On Premise 2Tier Infra 구성 및

AWS 마이그레이션 아키텍처 설계

ver. 1.0

2019년 02월 12일

목차

[1. 프로젝트 개요 4](#_Toc1550755)

[1-1. 프로젝트 명 4](#_Toc1550756)

[1-2. 프로젝트 기간 4](#_Toc1550757)

[1-3. 프로젝트 목적 4](#_Toc1550758)

[1-4. 프로젝트 기대효과 4](#_Toc1550759)

[2. 프로젝트 범위 5](#_Toc1550760)

[2.1 작업 명세서 5](#_Toc1550761)

[3. 클라우드 인프라 구성도 6](#_Toc1550762)

[3.1 클라우드 인프라 구성도 6](#_Toc1550763)

[3.2 서비스 구성 현황 6](#_Toc1550764)

[4. 프로젝트 추진체계 7](#_Toc1550765)

[4.1 프로젝트 수행 인원 7](#_Toc1550766)

[4.2 인원 별 역할 7](#_Toc1550767)

[5. 세부일정 8](#_Toc1550768)

[5.1 상세일정 8](#_Toc1550769)

[6. On Premise 9](#_Toc1550770)

[6.1 고려사항 9](#_Toc1550771)

[6.2 A회사의 On Premise Infra설계 9](#_Toc1550772)

[7. Migration 16](#_Toc1550773)

[7.1 Migration 시나리오 16](#_Toc1550774)

[7.2 VPN 생성 16](#_Toc1550775)

[7.3 Server Migration 22](#_Toc1550776)

[7.4 Database Migration 31](#_Toc1550777)

[7.5 Active Directory Migration 36](#_Toc1550778)

[8. AWS 42](#_Toc1550779)

[8.1 구성도 42](#_Toc1550780)

[8.2 VPC 구성 42](#_Toc1550781)

[8.3 Auto-Scaling 및 ELB구성 44](#_Toc1550782)

[8.4 CloudWatch를 이용한 SNS 구성 46](#_Toc1550783)

[8.5 Snapshot 저장을 위한 S3생성 48](#_Toc1550784)

[9. 최종 Infra 49](#_Toc1550785)

[9.1 최종Infra 구성도 49](#_Toc1550786)

[9.2 부하테스트 49](#_Toc1550787)

[9.3 무결성 검증 51](#_Toc1550788)

1. 프로젝트 개요

**1-1. 프로젝트 명**

On Premise 2 Tier Infra 구성 및 AWS 마이그레이션 아키텍처 설계

**1-2. 프로젝트 기간**

2019년 01월 02일 ~ 2019년 01월 24일

**1-3.** **프로젝트 목적**

▷ 기존의 데이터 센터를 AWS Cloud로 Migration

▷ 데이터 무결성 보장

▷ Migration 시, 다운타임 최소화

▷ 더 큰 트래픽 처리량을 위한 부하 안정성 설계

▷ 가용성 확보를 위한 구조화 설계

**1-4. 프로젝트 기대효과**

▷ 자본 비용을 가변 비용으로 대체

▷ 규모의 경제로 얻게 되는 경제적 효과

▷ 데이터 센터 운용 및 유지 관리에 대한 비용 절감

▷ 속도 및 민첩성 개선

▷ 데이터 무결성 보장

▷ 고가용성으로 인한 안정성 향상

2. 프로젝트 범위

**2.1 작업 명세서**

**AWS 환경으로 Migration**

|  |  |
| --- | --- |
| 업무 | 업무범위 |
| 사용할 서비스 | 글로벌 서비스를 위한 Route 53  고가용성을 위한 ELB와 Auto-Scaling  다중 가용영역을 이용한 고가용성 제공  Cloud Watch를 이용한 모니터링 환경 구축  SNS를 이용한 알림 서비스  RDS 서비스를 활용한 완전관리형 데이터베이스 구축 |
| 문서화 | AWS 아키텍처 설계 보고서 작성 |

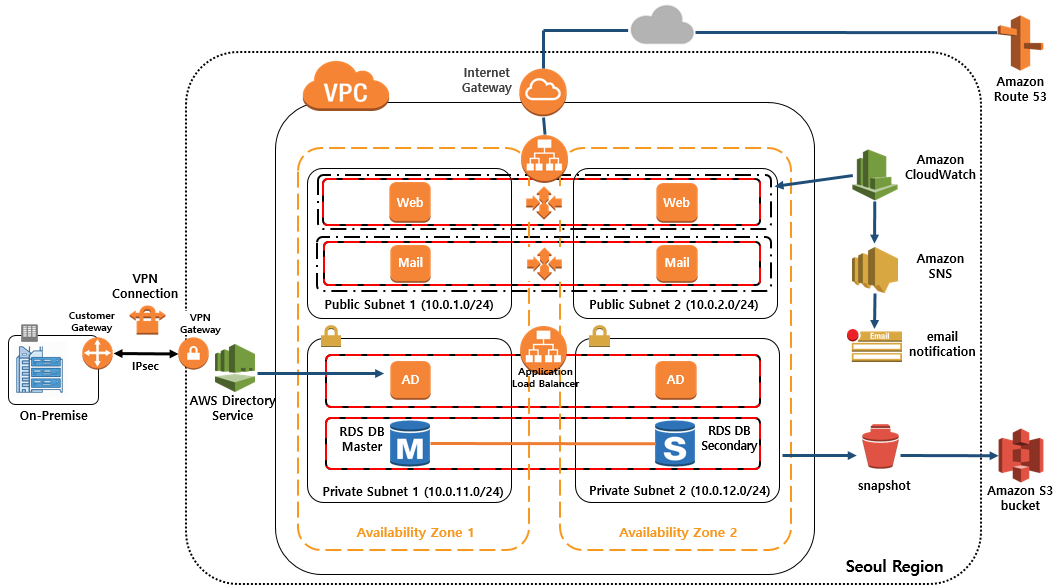
**시스템 구축**

|  |  |
| --- | --- |
| 업무 | 업무범위 |
| 시스템 구축 | * On Premise Infra 구축 * 적용할 아마존 서비스 분석 * Cloud Architecture 설계 * Cloud Architecture 구현 |
| 안정화 | * Database 무결성 검사 * EC2 Stress Test * Web 취약점 분석 |

3. 클라우드 인프라 구성도

**3.1 클라우드 인프라 구성도**

**전체 구성도**



**3.2 서비스 구성 현황**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **서비스명** | **시스템정보** | **Name Tag** | **AZ** | **비고** |
| **VPC** | 가상네트워크 | Migration-VPC |  |  |
| **Security Group** | 보안그룹 | Migration-SG |  | WEB용 보안그룹 |
| **Security Group** | 보안그룹 | Migration-SG |  | Mail용 보안그룹 |
| **Security Group** | 보안그룹 | Migration-SG |  | DB용 보안그룹 |
| **Security Group** | 보안그룹 | Migration-SG |  | AD용 보안그룹 |
| **EC2 #1**  **EC2 #2** | t2.small | Migration-WEB1  Migration-WEB2 | ap-northeast-2a  ap-northeast-2c | 서비스용 웹서버 |
| **EC2 #3**  **EC2 #4** | t2.small | Migration-Mail1  Migration-Mail2 | ap-northeast-2a  ap-northeast-2c | 서비스용 메일서버 |
| **EC2 #5**  **EC2 #6** | t2.small | Migration-AD1  Migration-AD2 | ap-northeast-2a  ap-northeast-2c | 서비스용 AD서버 |
| **ELB #1** | External Load Balancer | Migration-ELB1 | Ap-northeast-2a  Ap-northeast-2c | EC2 #1,#2,#3,#4로 부하분산 |
| **ELB #2** | Internal Load Balancer | Migration-ELB2 | Ap-northeast-2a  Ap-northeast-2c | EC2 #5, #6으로 부하분산 |
| **RDS Master** | MySQL | Migration-DB1 | Ap-northeast-2a | Master Database |
| **RDS Slave** | MySQL | Migration-DB2 | Ap-northeast-2c | Slave Database |
| **IGW** |  | Migration-IGW |  | Public Subnet 연결용 인터넷 라우터 |

4. 프로젝트 추진체계

**4.1 프로젝트 수행 인원**

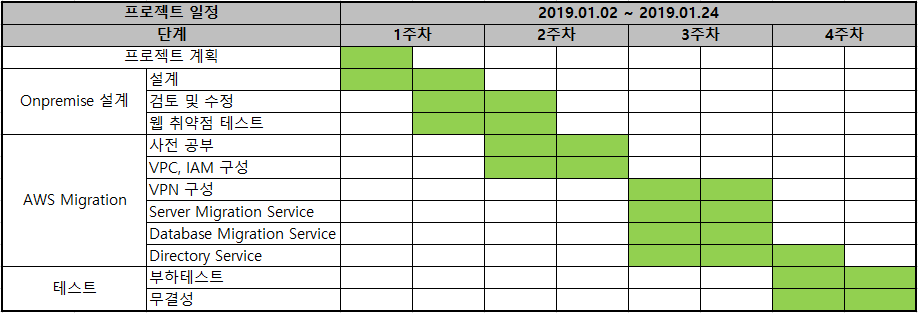


**4.2 인원 별 역할**

|  |  |
| --- | --- |
| 구분 | 역할 |
| **김 상 철** | Database Migration 및 Database 무결성 검사 |
| **이 한 섭** | Active Directory Migration 및 EC2 Stress Test |
| **최 승 원** | Active Directory Migration 및 VPN 구성 |
| **허 현** | Server Migration 및 AWS Auto-Scaling, ELB, Route 53 |
| **홍 덕 화** | Server Migration 및 AWS CloudWatch, SNS |

