

AWS Network Services:

VPC, ELB, Route 53

Hyunchan, Park

<http://oslab.jbnu.ac.kr>

Division of Computer Science and Engineering

Jeonbuk National University

(시작 전에..)

- 새로운 VPC 에 만들었던 Wordpress 3개와 DB 인스턴스 시작
- Community AMI 이용해서 새로운 Wordpress 인스턴스 2개 시작
 - Community AMI
 - Bitnami-wordpress-4.7.4.php56-0-linux-ubuntu-14.04.3-x86_64-hvm-ebs
 - SSH user id: bitnami (keypair 로 접속)
 - Wordpress ID//PW: user//bitnami



개인 과제 #7

- 제출: 하나의 ppt에 각 캡처 파일을 넣은 후, PDF 파일로 변환해 LMS 제출
 - 파일 이름: 학번.pdf
 - Page #1: 제목, 학번, 이름
 - Page #2~#7: 아래 캡처 화면 하나씩
- ELB 실습
 - 두 개 인스턴스 각각의 Public IP 로 접속한 화면. 두 화면이 함께 나오도록 (#2)
 - ELB 도메인 이름에 대해 nslookup 수행한 화면 (#3)
- 새로운 VPC 에서 ELB 설정
 - 새로운 VPC의 서브넷 설정 화면 (#4)
 - 보안 그룹 설정 화면 (#5)
 - ELB 설정 완료된 화면 (#6)
 - 해당 ELB 주소로 접근해서 Wordpress 접근한 화면 (#7)
- 기한: 11/23 (월) 23:59
 - 지각 감점: 5%p / 12H
 - 1주 이후 제출 차단



ELB (Elastic Load Balancing)

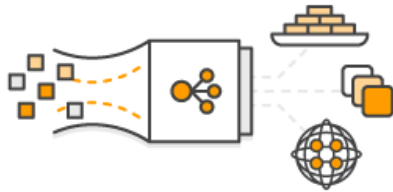
참고: AWS ELB Documentation

“ELB와 EBS의 아키텍처로 생각해보는 사용상 주의할 점들”, AWS 한국 사용자모임

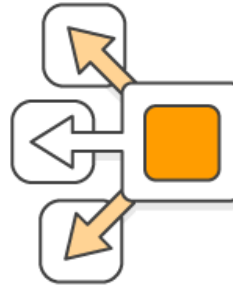


ELB

- 들어오는 애플리케이션 트래픽을 자동으로 분산
 - 대상: Amazon EC2 인스턴스, 컨테이너, IP 주소 등
 - 단일 가용 영역 또는 여러 가용 영역에 분산 가능
 - 애플리케이션의 내결함성 보장
 - 고가용성, 자동 확장/축소, 보안
 - 세 가지 로드 밸런서 제공



Application Load Balancer



Network Load Balancer

* 이전 세대



Classic Load Balancer

Application and Network Load Balancer

- 차이점: 각각 L7 과 L4 계층에서 동작
 - 프로토콜
 - ALB: HTTP, HTTPS
 - NLB: TCP
 - 성능
 - 더 낮은 계층에서 동작하므로 ALB에 비해 NLB의 처리 성능이 높음
- Classic LB
 - L4, L7 으로 명확히 구분하기 전, 통합된 형태로 서비스하였음
 - VPC 이전에 EC2-classic 이라 불리던 네트워크 서비스에서 사용됨
 - VPC 사용 시에는 더 이상 사용을 권장하지 않음

ELB 특징

- 고가용성
 - 인스턴스의 상태를 주기적으로 점검하여, 정상 상태의 인스턴스에만 트래픽을 전달함
- 탄력성
 - Auto-scaling 과 통합 가능
 - ELB 자체도 scale out 을 수행하여, 부하에 탄력적으로 대응함
- 기타
 - 보안: 보안 그룹으로 제어 가능
 - 모니터링: CloudWatch 와 통합
 - Hybrid load balancing: 사용자의 로컬 on premise 리소스와 통합 가능

ELB 특징

- ELB Scale-out
 - ELB는 인스턴스를 기반으로 제공되는 서비스
 - 인스턴스가 감당하지 못하는 수준의 트래픽이 몰리면, scale out 수행
 - 추가 인스턴스를 생성해 처리함
 - 생성까지 시간이 걸리므로 요청이 단기간에 급증하는 경우, 서비스 딜레이가 생길 수 있음
- MultiAZ 로 로드 밸런싱을 하는 경우
 - 각 AZ 별로 ELB 인스턴스가 생성됨
 - AZ별로 같은 비율로 트래픽이 분배됨
 - 따라서 각 AZ의 서비스 처리 성능은 유사하게 유지되어야 함

ELB 요금

<https://aws.amazon.com/ko/elasticloadbalancing/pricing/>

Application Load Balancer

Application Load Balancer가 실행된 시간 또는 부분 시간 그리고 시간당 사용된 로드 밸런서 용량 단위(LCU)에 대해 요금이 부과됩니다.

Network Load Balancer

Network Load Balancer가 실행된 시간 또는 부분 시간 그리고 시간당 Network Load Balancer에서 사용된 로드 밸런서 용량 단위(LCU)에 대해 요금이 부과됩니다.

AWS 리전별 Application Load Balancer 요금

AWS 리전별 Network Load Balancer 요금

LCU 세부 정보

LCU는 애플리케이션 로드 밸런서가 트래픽을 처리하는 차원을 측정합니다(시간당 평균). 4가지 차원이 측정됩니다.

