Node.js로 간단한 서버 만들기 with AWS

클라우드 서비스 | 클라우드 컴퓨팅 ×

완전함 https://aws.amazon.com/ko/

메뉴 amazon web services 문의처 제품 솔루션 요금 소프트웨어 지원 고객 파트너 엔터프라이즈 더 보기 한국어 내 계정 가입

AWS 솔루션 및 제품 탐색

Amazon EC2, S3 등 사용 – 1년 동안 무료

AWS 프리 티어에 대해 자세히 알아보기 »

무료로 AWS 시작하기

무료 기능 생성

Amazon S3
스토리지 5GB, Get 요청 20,000건 및 Put 요청 2,000건

AWS 프리 티어 세부 정보 보기 »

시작하기
몇 분 내에 AWS 사용을 시작하는 방법 알아보기

AWS 코리아 행사 및 웨비나
한국에서 열리는 다양한 클라우드 행사들을 살펴보세요.

AWS 기반 윈도우 서버 운영
백서를 다운받으시고 라이브 웨비나에 참여하세요!

AWS 기반 프로젝트
AWS 단계별 안내서를 사용하여 프로젝트 구축

AWS 제품 살펴보기

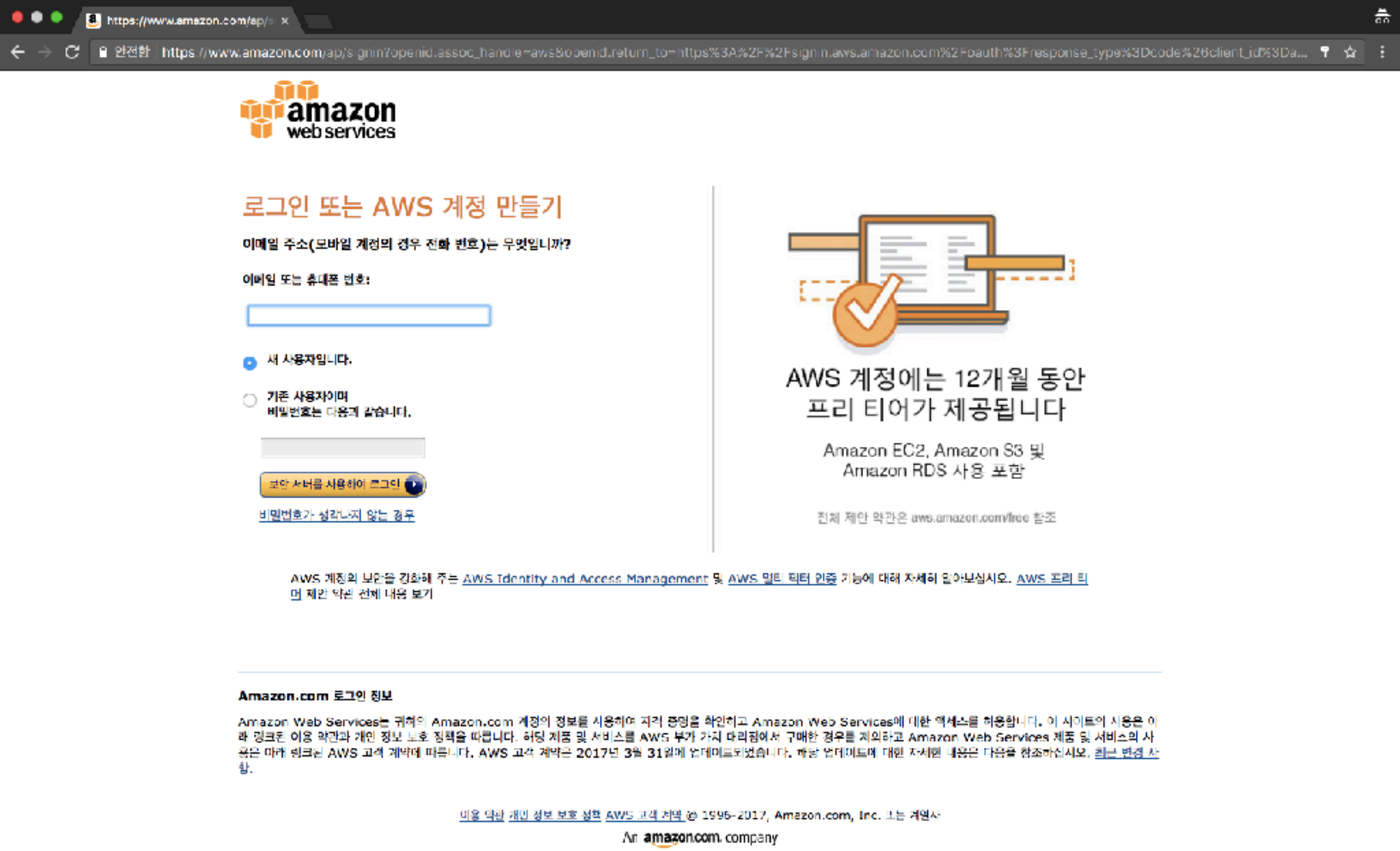
안전한 연결 설정 중...