5. 세부일정

**5.1 상세일정 : 2019년 01월 02일 ~ 2019년 01월 24일**



6. On Premise

**6.1 고려사항**

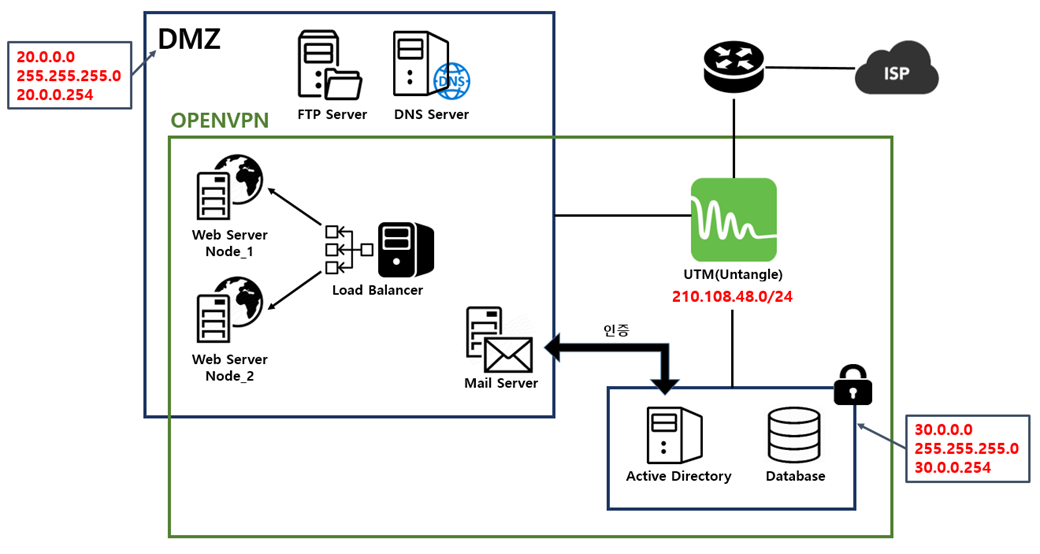
**보안**

▷ Untangle을 사용한 방화벽 구축으로 Private-Zone과 DMZ-Zone을 구분

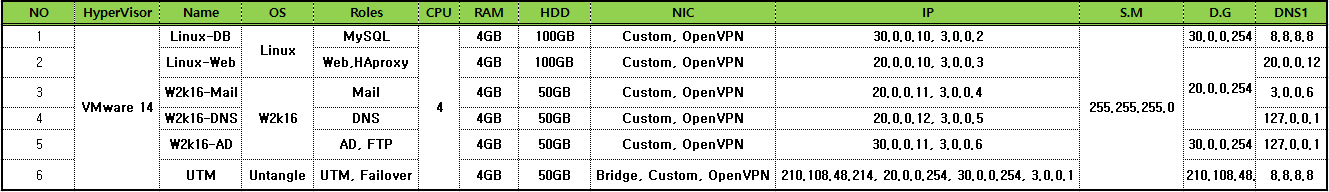
▷ OpenVPN으로 내부 네트워크를 구축하여 보안성 향상

**6.2 A회사의 On Premise Infra설계**

**구성도**

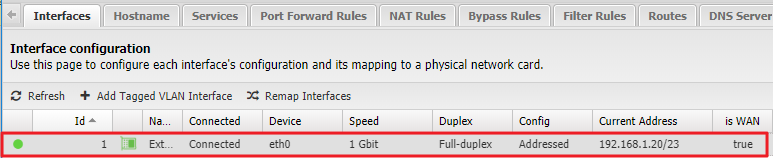


**구성표**

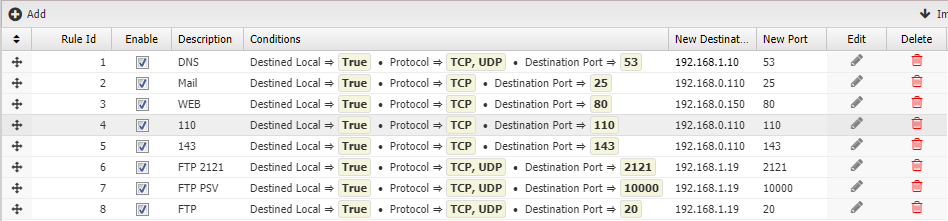


**Untangle**

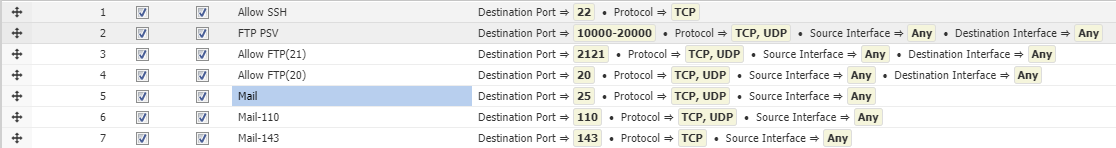
▷외부 통신을 위한 Network Card



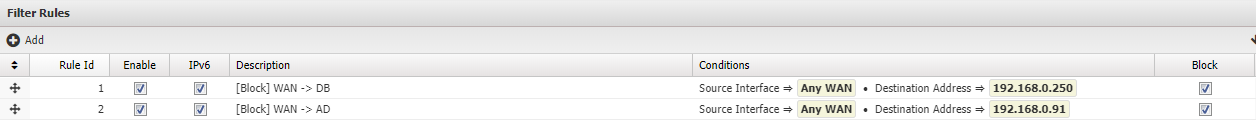
▷ 포트 포워딩



▷ Access Rule

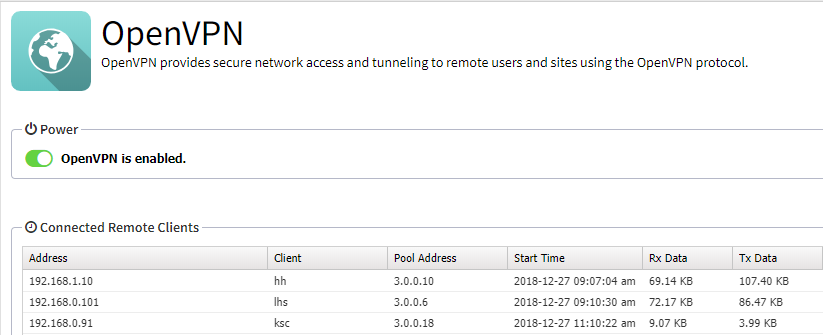


▷ Filter Rule



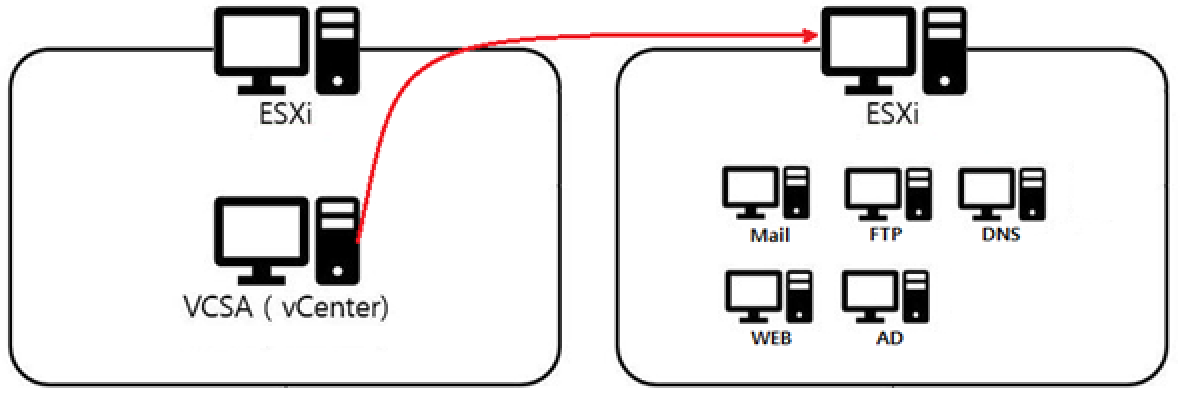
**OpenVPN**

▷ 내부 네트워크용 OpenVPN



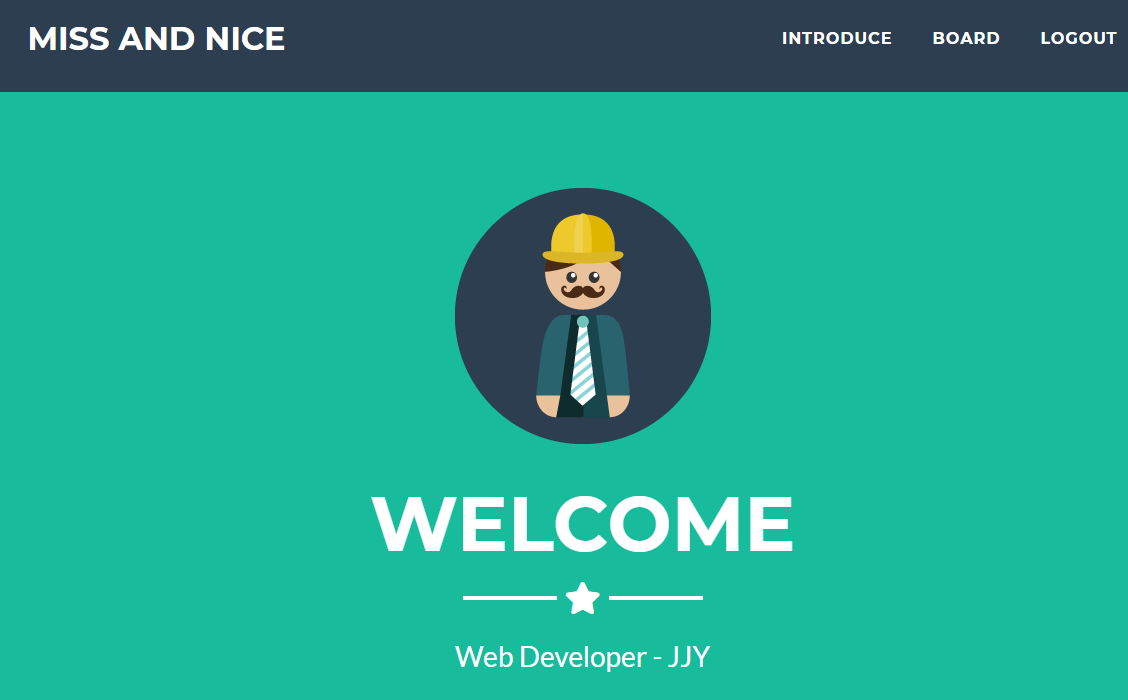
**ESXi, VCSA 구축**

▷ On Premise Server들을 관리할 ESXi와 VCSA를 설치한 뒤 서버 구축

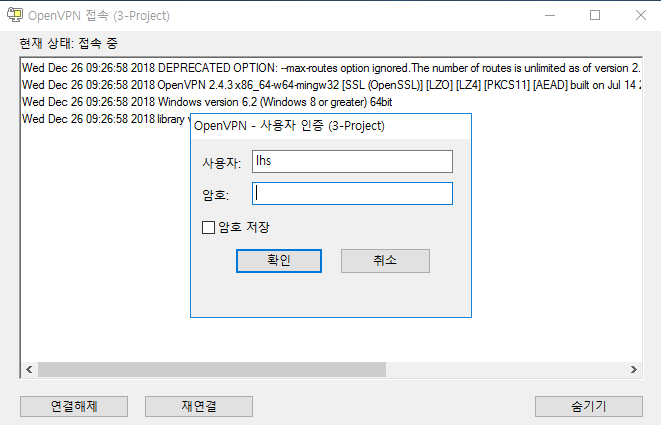
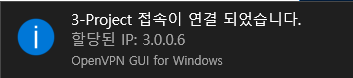


**Web Server**

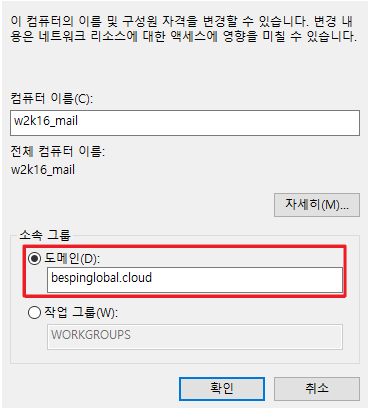
▷ Django를 활용한 Web Server 구축



**Active Directory Server 구축**

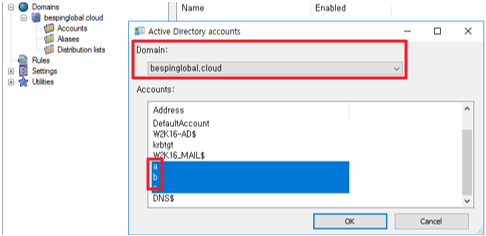
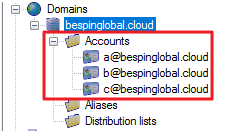
▷ OpenVPN 내에서 AD가입을 하기 위해 먼저 OpenVPN연결