- 새 연결 수: 초당 새로 설정된 연결 수. 일반적으로 연결마다 많은 요청이 전송됩니다.
 - 활성 연결 수: 분당 활성 연결 수.
 - 처리된 바이트: HTTP 요청 및 응답을 위해 로드 밸런서에서 처리된 바이트 수(GB).
 - 규칙 평가: 로드 밸런서가 처리한 규칙 수와 요청 속도의 곱. 처음 10개의 처리 규칙은 무료(규칙 평가 = 요청 속도 × 규칙 수). 가장 사용량이 많은 차원에 대해서만 요금이 부과됩니다. 1LCU는 다음을 포함합니다.
- 초당 25개의 새로운 연결.
 - 분당 3,000개의 활성 연결.
 - EC2 인스턴스, 컨테이너 및 IP 주소를 대상으로 하는 경우 시간당 1GB, Lambda 함수를 대상으로 하는 경우 0.4GB
 - 초당 1,000 규칙 평가

리전: 미국 동부(버지니아 북부) ↕

AWS 리전에서 Application Load Balancer를 사용하는 경우:

- Application Load Balancer 시간(또는 부분 시간)당 0.0225 USD
- LCU 시간(또는 부분 시간)당 0.008 USD

Outposts에서 Application Load Balancer를 사용하는 경우:

- Application Load Balancer 시간(또는 부분 시간)당 0.0225 USD
- LCU 시간(또는 부분 시간)당 0.00 USD

• 링크 내용의 요금 예를 살펴볼 것



실습 내용 (기존 VPC 과제와는 별개로)

- 로드밸런서를 이용하여 2개 이상의 인스턴스로 부하 분배
 - Wordpress 인스턴스 2개 생성
 - Community AMI
 - Bitnami-wordpress-4.7.4.php56-0-linux-ubuntu-14.04.3-x86_64-hvm-ebs
 - SSH user id: bitnami (keypair 로 접속)
 - Wordpress ID//PW: user//bitnami
 - 각 인스턴스에 웹페이지 접근 확인
 - 로드밸런서 생성
 - 로드밸런서에 인스턴스들 연결
 - 각 인스턴스의 서로 다른 글을 쓰고, 각각의 동작 확인

예시. 서로 다른 별개의 인스턴스 생성

인스턴스 시작

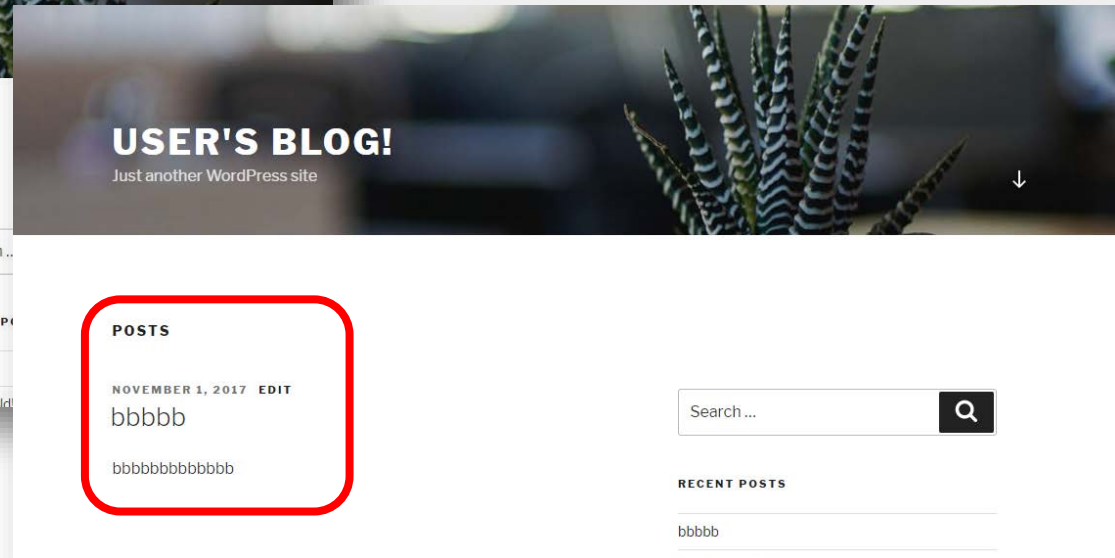
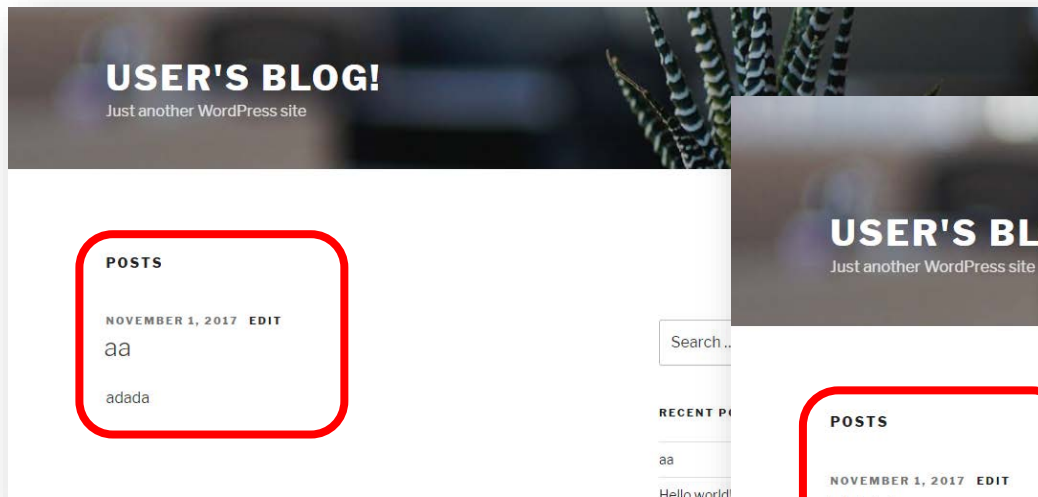
연결

