloudWatch]



- 모니터링 서비스로, 다양한 AWS 리소스 및 애플리케이션의 성능 데이터 및 로그를 수집하고, 분석, 저장 및 시각화할 수 있는 서비스
- 모니터링된 지표와 로그 데이터를 기반으로 경보를 생성하며 모니터링한 지표 및 로그 데이터를 대시보드에서 시각화

모니터링

경보

대시보드

이벤트

분석

SECURITY

[AWS ACM]

AWS Certificate Manager



- 완전 관리형 SSL/TLS 인증서 관리 서비스로 SSL/TLS 인증서의 발급, 관리 및 배포를 쉽게 처리 수 있으며, 무료 공인 SSL/TLS 인증서를 제공

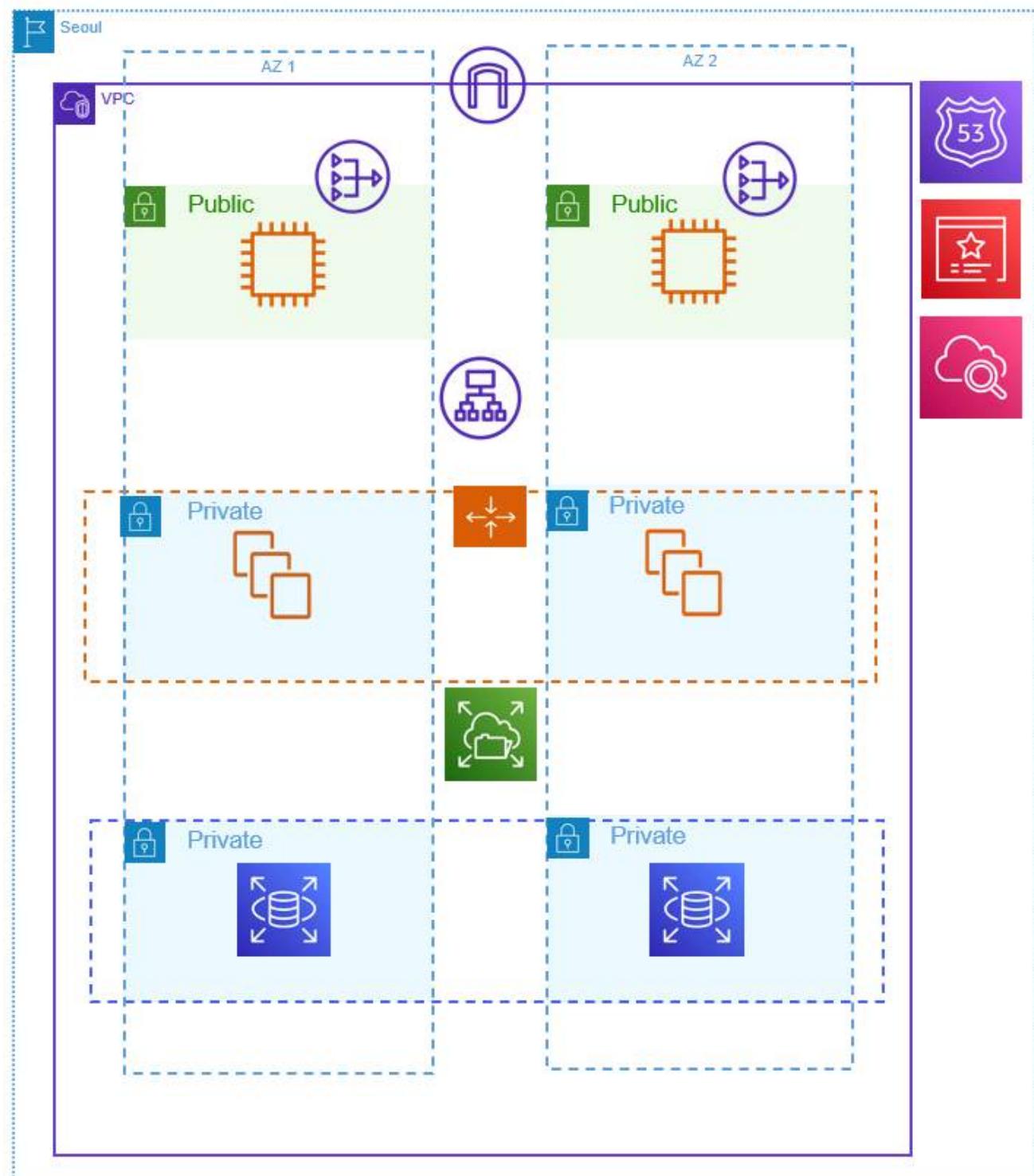
- ◆ AWS의 다른 서비스와 쉽게 통합되어 보안성과 신뢰성이 높은 웹 사이트 및 애플리케이션을 쉽게 배포

SSL/TLS 인증서 발급 및 관리

보안 강화

03 구축 및 검증

▣ 아키텍처 설계



▣ 아키텍처 분석

1 개의 VPC 에 이중화를 위해 2 개의 가용영역을 이용하고, 하나의 가용 영역 안에 퍼블릭 서브넷 1개와 2개의 프라이빗 서브넷으로 구성하였다. 퍼블릭 서브넷에는 외부 인터넷 통신이 가능하도록 NAT 게이트웨이와 Bastion Host를 배치하고, 프라이빗 서브넷에는 웹 서버 인스턴스와 RDS 데이터베이스를 구성하였다. 확장성과 가용성, 부하 분산의 목적으로 웹 서버 인스턴스를 오토스케일링 그룹으로 그룹화하고 ELB 를 생성하였다. 그리고 EFS 를 생성하여 자주 변경 및 사용되는 데이터나 이미지를 저장해두고 여러 인스턴스에서 공유해서 사용하여 시스템의 효율성과 안정성을 높혔다. 또한 ACM 서비스를 통해 데이터의 암호화를 구현하여 보안을 강화하고, CloudWatch 서비스를 이용하여 실시간 모니터링을 통해 안정적인 운영과 문제 예방을 실현하였다.

➤ Region & VPC & AZ

Seoul Region 1개의 VPC를 생성하고 2개의 가용 영역으로 구성하여 고가용성과 내구성을 보장할 수 있다. 하나의 가용 영역에 장애가 발생하더라도 다른 가용 영역에서 시스템의 지속할 수 있으므로 데이터의 백업 및 복원이 가능하며 안정성을 향상시킨다.

➤ Public Subnet & Private Subnet

각 가용 영역에는 Public Subnet 1개와 Private Subnet 2개로 구성하여 보안성과 확장성을 보장하였다. Public Subnet은 외부 인터넷과 연결되어 외부 통신 관련 서비스를 배치하였으며, Private Subnet은 내부 인프라나 데이터베이스 서버와 같이 중요한 서버들을 배치하여 외부에서의 직접적인 접근을 차단하였다.

➤ NAT Instance & Bastion Host

Public Subnet 에는 AWS NAT Instance 와 EC2 인스턴스를 사용하여 구성한 Bastion Host를 배치하였다. AWS NAT Instance 를 통하여 인터넷에 연결하여 private Subnet에서 생성된 리소스가 외부에 직접 노출하지 않고 외부 인터넷과 통신할 수 있다. 또한 AWS Bastion Host는 외부에서 접속이 가능한 인스턴스로 모든 외부 통신이 거쳐가게 되며 이를 통해 VPC 내부에 위치한 인스턴스의 보안성이 강화된다.

(※ 아키텍처 설계시 NAT Gateway 로 구성하였으나 실습시 비용 절감의 목적으로 NAT Instance 로 구축함)

➤ Autoscaling Group & AWS ELB

Private Subnet 의 웹 서버를 Autoscaling Group 으로 그룹화하고, ELB 와 함께 사용하여 웹 서버 인스턴스의 확장 및 축소에 따라 ELB 의 연결 수가 자동으로 조절되어 트래픽을 분배한다. ELB 는 웹서버의 주기적 Health check를 통해 문제가 발생한 인스턴스를 탐지하고 해당 인스턴스로의 연결을 차단한다. Autoscaling Group은 Health check 결과에 따라 인스턴스를 자동으로 추가 및 삭제하여 시스템의 가용성과 안정성을 보장하며 자원 활용도를 최적화한다.

➤ Storage > AWS EFS

AWS EFS는 분산 파일 시스템 프로토콜인 NFS (Network File System)를 사용하여 여러 개의 EC2 인스턴스에서 동시에 접근 가능한 공유 파일 시스템을 제공한다. AWS EFS 웹 어플리케이션의 /static과 /media 디렉토리를 마운트하여 정적 파일(static files)과 미디어 파일(media files)을 효율적으로 관리하고 사용자가 웹 어플리케이션의 /static 또는 /media URL에 대한 요청을 보내면 Django는 해당 파일을 EFS 파일 시스템에서 읽어와 사용자에게 HTTP 응답으로 반환한다. AWS EFS와 같은 외부 파일 시스템을 사용하여 파일 저장 및 처리에 대한 부하를 분산시키고, 확장성과 가용성을 높일 수 있다.

➤ Monitoring > AWS Cloudwatch

AWS Cloudwatch는 EC2 인스턴스, RDS 데이터베이스, Elastic Load Balancer 등과 같은 AWS 리소스의 상태 및 성능 지표를 모니터링하여 문제를 빠르게 감지하고 대응할 수 있다. AWS CloudWatch 대시보드를 생성하면 모니터링 대상 지표를 시각화하여 문제점을 쉽게 파악하고 분석할 수 있으며 이를 통해 안정적인 운영과 문제 예방, 비용 최적화를 실현할 수 있다.

▣ 구축 및 검증

아키텍쳐와 요구 사항을 바탕으로 실제 구축이 진행되었으며, 동시에 테스트가 필요한 부분에서 동작에 대한 검증을 진행하였다.

1. 인프라 | VPC 및 Subnet 생성

◆ VPC 생성

| VPC (1) 정보 | | | | |
|--------------------------|---------------|-----------------------|------------------------|---------------|
| | Name | VPC ID | 상태 | IPv4 CIDR |
| <input type="checkbox"/> | Buddy-Project | vpc-08e3486799aa74735 | Available | 172.31.0.0/16 |

◆ Subnet 생성

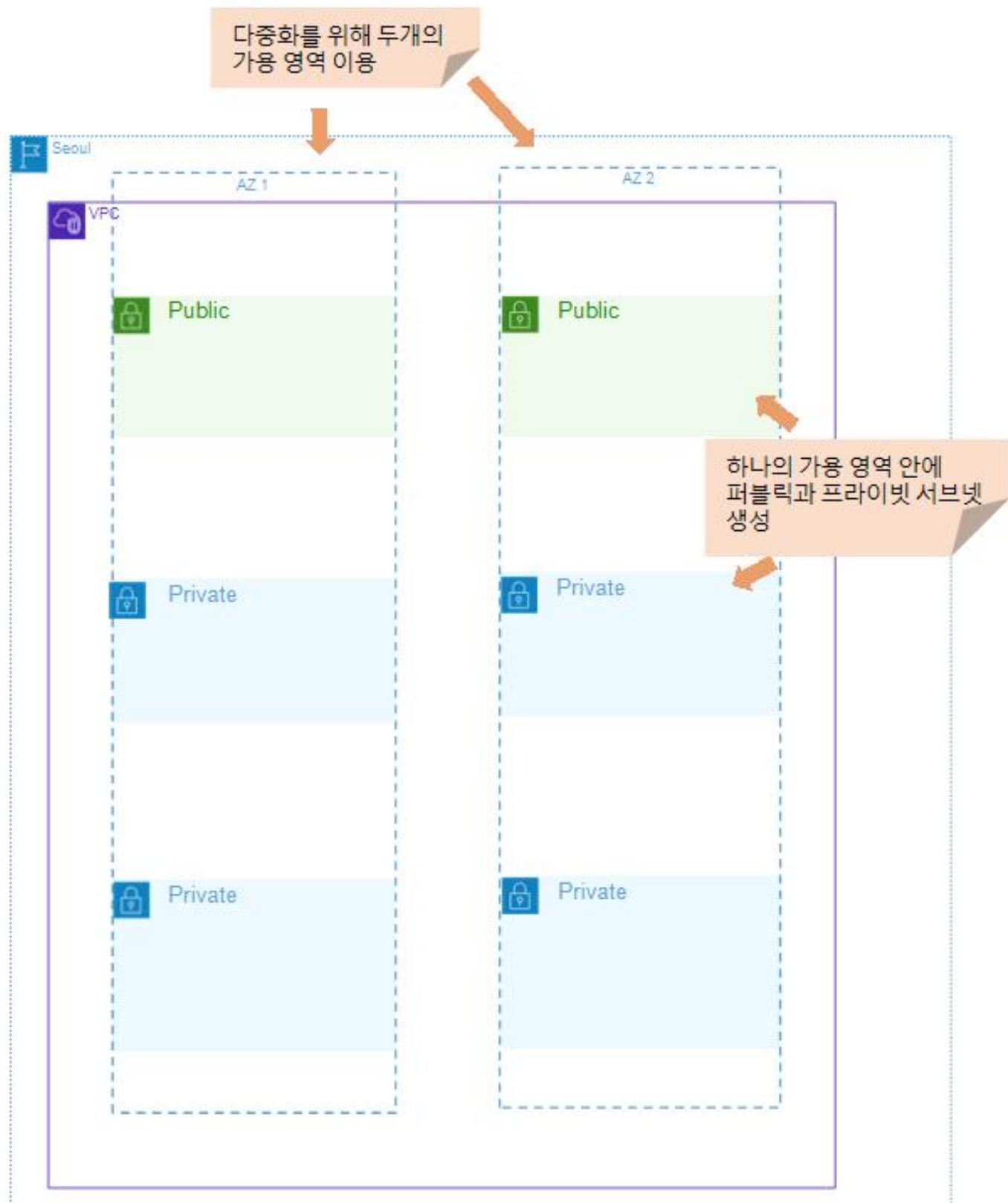
| AZ | Subnet | CIDR |
|-----------------|--------------------|----------------|
| ap-northeast-2a | Buddy-Public1 | 172.31.10.0/24 |
| | Buddy-Web-Private1 | 172.31.11.0/24 |
| | Buddy-DB-Private1 | 172.31.12.0/24 |
| ap-northeast-2c | Buddy-Public2 | 172.31.20.0/24 |
| | Buddy-Web-Private2 | 172.31.21.0/24 |
| | Buddy-DB-Private2 | 172.31.22.0/24 |

서브넷 생성 확인

| 서브넷 (6) 정보 | | | | |
|--------------------------|--------------------------|------------------------|-----------------------|----------------|
| | 서브넷 ID | 상태 | VPC | IPv4 CIDR |
| <input type="checkbox"/> | subnet-0f68b5444b86408b0 | Available | vpc-08e3486799aa74735 | 172.31.22.0/24 |
| <input type="checkbox"/> | subnet-0f3ce790bba29d4ad | Available | vpc-08e3486799aa74735 | 172.31.11.0/24 |
| <input type="checkbox"/> | subnet-065d7b26d1f8249c3 | Available | vpc-08e3486799aa74735 | 172.31.12.0/24 |
| <input type="checkbox"/> | subnet-069661c2fabd94d7d | Available | vpc-08e3486799aa74735 | 172.31.10.0/24 |
| <input type="checkbox"/> | subnet-0ce6b289fdbaf9bfd | Available | vpc-08e3486799aa74735 | 172.31.20.0/24 |
| <input type="checkbox"/> | subnet-065d7b26d1f8249c3 | Available | vpc-08e3486799aa74735 | 172.31.21.0/24 |
| <input type="checkbox"/> | subnet-0f3ce790bba29d4ad | Available | vpc-08e3486799aa74735 | 172.31.22.0/24 |

[생성 리소스 요약]

VPC를 생성하여 논리적으로 분리된 네트워크를 생성하고,
2개의 가용 영역에 각각 퍼블릭 서브넷 1개, 프라이빗 서브넷 2개를 구성하였다.



1. 인프라 | Internet Gateway & Nat Gateway 및 Routing table 설정

◆ Internet Gateway 생성

| 항목 | 값 |
|-------|---------------|
| 이름 태그 | Buddy-GW |
| VPC | Buddy-Project |

| Name | 인터넷 게이트웨이 ID | 상태 | VPC ID | 소유자 |
|----------|-----------------------|----------|--------------------------------------|--------------|
| Buddy-GW | igw-0e437e5ca59a2a24a | Attached | vpc-08e3486799aa74735 Buddy-Pro... | 567378238963 |

◆ NAT Gateway 생성

비용 절감의 목적으로 본 프로젝트에서는 NAT Instance를 사용하였다.