데이터베이스

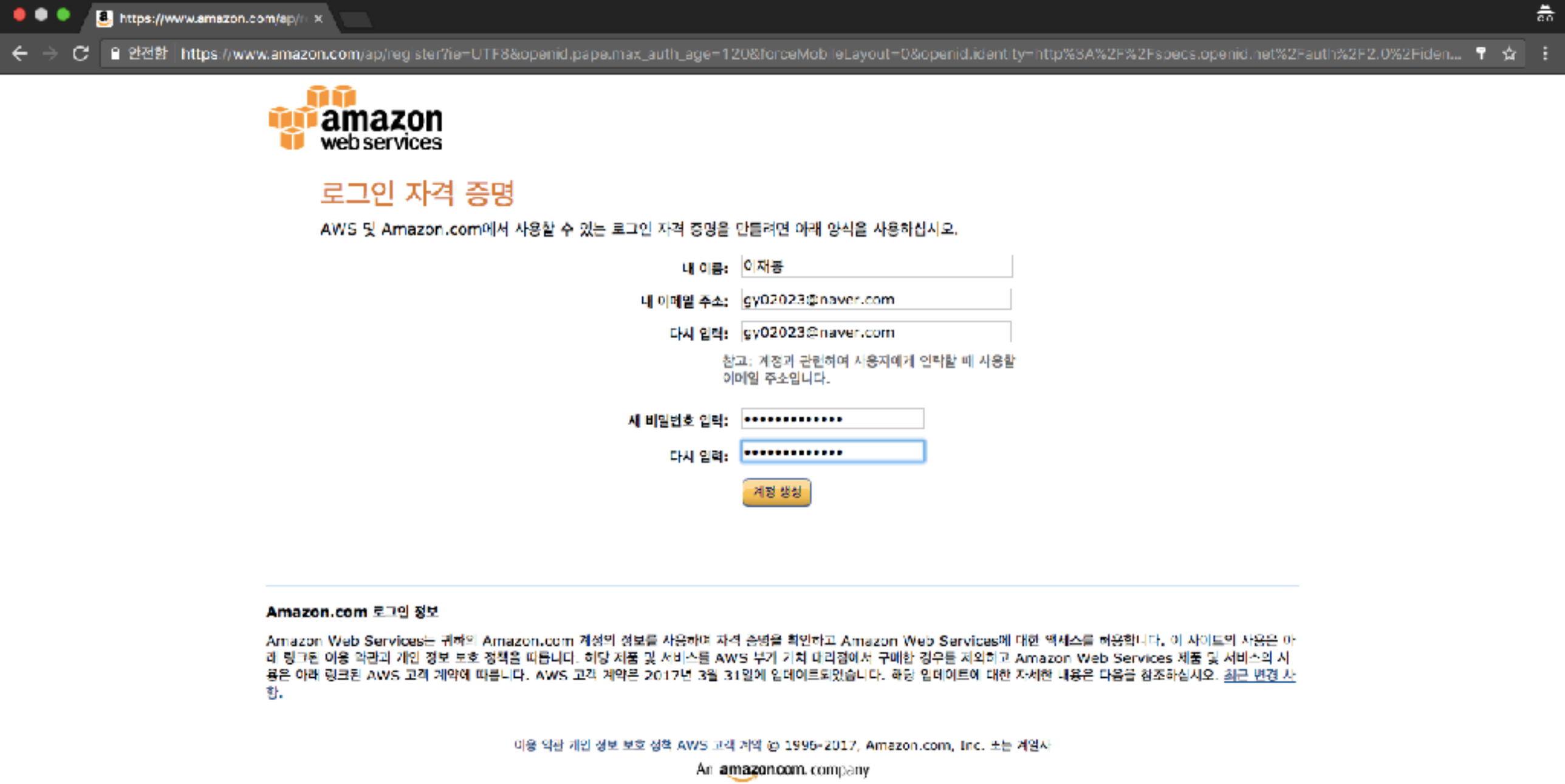
마이그레이션

네트워킹 및 콘텐츠 전송

https://aws.amazon.com/ko/

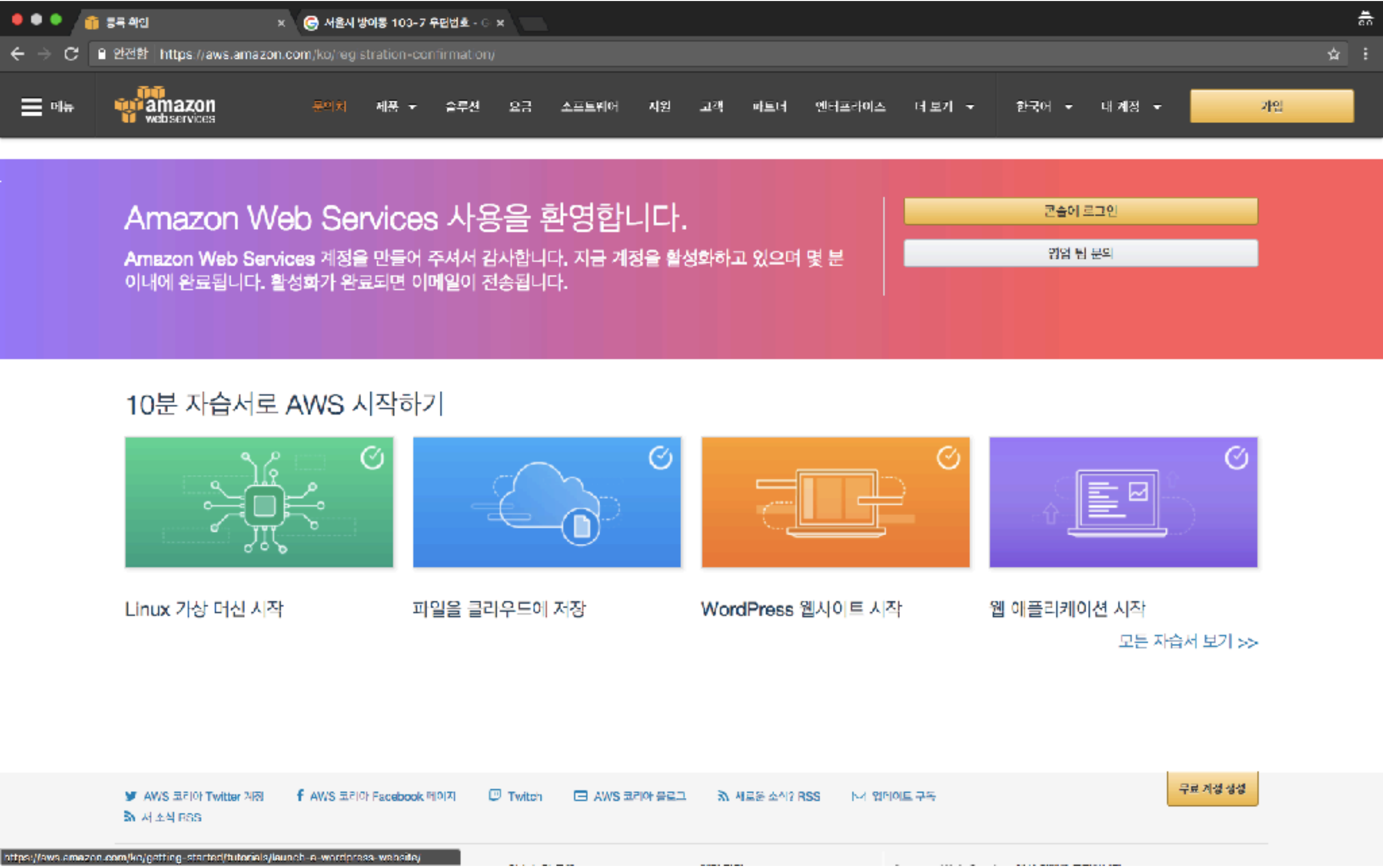

amazon
web services
로그인 또는 AWS 계정 만들기
이메일 주소(모바일 계정의 경우 전화 번호)는 무엇입니까?
이메일 또는 휴대폰 번호:
새 사용자입니다.
기존 사용자이며
비밀번호는 다음과 같습니다.
보안 서버를 사용하여 로그인
비밀번호가 설정되지 않은 경우
AWS 계정에는 12개월 동안
프리 티어가 제공됩니다
Amazon EC2, Amazon S3 및
Amazon RDS 사용 포함
전체 제안 약관은 aws.amazon.com/free 참조
AWS 계정의 보안을 강화해 주는 AWS Identity and Access Management 및 AWS 멀티 팩터 인증 기능에 대해 자세히 알아보십시오. AWS 프리 티
어 제안 약관 전체 내용 보기
Amazon.com 로그인 정보
Amazon Web Services는 귀하의 Amazon.com 계정의 정보를 사용하여 자격 증명을 확인하고 Amazon Web Services에 대한 액세스를 허용합니다. 이 사이트의 사용은 아
래 링크된 이용 약관과 개인 정보 보호 정책을 따릅니다. 허딩 제품 및 서비스를 AWS 부가 가치 리셀러에서 구매한 경우를 제외하고 Amazon Web Services 제품 및 서비스의 사
용은 아래 링크된 AWS 고객 계약에 따릅니다. AWS 고객 계약은 2017년 3월 31일에 업데이트되었습니다. 해당 업데이트에 대한 자세한 내용은 다음을 참조하십시오. 최근 변경 사
항.
이용 약관 개인 정보 보호 정책 AWS 고객 계약 © 1996-2017, Amazon.com, Inc. 또는 계열사
An amazon.com company