작업 ▾

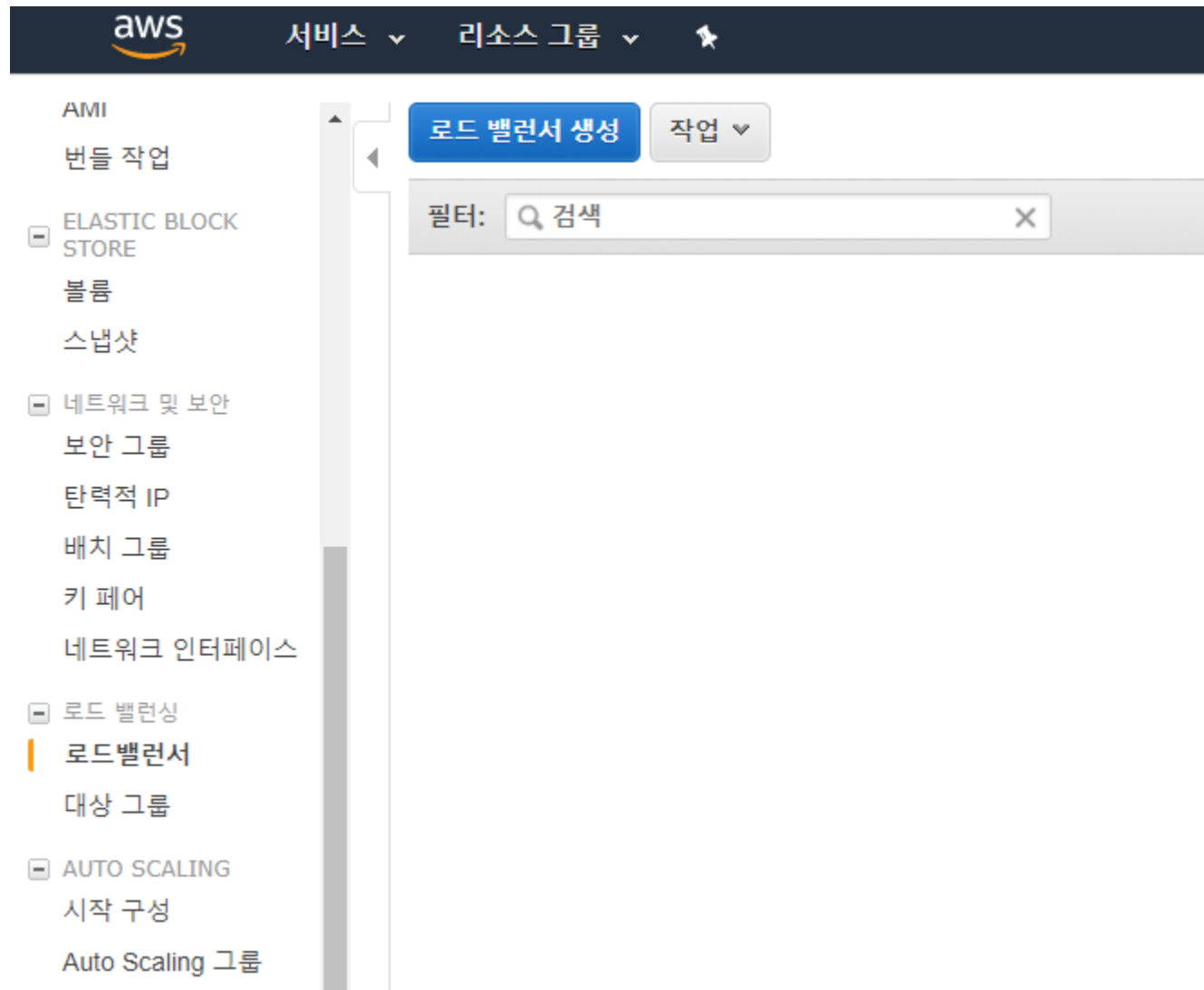
search : WP

필터 추가

<input type="checkbox"/>	Name ▾	인스턴스 ID ▴	인스턴스 유형 ▾	가용 영역 ▾	인스턴스 상태 ▾	상태 검사 ▾
<input type="checkbox"/>	WPBitnamiMarketplace2	i-0b02559342983e5...	t2.micro	ap-northeast-2a	● running	✓ 2/2 검사 통과
<input type="checkbox"/>	WPBitnamiMarketplace	i-0c69b9b128d7c9c92	t2.micro	ap-northeast-2a	● running	✓ 2/2 검사 통과



로드 밸런서 생성



The screenshot displays the AWS Management Console interface. At the top, the AWS logo is visible, followed by navigation tabs for '서비스' (Services) and '리소스 그룹' (Resource Groups). The left-hand navigation pane lists various AWS services, including 'AMI', 'ELASTIC BLOCK STORE', '네트워크 및 보안' (Network & Security), '로드 밸런싱' (Load Balancing), and 'AUTO SCALING'. The '로드 밸런싱' section is expanded, and '로드밸런서' (Load Balancers) is selected. The main content area features a blue button labeled '로드 밸런서 생성' (Create Load Balancer) and a dropdown menu for '작업' (Actions). Below this is a search bar with the placeholder text '필터: 검색' (Filter: Search).