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard. In the 'Name and tags' step, the stack name is 'Buddy-NAT1'. Under 'Applying configuration', the 'Template' is selected. In the 'Configure template' section, the 'Template' dropdown is set to 'Amazon Machine Image (AMI)'. The search bar contains 'NAT'. A red box highlights the search results for 'Amazon Machine Image (AMI)', showing the selected item: 'amzn-ami-vpc-nat-2018.03.0.20220609.0-x86_64-ebs'. To the right, there's a note about the instance type: 't2.micro' (free tier). The 'Review and Create' button is at the bottom.

◆ 라우팅 테이블 생성

라우팅 테이블 생성

| Name | 라우팅 테이블 ID | 명시적 서브넷 연결 | 엣지 연결 | 기본 | VPC |
|---------------|-----------------------|------------|-------|----|-------------------------------|
| Buddy-Routing | rtb-0b62985b128b39a03 | - | - | 예 | vpc-08e3486799aa74735 Bu... |

라우팅 테이블 생성 후 퍼블릭 서브넷과 연결

| 서브넷 연결 편집 | | | |
|---|--------------------------|----------------|--|
| 이용 가능한 서브넷 (2/6) | | | |
| <input type="text"/> 서브넷 연결 필터링 | | | |
| 이름 | 서브넷 ID | IPv4 CIDR | |
| <input type="checkbox"/> Buddy-DB-Private2 | subnet-0ce6b289fdbaf9bfd | 172.31.22.0/24 | |
| <input type="checkbox"/> Buddy-Web-Private1 | subnet-065d7b26d1f8249c3 | 172.31.11.0/24 | |
| <input type="checkbox"/> Buddy-DB-Private1 | subnet-069661c2fabd94d7d | 172.31.12.0/24 | |
| <input checked="" type="checkbox"/> Buddy-Public1 | subnet-0f68b5444b86408b0 | 172.31.10.0/24 | |
| <input checked="" type="checkbox"/> Buddy-Public2 | subnet-0f3ce790bba29d4ad | 172.31.20.0/24 | |
| <input type="checkbox"/> Buddy-Web-Private2 | subnet-06aba022a87117a3b | 172.31.21.0/24 | |

퍼블릭 라우팅 테이블 연결 확인

| rtb-0b62985b128b39a03 / Buddy-Routing | | | |
|---------------------------------------|--------------------------|----------------|-------|
| 세부 정보 | 라우팅 | 서브넷 연결 | 엣지 연결 |
| 라우팅 편집 | | | |
| 엔지 연결 편집 | | | |
| 명시적 서브넷 연결 (2) | | | |
| <input type="text"/> 서브넷 연결 검색 | | | |
| Name | 서브넷 ID | IPv4 CIDR | |
| Buddy-Public1 | subnet-0f68b5444b86408b0 | 172.31.10.0/24 | |
| Buddy-Public2 | subnet-0f3ce790bba29d4ad | 172.31.20.0/24 | |

프라이빗 서브넷용 라우팅 테이블 생성

| Name | 라우팅 테이블 ID | 명시적 서브넷 연결 | 엣지 연결 | 기본 | VPC |
|---|-----------------------|--------------------------|-------|-----|-------------------------------|
| <input type="checkbox"/> Buddy-Routing | rtb-0b62985b128b39a03 | 2 서브넷 | - | 예 | vpc-08e3486799aa74735 Bu... |
| <input type="checkbox"/> Buddy-Private-GW2 | rtb-0781123f751f4bed8 | subnet-06aba022a87117a3b | - | 아니요 | vpc-08e3486799aa74735 Bu... |
| <input checked="" type="checkbox"/> Buddy-Private-GW1 | rtb-0ea1b76a4dfac7a46 | subnet-065d7b26d1f8249c3 | - | 아니요 | vpc-08e3486799aa74735 Bu... |

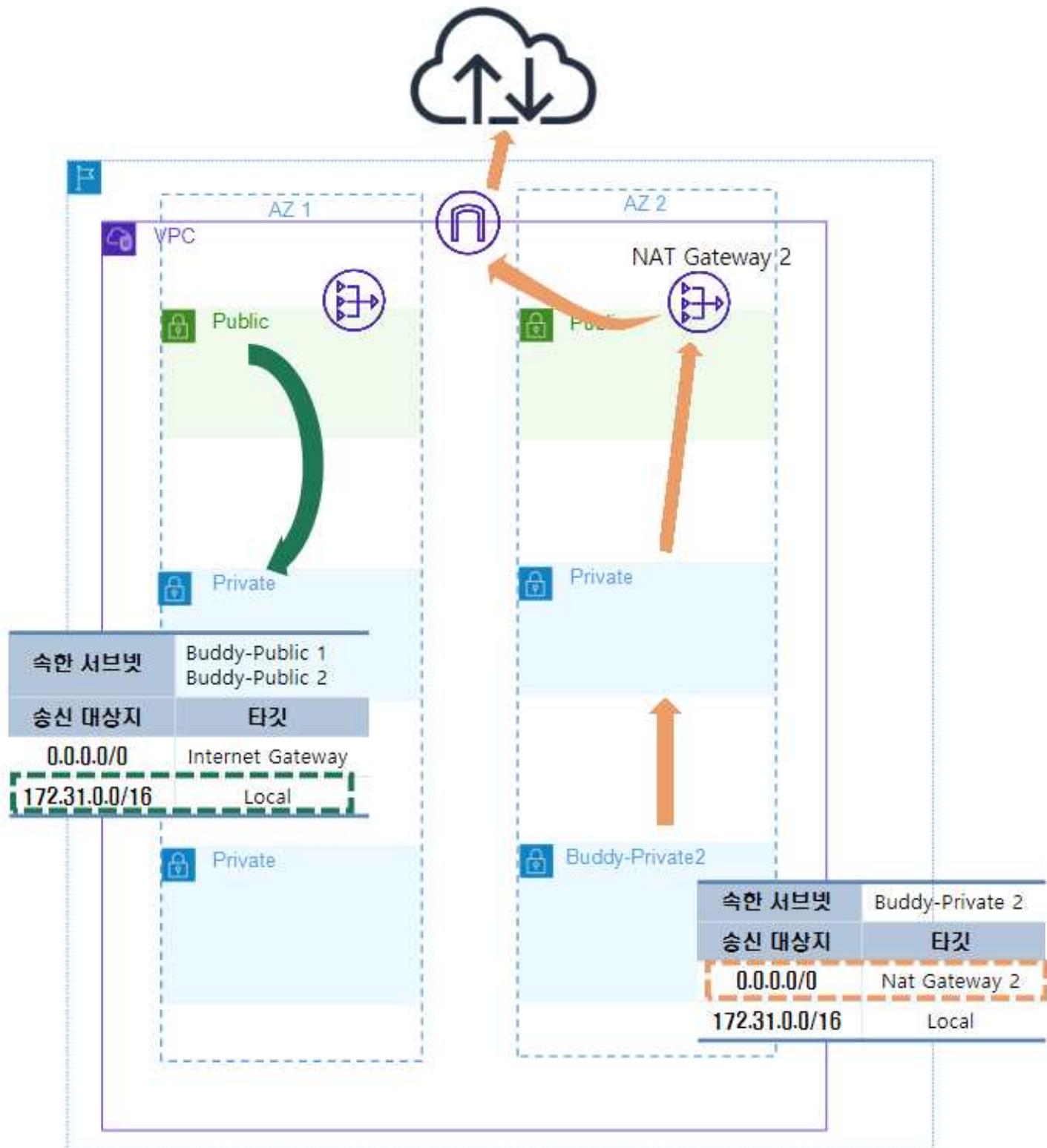
라우팅 테이블 생성 확인

| | |
|---------------|-------------------------------|
| 속한 서브넷 | Buddy-Public1 , Buddy-Public2 |
| 송신 대상지 | 타깃 |
| 0.0.0.0/0 | Internet Gateway |
| 172.31.0.0/16 | Local |

| rtb-0b62985b128b39a03 / Buddy-Routing | | | |
|--|-----------------------|--|-----|
| 세부 정보 | 라우팅 | | |
| 서브넷 연결 | 엣지 연결 | | |
| 라우팅 전파 | 태그 | | |
| 라우팅 (2) | | | |
| <input type="text" value="라우팅 필터링"/> 모두 ▾ | | | |
| 대상 | 대상 | 상태 | 전파됨 |
| 0.0.0.0/0 | igw-0e437e5ca59a2a24a | <input checked="" type="checkbox"/> 활성 | 아니요 |
| 172.31.0.0/16 | local | <input checked="" type="checkbox"/> 활성 | 아니요 |

[생성 리소스 요약]

Internet Gateway 와 Nat Gateway를 생성하여 리소스가 인터넷과 통신할 수 있는 출입구를 생성하였다. 이 후 라우팅 테이블을 생성하여 서브넷과 서브넷, 서브넷과 각 게이트웨이가 통신할 수 있는 경로를 설정하였다.



1. 인프라 | 보안그룹 및 Bastion Host 생성

◆ Private Subnet에 WEB 인스턴스 추가

Name and tags 정보

이름: **Buddy-WEB1**

Add additional tags

소프트웨어 이미지(AMI)

Canonical, Ubuntu, 22.04 LTS, ... 더 보기
ami-04cebc8d6c4f297a3

가상 서버 유형(인스턴스 유형)

t2.micro

방화벽(보안 그룹)

새 보안 그룹

스토리지(볼륨)

1개의 볼륨 – 8GiB

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

◆ Bastion Host를 통해서 들어오는 트래픽만 허용하는 보안그룹 설정

보안 그룹 (1/4) 정보

보안 그룹 필터링

| Name | 보안 그룹 ID | 보안 그룹 이름 | VPC ID | 설명 |
|---------------------------------------|----------------------|-----------|-----------------------|-----------|
| <input checked="" type="checkbox"/> - | sg-03abbfe06b1f7a10f | Buddy-WEB | vpc-08e3486799aa74735 | Buddy-WEB |

웹 페이지 요청 접근 허가를 위하여 포트 번호 80번과 443번 허용, Bastion Host에서 SSH를 이용해 접근 할 수 있도록 22번 포트를 허용하였다. 통신 테스트를 위하여 모든 ICMP 요청은 허가하였다.

인바운드 규칙 (4)

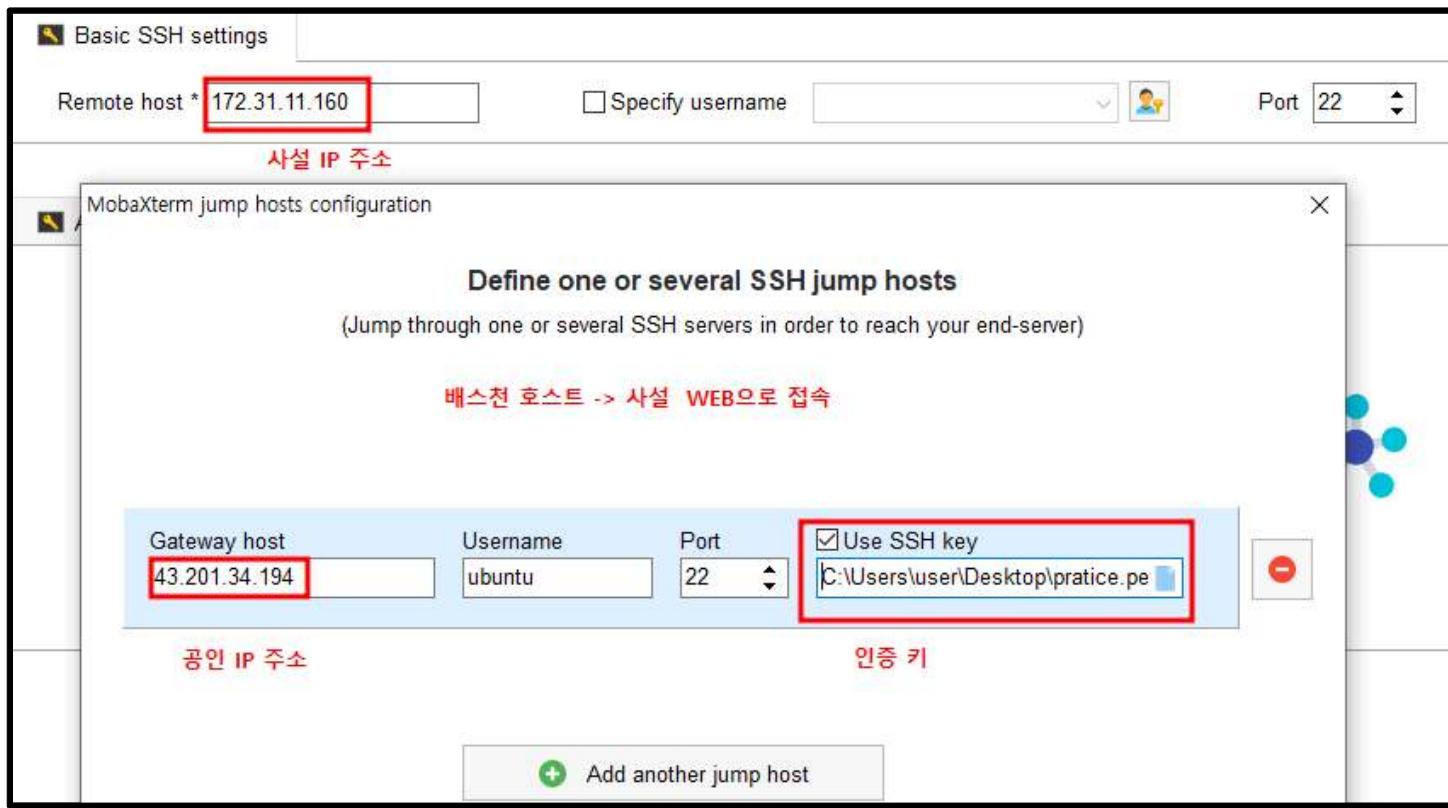
보안 그룹 규칙 필터

| Name | 보안 그룹 규칙 ID | IP 버전 | 유형 | 프로토콜 | 포트 범위 |
|------|------------------------|-------|----------------|------|-------|
| - | sgr-04ecb44820a5a78... | - | HTTPS | TCP | 443 |
| - | sgr-013a0770c314dc2... | - | HTTP | TCP | 80 |
| - | sgr-0524f1e2c4c473991 | - | SSH | TCP | 22 |
| - | sgr-08b09d1012ded9f8c | - | 모든 ICMP - IPv4 | ICMP | 전체 |

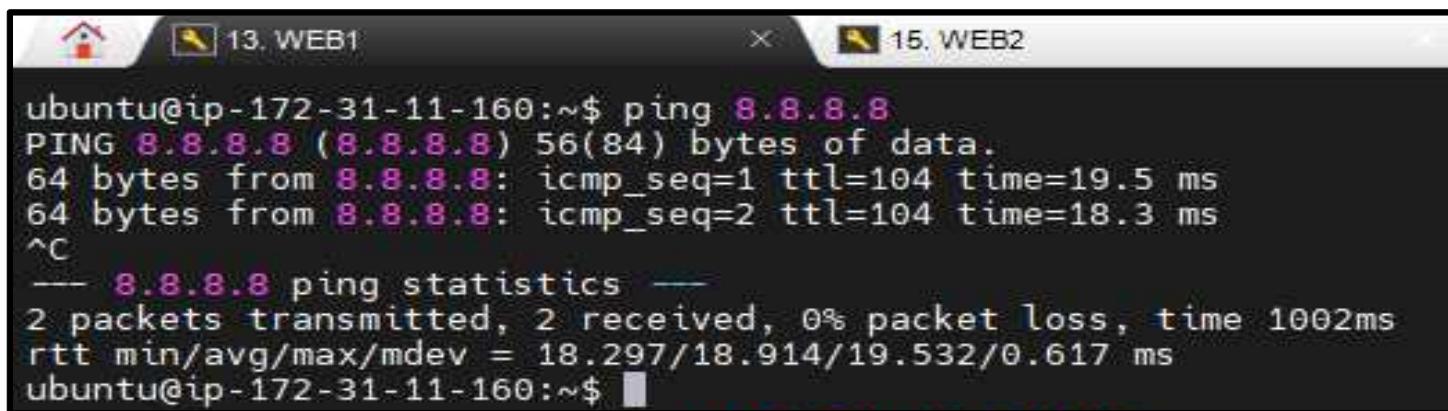
◆ Bastion Host 테스트

| AZ | Subnet | CIDR |
|-----------------|--------------------|----------------|
| ap-northeast-2a | Buddy-Public1 | 172.31.10.0/24 |
| | Buddy-Web-Private1 | 172.31.11.0/24 |
| | Buddy-DB-Private1 | 172.31.12.0/24 |