▷ Mail Server에서 AD Member Server 가입



**AD 인증을 통한 Mail Server 구축**

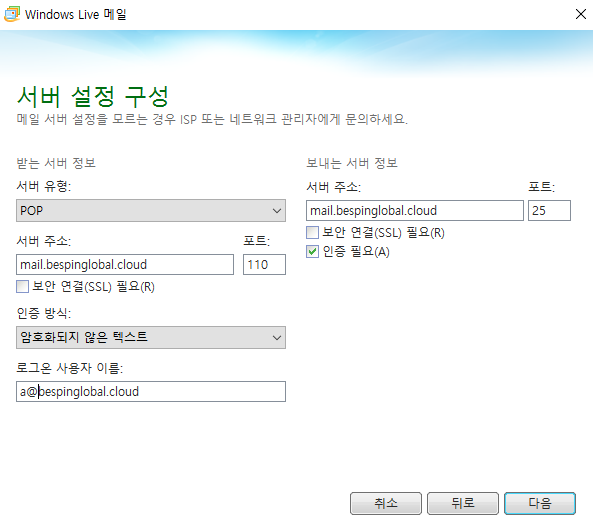
▷ AD 인증을 통한 사용자 계정 추가

****

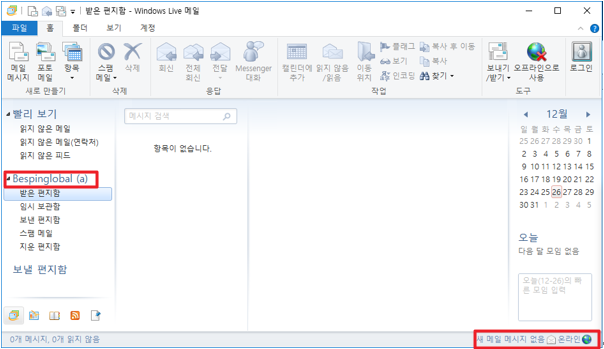
**DNS Server구축**

**FTP 구축**

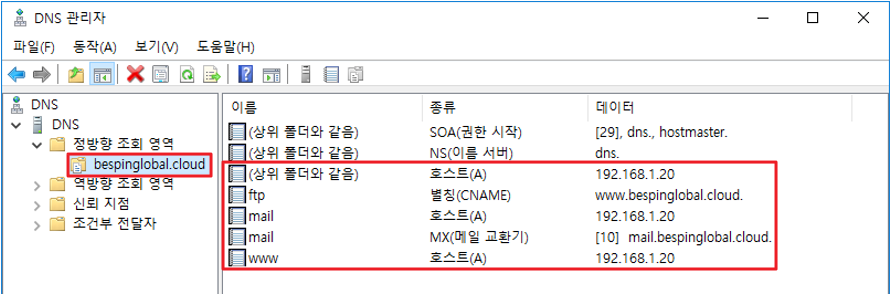
▷ Windows Live 메일에 계정 추가



▷ 계정 정상 접속 확인

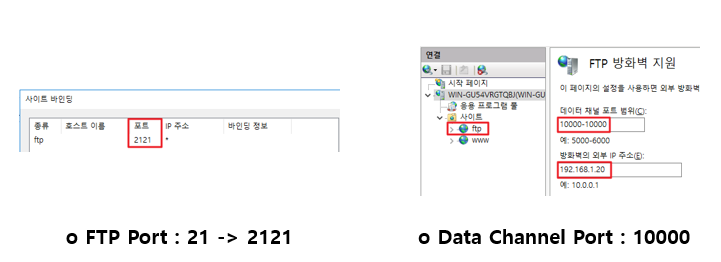


**DNS Server 구축**

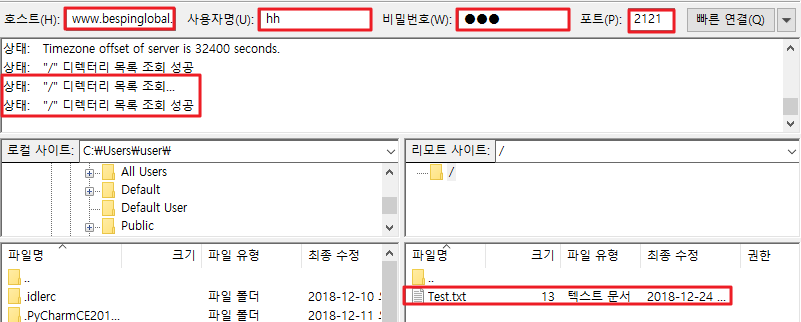


**FTP Server 구축**

▷ 보안상 21번 port를 사용하지 않고 2121번port로 바인딩해서 구축



▷ 디렉토리 목록 조회 확인



**DB Server 구축**

▷MySQL을 사용한 Database



7. Migration

**7.1 Migration 시나리오**

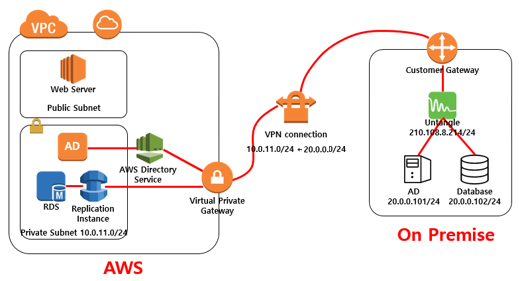
Active Directory와 Database의 보안을 유지하며 Migration 하기위해 VPN을 생성

생성한 VPN을 통해 Active Directory와 Database를 Migration. On Premise Server

Migration을 하기 위해 ESXi에 AWS Connector를 설치한 뒤 Migration을 진행

**7.2 VPN 생성**

**구성도**



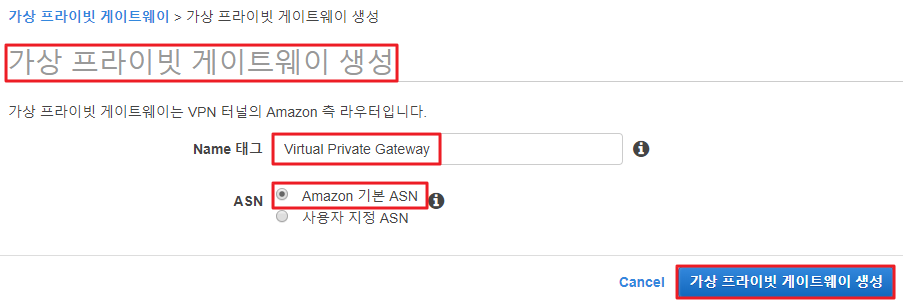
**AWS 구성**

▷ 고객 게이트웨이(Customer Gateway)를 생성.

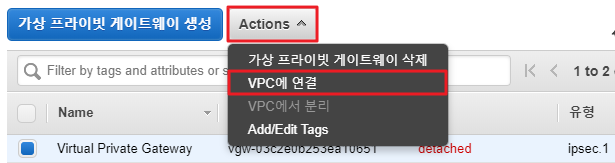
IP주소 부분에 On Premise UTM 장비인 Untangle의 External IP(외부 통신)를 설정



▷ 가상 프라이빗 게이트웨이(Virtual Private Gateway)를 생성

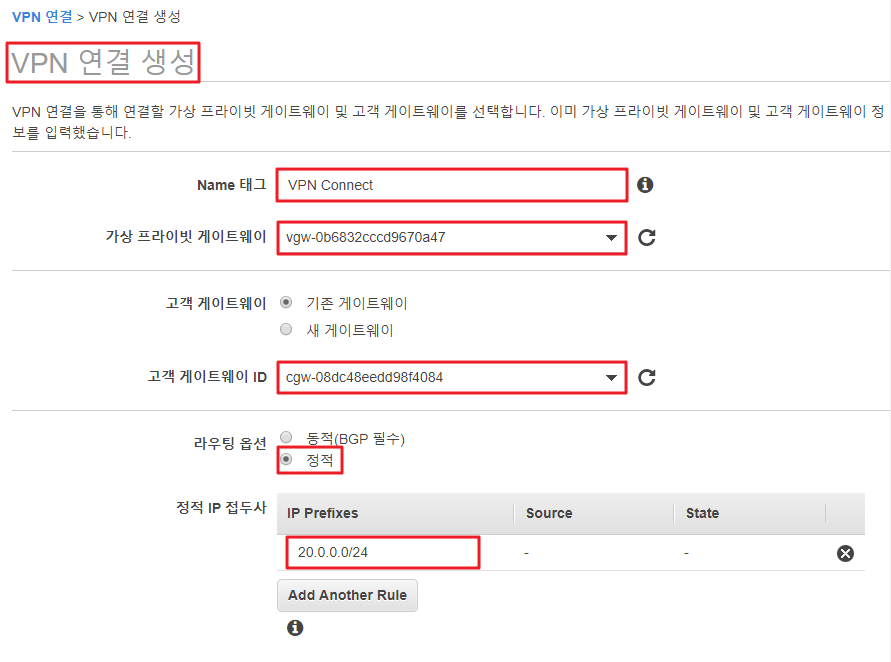


▷ 생성한 가상 프라이빗 게이트웨이를 AWS내에 VPC로 연결

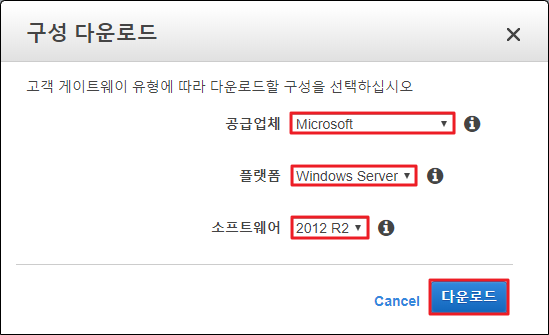
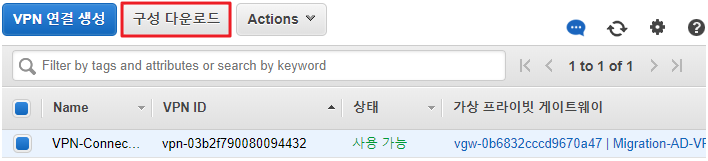


▷ VPN 연결 생성을 클릭하여 가상 프라이빗 게이트웨이와 고객 게이트웨이를 지정

라우팅 옵션은 정적으로 설정하고 정적 IP 접두사에 On Premise 내부 IP 대역 설정

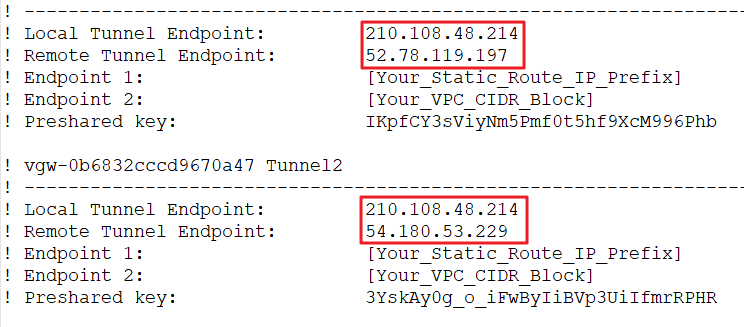


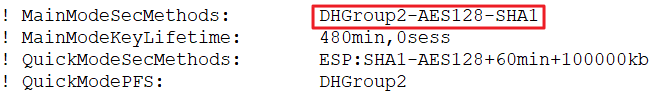
▷ VPN 연결 생성 탭에서 [구성 다운로드]를 누름



▷ Local Tunnel Endpoint : 210.108.48.214

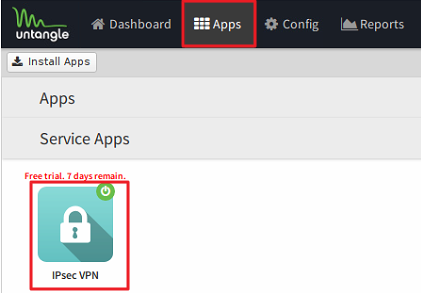
▷ Remote Tunnel Endpoint : 52.78.119.197