로드 밸런서 생성



서비스 ▾

리소스 그룹 ▾



Hyunchan Park ▾

서울 ▾

지원 ▾

로드 밸런서 유형 선택

탄력적 로드 밸런싱은 다음 세 유형의 로드 밸런서를 지원합니다 - 애플리케이션 로드 밸런서, 네트워크 로드 밸런서(신규) 및 클래식 로드 밸런서. 요구 사항을 충족하는 로드 밸런서 유형을 선택하십시오. [내게 적합한 로드 밸런서에 대해 자세히 알아보십시오.](#)

애플리케이션 로드 밸런서

HTTP
HTTPS

생성

HTTP 및 HTTPS 트래픽을 사용하는 웹 애플리케이션에 대해 유연한 기능을 설정해야 할 경우 애플리케이션 로드 밸런서를 선택합니다. 요청 수준에서 작동하는 애플리케이션 로드 밸런서는 마이크로서비스 및 컨테이너를 비롯해 애플리케이션 아키텍처를 목표로 한 고급 라우팅, TLS 종료 및 표시 기능을 제공합니다.

[자세히 알아보기 >](#)

네트워크 로드 밸런서

TCP

생성

애플리케이션에 초고성능과 정적 IP 주소가 필요한 경우 네트워크 로드 밸런서를 선택합니다. 연결 수준에서 작동하는 네트워크 로드 밸런서는 초당 수백만 개의 요청을 처리하면서도 극히 낮은 지연 시간을 유지할 수 있습니다.

[자세히 알아보기 >](#)

클래식 로드 밸런서

이전 세대
HTTP, HTTPS 및 TCP용

생성

EC2-Classic 네트워크에서 구축된 기존 애플리케이션이 있는 경우 클래식 로드 밸런서를 선택합니다.

[자세히 알아보기 >](#)



로드 밸런서 생성

1. 로드 밸런서 구성 2. 보안 설정 구성 3. 보안 그룹 구성 4. 라우팅 구성 5. 대상 등록 6. 검토

단계 1: 로드 밸런서 구성

기본 구성

로드 밸런서를 구성하려면 이름을 입력하고, 스키마를 선택하고, 하나 이상의 리스너를 지정하고, 네트워크를 선택합니다. 기본 구성은 선택한 네트워크에서의 인터넷 연결 로드 밸런서 및 포트 80에서 HTTP 트래픽을 수신하는 리스너입니다.

이름 ⓘ

스키마 ⓘ ☒ 인터넷 연결 ☐ 내부

IP 주소 유형 ⓘ

리스너

리스너는 구성된 프로토콜 및 포트를 사용하여 연결 요청을 확인하는 프로세스입니다.

로드 밸런서 프로토콜	로드 밸런서 포트
<input type="text" value="HTTP"/>	<input type="text" value="80"/>

가용 영역

로드 밸런서에서 활성화할 가용 영역을 지정합니다. 로드 밸런서는 지정한 가용 영역에 위치한 대상으로만 트래픽을 라우팅합니다. 가용 영역당 1개의 서브넷만 지정할 수 있습니다. 로드 밸런서의 가용성을 높이려면 2개 이상의 가용 영역에서 서브넷을 지정해야 합니다.

VPC ⓘ

<input checked="" type="checkbox"/> 가용 영역	서브넷 ID	서브넷 IPv4 CIDR	이름
<input checked="" type="checkbox"/> ap-northeast-2a	subnet-4359742a	172.31.0.0/20	
<input checked="" type="checkbox"/> ap-northeast-2c	subnet-cf1b7f82	172.31.16.0/20	



로드 밸런서 생성

1. 로드 밸런서 구성

2. 보안 설정 구성

3. 보안 그룹 구성

4. 라우팅 구성

5. 대상 등록

6. 검토

단계 2: 보안 설정 구성



로드 밸런서의 보안을 개선하십시오. 로드 밸런서가 보안 리스너를 사용하고 있지 않습니다.

로드 밸런서로 전송되는 트래픽에 대한 보안이 필요한 경우 프런트 엔드 연결에 HTTPS 프로토콜을 사용하십시오. 첫 번째 단계로 돌아가 기본 구성 섹션에서 보안 리스너를 추가/구성할 수 있습니다. 현재 설정을 사용하여 계속할 수도 있습니다.

1. 로드 밸런서 구성

2. 보안 설정 구성

3. 보안 그룹 구성

4. 라우팅 구성

5. 대상 등록

6. 검토

단계 3: 보안 그룹 구성

보안 그룹은 로드 밸런서에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 로드 밸런서에 도달하도록 허용할 규칙을 추가할 수 있습니다. 먼저 새 보안 그룹을 생성할지 아니면 기존 보안 그룹을 선택할지 결정합니다.

- 보안 그룹 할당:
- ☒ 새 보안 그룹 생성
 - ☐ 기존 보안 그룹 선택

보안 그룹 이름:

설명:

유형 ⓘ

프로토콜 ⓘ

포트 범위 ⓘ

원본 ⓘ

HTTP ▼

TCP

80

위치 무관 ▼

0.0.0.0/0, ::/0



규칙 추가



로드 밸런서 생성

단계 4: 라우팅 구성

로드 밸런서는 지정된 프로토콜 및 포트를 사용하여 이 대상 그룹의 대상으로 요청을 라우팅하며, 상태 검사 설정을 사용하여 이를 수행합니다. 각 대상 그룹은 하나의 로드 밸런서에만 연결될 수 있습니다.