Bastion Host를 통해서 연결되어 있는 Private Subnet으로 SSH 접속

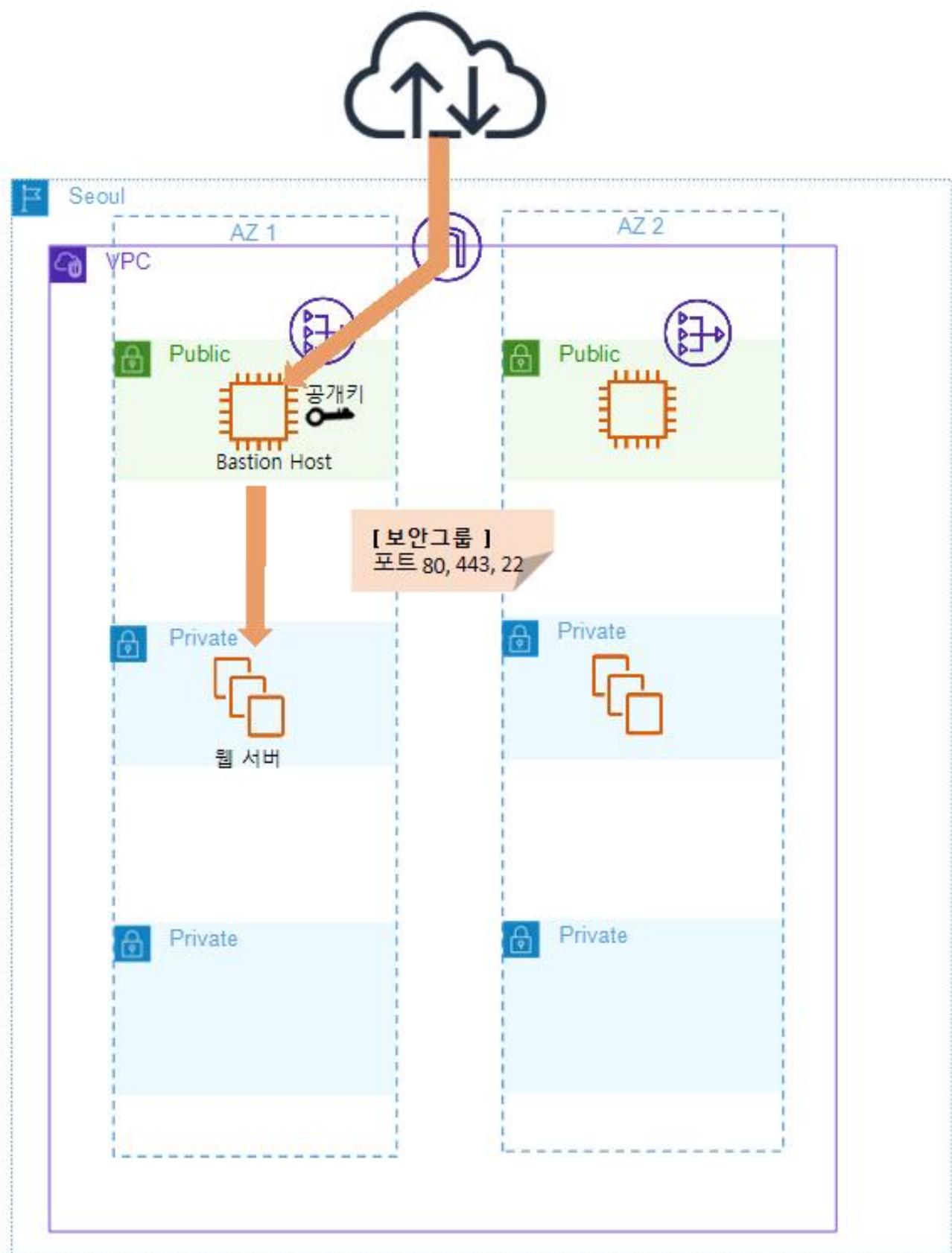


Bastion Host의 NAT Instance 를 통하여 Private Subnet에서 외부 통신 가능 여부 확인



[생성 리소스 요약]

보안그룹을 생성하여 외부로부터 접근을 제한하여 허용된 포트에서만 접근이 가능하도록 설정하였다. 퍼블릭 서브넷에는 Bastion Host를 생성하여 외부에서 해당 서버를 경유하여 리소스에 접근할 수 있도록 하였다.



1. 인프라 | Elastic Load Balancer

| 항목 | 값 |
|-------|--------------------------------------|
| 이름 | Buddy-LoadBalance |
| VPC | Buddy-Project |
| 가용 영역 | Buddy-Public1 Buddy-Public2 |
| 보안 그룹 | LoadBalance-Security |
| | 이름 Buddy-LoadBalancer-TargetGroup |
| | 프로토콜 HTTP |
| 대상 그룹 | 포트 80 |
| | 등록 완료 타깃 Buddy-Web1 Buddy-Web2 |

◆ ELB를 적용할 보안 그룹 생성 및 인바운드 규칙 편집

보안 그룹 생성 [정보](#)

보안 그룹은 인바운드 및 아웃바운드 트래픽을 관리하는 인스턴스의 가상 방화벽 역할을 합니다. 새 보안 그룹을 생성하려면 아래의 필드를 작성하십시오.

기본 세부 정보

보안 그룹 이름 [정보](#)
 생성 후에는 이름을 관리할 수 없습니다.

설명 [정보](#)

VPC [정보](#)
 X

HTTP, HTTPS 인바운드 규칙 설정

인바운드 규칙 정보

| 유형 | 정보 | 프로토콜 | 정보 | 포트 범위 | 정보 | 소스 | 정보 | 설명 | 선택 사항 | 정보 |
|-------|----|------|----|-------|----|-------------|----|-----------|-------|-------------|
| HTTP | | TCP | | 80 | | Anywhere... | | 0.0.0.0/0 | X | Buddy-HTTP |
| HTTPS | | TCP | | 443 | | Anywhere... | | 0.0.0.0/0 | X | Buddy-HTTPS |

ELB 보안그룹 생성 확인

보안 그룹 (5) 정보

| Name | 보안 그룹 ID | 보안 그룹 이름 | VPC ID |
|----------------------|-----------------------|----------------------|-----------------------|
| LoadBalance-Security | sg-06bbbbeeb5bf48e45b | LoadBalance-Security | vpc-08e3486799aa74735 |
| BastionHost | sg-0ac22c75dff4ec376 | BastionHost | vpc-08e3486799aa74735 |

◆ Target Group 생성

Create target group 선택

Target groups Info

No target groups

You don't have any target groups in ap-northeast-2

Create target group

Target type 선택

Target group name
Buddy-LoadBalance-TargetGroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port
HTTP : 80

VPC
Select the VPC with the instances that you want to include in the target group.
Buddy-Project
vpc-08e3486799aa74735
IPv4: 172.31.0.0/16

Target group 지정 대상 선택 후 “Including as pending below” 클릭

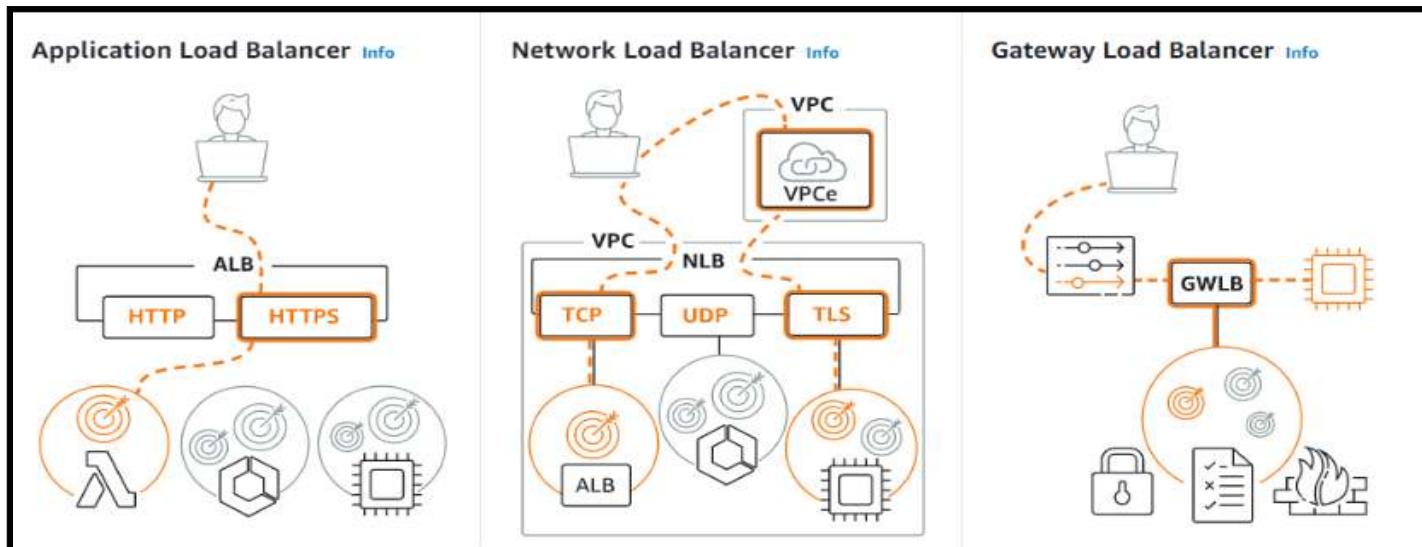
| | | | | | | |
|-------------------------------------|---------------------|------------|----------------------|-------------|-----------------|--------------------------|
| <input checked="" type="checkbox"/> | i-0d6f1e78a4c6a9ae3 | Buddy-WEB2 | running | Buddy-WEB | ap-northeast-2c | subnet-06aba022a87117a3b |
| <input type="checkbox"/> | i-001b13f1a268dae8a | Buddy-NAT1 | running | NAT-GateWay | ap-northeast-2a | subnet-0f68b5444b86408b0 |
| <input checked="" type="checkbox"/> | i-0c630b03a33edbe9e | Buddy-WEB1 | running | Buddy-WEB | ap-northeast-2a | subnet-065d7b26d1f8249c5 |

생성된 Target Groups 확인

| Target groups (1/1) Info | | | | | | |
|--|-------------------------------|---|------|----------|-------------|-------------------|
| <input checked="" type="checkbox"/> | Name | ARN | Port | Protocol | Target type | Load balancer |
| <input checked="" type="checkbox"/> | Buddy-LoadBalance-TargetGroup | arn:aws:elasticloadbalanci... | 80 | HTTP | Instance | Buddy-LoadBalance |

◆ Elastic Load Balancer 생성

3가지 종류의 로드밸런스 선택이 가능하고, 프로젝트에서는 웹 서버 인스턴스를 대상으로 로드밸런싱을 하기 때문에 HTTP/HTTPS 통신에 특화한 Application Load Balancer를 선택한다.



Load Balancer의 이름을 지정한 후 Scheme와 IPv4는 기본값 유지

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

Buddy-LoadBalance

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme cannot be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

Round robin 알고리즘 방식을 사용 선택 및 ELB에서 생성되는 쿠키인 Load Balancer Generated Cookie를 사용하여 스티키 세션 유지

Load balancing algorithm

Determines how the load balancer selects targets from this target group when routing requests.

Round robin

Least outstanding requests

Cannot be combined with the Slow start duration attribute.

Target selection configuration

Stickiness Info

Stickiness allows the load balancer to bind a user's session to a specific target. The stickiness type differs based on the type of cookie used. You can use different stickiness types across your target groups, or even turn off stickiness. Stickiness cannot be turned on if cross-zone load balancing is turned off.

Stickiness type

Load balancer generated cookie

Application-based cookie

Network mapping 카테고리에서 VPC 와 가용 영역을 설정하여 VPC 와 로드밸런서를 연결함

| | | |
|---|---------------|---|
| <input checked="" type="checkbox"/> ap-northeast-2a (apne2-az1) | | |
| Subnet | | |
| subnet-0f68b5444b86408b0 | Buddy-Public1 | ▼ |
| IPv4 settings | | |
| Assigned by AWS | | |
| <hr/> | | |
| <input checked="" type="checkbox"/> ap-northeast-2c (apne2-az3) | | |
| Subnet | | |
| subnet-0f3ce790bba29d4ad | Buddy-Public2 | ▼ |
| IPv4 settings | | |
| Assigned by AWS | | |

- ◆ ELB를 통한 부하분산 & Sticky Session 동작 검증

- ELB (Elastic Load Balancer)를 사용하여 EC2 인스턴스 2개의 웹 서버를 연결하였고, ELB 가 지속적인 상태 확인을 하고 있다 – 200 OK
 - access.log 로그 확인

(왼쪽화면 WEB1 / 오른쪽화면 WEB2)

```
ubuntu@ip-172-31-11-244:~$ sudo tail -f /var/log/apache2/access.log
172.31.10.234 - - [02/Apr/2023:03:26:12 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - - [02/Apr/2023:03:26:13 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - - [02/Apr/2023:03:26:14 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - - [02/Apr/2023:03:26:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - - [02/Apr/2023:03:27:13 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - - [02/Apr/2023:03:27:14 +0000] "GET / HTTP/1.1" 200 5128 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - - [02/Apr/2023:03:27:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - - [02/Apr/2023:03:27:44 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - - [02/Apr/2023:03:28:13 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - - [02/Apr/2023:03:28:14 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
```

```
ubuntu@172-31-21-10: ~$ sudo tail -f /var/log/apache2/access.log
172.31.20.74 - [02/Apr/2023:03:26:14 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:26:17 +0000] "GET / HTTP/1.1" 400 6319 "-" "Xpanse, a Palo Alto Networks Application Network Security Platform"
172.31.20.74 - [02/Apr/2023:03:27:14 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:18 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:24 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:44 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:45 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:28:14 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:28:18 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
```

웹 페이지 요청. 이 때, Stiky Session 동작 검증을 위해 웹페이지를 계속해서 새로고침



WEB2 서버가 요청에 응답하여 웹페이지를 출력하고, 이 후 Sticky Session 동작으로 WEB2 서버에서 세션이 유지됨

```
ubuntu@ip-172-31-11-242:~$ sudo tail -f /var/log/apache2/access.log
172.31.10.234 - [02/Apr/2023:03:26:12 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:26:13 +0000] "GET / HTTP/1.1" 200 5126 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:26:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:03 +0000] "GET / HTTP/1.1" 200 5126 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:27:12 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:13 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:27:43 +0000] "GET / HTTP/1.1" 200 5126 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:27:44 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:27:45 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:28:13 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:28:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:28:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:29:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:29:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.10.234 - [02/Apr/2023:03:29:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
172.31.20.74 - [02/Apr/2023:03:29:43 +0000] "GET / HTTP/1.1" 200 5163 "-" "ELB-HealthChecker/2.0"
```

17_0.0 Safari/[57.36 Edg/[11.0.1601.54]
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/music1.png HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/music2.png HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/music3.png HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/music4.png HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/music5.png HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/images/count_Back.gif HTTP/1.1" 304 195 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/bootstrap/s/bootstrap.bundle.min.js HTTP/1.1" 200 357 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/smoothscroll/smooth-scroll.js HTTP/1.1" 200 355 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/ytplayer/index.js HTTP/1.1" 200 6517 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/dropdown.js HTTP/1.1" 200 194 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"
17_0.0.20 Safari/[57.36 Edg/[11.0.1601.54] - { [02/Apr/2023:03:29:25 +0000] "GET /static/assets/webia_min.js HTTP/1.1" 200 6196 "http://buddyverse.link/" }
"Mozilla/.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5580.106 Safari/[57.36 Edg/[11.0.1601.54]"