새 사용자로 계정 생성

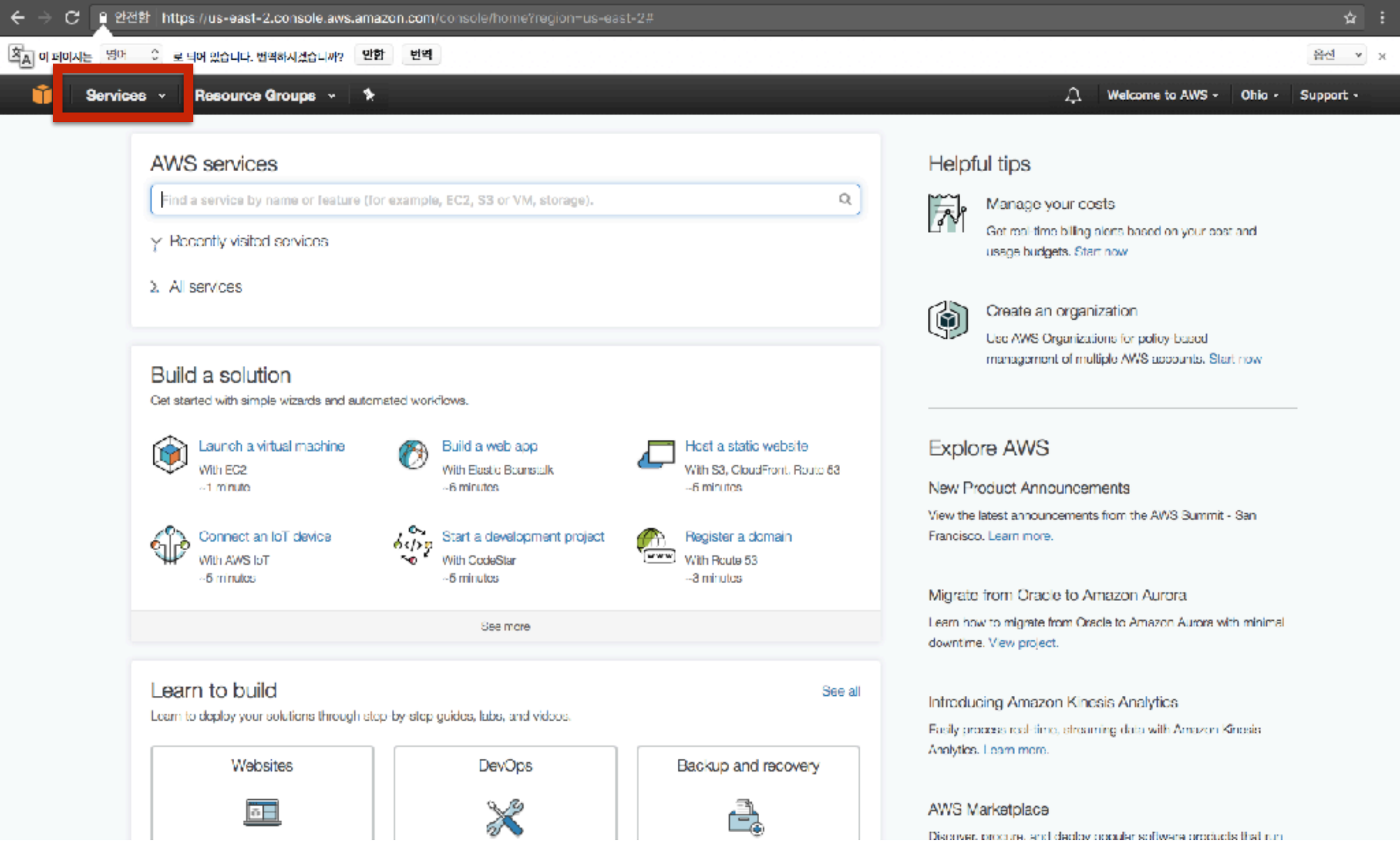


https://aws.amazon.com/ko/