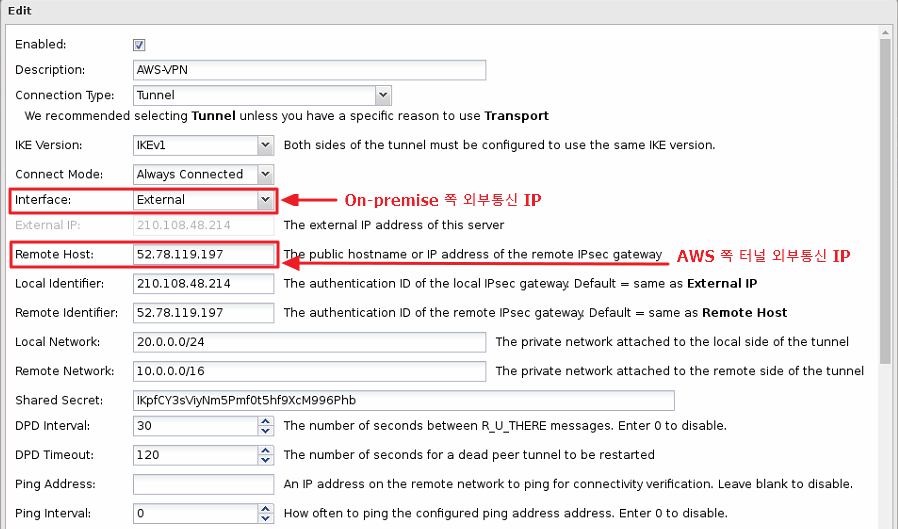
**On Premise 구성**

▷ Untangle에서 IPsec VPN 설치

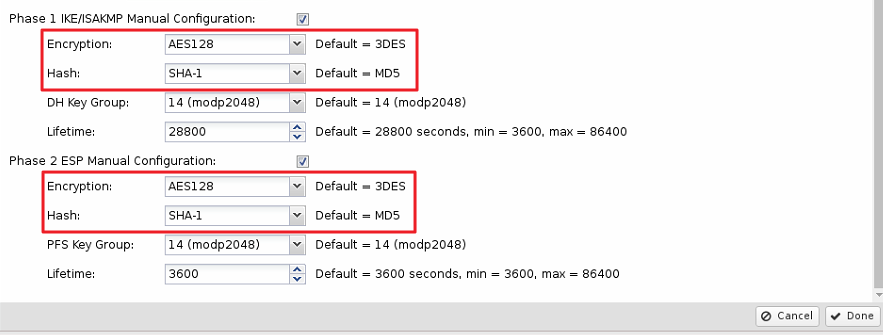


▷ IPsec VPN 설정에서 Interface는 On Premise 외부 통신 IP 설정

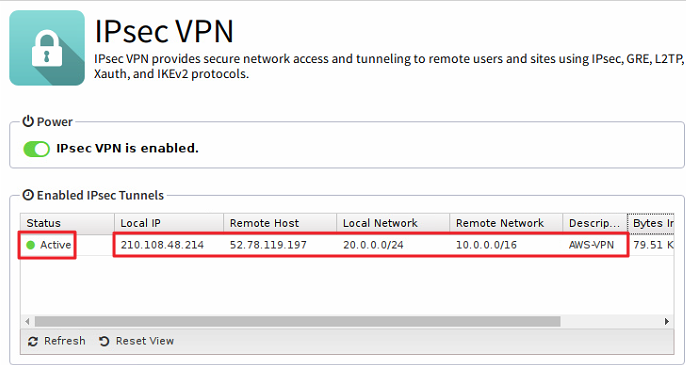
Remote Host 는 AWS 터널 외부 통신 IP 설정



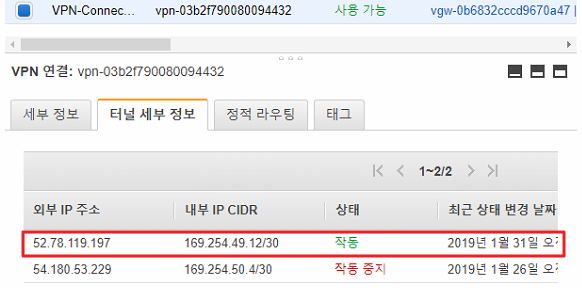
▷ 구성 다운로드한 txt 파일을 참고하여 Phase 1, Phase 2 설정



▷ 모든 설정이 완료되면 On Premise쪽 Untangle에서 Status가 Active로 변경



▷ 설정이 끝나면 VPC-VPN 설정에서 터널 세부 정보에 외부 IP 주소 상태가 작동으로 변경



**7.3 Server Migration**

**Migration 계획**

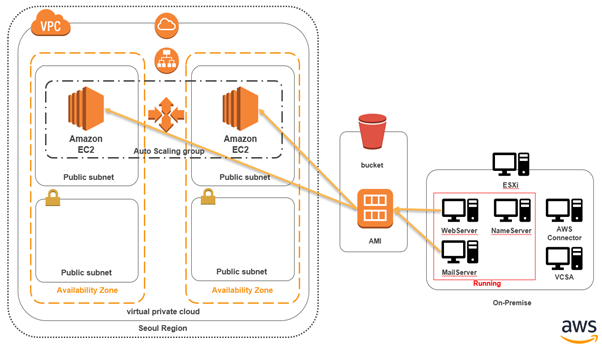
**시나리오**

VCSA에 AWS Connector를 설치 및 구성을 완료 후 AWS로 Server를 Migration 실행

**다운타임 최소화**

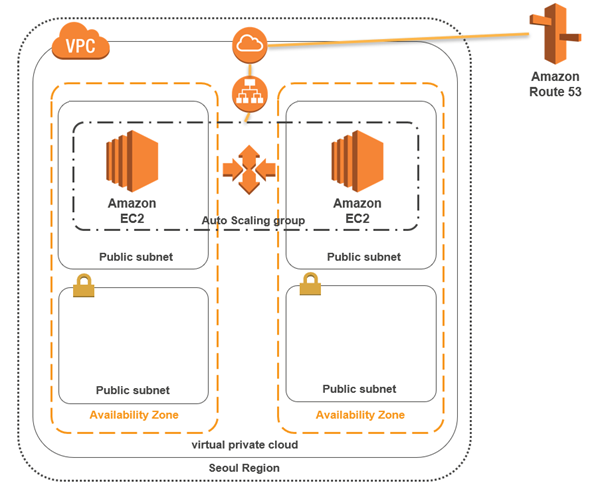
다운타임 최소화를 위해 On Premise Server에서 서비스를 유지한 상태로 Migration Name Server를 변경. Name Server가 변경되는 그 순간에만 다운타임이 발생

1. Web Server와 Mail Server는Migration이 될 때까지 On Premise에서 서비스를 제공

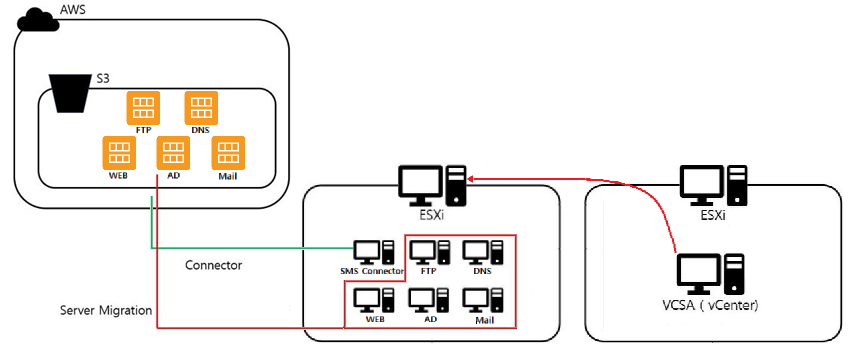


1. Migration이 성공적으로 완료되면 생성된 AMI를 이용하여 인스턴스를 생성

인스턴스가 생성되면 On Premise의 Name Server를 Route53으로 Migration

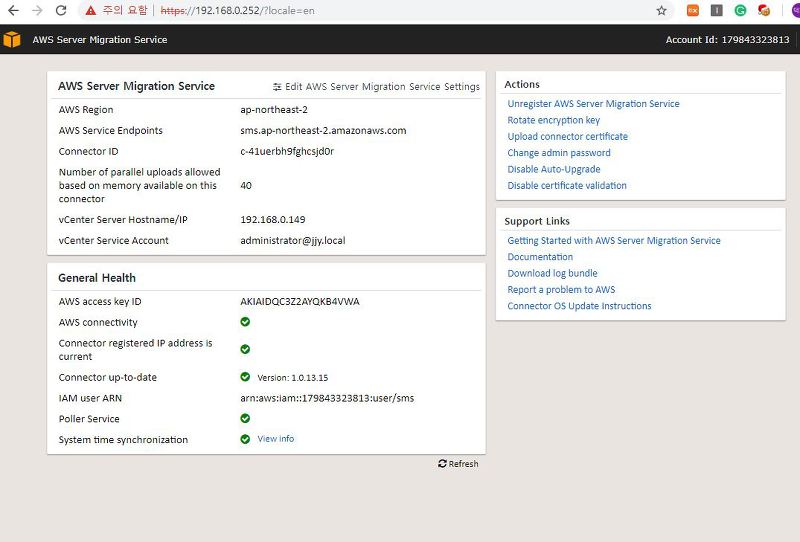


**AWS Connector 설치**



▷ ESXi에 AWS Connector를 설치

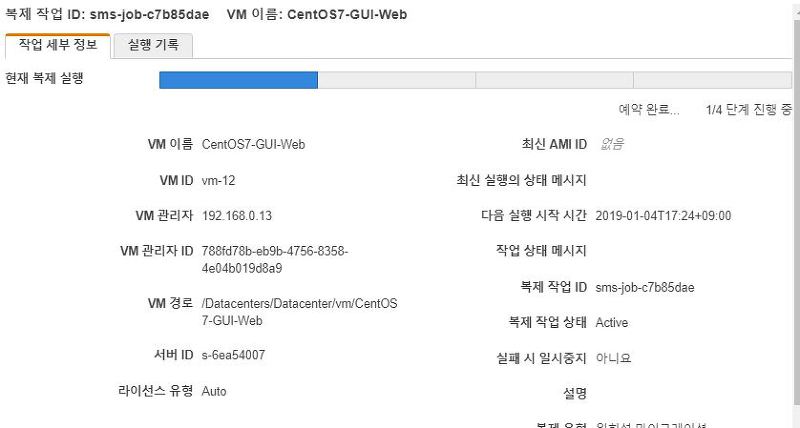
AWS Connector는 Connector가 설치된 VCSA의 VM을 AWS상에 올릴 수 있도록 연결