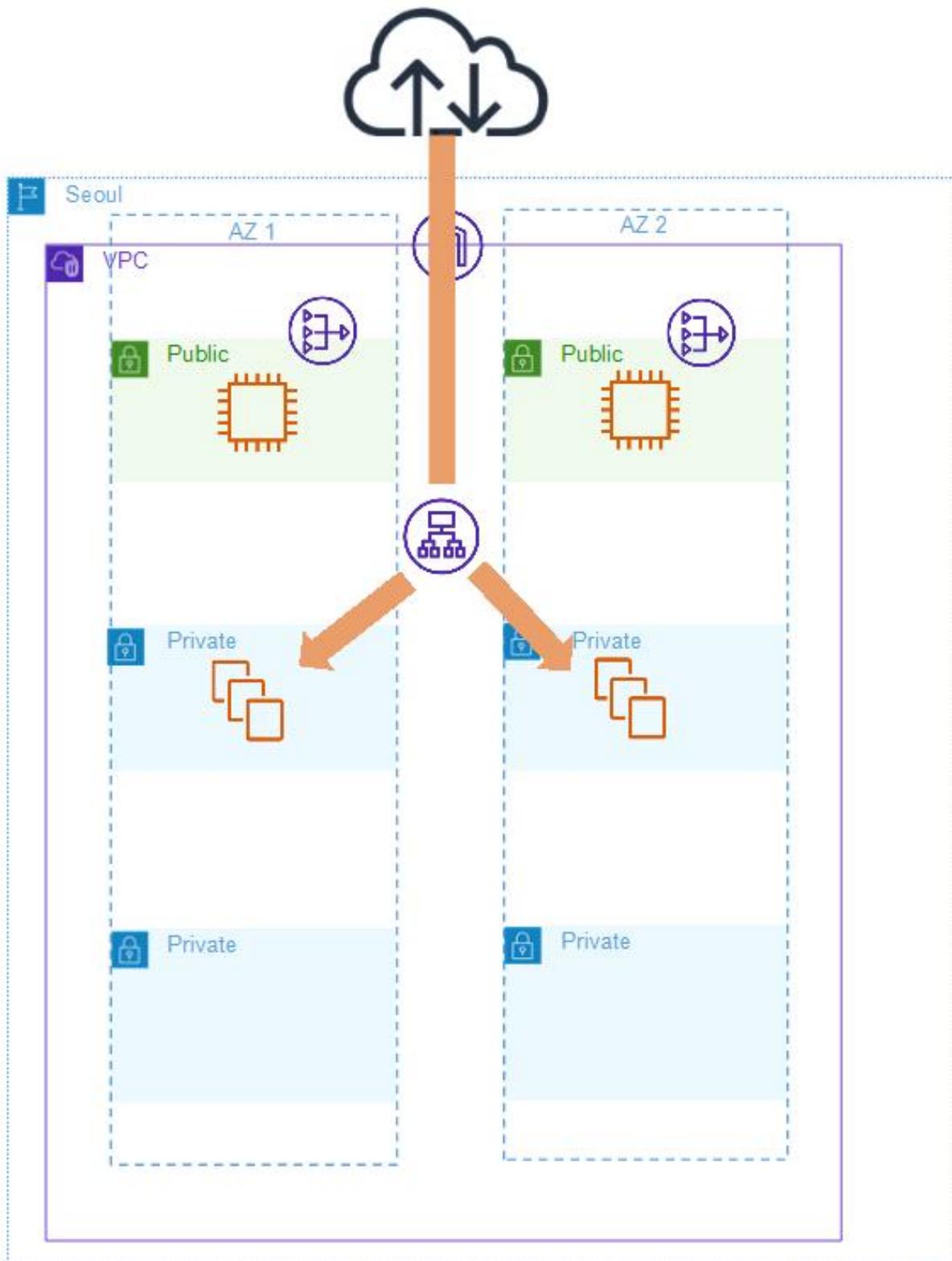
새로 웹브라우저를 시작하여 웹 페이지 요청

The image shows the homepage of Buddyverse. At the top, there's a navigation bar with links like 'CHALLENGE', 'CLUB', 'SOCIALING', 'BOARD', 'REVIEW', and 'LOGOUT'. Below the bar is a large banner featuring three cartoon characters: a man in a blue suit with his fist raised, a man with glasses pointing upwards, and a woman in a purple shirt. They are set against a background of a clear sky with several seagulls. Speech bubbles around them say 'WOW' in black and green. The overall theme is a mix of social interaction and achievement.

Round Robin 알고리즘 방식을 적용하여 이번에는 WEB1 서버에서 요청에 응답하는 것을 확인

[생성 리소스 요약]

외부에서 웹 서버 인스턴스로 접근 시 ELB가 TargetGroup으로 등록한 Web1과 Web2로 요청을 분산시킨다.



1. 인프라 | Auto Scaling

◆ 이미지 생성

| 인스턴스를 속성 또는 (case-sensitive) 태그로 찾기 | | | | | | | | | | 인스턴스 종류 |
|-------------------------------------|---------------------|---------|----------|---------------|-------|-------------------|---------------------------|--|--|---------|
| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 | 가용 영역 | | | | |
| Bastion2 | i-015173cc5cfe38523 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2c | | | | |
| Buddy-NAT2 | i-057e5abe45e8e94cc | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2c | | | | |
| Buddy-WEB2 | i-0d6f1e78a4c6a9ae3 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2c | | | | |
| Buddy-NAT1 | i-001b13f1a268dae8a | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | ec2-52-79-59-1.ap.nort... | | | |
| Buddy-WEB1 | i-0c630b03a33edbe9e | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | - | | | |
| Bastion1 | i-0d988c79c9a161330 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | ec2-43-201-34-194.ap.... | | | |

EC2 인스턴스에 설정을 완료한 후, 인스턴스 중지

| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 | 가용 영역 | 피블릭 IPv4 DNS | 제어 콘솔 보기 |
|-------------------|---------------------|---------|----------|---------------|-------|-------------------|--------------------------|--------------|
| Bastion2 | i-015173cc5cfe38523 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2c | ec2-3-39-187-93.ap-no... | 인스턴스 상태 관리 |
| Buddy-NAT2 | i-057e5abe45e8e94cc | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2c | ec2-15-164-67-241.ap... | 인스턴스 설정 |
| Buddy-WEB2 | i-0d6f1e78a4c6a9ae3 | 중지됨 | t2.micro | - | 경보 없음 | + ap-northeast-2c | - | 네트워킹 |
| Buddy-NAT1 | i-001b13f1a268dae8a | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | 2.67.20.50.1 | 보안 |
| Buddy-WEB1 | i-0c630b03a33edbe9e | 중지됨 | t2.micro | - | 경보 없음 | + ap-northeast-2a | 이미지 생성 | 이미지 및 템플릿 |
| Bastion1 | i-0d988c79c9a161330 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | 인스턴스에서 템플릿 생성 | 모니터링 및 문제 해결 |

인스턴스가 중지되었음을 확인한 후, 이미지 생성 선택

이미지 생성 정보
이미지(AMI라고도 함)는 EC2 인스턴스를 시작할 때 적용되는 프로그램 및 설정을 정의합니다. 기존 인스턴스의 구성에서 이미지를 생성할 수 있습니다.

인스턴스 ID
 i-0d6f1e78a4c6a9ae3 (Buddy-WEB2)

이미지 이름

최대 127자. 생성 후에는 수정할 수 없습니다.

이미지 설명 - 선택 사항

최대 255자

◆ 시작 구성 생성

생성할 시작 구성의 이름, 이미지, 보안그룹, 키페어를 각각 설정

시작 구성 생성 정보

⚠️ 시작 구성을 사용하여 EC2 Auto Scaling 그룹을 생성하는 대신, 시작 템플릿을 사용하고 Auto Scaling 지침 옵션을 사용하는 것이 좋습니다. 시작 구성 마이그레이션 및 시작 템플릿 사용에 대한 자세한 내용은 [설명서를 참조하세요](#)

시작 구성 이름

이름

Amazon Machine Image(AMI) 정보

AMI

인스턴스 유형 정보

인스턴스 유형 인스턴스 유형 선택

| 보안 그룹 ID | 이름 | VPC ID | 설명 |
|--|----------------------|-----------------------|----------------------------|
| <input type="checkbox"/> sg-06bbbeeb5bf48e45b | LoadBalance-Security | vpc-08e3486799aa74735 | LoadBalance-Security |
| <input type="checkbox"/> sg-0ac22c75dff4ec376 | BastionHost | vpc-08e3486799aa74735 | BastionHost |
| <input type="checkbox"/> sg-01a0ba32372207ad6 | default | vpc-08e3486799aa74735 | default VPC security group |
| <input checked="" type="checkbox"/> sg-03abbfe06b1f7a10f | Buddy-WEB | vpc-08e3486799aa74735 | Buddy-WEB |
| <input type="checkbox"/> sg-061ff277093a61525 | NAT-GateWay | vpc-08e3486799aa74735 | NAT-GateWay |

키 페어(로그인) 정보

키 페어 옵션

기존 키 페어

생성된 시작 구성 확인

| 시작 구성 (1) 정보 | | | | | |
|--------------------------|-----------------|--------------------|----------|-------|--|
| | 이름 | AMI ID | 인스턴스 유형 | 스팟 가격 | |
| <input type="checkbox"/> | Buddy-WEB-Im... | ami-057aca1f506... | t2.micro | - | |

◆ Auto Scaling 구성

Auto Scaling의 VPC 및 서브넷 지정

네트워크 Info

대부분의 애플리케이션에서는 여러 가용 영역을 사용할 수 있으며 EC2 Auto Scaling이 여러 영역 간에 인스턴스를 균일하게 분산할 수 있습니다. 기본 VPC와 기본 서브넷은 빠르게 시작하는 데 적합합니다.

VPC
Auto Scaling 그룹의 가상 네트워크를 정의하는 VPC를 선택합니다.

vpc-08e3486799aa74735 (Buddy-Project)
172.31.0.0/16 Default

C

[VPC 생성](#)

가용 영역 및 서브넷
선택한 VPC에서 Auto Scaling 그룹이 사용할 수 있는 가용 영역과 서브넷을 정의합니다.

가용 영역 및 서브넷 선택

C

ap-northeast-2a | subnet-065d7b26d1f8249c3 X
 (Buddy-Web-Private1)
 172.31.11.0/24

ap-northeast-2c | subnet-06aba022a87117a3b X
 (Buddy-Web-Private2)
 172.31.21.0/24

기존 Load Balancer에 만들어 둔 대상 그룹을 선택

로드 밸런싱 Info

아래 옵션을 사용하여 Auto Scaling 그룹을 기존 로드 밸런서 또는 사용자가 정의한 새 로드 밸런서에 연결합니다.

- 로드 밸런서 없음**
Auto Scaling 그룹에 대한 트래픽은 로드 밸런서가 앞에 있지 않습니다.
- 기존 로드 밸런서에 연결**
기존 트로드 밸런서 중에서 선택합니다.
- 새 로드 밸런서에 연결**
Auto Scaling 그룹에 연결할 기본 로드 밸런서를 빠르게 생성합니다.

기존 로드 밸런서에 연결

Auto Scaling 그룹에 연결할 로드 밸런서를 선택합니다.

로드 밸런서 대상 그룹에서 선택
이 옵션을 사용하면 Application Load Balancer, Network Load Balancer 또는 Gateway Load Balancer를 연결할 수 있습니다.

Classic Load Balancer에서 선택

기존 로드 밸런서 대상 그룹
Auto Scaling 그룹과 동일한 VPC에 속하는 인스턴스 대상 그룹만 선택할 수 있습니다.

대상 그룹 선택 ▼ 

Buddy-LoadBalance-TargetGroup | HTTP X
Application Load Balancer: Buddy-LoadBalance

필요한 용량에 맞춰서 그룹 크기 설정

그룹 크기 - 선택 사항 Info

원하는 용량을 변경하여 Auto Scaling 그룹의 크기를 지정합니다. 최소 및 최대 용량 한도를 지정할 수도 있습니다. 원하는 용량은 한도 범위 내에 있어야 합니다.

| | |
|--------|---|
| 원하는 용량 | 2 |
| 최소 용량 | 2 |
| 최대 용량 | 4 |

크기 조정 정책 선택

크기 조정 정책 - 선택 사항

크기 조정 정책을 사용하여 수요의 변화를 감지하도록 Auto Scaling 그룹의 크기를 동적으로 조정할지 여부를 선택합니다. [Info](#)

대상 추적 크기 조정 정책

원하는 결과를 선택하고 크기 조정 정책에 따라 필요한 경우 용량을 추가 및 제거하여 해당 결과를 달성합니다.

없음

생성된 Auto Scaling 그룹을 확인

Auto Scaling 그룹 (1) [Info](#)

| Auto Scaling 그룹 검색 | | | | | | |
|--|-----------------------|------|---------------------------|--------|--------|------|
| 이름 | 시작 템플릿/구성 | 인스턴스 | 상태 | 원하는 용량 | 최... ▾ | 최대 ▾ |
| <input type="checkbox"/> Buddy-AutoScaling | Buddy-WEB-Image-Start | 0 | ● 웹 앱 업데이트 중 | 2 | 2 | 4 |

ELB를 통한 상태 확인 활성화

* Auto Scaling은 자체적으로 서비스 상태를 확인할 수 없기 때문에 인스턴스가 실행되고 있다면 이상을 감지할 수 없다. 따라서 상태 확인 유형을 ELB로 설정하여 상태 확인에 실패한 인스턴스를 자동으로 교체한다.

상태 확인 - 선택 사항

상태 확인 유형 [Info](#)

EC2 Auto Scaling은 상태 확인에 실패한 인스턴스를 자동으로 교체합니다. 로드 밸런서를 활성화한 경우 항상 활성화된 EC2 상태 확인 외에도 ELB 상태 확인을 활성화할 수 있습니다.

EC2 ELB

◆ Auto Scaling 동작 검증

Auto Scaling 동작 확인을 위하여 CloudWatch 대시보드 생성



1) 확장성 테스트 – CPU 스트레스 부하로 Autoscaling 동작 확인

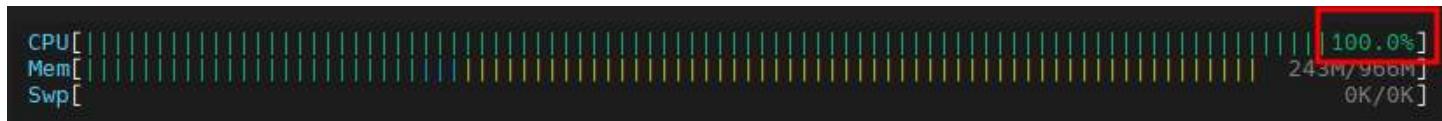
부하 전 >> EC2 인스턴스 개수 2개 확인

| Name | ▲ 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|------------|---------------------|---------|----------|------------|-------|
| - | i-0881d23e330cb8750 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| - | i-0edf18b175387d4a4 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Bastion1 | i-0d988c79c9a161330 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Bastion2 | i-015173cc5cfe38523 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Buddy-NAT1 | i-001b13f1a268dae8a | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Buddy-NAT2 | i-057e5abe45e8e94cc | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |

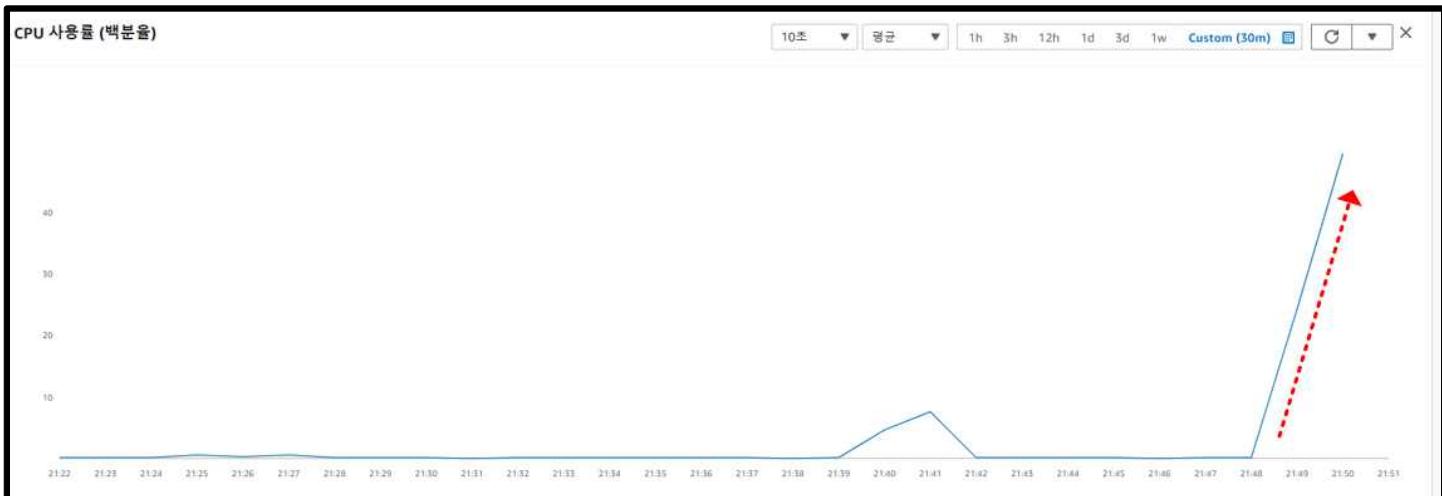
부하검증을 위한 패키지 다운로드*stress-ng : 다양한 리소스 (CPU, 메모리, 디스크 등)에 대한 부하 테스트를 수행하는 유틸리티 프로그램CPU 스트레스 테스트를 4 개의 스레드로 900초 동안 실행

```
ubuntu@ip-172-31-21-151:~$ sudo apt-get install stress-ng -y
ubuntu@ip-172-31-21-151:~$ stress-ng --cpu 4 --timeout 900s
stress-ng: info: [1293] setting to a 900 second (5 mins, 0.00 secs) run per stressor
stress-ng: info: [1293] dispatching hogs: 4 cpu
```

EC2 인스턴스에 부하를 가하고 CPU의 상태를 확인



CloudWatch 대시보드를 통하여 EC2 Instance의 부하가 증가한 것을 시각적 지표로 확인



부하 후 >> Auto Scaling에 의하여 EC2 Instance 의 개수가 4개로 증가한 것 확인

| Auto Scaling 그룹 (1) Info | | | | | | | | | |
|--|-------------------------|----------------------------|------|----|---------|----|----|----|----|
| <input type="text"/> Auto Scaling 그룹 검색 | | | | | | | | | |
| | 이름 | 시작 템플릿/구성 | 인스턴스 | 상태 | 원하는 ... | 최소 | 최대 | 최대 | 최소 |
| | Buddy-AutoScaling-Group | Buddy-WEB-Image-Start(EFS) | 4 | - | 4 | 2 | 4 | 4 | 2 |

| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|------------|-----------------------|---------|----------|------------|-------|
| - | 1 i-007b4bff1dc6450e3 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| - | 2 i-0881d23e330cb8750 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| - | 3 i-019ea01a6180e1327 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| - | 4 i-0edf18b175387d4a4 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Bastion1 | i-0d988c79c9a161330 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Bastion2 | i-015173cc5cf38523 | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Buddy-NAT1 | i-001b13f1a268dae8a | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |
| Buddy-NAT2 | i-057e5abe45e8e94cc | 실행 중 | t2.micro | 2/2개 검사 통과 | 경보 없음 |

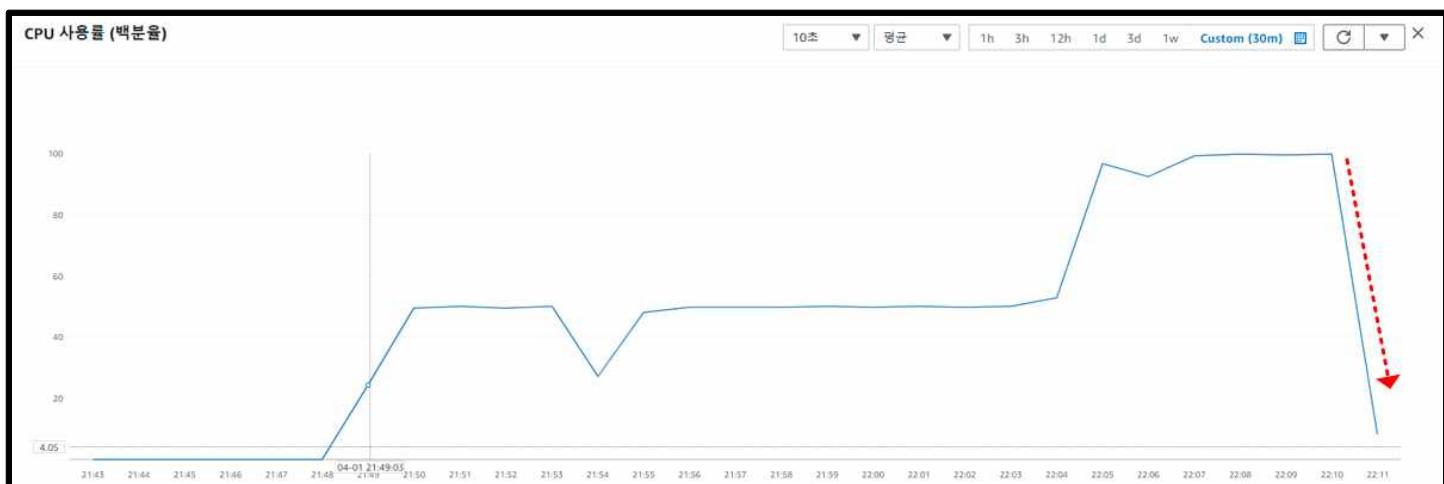
부하 분산 종료

^Cstress-ng: info: [1399] successful run completed in 249.76s (4 mins, 9.76 secs)

EC2 인스턴스의 부하가 감소한 후 Instance의 개수가 감소한 것을 확인



CloudWatch 대시보드를 통하여 EC2 Instance의 부하가 감소한 것 확인



CPU 부하도가 감소 수, 인스턴스의 개수 최소 개수인 2개로 변동된 것을 확인

| | Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|---|----------|---------------------|---------|----------|--------------|-------|
| □ | - | i-007b4bff1dc6450e3 | ⓧ 종료됨 | t2.micro | - | 경보 없음 |
| □ | - | i-0e9ac8063bbc812ac | ⓧ 실행 중 | t2.micro | ⓧ 2/2개 검사 통과 | 경보 없음 |
| □ | - | i-0881d23e330cb8750 | ⓧ 종료됨 | t2.micro | - | 경보 없음 |
| □ | - | i-019ea01a6180e1327 | ⓧ 종료됨 | t2.micro | - | 경보 없음 |
| □ | - | i-0edf18b175387d4a4 | ⓧ 종료됨 | t2.micro | - | 경보 없음 |
| □ | - | i-009f60e1fa39249f1 | ⓧ 실행 중 | t2.micro | ⓧ 2/2개 검사 통과 | 경보 없음 |
| □ | Bastion1 | i-0d988c79c9a161330 | ⓧ 실행 중 | t2.micro | ⓧ 2/2개 검사 통과 | 경보 없음 |
| □ | Bastion2 | i-015173cc5cf38523 | ⓧ 실행 중 | t2.micro | ⓧ 2/2개 검사 통과 | 경보 없음 |

2-1) 안정성 테스트 – 인스턴스 장애시 Autoscaling 동작 확인

EC2 인스턴스를 강제 종료

```
ubuntu@ip-172-31-11-121:~$ init 0
```

강제 종료한 인스턴스가 중지되고, 새로운 인스턴스가 생성되는 과정

| | Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|---|------------|---------------------|---------|----------|-----------------|-------|
| □ | Bastion2 | i-015173cc5fce38523 | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| □ | Buddy-NAT2 | i-057e5abe45e8e94cc | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| □ | - | i-0e9ac8063bbc812ac | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| □ | Buddy-NAT1 | i-001b13f1a268dae8a | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| □ | Bastion1 | i-0d988c79c9a161330 | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| □ | - | i-009f60e1fa39249f1 | ● 중지 중 | t2.micro | - | 경보 없음 |
| □ | - | i-020f45f7e9b33b2db | ● 대기 중 | t2.micro | - | 경보 없음 |

Autoscaling 이 동작하여 새로운 인스턴스가 실행됨

| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|------------|---------------------|---------|----------|-----------------|-------|
| Bastion2 | i-015173cc5fce38523 | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| Buddy-NAT2 | i-057e5abe45e8e94cc | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| - | i-0e9ac8063bbc812ac | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| Buddy-NAT1 | i-001b13f1a268dae8a | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| Bastion1 | i-0d988c79c9a161330 | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| - | i-020f45f7e9b33b2db | ● 실행 중 | t2.micro | ● 2/2개 검사 통과... | 경보 없음 |
| - | i-009f60e1fa39249f1 | ● 중지됨 | t2.micro | - | 경보 없음 |

2-2) 안정성 테스트 – WEB 어플리케이션 장애시 Autoscaling 동작 확인

```
ubuntu@ip-172-31-11-79:~$ sudo service apache2 stop
```

Apache HTTP 서버가 현재 동작하지 않는 것 확인

```
ubuntu@ip-172-31-11-79:~$ sudo service apache2 status
```

○ apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Active: inactive (dead) since Sat 2023-04-01 17:01:24 UTC; 45s ago

Docs: <https://httpd.apache.org/docs/2.4/>

Process: 541 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)

Process: 1049 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)

Main PID: 607 (code=exited, status=0/SUCCESS)

CPU: 855ms

Apr 01 16:56:10 ip-172-31-11-79 systemd[1]: Starting The Apache HTTP Server...

Apr 01 16:56:10 ip-172-31-11-79 systemd[1]: Started The Apache HTTP Server.

Apr 01 17:01:21 ip-172-31-11-79 systemd[1]: Stopping The Apache HTTP Server...

Apr 01 17:01:24 ip-172-31-11-79 systemd[1]: apache2.service: Deactivated successfully.

Apr 01 17:01:24 ip-172-31-11-79 systemd[1]: **Stopped The Apache HTTP Server.**

ELB에 의한 헬스체크에 의해 인스턴스의 상태가 unhealthy로 변경

| 인스턴스 ID | 수명 주기 | 인스턴스 유형 | 가중치 기반 용량 | 시작 템플릿/구성 | 가용 영역 | 상태 |
|---------------------|-----------|----------|-----------|------------------|-----------------|---|
| i-020f45f7e9b33b2db | InService | t2.micro | - | Buddy-WEB-Img... | ap-northeast-2a | ⚠️ Unhealthy |
| i-0e9ac8063bbc812ac | InService | t2.micro | - | Buddy-WEB-Img... | ap-northeast-2c | Healthy |

EC2 인스턴스 상태는 실행 중이지만 Apache HTTP 서버가 동작하지 않으므로 Autoscaling이 이를 감지하고 새로운 인스턴스를 생성함

| | | | | | | |
|------------|---------------------|---|---------------------------------------|----------|---|-------|
| - | i-0e9ac8063bbc812ac | ✔️ 실행 중 | ✔️ | t2.micro | ✔️ 2/2개 검사 통과... | 경보 없음 |
| Buddy-NAT1 | i-001b13f1a268dae8a | ✔️ 실행 중 | ✔️ | t2.micro | ✔️ 2/2개 검사 통과... | 경보 없음 |
| Bastion1 | i-0d988c79c9a161330 | ✔️ 실행 중 | ✔️ | t2.micro | ✔️ 2/2개 검사 통과... | 경보 없음 |
| - | i-0ad2c90e47244a6d5 | ✔️ 실행 중 | ✔️ | t2.micro | ✔️ 2/2개 검사 통과... | 경보 없음 |
| - | i-020f45f7e9b33b2db | ✖️ 실행 중 | ✖️ | t2.micro | ✔️ 2/2개 검사 통과... | 경보 없음 |

새로운 인스턴스가 생성되어 인스턴스의 상태가 Health로 변경됨

| 인스턴스 ID | 수명 주기 | 인스턴스 유형 | 가중치 기반 용량 | 시작 템플릿/구성 | 가용 영역 | 상태 |
|---------------------|-----------|----------|-----------|-----------------|-----------------|--|
| i-0ad2c90e47244a6d5 | InService | t2.micro | - | Buddy-WEB-Image | ap-northeast-2a | Healthy |
| i-0e9ac8063bbc812ac | InService | t2.micro | - | Buddy-WEB-Image | ap-northeast-2c | Healthy |

종료했던 Apache HTTP 서버를 재시작하고 상태가 active (running) 상태

```
ubuntu@ip-172-31-11-79:~$ sudo service apache2 restart
```

```
ubuntu@ip-172-31-11-79:~$ sudo service apache2 status
```

● apache2.service - The Apache HTTP Server

```
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Sat 2023-04-01 17:07:26 UTC; 3s ago
```

```
Docs: https://httpd.apache.org/docs/2.4/
```

```
Process: 1093 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
```

```
Main PID: 1097 (apache2)
```

```
Tasks: 73 (limit: 1141)
```

```
Memory: 16.1M
```

```
CPU: 92ms
```

```
CGroup: /system.slice/apache2.service
```

```
    └─1097 /usr/sbin/apache2 -k start
    └─1098 /usr/sbin/apache2 -k start
    └─1099 /usr/sbin/apache2 -k start
    └─1100 /usr/sbin/apache2 -k start
```

```
Apr 01 17:07:26 ip-172-31-11-79 systemd[1]: Starting The Apache HTTP Server...
```

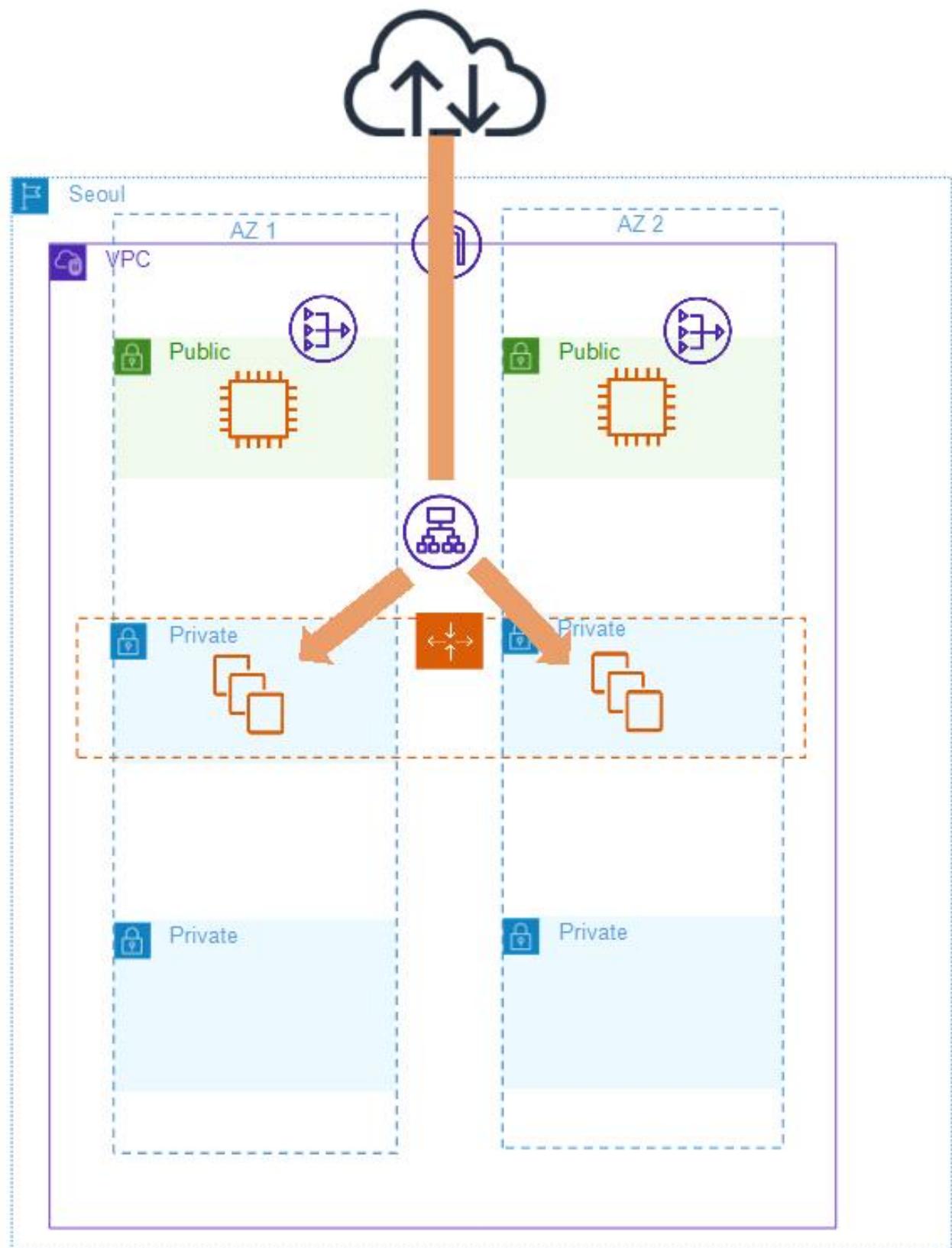
```
Apr 01 17:07:26 ip-172-31-11-79 systemd[1]: Started The Apache HTTP Server.
```

최소 인스턴스 개수에 맞추어 재시작했던 인스턴스가 종료됨

| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 |
|------------|---------------------|---|----------|--|-------|
| Bastion2 | i-015173cc5fce38523 | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| Buddy-NAT2 | i-057e5abe45e8e94cc | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| - | i-0e9ac8063bbc812ac | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| Buddy-NAT1 | i-001b13f1a268dae8a | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| Bastion1 | i-0d988c79c9a161330 | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| - | i-009f60e1fa39249f1 | 종료됨 | t2.micro | - | 경보 없음 |
| - | i-0ad2c90e47244a6d5 | �行 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 |
| - | i-020f45f7e9b33b2db | 종료됨 | t2.micro | - | 경보 없음 |

[생성 리소스 요약]

Auto Scaling Group은 설정된 값을 기반으로 EC2 인스턴스를 자동으로 생성하거나 종료하여 로드 밸런싱을 유지한다. 또한 상태 확인 유형을 ELB로 설정하여 웹 서버 인스턴스의 장애나 문제가 발생한 경우 ELB가 이를 감지하고 해당 인스턴스를 제거하고 새로운 인스턴스를 생성하여 대체한다.