대상 그룹

대상 그룹 ⓘ	새 대상 그룹 ▼
이름 ⓘ	WP-Instances
프로토콜 ⓘ	HTTP ▼
포트 ⓘ	80
대상 유형 ⓘ	instance ▼

상태 검사

프로토콜 ⓘ	HTTP ▼
경로 ⓘ	/

▼ 고급 상태 검사 설정

포트 ⓘ	<input checked="" type="radio"/> 트래픽 포트 <input type="radio"/> 재정의
정상 임계 값 ⓘ	5
비정상 임계 값 ⓘ	2
제한 시간 ⓘ	5 초
간격 ⓘ	30 초
성공 코드 ⓘ	200



로드 밸런서 생성

단계 5: 대상 등록

대상 그룹에 대상을 등록합니다. 활성화된 가용 영역에서 대상을 등록할 경우, 로드 밸런서는 등록 프로세스가 완료되고 대상이 초기 상태 검사를 통과하자마자 해당 대상으로 요청을 라우팅하기 시작합니다.

등록된 대상

인스턴스를 등록 취소하려면 등록된 인스턴스를 하나 이상 선택한 다음 제거를 클릭합니다.

제거

<input type="checkbox"/>	인스턴스	이름	포트	상태	보안 그룹	영역
사용 가능한 인스턴스가 없습니다.						

인스턴스

추가 인스턴스를 등록하려면 하나 이상의 실행 인스턴스를 선택하고 포트를 지정한 다음 추가를 클릭하십시오. 기본 포트는 대상 그룹에 지정된 포트입니다. 인스턴스가 지정된 포트에 이미 등록되어 있는 경우 다른 포트를 지정해야 합니다.

등록된 항목에 추가

포트: 80

WP

<input type="checkbox"/>	인스턴스	이름	상태	보안 그룹	영역	서브넷 ID	서브넷 CIDR
<input type="checkbox"/>	i-0b025593...	WPBitnami...	running	WordPress...	ap-northea...	subnet-4359742a	172.31.0.0/20
<input type="checkbox"/>	i-0c69b9b1...	WPBitnami...	running	WordPress...	ap-northea...	subnet-4359742a	172.31.0.0/20



로드 밸런서 생성

단계 5: 대상 등록

대상 그룹에 대상을 등록합니다. 활성화된 가용 영역에서 대상을 등록할 경우, 로드 밸런서는 등록 프로세스가 완료되고 대상이 초기 상태 검사를 통과하자마자 해당 대상으로 요청을 라우팅하기 시작합니다.

등록된 대상

인스턴스를 등록 취소하려면 등록된 인스턴스를 하나 이상 선택한 다음 제거를 클릭합니다.

제거

<input type="checkbox"/>	인스턴스	이름	포트	상태	보안 그룹	영역
<input type="checkbox"/>	i-0b02559342983...	WPBitnamiMa...	80	running	WordPress Certified by Bitnami-4...	ap-northeast-2a
<input type="checkbox"/>	i-0c69b9b128d7c...	WPBitnamiMa...	80	running	WordPress Certified by Bitnami-4...	ap-northeast-2a

VPC 과제에서 만든 웹서버 인스턴스 (3개) 등록

인스턴스

추가 인스턴스를 등록하려면 하나 이상의 실행 인스턴스를 선택하고 포트를 지정한 다음 추가를 클릭하십시오. 기본 포트는 대상 그룹에 지정된 포트입니다. 인스턴스가 지정된 포트에 이미 등록되어 있는 경우 다른 포트를 지정해야 합니다.

등록된 항목에 추가 포트:

<input checked="" type="checkbox"/>	인스턴스	이름	상태	보안 그룹	영역	서브넷 ID	서브넷 CIDR
<input checked="" type="checkbox"/>	i-0b025593...	WPBitnami...	● running	WordPress...	ap-northea...	subnet-4359742a	172.31.0.0/20
<input checked="" type="checkbox"/>	i-0c69b9b1...	WPBitnami...	● running	WordPress...	ap-northea...	subnet-4359742a	172.31.0.0/20



로드 밸런서 생성

단계 6: 검토

계속하기 전에 로드 밸런서 세부 정보를 검토하십시오.

▼ 로드 밸런서

[편집](#)

이름 TestELB
스키마 internet-facing
리스너 포트:80 - 프로토콜:HTTP
IP 주소 유형 ipv4
VPC vpc-76d28f1f
서브넷 subnet-4359742a, subnet-cf1b7f82
태그

▼ 보안 그룹

[편집](#)

보안 그룹 load-balancer-wizard-1

▼ 라우팅

[편집](#)

대상 그룹 새 대상 그룹
대상 그룹 이름 WP-Instances
포트 80
대상 유형 instance
프로토콜 HTTP
상태 검사 프로토콜 HTTP
경로 /
상태 검사 포트 traffic port
정상 임계 값 5
비정상 임계값 2
제한 시간 5
간격 30
성공 코드 200

▼ 대상

[편집](#)

인스턴스 i-0b02559342983e5e8 (WPBitnamiMarketplace2):80, i-0c69b9b128d7c9c92 (WPBitnamiMarketplace):80



로드 밸런서 완성

로드 밸런서 생성 상태



성공적으로 로드 밸런서 생성 완료

성공적으로 로드 밸런서 TestELB 을(를) 생성했습니다.