**Web Server Migration(Linux)**

▷ 서버 카탈로그 가져오기 한 뒤 Web Server를 선택한 뒤 복제 작업 생성

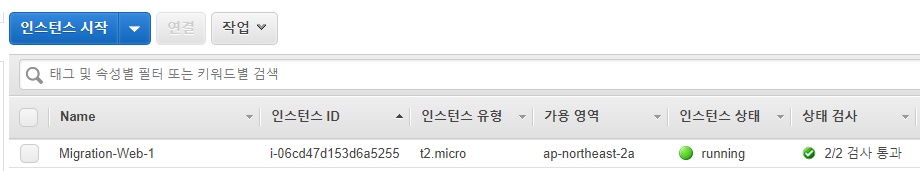




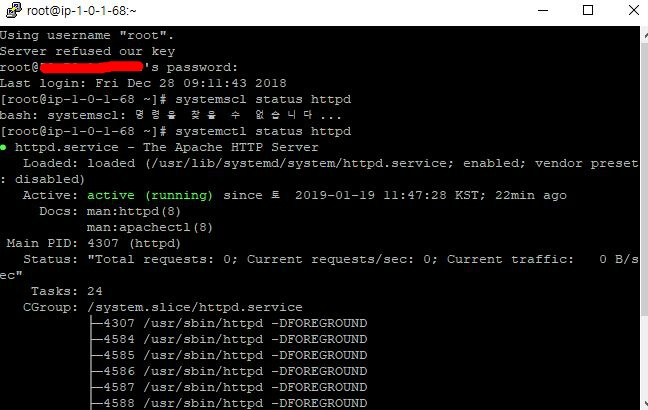
▷ Migration이 완료되면 AMI가 생성



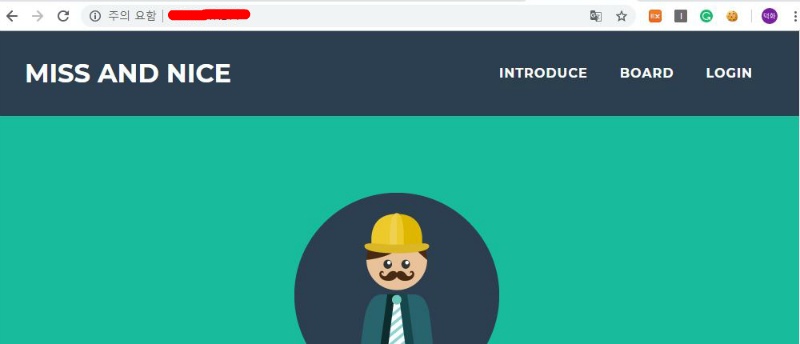
▷ 생성된 AMI를 사용하여 인스턴스를 생성



▷ Migration된 Web Server의 상태를 확인

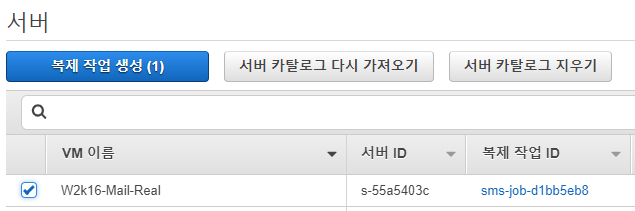


▷ 인스턴스에 할당된 IP로 접속하면 On Premise의 Web Server가 AWS로 Migration된 것을 확인

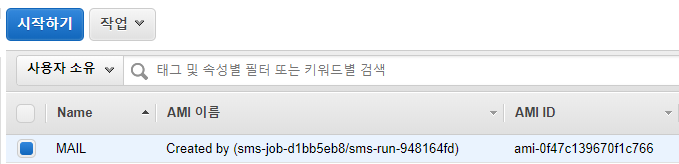


**Mail Server Migration(Windows)**

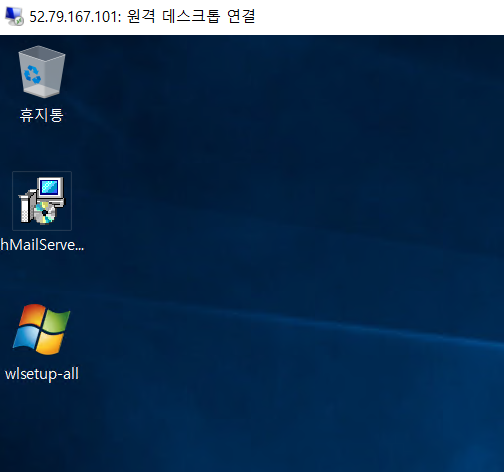
▷ 복제 작업 생성 후 Migration



▷ 생성된 AMI를 사용하여 Instance 생성



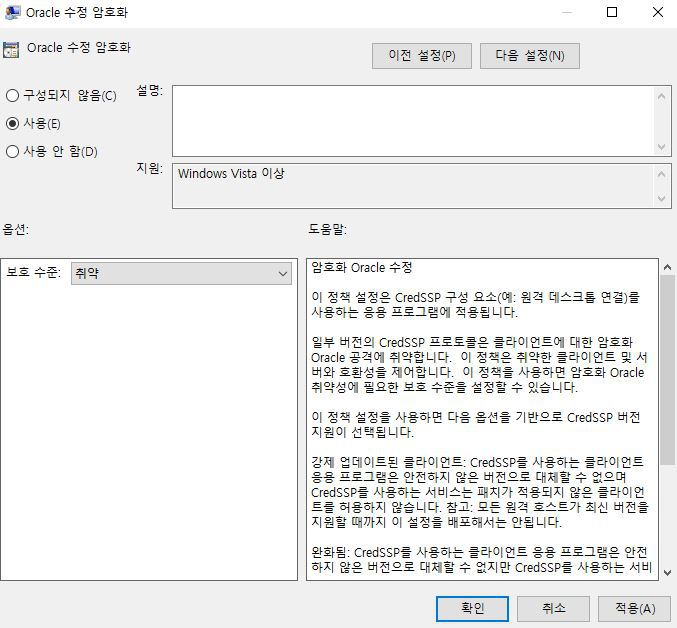
▷ 생성된 인스턴스에 원격 접속해서 On Premise에서 한 구성과 동일한지 확인.



**Windows Server Migration시 주의사항**

1. Migration 작업 전에 On Premise의 VM에서 원격 접속을 위한 3389포트와 원격 접속을 허용
2. VM 생성 시 Firmware 타입을 반드시 BIOS로 생성
3. 원격 접속 시 아래와 같은 오류가 뜨면 원격 접속을 하려는 PC의 Oracle 수정 암호화 보호 수준을 취약으로 수정





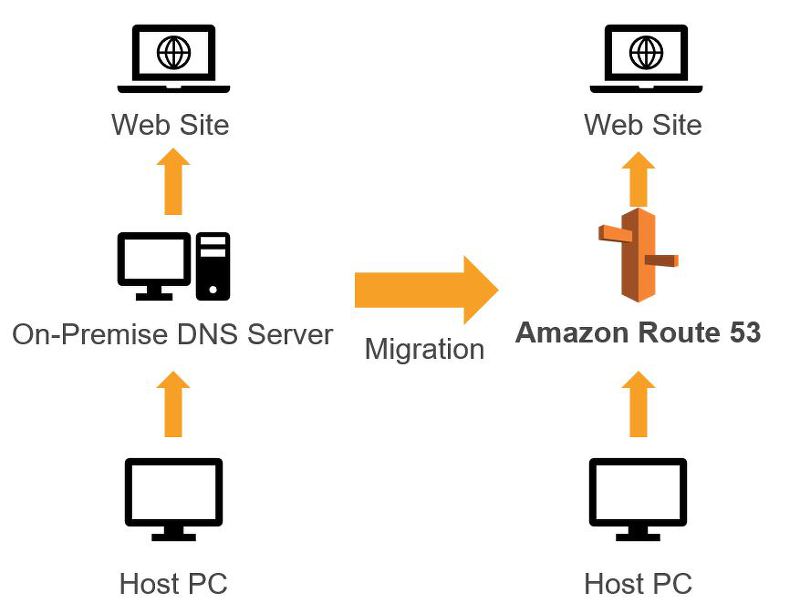
접속 경로 – 실행창 gpedit.msc 뒤

컴퓨터 구성 -> 관리 템플릿 -> 시스템 -> 자격 증명 위임

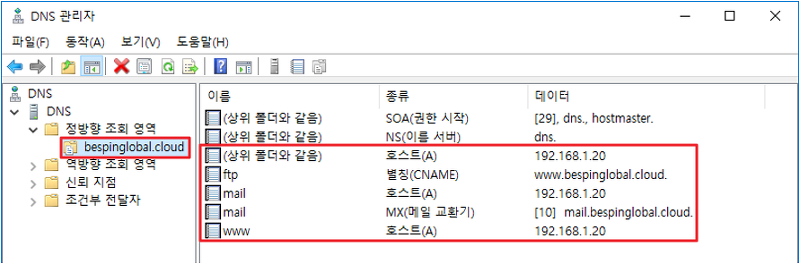
-> Oracle 수정 암호화

**Name Server Migration**

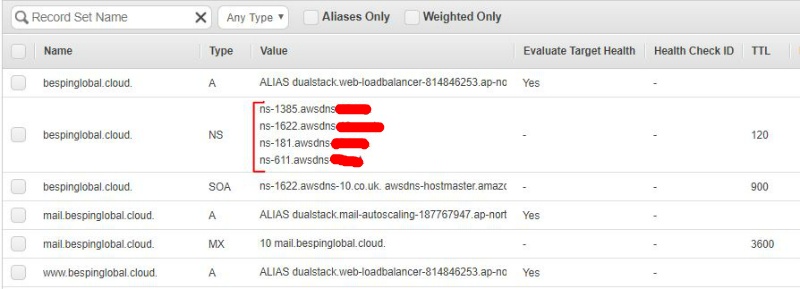
▷ On Premise의 Name Server를 AWS에서 제공하는 Name Server로 Migration 한 뒤 Route53에 레코드 값 입력.



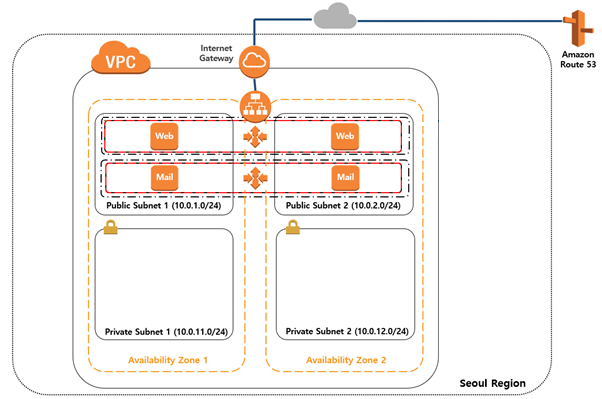
▷ On Premise Name Server 레코드



▷ AWS Route53 Name Server의 레코드는 위와 같이 구성한 뒤 도메인 호스팅 업체에서AWS의 Route53에서 제공하는 Name Server로 Name Server주소를 변경



**최종 Infra**

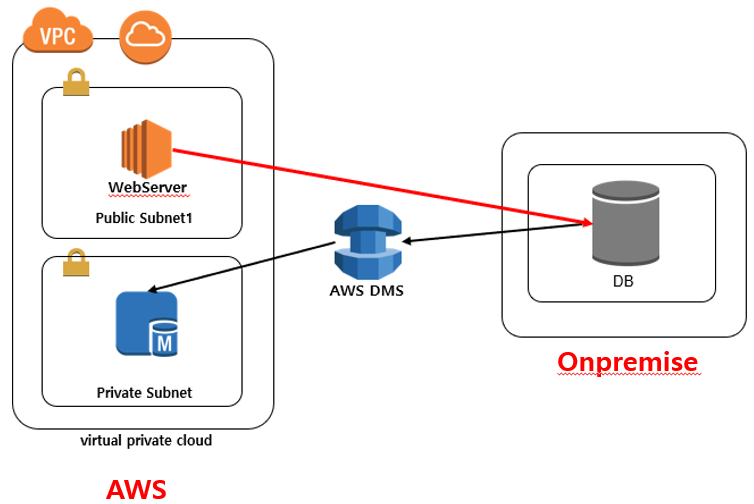


**7.4 Database Migration**

**Migration 계획**

**다운타임 최소화**

▷다운타임 최소화를 위해 On Premise의 DB 연결을 유지한 상태로 AWS의 RDS에 데이터를 이전



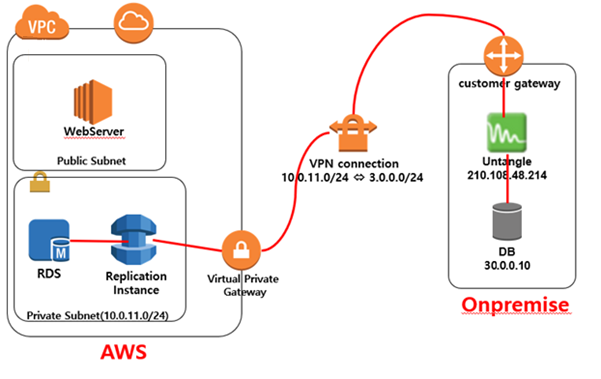
▷ DB의 데이터가 다 옮겨진 후 Web-Server가 RDS를 바라보도록 변경. 변경되는 순간에만 다운타임이 발생



**안전한 데이터 전송 (AWS VPN)**

▷ Untangle의 IpSec과 AWS의 관리형 VPN서비스를 이용하여 터널링을 통해

데이터를 전송

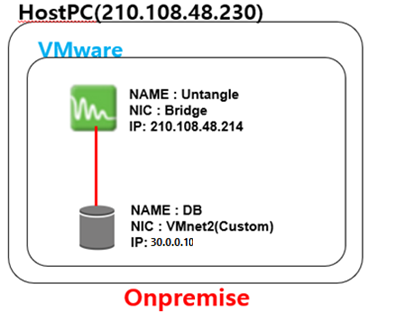


**On Premise Database**

**네트워크 구성**

▷ VPN 연결을 위해 Untangle의 NIC를 Bridge로 하여 공인 IP를 할당

DB는 Untangle의 내부 대역으로써 연결되며 VMnet2(Custom)인 30.0.0.10를 사용



**데이터 확인**

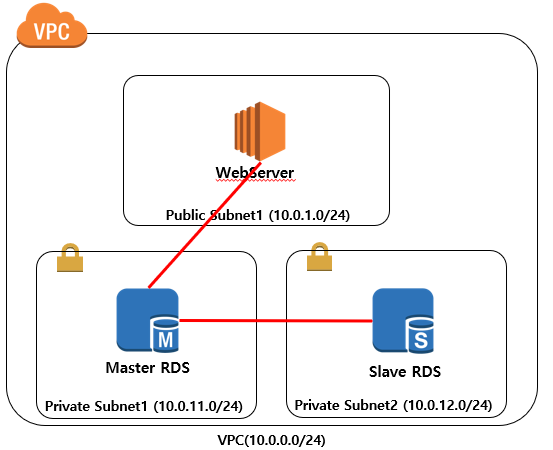
▷ On Premise의 Table들을 AWS로 Migration



**AWS**

**RDS**

▷ RDS는 VPC(10.0.0.0/16)의 Private Subnet인 10.0.11.0/24 대역에 생성.