1. 인프라 | AWS EFS

◆ EFS 파일 시스템 생성

EFS 보안그룹 생성한다.

NFS 프로토콜을 선택하고, WEB 서버의 보안그룹을 대상으로 선택

The screenshot shows a CloudFormation template for creating an EFS file system. It includes sections for security group creation (Buddy-EFS), VPC configuration (VPC ID: vpc-08e3486799aa74735), and ingress rules (NFS protocol, port 2049, target security group sg-03abbfe06b1f7a10f).

```

    {
        "Resources": {
            "BuddyEFS": {
                "Type": "AWS::EFS::FileSystem",
                "Properties": {
                    "Tags": [
                        {"Key": "Name", "Value": "Buddy-EFS"}
                    ],
                    "SubnetIds": [
                        "subnet-01234567890abcdef"
                    ],
                    "PerformanceMode": "generalPurpose"
                }
            },
            "BuddyEFSIngress": {
                "Type": "AWS::EFS::IngressRule",
                "Properties": {
                    "FileSystemId": "fs-01234567890abcdef",
                    "Port": 2049,
                    "Protocol": "NFS",
                    "SecurityGroups": [
                        "sg-03abbfe06b1f7a10f"
                    ]
                }
            }
        }
    }
  
```

| 보안 그룹 (1/7) 정보 | | | | |
|-------------------------------------|-----------|----------------------|-------------|-----------------------|
| | 포트 번호 필터링 | 선택 | 설명 | 선택 |
| <input type="checkbox"/> | Buddy-WEB | sg-03abbfe06b1f7a10f | Buddy-WEB | vpc-08e3486799aa74735 |
| <input checked="" type="checkbox"/> | Buddy-EFS | sg-03589fa92687ca2c1 | Buddy-EFS | vpc-08e3486799aa74735 |
| <input type="checkbox"/> | Buddy-NAT | sg-061#277093a61525 | NAT-GateWay | vpc-08e3486799aa74735 |

AWS 콘솔 홈에서 EFS를 검색해 이동 후 파일 시스템 생성 클릭

The screenshot shows the AWS CloudFront console with the 'Amazon Elastic File System' section selected. It displays information about EFS and provides a 'File System Creation' button.

Amazon Elastic File System
조정 가능하고 탄력적인
클라우드 네이티브 NFS
파일 시스템

파일 시스템 생성
서비스 권장 설정으로 EFS 파일 시스템을 생성합니다.
파일 시스템 생성

이름과 VPC를 설정하고 사용자 지정 클릭

파일 시스템 생성

서비스 권장 설정으로 EFS 파일 시스템을 생성합니다. [자세히 알아보기](#)

이름 - 선택 사항
파일 시스템 이름을 지정합니다.

Buddy-EFS

이름에는 문자, 숫자 및+-_=:_/ 기호로 최대 256자까지 포함될 수 있습니다.

Virtual Private Cloud(VPC)
EC2 인스턴스를 파일 시스템에 연결하고자 하는 VPC를 선택합니다. [자세히 알아보기](#)

vpc-08e3486799aa74735

기본값

스토리지 클래스 자세히 알아보기

- Standard
여러 AZ에 걸쳐 데이터를 중복 저장
- One Zone
단일 AZ 내에 데이터를 중복 저장

취소 [사용자 지정](#) **생성**

원하는 세부 설정 체크하고 다음 클릭

처리량 모드
파일 시스템의 처리량 제한에 대한 방법을 선택합니다. [자세히 알아보기](#)

- 버스트
기본적인 성능 요구 사항이 있는 워크로드를 위해 스토리지 양에 따라 확장되는 처리량을 제공합니다.
- 개선됨
다양한 성능 요구 사항이 있는 워크로드에 더 많은 유연성과 더 높은 처리량 수준을 제공합니다.

▼ 주가 설정

성능 모드
필요한 IOPS를 기반으로 파일 시스템의 성능 모드를 설정합니다. [자세히 알아보기](#)

- 범용 (권장)
고성능 및 시연 시간에 민감한 애플리케이션을 비롯하여 다양한 워크로드에 적합
- 최대 I/O
긴 지연 시간을 허용할 수 있는 고도로 병렬화된 워크로드를 위해 설계

마운트 할 EC2 인스턴스가 위치하는 VPC, 가용영역, 서브넷 선택

네트워크

Virtual Private Cloud(VPC)
EC2 인스턴스를 파일 시스템에 연결하고자 하는 VPC를 선택합니다. 자세히 알아보기 [\[링크\]](#)

vpc-08e3486799aa74735
기본값

▼

탑재 대상

탑재 대상은 Amazon EFS 파일 시스템을 탑재할 수 있는 NFSv4 엔드포인트를 제공합니다. 가용 영역마다 탑재 대상을 한 개씩 생성하는 것이 좋습니다. 자세히 알아보기 [\[링크\]](#)

| 가용 영역 | 서브넷 ID | IP 주소 | 보안 그룹 |
|-----------------|--------------------------|-------|---|
| ap-northeast-2a | subnet-065d7b26d1f8249c3 | 자동 | 보안 그룹 선택 sg-03589fa92687ca2c1 X Buddy-EFS |
| ap-northeast-2c | subnet-06aba022a87117a3b | 자동 | 보안 그룹 선택 sg-03589fa92687ca2c1 X Buddy-EFS |

제거

마운트 EFS 파일시스템 생성 완료

| 파일 시스템 (1) | | | | | | | | | 파일 | |
|------------|----------------------|------|----------|-----------------------|-----------------------------|-------------------|-----------|-------------------------------|----------|----|
| 이름 | 파일 시스템 ID | 암호화됨 | 총 크기 | Standard/One Zone의 크기 | Standard-IA/One Zone-IA의 크기 | 프로비저닝된 처리량(MiB/s) | 파일 시스템 상태 | 생성 시간 | 세부 정보 보기 | 삭제 |
| Buddy-EFS | fs-0e922dc34f9e8f2f6 | 암호화됨 | 6.00 KiB | 6.00 KiB | 0바이트 | - | 사용 가능 | Sat, 01 Apr 2023 09:55:32 GMT | | |

EFS 의 세부정보 확인

Buddy-EFS (fs-0e922dc34f9e8f2f6)

일반

| | |
|---------------------------|--|
| 성능 모드 | 자동 백업 |
| 범용 | 활성화됨 |
| 처리량 모드 | 암호화됨 |
| 버스트 | 0461d3c3-3f1e-48fc-bfe0-38a5bb086637 (aws/elasticfilesystem) |
| 수명 주기 관리 | 파일 시스템 상태 |
| IA로 전환: 마지막 액세스 이후 30일 경과 | 사용 가능 |
| IA 외부로 전환: 없음 | DNS 이름 |
| 가용 영역 | fs-0e922dc34f9e8f2f6.efs.ap-northeast-2.amazonaws.com |
| Standard | |

EFS 의 WEB 서버에서 정적 파일을 저장하는 /static 디렉토리와 사용자가 업로드한 미디어 파일을 저장하는 /media 디렉토리를 EFS 에 마운트하고 오토마운트를 설정한다.

```
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo mkdir -p /mnt/efs
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo mkdir /mnt/efs/media
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo mkdir /mnt/efs/static
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo apt-get install -y nfs-common
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,tcp fs-0e922dc34f9e8f2f6.efs.ap-northeast-2.AWSaws.com:/ /mnt/efs
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/root                                 7.6G  2.6G  5.1G  34% /
tmpfs                                    484M   0    484M   0% /dev/shm
tmpfs                                    194M  844K  193M   1% /run
tmpfs                                    5.0M   0    5.0M   0% /run/lock
/dev/xvda15                               105M  6.1M   99M   6% /boot/efi
tmpfs                                    97M  4.0K   97M   1% /run/user/1000
fs-0e922dc34f9e8f2f6.efs.ap-northeast-2.AWSaws.com:/  8.0E   0  8.0E  0% /mnt/efs
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ vi config/settings.py
```

```
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join('/mnt/efs', 'static')
```

```
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join('mnt/efs', 'media')
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo service apache2 status
```

```
ubuntu@ip-172-31-11-23:~/django/buddy$ sudo vi /etc/fstab
```

```
LABEL=cloudimg-rootfs   /          ext4  discard,errors=remount-ro      0 1
LABEL=UEFI      /boot/efi      vfat   umask=0077      0 1
```

```
fs-0e922dc34f9e8f2f6.efs.ap-northeast-2.AWSaws.com:/          /mnt/efs
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0 0
```

EC2 인스턴스를 시작할 때 Amazon EFS 파일 시스템을 자동으로 마운트하도록 시작 템플릿 생성

| 시작 구성 (2) 정보 | | | | | | | | 작업 ▾ | 시작 템플릿에 복사 ▾ | 시작 구성 생성 |
|-------------------------------|----------------------------|---|--------------------|---|----------|---|-------|------|--|----------|
| | | | | | | | < | 1 | > | 닫기 |
| <input type="text"/> 시작 구성 검색 | | | | | | | | | | |
| □ | 이름 | ▼ | AMI ID | ▼ | 인스턴스 유형 | ▼ | 스팟 가격 | ▼ | 생성 시간 | ▼ |
| □ | Buddy-WEB-Image-Start(EFS) | | ami-07a46d89c9f... | | t2.micro | | - | | Sat Apr 01 2023 19:58:01 GMT+0900 (한국 표준시) | |
| □ | Buddy-WEB-Image-Start | | ami-021ffe45b93... | | t2.micro | | - | | Sat Apr 01 2023 17:02:49 GMT+0900 (한국 표준시) | |

시작 구성은 EC2 Auto Scaling 그룹의 런처 구성으로 지정하여, Auto Scaling 그룹에서 새로운 인스턴스를 시작할 때 자동으로 Amazon EFS 파일 시스템이 마운트되도록 한다.

| 시작 구성 | Info | 시작 템플릿으로 전환 |
|---|----------------------------|---|
| ⚠ 시작 구성을 사용하여 EC2 오토 스케일링 그룹을 생성하는 대신, 시작 템플릿을 사용하고 오토 스케일링 지침 옵션을 활용하는 것이 좋습니다. 시작 구성 마이그레이션 및 시작 템플릿 사용에 대한 자세한 내용은 설명서를 참조하세요. [x] | | |
| 시작 구성 | | |
| Amazon Machine Image(AMI), 인스턴스 유형, 키 페어 및 보안 그룹과 같은 인스턴스 수준 설정이 포함된 시작 구성을 선택합니다. | Buddy-WEB-Image-Start(EFS) | ▼ C |
| 시작 구성 생성 [x] | | |
| 시작 구성 | AMI ID | 생성된 날짜 |
| Buddy-WEB-Image-Start(EFS) | ami-07a46d89c9f498b89 | Sat Apr 01 2023 19:58:01 GMT+0900 (한국 표준시) |
| 보안 그룹 | 인스턴스 유형 | 키 페어 이름 |
| sg-03abbfe06b1f7a10f [x] | t2.micro | Buddy-Project |

| Auto Scaling 그룹 (1/1) Info | | C | 편집 | 삭제 | | | |
|--|-------------------------|----------------------------|--------------------|--------------------|------------|----------|--------|
| <input type="text"/> Auto Scaling 그룹 검색 | | | | | | | |
| <input checked="" type="checkbox"/> | 이름 | ▼ 시작 템플릿/구성 | ▼ 인스턴스 | 상태 | ▼ 원하는 용량 | ▼ 최... | ▼ 최대 |
| <input checked="" type="checkbox"/> | Buddy-AutoScaling-Group | Buddy-WEB-Image-Start(EFS) | 4 | 용량 업데이트 중 | 1 | 1 | 2 |

◆ 웹 서버 게시판 첨부파일 게시를 통한 EFS 동작 검증

게시판에 사진 파일을 첨부한 글 게시

The screenshot shows a web application interface. At the top, there is a navigation bar with icons for user profile, buddy, challenge, club, socialing, board, review, and logout. Below the navigation bar, the main content area displays a post titled "쿠버네티스를 공부하자" (Learning Kubernetes) created by "test" on April 2, 2023, at 12:45 PM. A red box highlights the "첨부파일 : Test.png" (Attached file: Test.png) link. Below the title, there is a text area containing the message "고래 귀여워" (Whale is cute).