참고: 로드 밸런서가 완전히 설정되어 트래픽을 라우팅할 준비가 완료되고 대상이 등록 프로세스를 완료한 후 초기 상태 검사를 통과하려면 몇 분이 걸릴 수 있습니다.

로드 밸런서 생성

작업 ▾



닫기

필터: 🔍 검색



1~1/1

<input checked="" type="checkbox"/>	이름	DNS 이름	상태	VPC ID
<input checked="" type="checkbox"/>	TestELB	TestELB-1144714255.ap-nor...	provisioning	vpc-76d2i

필터: 🔍 검색



<input checked="" type="checkbox"/>	이름	DNS 이름	상태	VPC ID	가용 영역
<input checked="" type="checkbox"/>	TestELB	TestELB-1144714255.ap-nor...	active	vpc-76d28f1f	ap-northeast-2a, ap-nor...



현재 상황

- 현재 세 개의 접속 가능한 IP와 Domain name 이 존재
 - 2개의 인스턴스: 개별적인 DB를 갖고 있으므로 서로 다른 블로그
 - 1개의 ELB
- ELB의 주소로 접근하면 자동으로 2개의 인스턴스로 전달
 - 50:50 확률
- 이를 확인하기 위해
 - 각 인스턴스 domain name으로 접근해서,
 - 1번 인스턴스에는 “AAAA”라고 글을 쓰고,
 - 2번 인스턴스에는 “BBBB” 라고 글을 쓴 후,
 - ELB 로 접속해 내용을 확인한다.

ELB의 DNS 주소로 접속



POSTS

NOVEMBER 1, 2017

aa

adada



RECENT POSTS



ELB의 DNS 주소로 재접속 (새로고침 연속 수행)



POSTS

NOVEMBER 1, 2017

bbbbbb

bbbbbbbbbbbbbb

OCTOBER 27, 2017



RECENT POSTS

bbbbbb

Another article



로드 밸런서 IP 확인

- 실제 ELB 처리를 수행하는 인스턴스의 IP를 확인할 수 있음
 - 보통 가용 영역 별로 하나씩 생성됨

```
C:\Users\whcpark>nslookup chanswp.tk
서버:      cbdns.chonbuk.ac.kr
Address:   203.254.128.11

권한 없는 응답:
이름:      chanswp.tk
Addresses: 52.78.137.187
           52.78.198.45

C:\Users\whcpark>
```