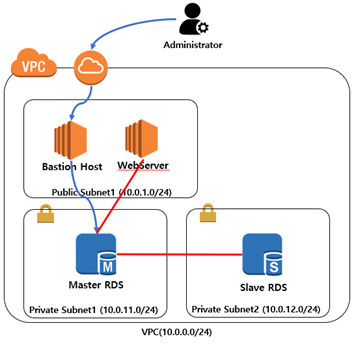


**Migration 결과**

**최종 인프라**

▷ AWS의 Private Subnet으로 옮겨진 RDS를 관리하기 위해 Administrator가

Bastion Host를 통해 접근할 수 있도록 추가



**RDS**

▷ Bastion Host를 통해 RDS에 접근하여 확인한 결과

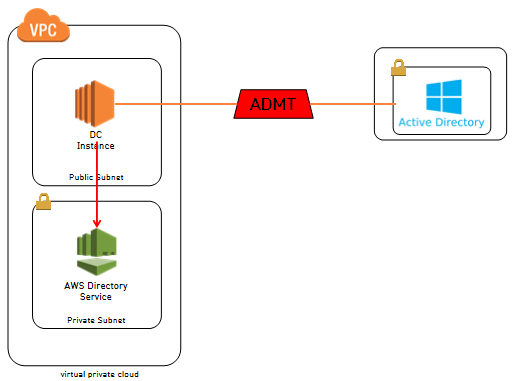


**7.5 Active Directory Migration**

**Migration 계획**

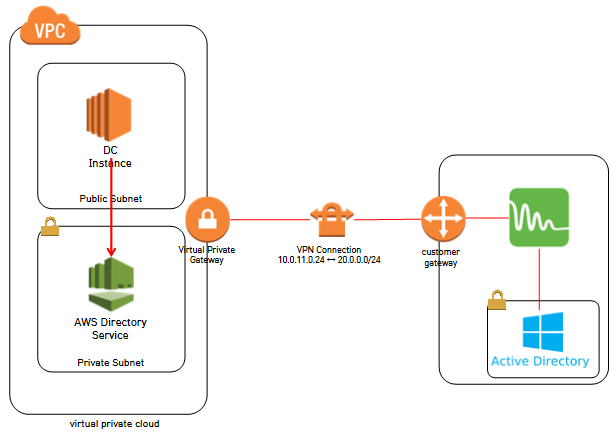
**계획 구성도**

▷ ADMT(Active Directory Migration Tool)을 사용하여 Forest Migration



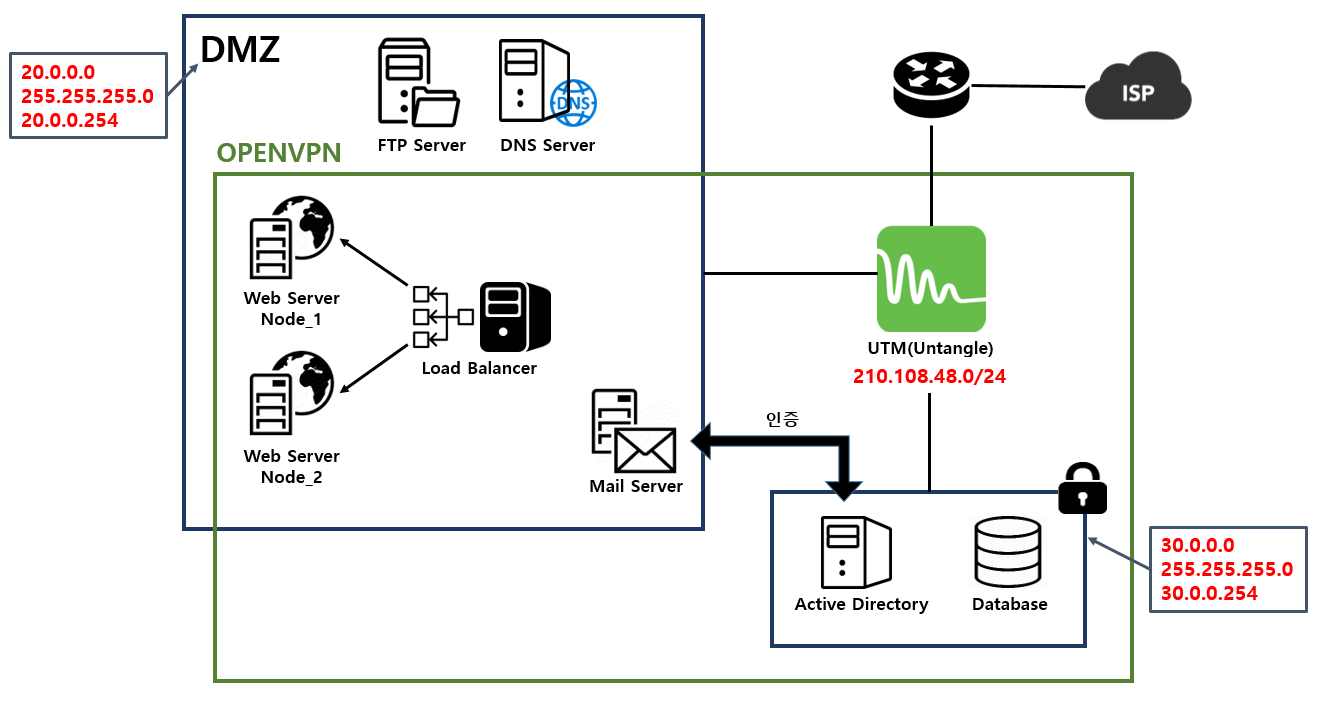
**AD Migration 간 VPN 연결**

▷ 안전한 데이터 전송을 위해 Untangle의 IpSec과 AWS의 관리형 VPN 서비스를 이용한 터널링을 통해 데이터를 전송

****

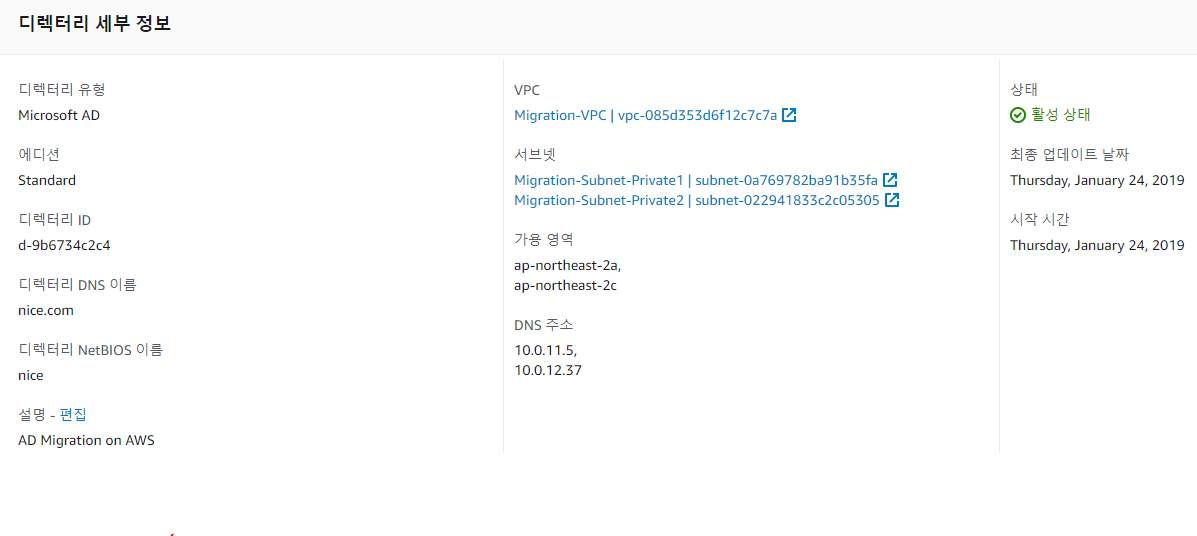
**On Premise AD 환경**

**On Premise AD 환경 구성도**



**AWS AD On Premise AD Migration**

**AWS Active Directory 생성**

****

**AWS AD 관리 인스턴스 생성**

관리형 인스턴스를 생성하는 이유

AWS Active Directory Service에서 제공하는 AD는 관리형 서비스로 Web콘솔로 제어

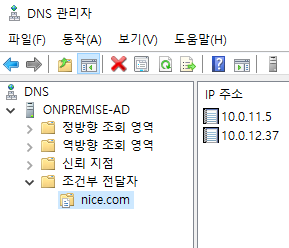
Web콘솔에서는 따로 Migration에 관한 항목들이 없기 때문에 제어 할 수 없음

그렇기 때문에 EC2 자체 관리형 인스턴스로 원격 접속하여 Migration

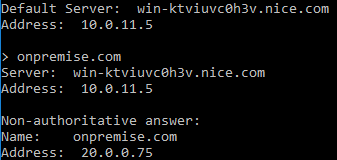
**AWS AD - On Premise AD간의 Trust 관계 형성**

서로 다른 AD Domain 사이에 Trust 관계를 형성하려면 양방향으로 AD 조회 가능해야 함

AWS AD - On Premise AD 의 DNS 조건부 전달자에 서로의 DNS를 추가

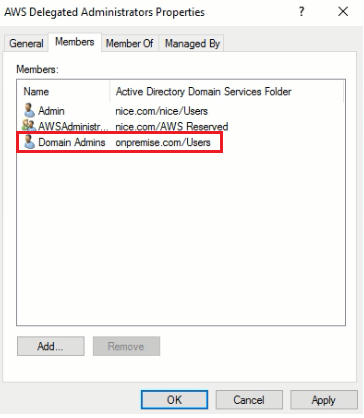
 

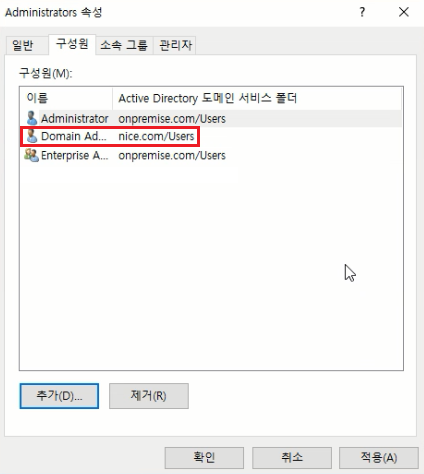
AWS DNS 조건 전달 부 설정-------------------------------------On Premise 조건 전달 부 설정

**ADMT(AD Migration Tool)을 활용하여 사용자 계정 Migration**

ADMT를 사용하기 위해서 관리자의 권한이 필요하므로 권한 설정

****

****

**ADMT 설치**

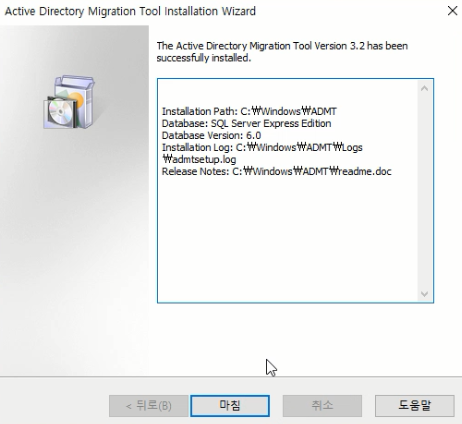
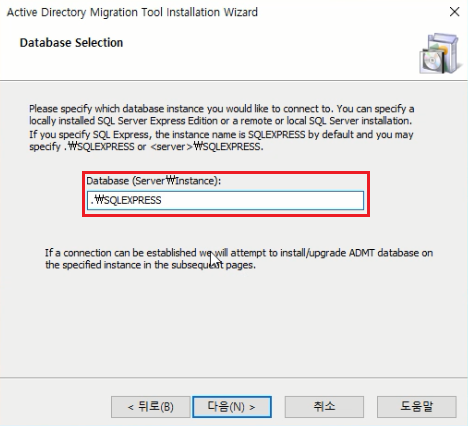
ADMT는 Source Domain에서 Target Domain으로 AD 자원을 복제하는 기능을 제공

ADMT를 사용하기 위해선 SQL Express로 내부 Database를 구축해야 하므로 ADMT 설치 후 Migration 시 Password Migration을 위해 ADMT Password Migration DLL을 설치해야 함

ADMT는 On Premise AD의 DC에 설치

※ ADMT 설치 시 유의사항

ADMT 설치 시 사용 데이터베이스를 SQL Express로 입력



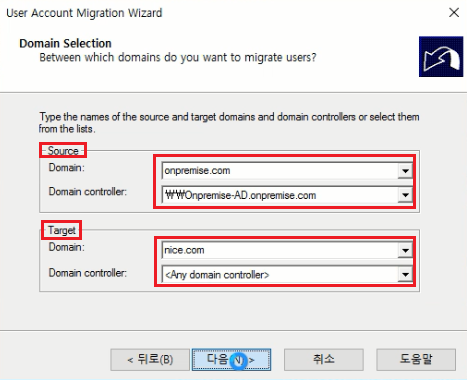
.※ ADMT Password Migration DLL 설치 시 유의 사항

admt key 명령어를 이용하여 암호키를 생성

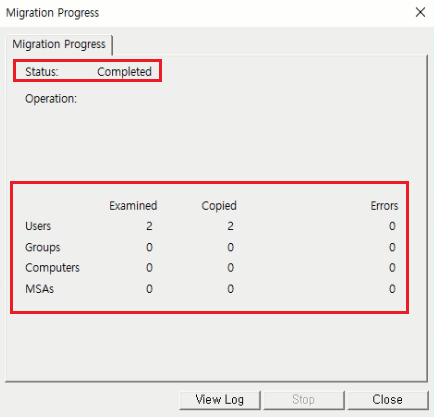


**ADMT 실행**

▷ Source의 Domain은 On Premise AD의 Domain으로 입력하고 Target의 Domain은 AWS AD의 Domain으로 입력