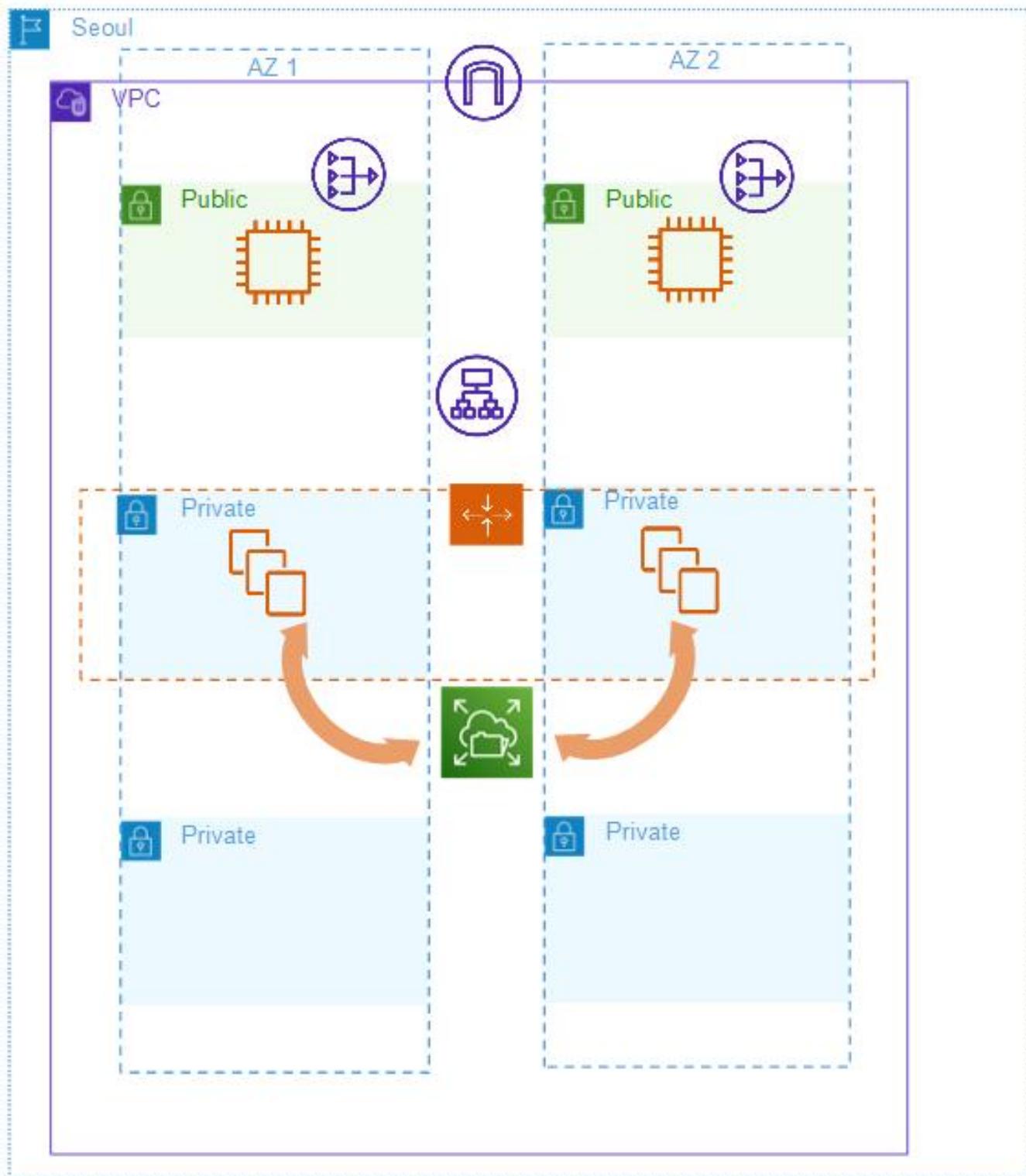
WEB1 인스턴스와 WEB2 인스턴스가 같은 EFS 파일 시스템을 공유하고 있어 방금 첨부한 사진에 동시에 접근 가능

```
ubuntu@ip-172-31-11-121:~/django/buddy$ ls -l /mnt/efs/media/review/data/2023/04/02/Test.png
-rw-r--r-- 1 www-data www-data 7568 Apr  1 15:45 /mnt/efs/media/review/data/2023/04/02/Test.png

ubuntu@ip-172-31-21-18:~$ ls -l /mnt/efs/media/review/data/2023/04/02/Test.png
-rw-r--r-- 1 www-data www-data 7568 Apr  1 15:45 /mnt/efs/media/review/data/2023/04/02/Test.png
-----+-----+
2 rows in set (0.00 sec)
```

[생성 리소스 요약]

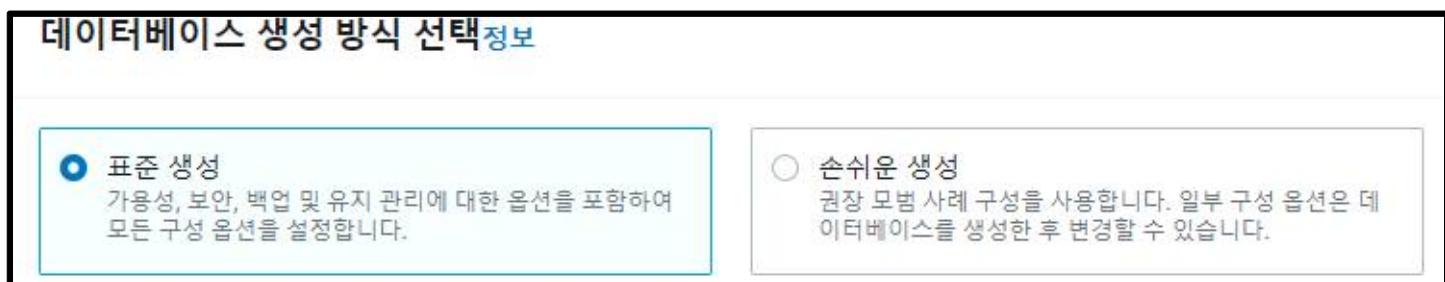
AWS EFS를 생성하여 웹 서버의 /static 디렉토리와 /media 디렉토리를 마운트하여 WEB1 서버와 WEB2 서버에서 EFS에 저장된 파일 시스템에 동시에 접근이 가능하다.



1. 인프라 | AWS RDS

◆ 데이터베이스 생성

AWS 대시보드에서 RDS 검색 후 데이터베이스 생성 클릭



생성할 데이터베이스 엔진으로 MySQL 선택



생성한 데이터베이스의 읽기 전용 복제본 생성



RDS 로 접근 가능한 보안 그룹을 생성하여 WEB 서버와 Bastion Host에서의 접근만 RDS 로 접근 허용

인바운드 규칙 정보

| 보안 그룹 규칙 ID | 유형 | 정보 | 프로토콜 | 정보 | 포트 범위 | 정보 | 소스 | 정보 | 설명 - 선택 사항 | 정보 |
|-------------|--------------|----|------|----|-------|----|--------|----|------------------------|---------------------------|
| - | MySQL/Aurora | | TCP | | 3306 | | 사용자 지정 | | Q | |
| - | MySQL/Aurora | | TCP | | 3306 | | 사용자 지정 | | Q | Buddy-Database-To-WEB |
| | | | | | | | | | sg-03abbfe06b1f7a10f X | |
| | | | | | | | | | sg-0ac22c75dff4ec376 X | Buddy-Database-To-Bastion |

데이터 베이스 파라미터 그룹 지정

RDS > 파라미터 그룹 > utf8encoding > 변경 사항 미리 보기

변경 사항 미리 보기

| 파라미터 | 현재 값 | 새 값 |
|--------------------------|------|--------------------|
| character_set_client | | utf8mb4 |
| character_set_connection | | utf8mb4 |
| character_set_database | | utf8mb4 |
| character_set_results | | utf8mb4 |
| character_set_server | | utf8mb4 |
| collation_connection | | utf8mb4_general_ci |
| collation_server | | utf8mb4_unicode_ci |

◆ WEB 서버 게시판 글 작성을 통한 RDS 동작 검증

The screenshot shows a web application interface with a navigation bar at the top containing links for BUDDY, CHALLENGE, CLUB, SOCIACLING, BOARD, REVIEW, and LOGOUT. The main content area displays a post titled "쿠버네티스를 공부하자" created by "test" on April 2, 2023, at 12:45 PM. The post content is "고래 귀여워". Below the post are two buttons: "목록" (List) on the left and "수정" (Edit) and "삭제" (Delete) on the right.

마스터 RDS 에 접속하여 게시판에 작성한 글 확인

```
ubuntu@ip-172-31-11-121:~/django/buddy$ mysql -h database1.ctwmhbcednmg.ap-northeast-2.rds.amazonaws.com -u admin -p

mysql> use django_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from review_review;
+----+-----+-----+-----+
| id | title           | content          | file_upload      | create_date    |
|    | modify_date     |                  | author_id       |
+----+-----+-----+-----+
| 1  | az              | az               |                 | 2023-04-01    |
| 07:56:42.740357 | 2023-04-01 07:56:42.740403 | 1               |
| 2  | 쿠버네티스를 공부하자 | 고래 귀여워 | review/data/2023/04/02/Test.png | 2023-04-01    |
| 15:45:37.277904 | 2023-04-01 15:45:37.277950 | 2               |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Slave DB 는

```
ubuntu@ip-172-31-11-121:~/django/buddy$mysql -h database1-replica.ctwmhbcednmg.ap-northeast-2.rds.amazonaws.com -u admin -p
```

```
mysql> use django_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> select * from review_review;
```

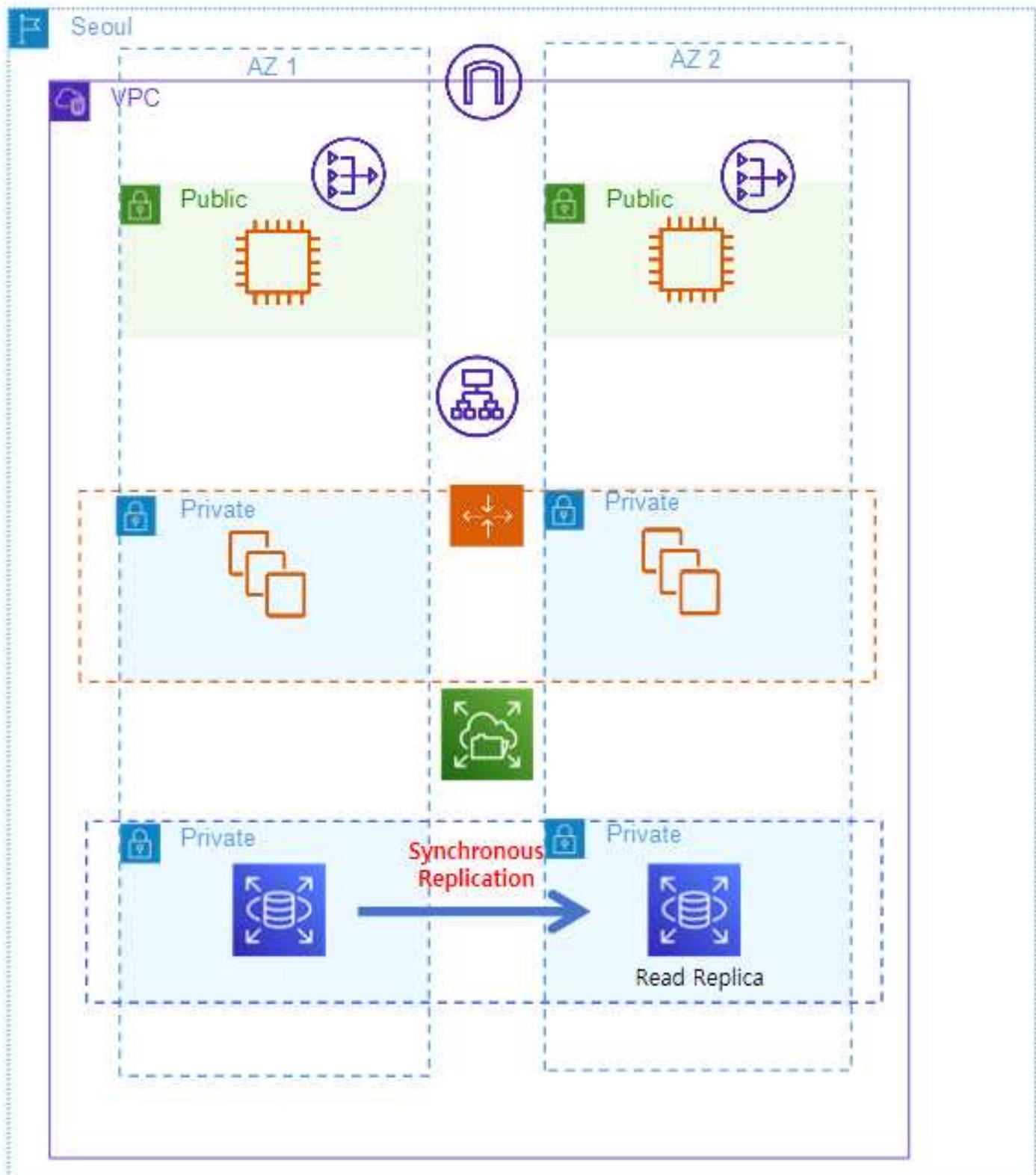
| id title content file_upload create_date |
|--|
| modify_date author_id |
| 1 az az 2023-04-01 07:56:42.740357 2023-04-01 07:56:42.740403 1 |
| 2 쿠버네티스를 공부하자 고래 귀여워 review/data/2023/04/02/Test.png 2023-04-01 15:45:37.277904 2023-04-01 15:45:37.277950 2 |