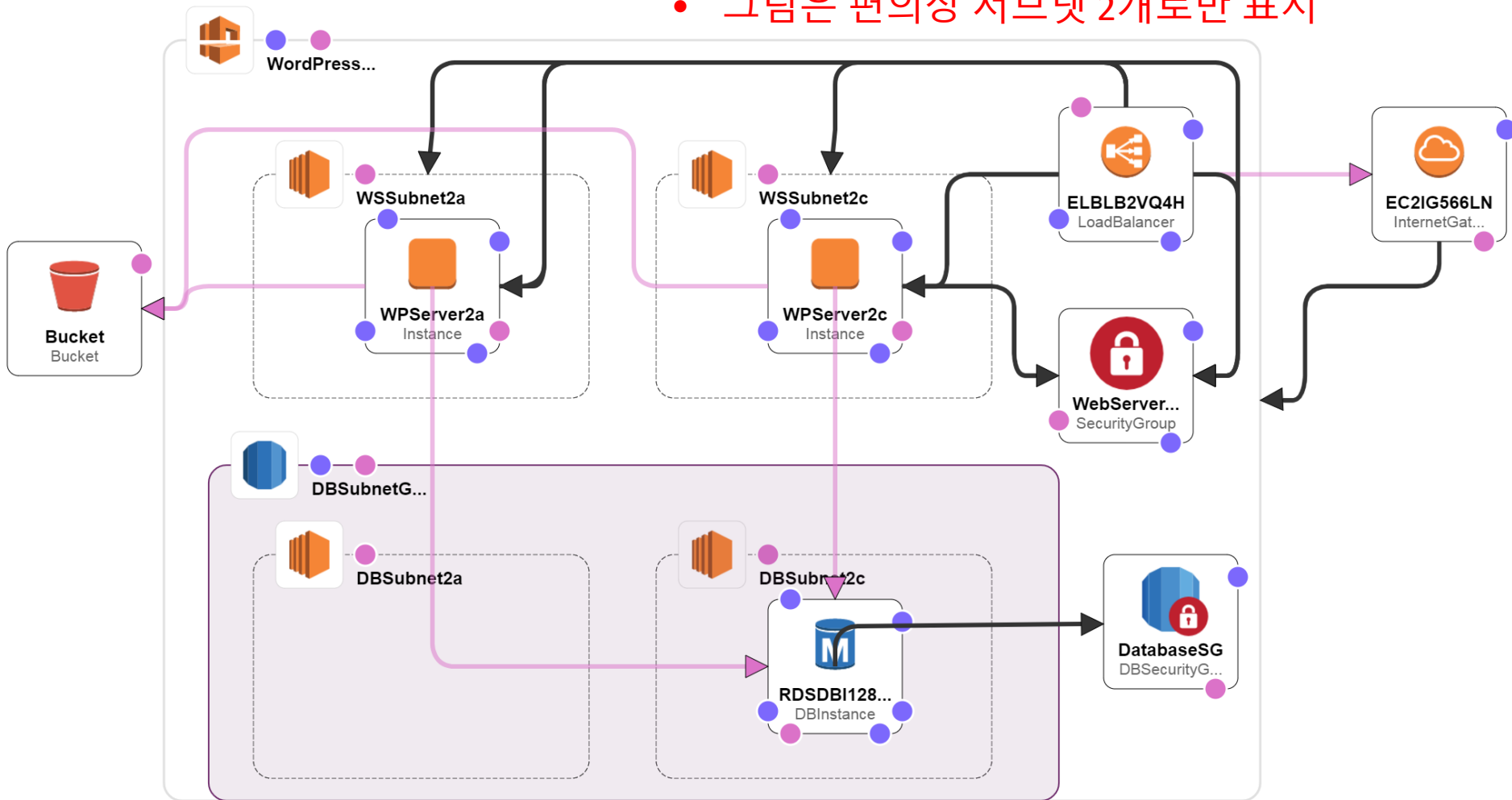


Wordpress VPC 에 추가



Wordpress VPC 구성

- 웹서버 인스턴스는 각 서브넷 별로 3개씩 존재함
- 그림은 편의상 서브넷 2개로만 표시



인스턴스 상태 확인

- 현재 인스턴스 3개가 동작 중 (VPC 과제에서 작성한 본인의 AMI 기반)
- 각 인스턴스에서 각각 wordpress 서비스 상태 확인
 - `$ sudo /opt/bitnami/ctlscript.sh status`
 - Mysql 이 수행 중이라면 중단
 - `$ sudo /opt/bitnami/ctlscript.sh stop mysql`
- 각 인스턴스의 EIP로 접속하여 내용 확인
 - 첫 화면 확인
 - 각각 수정한 후, 다시 최종 내용 확인할 것
 - 모두 같은 정보를 보여주어야 함



구분 가능하도록 글 작성



로드밸런서를 통하지 않고, 각 인스턴스의 EIP 로 직접 접근하여 확인

POSTS

NOVEMBER 2, 2017 EDIT

bbbb from Instance B

bbbb

NOVEMBER 2, 2017 EDIT

aaaa from Instance A

aaaa

AUGUST 4, 2017 EDIT

Hello world!

Search ...



RECENT POSTS

bbbb from Instance B

aaaa from Instance A

Hello world!

RECENT COMMENTS

A WordPress Commenter on Hello world!

ARCHIVES

November 2017



S3 Offload plugin 동작 확인

- 이미지의 주소 확인: <http://s3-ap-northeast-2.amazonaws.com/hcpark.wpbucket/wp-content/uploads/2017/11/02045847/%ED%9D%99%EA%BC%AD%EB%91%90.jpg>



NOVEMBER 2, 2017 BY USER

cccc with a file



RECENT POSTS

cccc with a file

bbbb from Instance B

aaaa from Instance A

Hello world!



새로운 ELB 생성 및 대상 그룹 등록

- 보안 그룹에서 ELB 보안 그룹 새로 생성
 - SSH나 DB 연결을 허용하지 않고, HTTP 만 오픈

1. Load Balancer 구성 2. 보안 설정 구성 3. 보안 그룹 구성 4. 라우팅 구성 5. 대상 등록 6. 검토

3단계: 보안 그룹 구성

보안 그룹은 Load Balancer에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 Load Balancer에 도달하도록 허용할 규칙을 추천합니다. 먼저 새 보안 그룹을 생성할지 아니면 기존 보안 그룹을 선택할지 결정합니다.