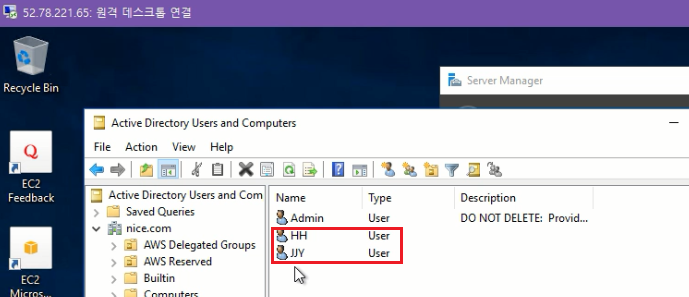


▷ User Migration을 진행



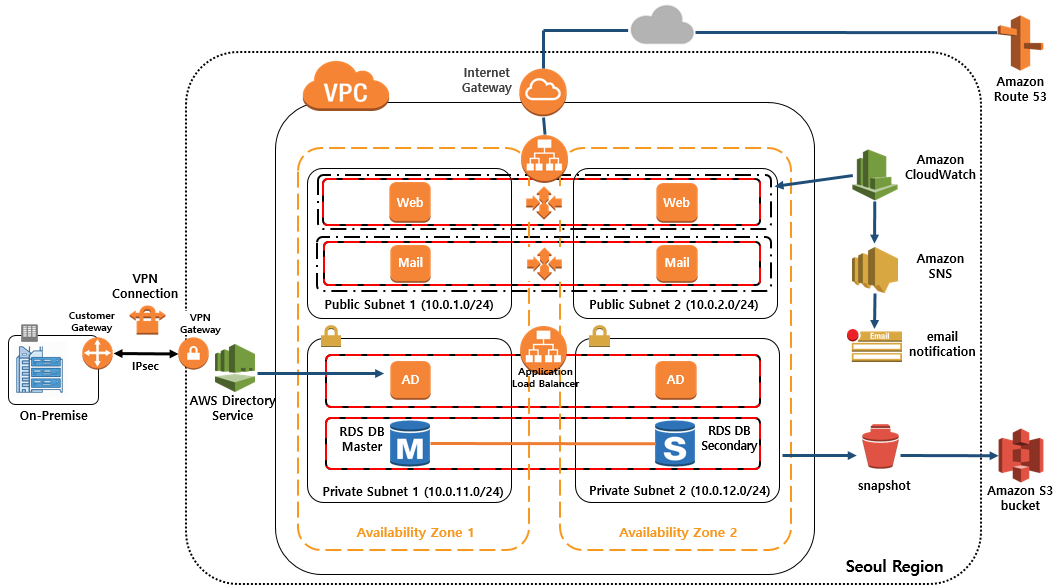
**Migration 결****과**

▷ Migration이 성공적으로 되었는지 확인



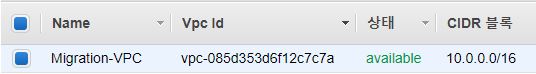
8. AWS

**8.1 구성도**

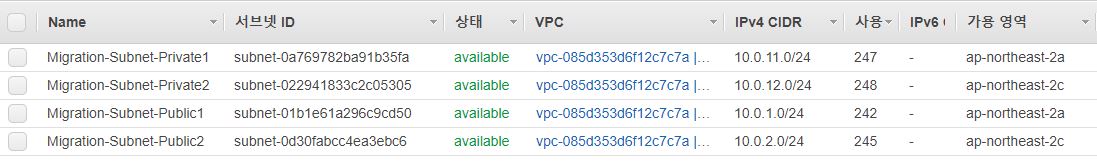


**8.2 VPC 구성**

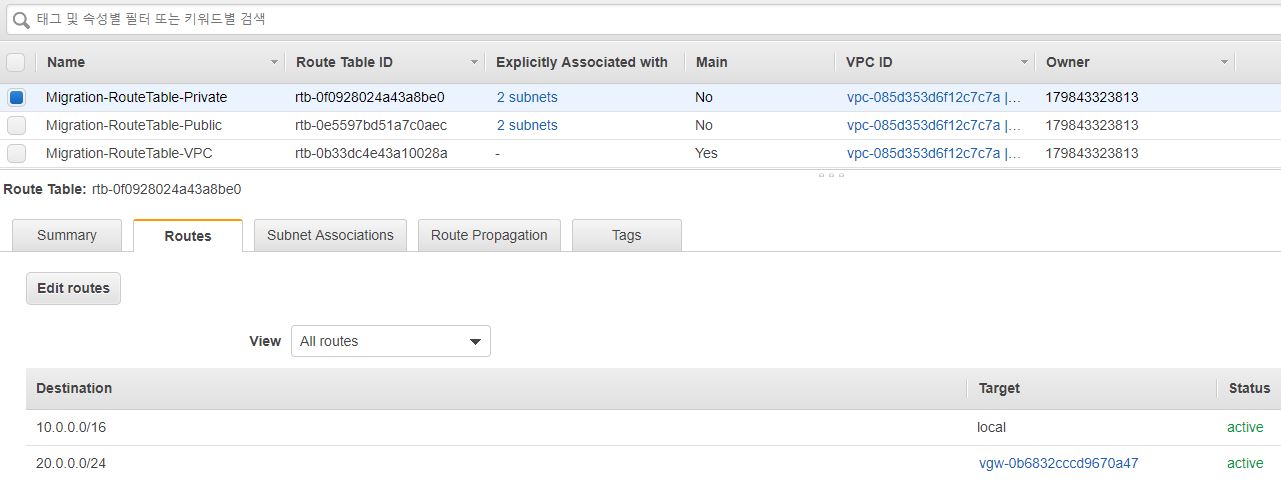
▷ 서울 리전에 사용할 VPC대역을 생성



▷ 두 개의 가용 영역에 Public, Private Subnet을 하나씩 생성



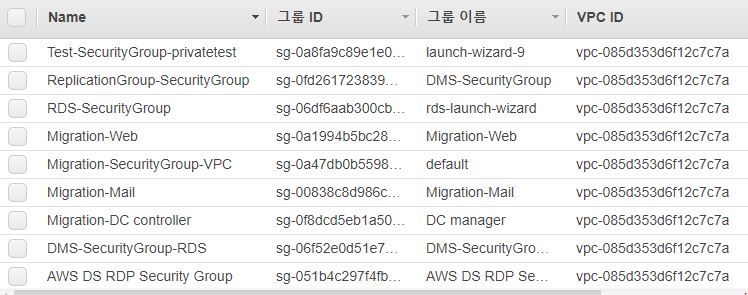
▷ Private Subnet의 Routing Table은 내부 통신을 위한 local 10.0.0.0/16과 On Premise에서 Migration을 위해 연결한 VPN의 VPG가 존재



▷ Public Subnet의 Routing Table은 외부 통신을 위한 igw(Internet Gateway)를 추가적으로 설정



▷ 보안 그룹은 각각의 서비스 별로 다르게 구성하여서 보안 수준을 향상



**8.3 Auto-Scaling 및 ELB구성**

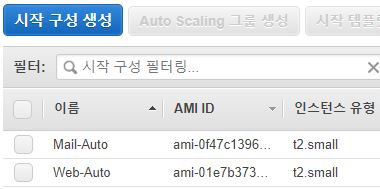
Auto-Scaling 기능을 사용하여 임계치 이상의 트래픽이 측정되면 자동으로 Scale Out하여 서버의 부하를 줄이고 ELB를 사용하여 Auto-Scaling 그룹의 인스턴스들로 트래픽을 분산하여 분배할 수 있도록 하여 고가용성을 제공

▷ Load Balancing을 할 대상 그룹을 생성



▷ Scale Out될 때 생성될 Image를 생성.

SMS로 Migration한 Server의 AMI를 사용하여 Image를 생성

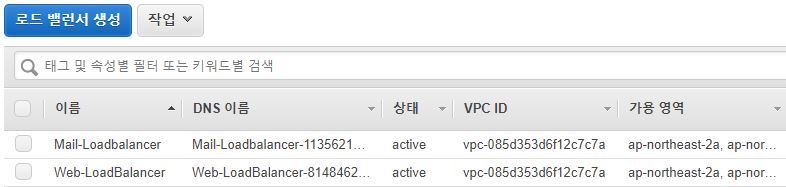


▷ Auto-Scaling Group을 생성

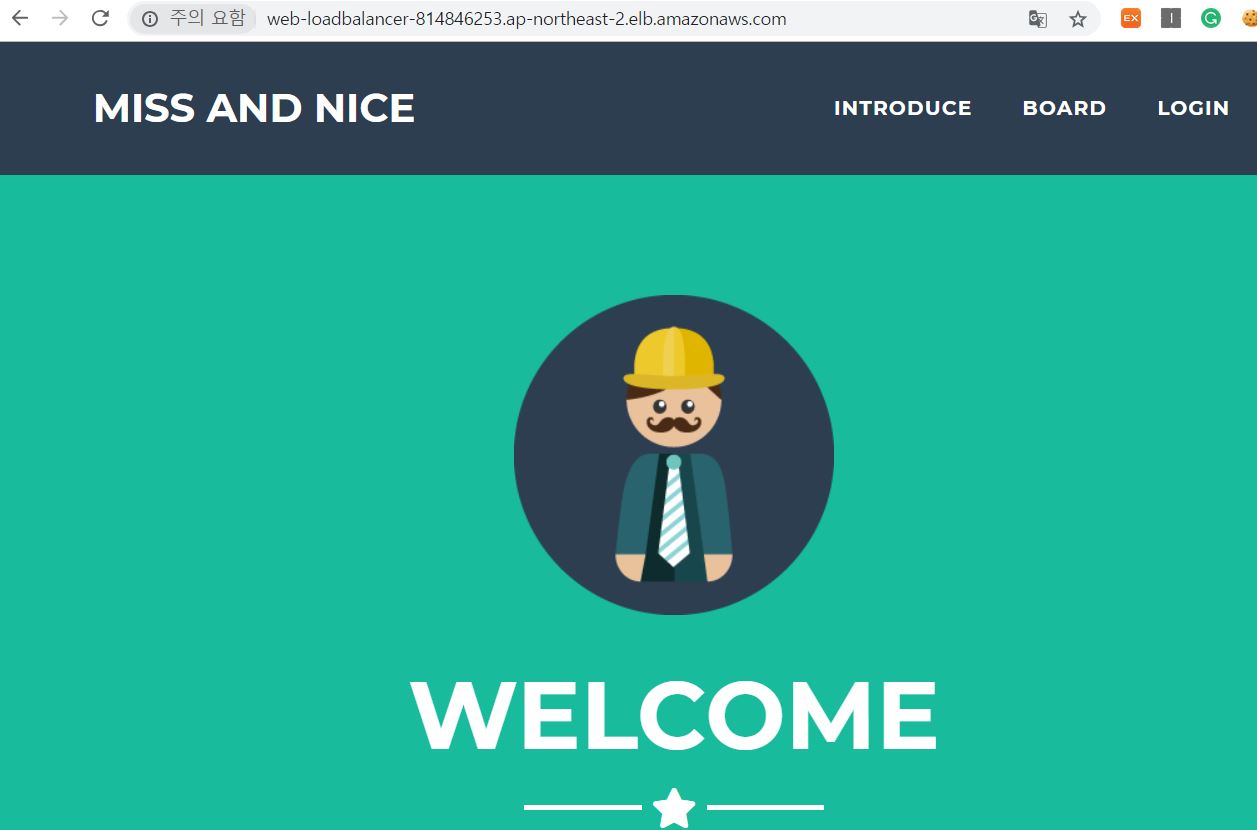


▷ Web, Mail의 트래픽을 분산시켜줄 ELB를 생성

또한 ELB를 사용하여 인스턴스들의 Health Check 가능



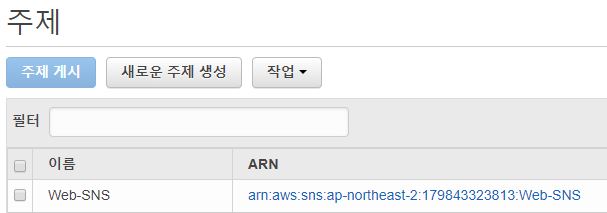
▷ 생성한 ELB의 DNS 주소를 사용하여 Web에 접근을 하였을 때 Web-Server로 정상적으로 접근



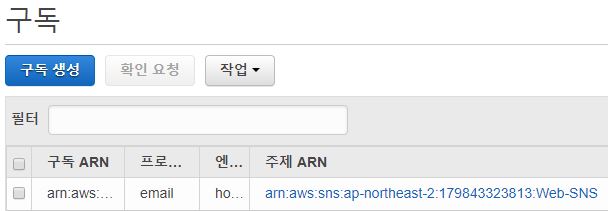
**8.4 CloudWatch를 이용한 SNS 구성**

CloudWatch로 ELB의 특정 임계치를 넘는 트래픽이 발생하면 SNS로 구독자들에게 알림 메일을 보내도록 설정.