```
+-----+-----+-----+-----+-----+-----+
| 1 | az | az | | | 2023-04-01 07:56:42.740357 | 2023-04-01 07:56:42.740403 | 1 |
| 2 | 쿠버네티스를 공부하자 | 고래 귀여워 | review/data/2023/04/02/Test.png | 2023-04-01 15:45:37.277904 | 2023-04-01 15:45:37.277950 | 2 |
+-----+-----+-----+-----+-----+-----+
```

2 rows in set (0.00 sec)

[생성 리소스 요약]

AZ1 에 RDS를 생성하고 읽기 복제본 RDS 를 AZ2 에 배치하였다. RDS 의 인바운드 규칙 설정으로 WEB 인스턴스와 Bastion Host 에서의 접근만 허용하였다.



1. 인프라 | AWS Route 53

대시보드에서 구매한 도메인 확인

The screenshot shows the 'Registered Domains' section of the AWS CloudFront console. It includes a search bar, tabs for 'Domain List', 'Domain Preview', and 'Domain Report'. A table displays domain information: 'buddyverse.link' (Type: Standard, Status: Enabled, Last Refreshed: 2024-03-30). The table also includes columns for 'Personal Information Protection' and 'Expiration Date'.

레코드값 설정

The screenshot shows the 'Records (4) Settings' section of the AWS Route 53 console. It lists four records for the domain 'buddyverse.link': 1. A record of type 'A' with value 'dualstack.k8s-default-flaskb...', 2. An NS record with values 'ns-1226.awsdns-25.org.', 'ns-638.awsdns-15.net.', 'ns-251.awsdns-31.com.', and 'ns-1560.awsdns-03.co.uk.', 3. An SOA record with value 'ns-1226.awsdns-25.org.', and 4. A CNAME record with value '_51c5db4278c58a976f93f5e...'. The table includes columns for 'Record Name', 'Type', 'Value', 'TTL', 'Status', and 'Last Refreshed'.

구매한 도메인을 입력하여 웹 페이지 요청이 가능

The screenshot shows a web browser window with the URL 'buddyverse.link' in the address bar. The page content features three cartoon characters (two men and one woman) against a background of flying birds. Speech bubbles say 'WOW'. A red arrow points to the status bar at the bottom of the browser, which shows '안전하지 않음' (Not Safe).

2. 보안 | AWS ACM

SSL 서버 인증서 발급

퍼블릭 인증서 요청

도메인 이름
인증서에 대해 하나 이상의 도메인 이름을 제공합니다.

완전히 정규화된 도메인 이름 정보
buddyverse.link

이 인증서에 다른 이름 추가
이 인증서에 이름을 추가할 수 있습니다. 예를 들어, 'www.example.com'에 대한 인증서를 요청하는 경우 고객이 두 이름 중 하나로 사이트에 접속할 수 있도록 'example.com'이라는 이름을 추가할 수 있습니다.

검증 방법 정보
도메인 소유권을 검증하기 위한 방법 선택

DNS 검증 – 권장
인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한이 있는 경우 이 옵션을 선택합니다.

이메일 검증
인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한을 소유하지 않거나 획득할 수 없는 경우 이 옵션을 선택합니다.

키 알고리즘 정보
암호화 알고리즘을 선택합니다. 일부 알고리즘은 일부 AWS 서비스에서 지원되지 않을 수 있습니다.

RSA 2048
RSA는 가장 널리 사용되는 키 유형입니다.

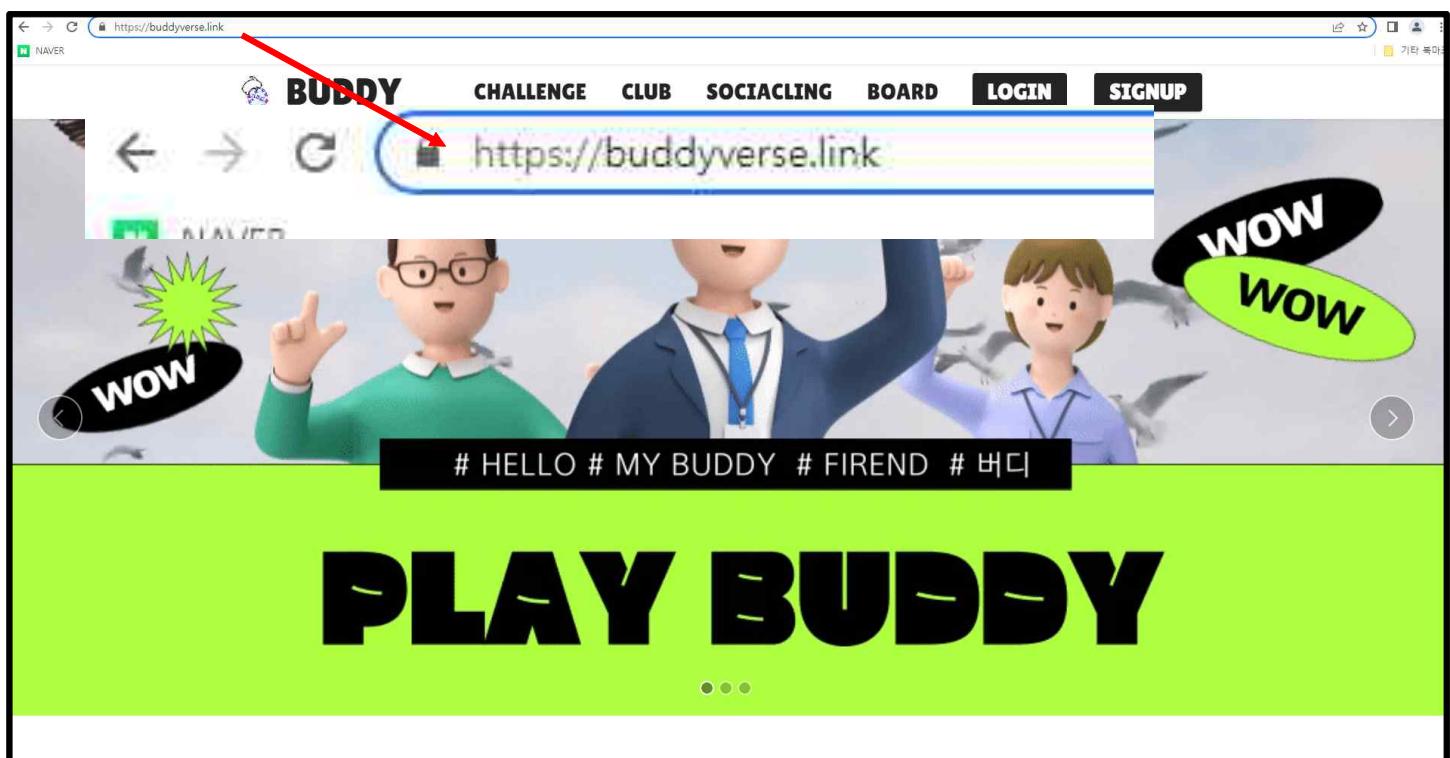
ECDSA P256
암호화 강도는 RSA 3072와 동일합니다.

ECDSA P384
암호화 강도는 RSA 7680와 동일합니다.

발행한 서버 인증서를 이용해 HTTPS를 받는 리스너를 로드 밸런서에 추가

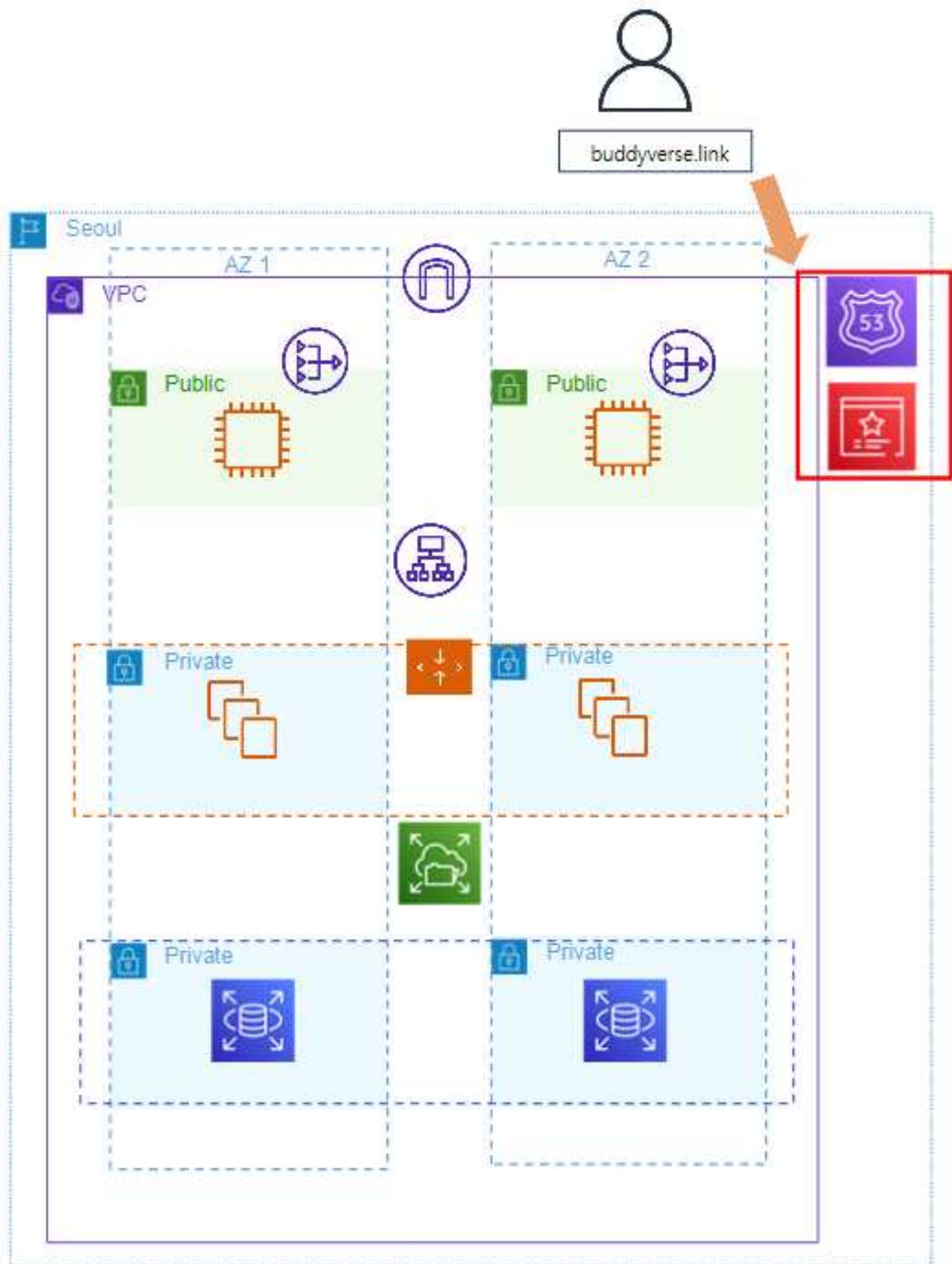
| Listeners (2) | | | | | |
|--|---------------|------------------------------------|---------------------------------|--------------------------------------|---|
| A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules. | | | | | |
| | Protocol:Port | ARN | Security policy | Default SSL cert | Default routing rule |
| <input type="checkbox"/> | HTTP:80 | <input type="button" value="ARN"/> | Not applicable | Not applicable | 1. Return fixed response <ul style="list-style-type: none"> <input type="radio"/> Response code: 503 <input type="radio"/> Response body <input type="radio"/> Response content type: text/plain |
| <input type="checkbox"/> | HTTPS:443 | <input type="button" value="ARN"/> | ELBSecurityPolicy-TLS13-1-2-... | buddyverse.link (Certificate ID:...) | 1. Forward to <ul style="list-style-type: none"> <input type="radio"/> k8s-default-demosflas-717729a4c <input type="radio"/> Group-level stickiness: Off |

Client에서 <http://www.buddyverse.link> 으로 접근했을 때, HTTPS 로 Redirection 하는 것 확인



[생성 리소스 요약]

도메인 네임을 생성하여 buddyverse.link URL을 입력하여 웹 서버로 접근할 수 있으며, ACM 서비스를 이용하여 SSL/TLS 접속이 가능하다.



3. 통합 모니터링 및 관리 | AWS CloudWatch

대시보드 생성 > 대시보드 이름 입력

새 대시보드 생성

대시보드 이름

CPU-Per-Second

대시보드 이름에 유효한 문자는 "0~9A~Za~z~_"입니다.

이 대시보드에 추가

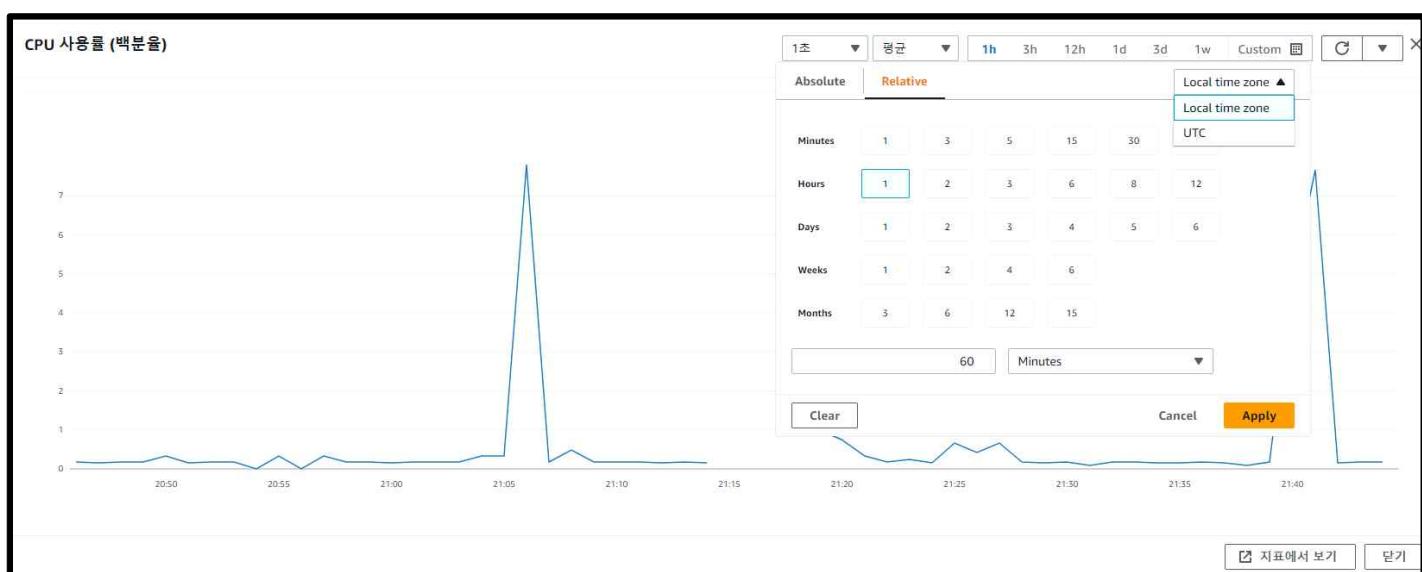
미리 채워진 Explorer 위젯
미리 만들어진 Explorer 템플릿 중 하나를 선택합니다.

빈 Explorer 위젯
처음부터 새로 Explorer 구축을 시작합니다.

템플릿 선택

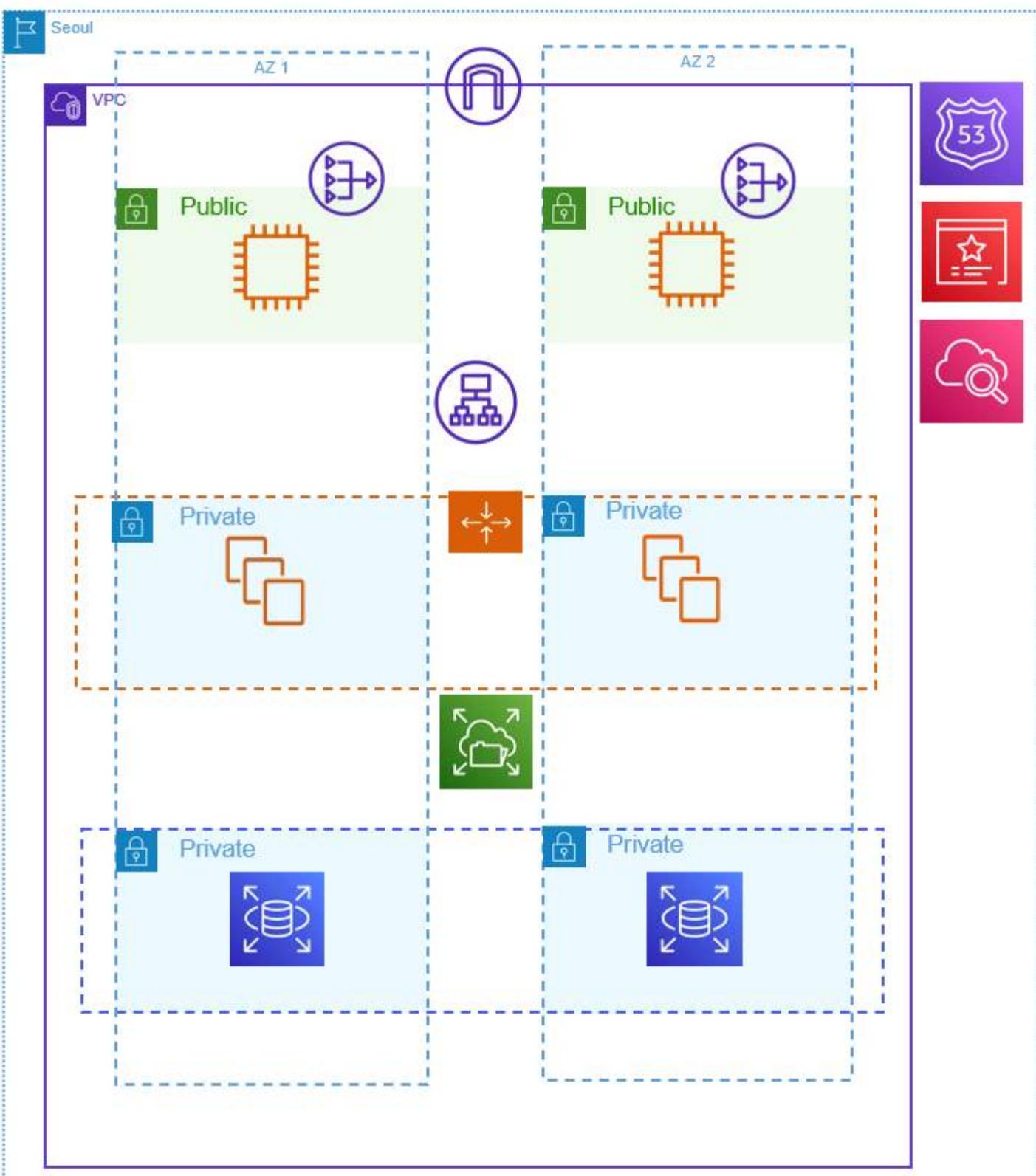
유형별 EC2 인스턴스 ▾

선택한 지표에 대한 모니터링 가능



[생성 리소스 요약]

도메인 네임을 생성하여 buddyverse.link URL을 입력하여 웹 서버로 접근할 수 있으며, ACM 서비스를 이용하여 SSL/TLS 접속이 가능하다.



04 기대효과

▣ 고객 요구 사항 확인 및 기대효과



인프라의 유연한 확장성

CPU 부하에 따른 Auto scaling 동작 검증을 통하여 부하도 증가 및 감소에 따라 웹 서버 인스턴스의 개수가 탄력적으로 운영됨을 확인하였다.

초기 인프라 구축 및 유지 비용 절감

클라우드는 초기 인프라에 대한 구축 비용을 절감할 수 있으며 인프라의 크기를 필요에 따라 조정할 수 있어서 사용한 만큼의 비용만 지불하므로 비용을 절감할 수 있다.

장애를 대비한 가용성 보장으로 지속적인 서비스 제공

2개의 가용 영역 사용으로 단일 가용 영역의 장애가 전체 시스템에 영향을 미치지 않도록 보장하며 Auto scaling이나 ELB 서비스를 이용하여 웹 서버 인스턴스의 가용성을 제공하였다.

서버 과부하로 인한 속도 저하 문제 해결

읽기 요청이 많이 일어나는 문제를 해결하기 위해 AWS RDS 읽기 전용 복제본을 사용하여 읽기 요청에 대해 부하를 분산하였다. 이를 통해 읽기 요청에 빠르게 응답하여 속도 저하 문제를 해결하였다.

인프라 및 어플리케이션의 성능, 가용성 및 오류 감시

AWS CloudWatch를 이용하여 어플리케이션 및 인프라 구성 요소의 상태를 실시간 모니터링하여 장애에 대비하며 대시 보드를 이용하여 다양한 방식으로 시각화하여 리소스 상태를 쉽게 파악할 수 있다.

사용자 개인 정보 및 민감한 정보에 대한 보안

AWS ACM을 이용하여 SSL/TLS 인증서를 사용하면 데이터가 암호화되어 전송되기 때문에 중간에서 정보를 가로채더라도 복호화할 수 없다. 또한 인증서를 통해 데이터를 보내는 서버의 신원이 검증되므로, 중간자 공격 등의 위협으로부터 보호될 수 있다.