보안 그룹 할당: ☒ 새 보안 그룹 생성
☐ 기존 보안 그룹 선택

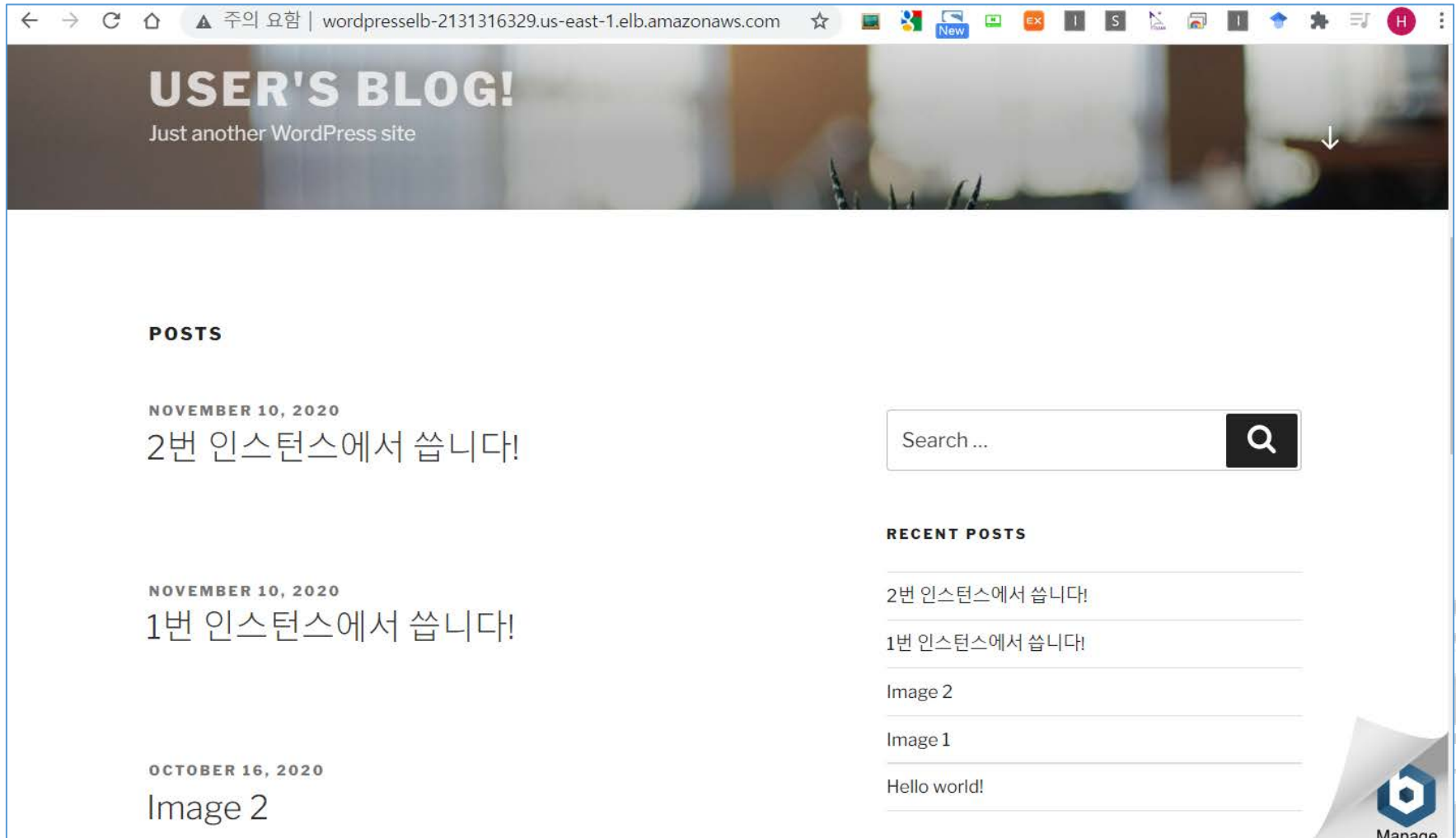
보안 그룹 이름:

설명:

유형 ⓘ	프로토콜 ⓘ	포트 범위 ⓘ	소스 ⓘ
<input type="text" value="HTTP"/>	<input type="text" value="TCP"/>	<input type="text" value="80"/>	<input type="text" value="사용자 지정"/> <input type="text" value="0.0.0.0/0, ::/0"/>



완성! ELB 주소로 접근하여 내용 확인



The screenshot shows a web browser window with the address bar displaying the URL: wordpresselb-2131316329.us-east-1.elb.amazonaws.com. The page title is "USER'S BLOG!" with the subtitle "Just another WordPress site". The main content area is titled "POSTS" and lists three blog entries. The first entry is dated "NOVEMBER 10, 2020" and titled "2번 인스턴스에서 씁니다!". The second entry is also dated "NOVEMBER 10, 2020" and titled "1번 인스턴스에서 씁니다!". The third entry is dated "OCTOBER 16, 2020" and titled "Image 2". On the right side of the page, there is a search bar with the placeholder text "Search ..." and a magnifying glass icon. Below the search bar, the section "RECENT POSTS" lists the same three entries: "2번 인스턴스에서 씁니다!", "1번 인스턴스에서 씁니다!", "Image 2", "Image 1", and "Hello world!". In the bottom right corner, there is a "Manage" button with a blue hexagonal icon.

← → ↻ ⌂ 주의 요함 | wordpresselb-2131316329.us-east-1.elb.amazonaws.com ☆

USER'S BLOG!

Just another WordPress site

POSTS

NOVEMBER 10, 2020

2번 인스턴스에서 씁니다!

NOVEMBER 10, 2020

1번 인스턴스에서 씁니다!

OCTOBER 16, 2020

Image 2

Search ...

RECENT POSTS

2번 인스턴스에서 씁니다!

1번 인스턴스에서 씁니다!

Image 2

Image 1

Hello world!

Manage



정리: HA (high-availability) system 완성!

- 3개의 서로 다른 AZ에서 수행되는 3개의 인스턴스가 동일한 서비스를 사용자에게 제공 중
 - 모두 같은 DB와 S3 버킷을 공유하므로, 똑같은 블로그를 볼 수 있음
- 사용자는 ELB 주소로 접근하므로,
3개 중 어느 인스턴스에서든 서비스를 받을 수 있음
 - 할당은 ELB가 자동으로 관리하며,
이때 동작 중이지 않은 인스턴스로는 요청이 전달되지 않음
- 3개의 인스턴스가 사용자들의 요청을 분산 처리하므로 성능도 높음
 - HA는 성능을 향상시키는 방식으로 활용할 수도 있고,
 - 혹은 Active-Standby 형태로 문제가 생겼을 때만 교체하도록 구성할 수 있음

(끝내기 전에..)

- 완료 후, 리소스 중지 및 EIP 해제
 - 인스턴스들 3개 모두 중지
 - 연결된 EIP 3개 삭제
 - DB 인스턴스 1개 중지
- 현재 남아있는 유료 서비스
 - EBS: Wordpress AMI
 - DB 인스턴스: DB 인스턴스의 저장소 및 스냅샷
 - ELB: \$0.0225/h (하루에 약 \$0.5)
 - 지웠다 또 새로 만들어도 되지만...그냥 놔두자
 - 모두 학기 종료 후, 최종 성적 공지 이후 삭제