▷ Mail을 보내기 위한 주제를 생성

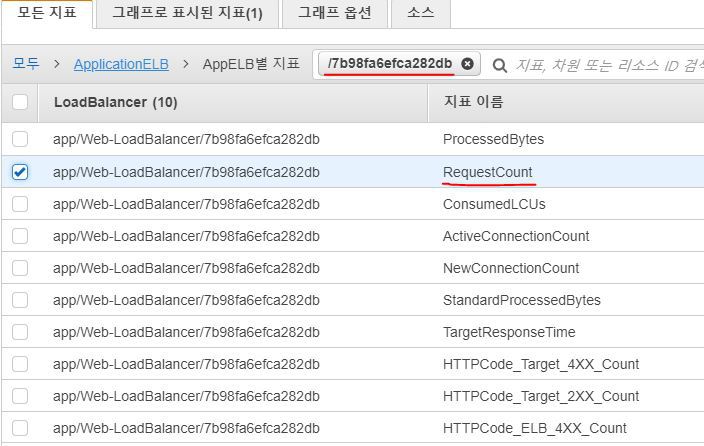


▷ 주제에 대한 Mail을 받기 위한 구독을 생성

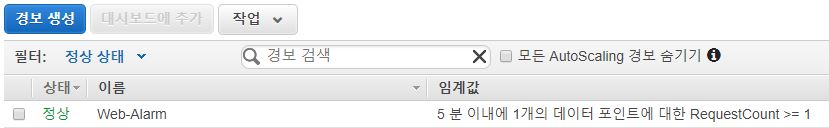


▷ 생성한 ELB의 지표를 선택

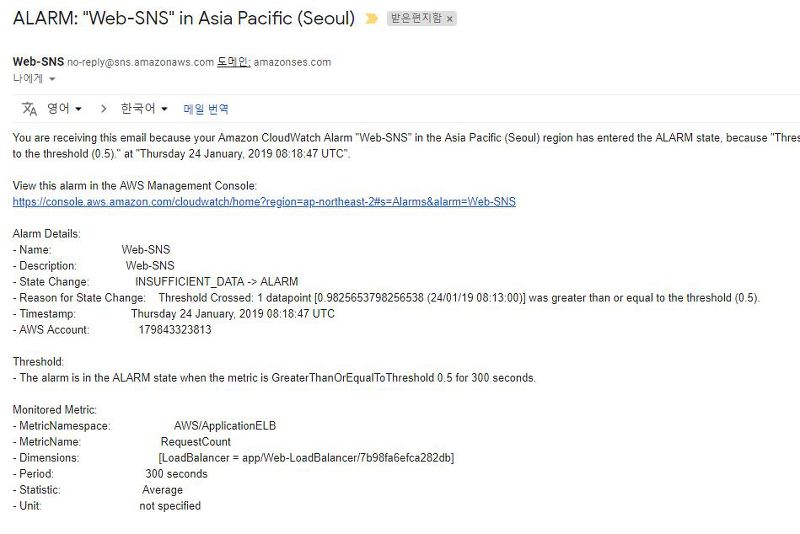
IPv4를 통해 처리된 요청의 수를 지표로 하는 RequestCount를 선택



▷ 경보(Alarm)을 생성

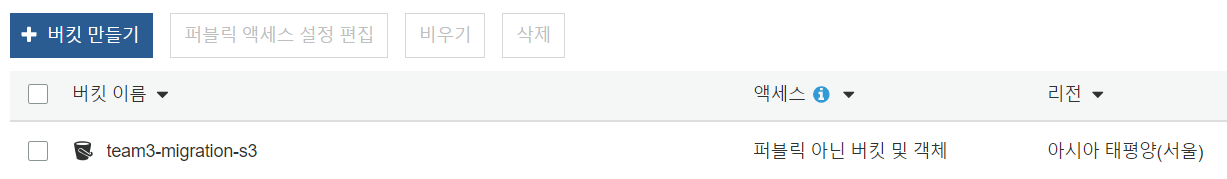


▷ 그 후 트래픽을 향상시켜 구독자에게 메일이 잘 가는지 확인



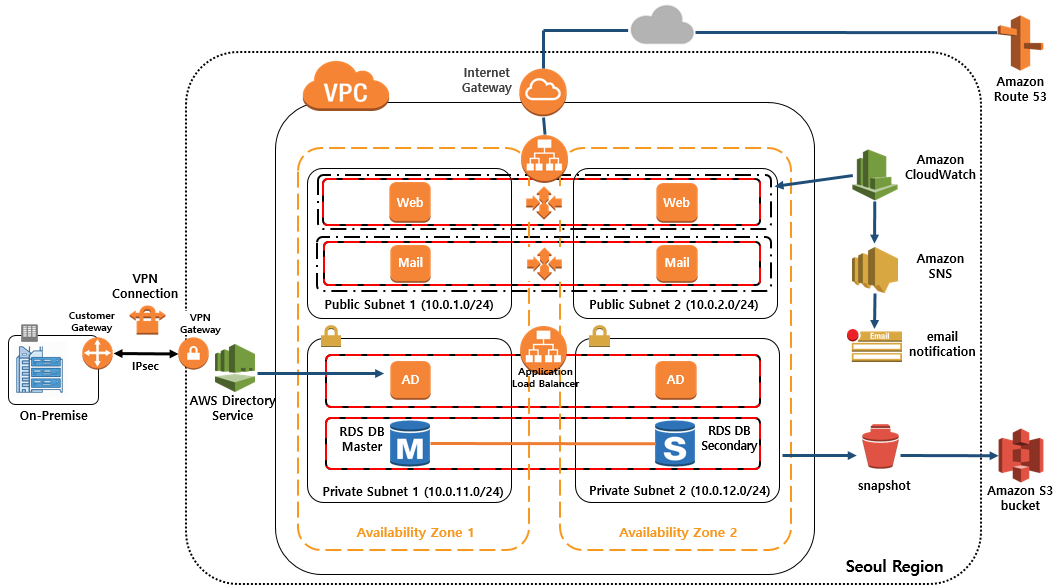
**8.5 Snapshot 저장을 위한 S3생성**

EC2 Instance의 스냅샷을 저장하기위한 별도의 S3 버킷을 생성



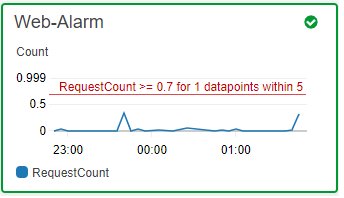
9. 최종 Infra

**9.1 최종Infra 구성도**



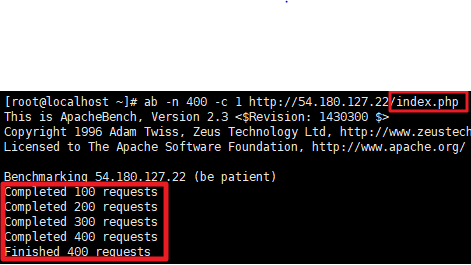
**9.2 부하테스트**

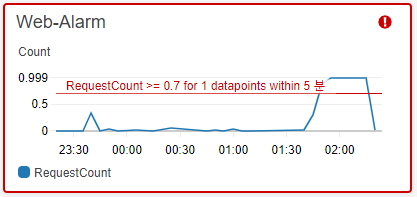
AWS CloudWatch를 사용하여 부하 시, 설정해둔 임계 값을 넘어서면 지정해둔 관리자의 E-mail로 알림이 가고 Auto Scaling 기능을 통해 확장



<평상시의 CloudWatch 대시보드의 모습>

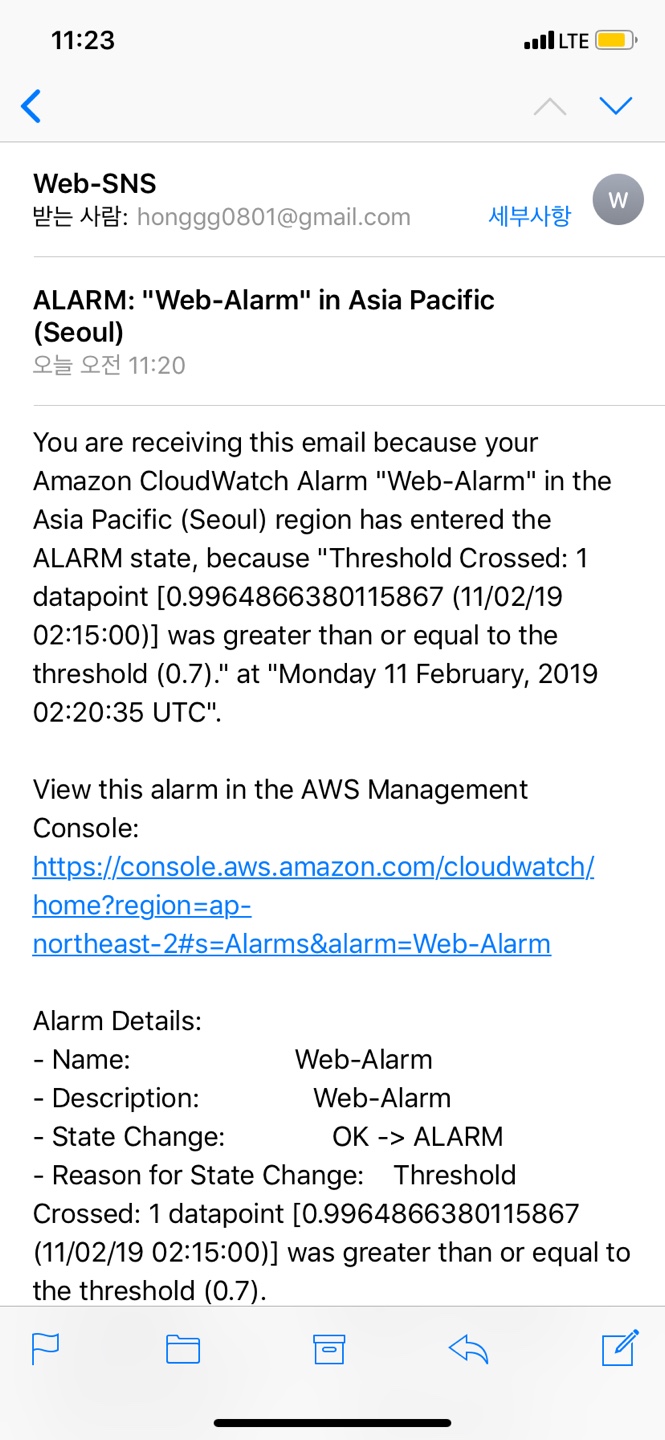
▷ ab 명령어를 통해 부하테스트 진행



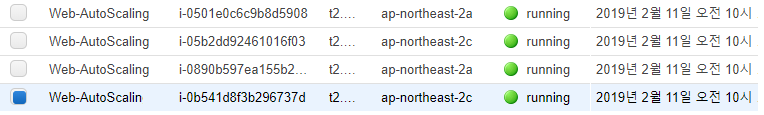


<웹 서버 부하 시 CloudWatch 대시보드>

▷ CloudWatch 대시보드에서 부하에 대한 경고 메시지가 나오면서 지정해둔 관리자의 메일로 알람 전송



▷ Auto Scaling group설정으로 자동으로 확장된 EC2를 확인



**9.3 무결성 검증**