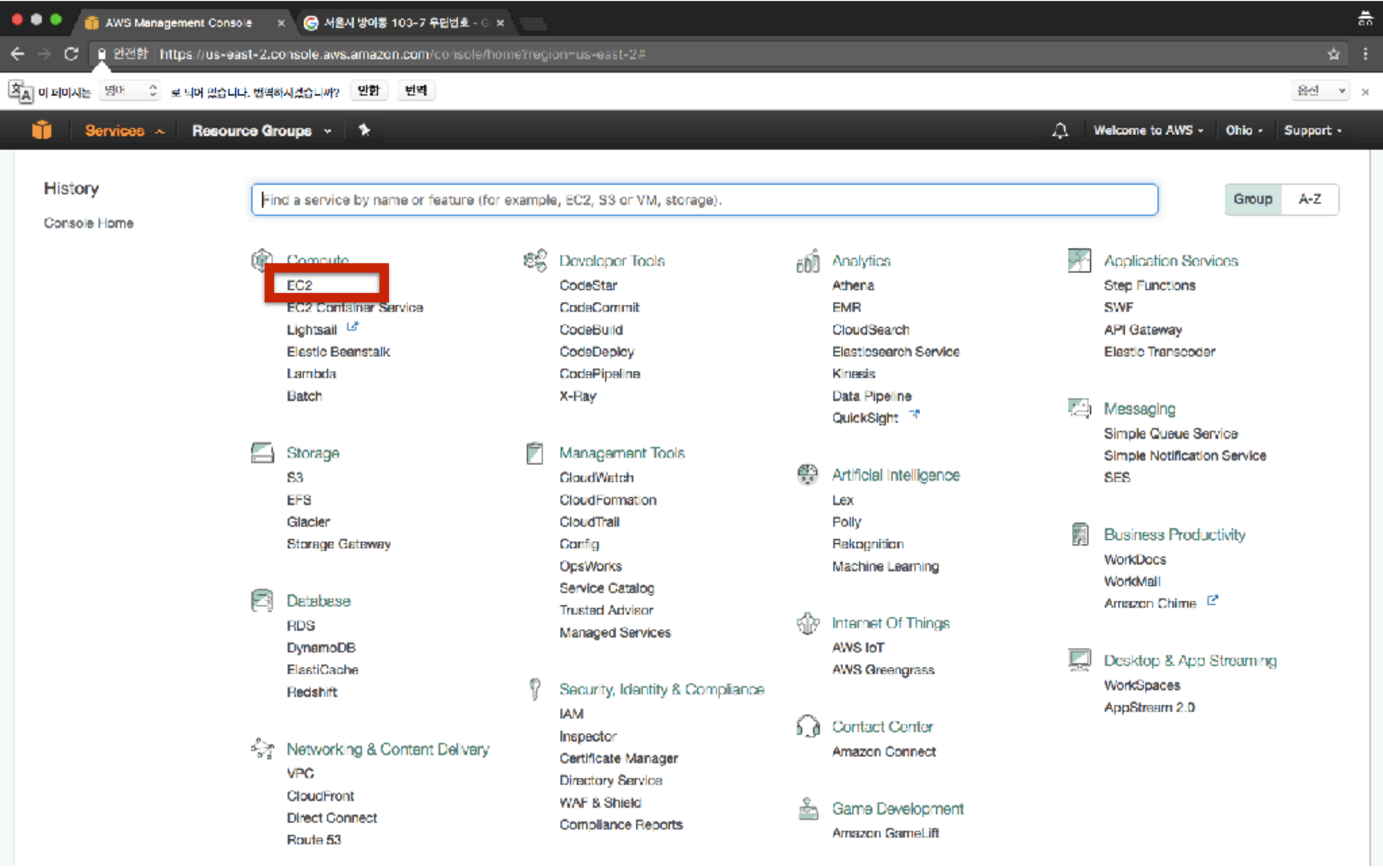
개인 계정으로 생성 (카드생성 과정중 1달라쯤 빠졌다가 들어옴)



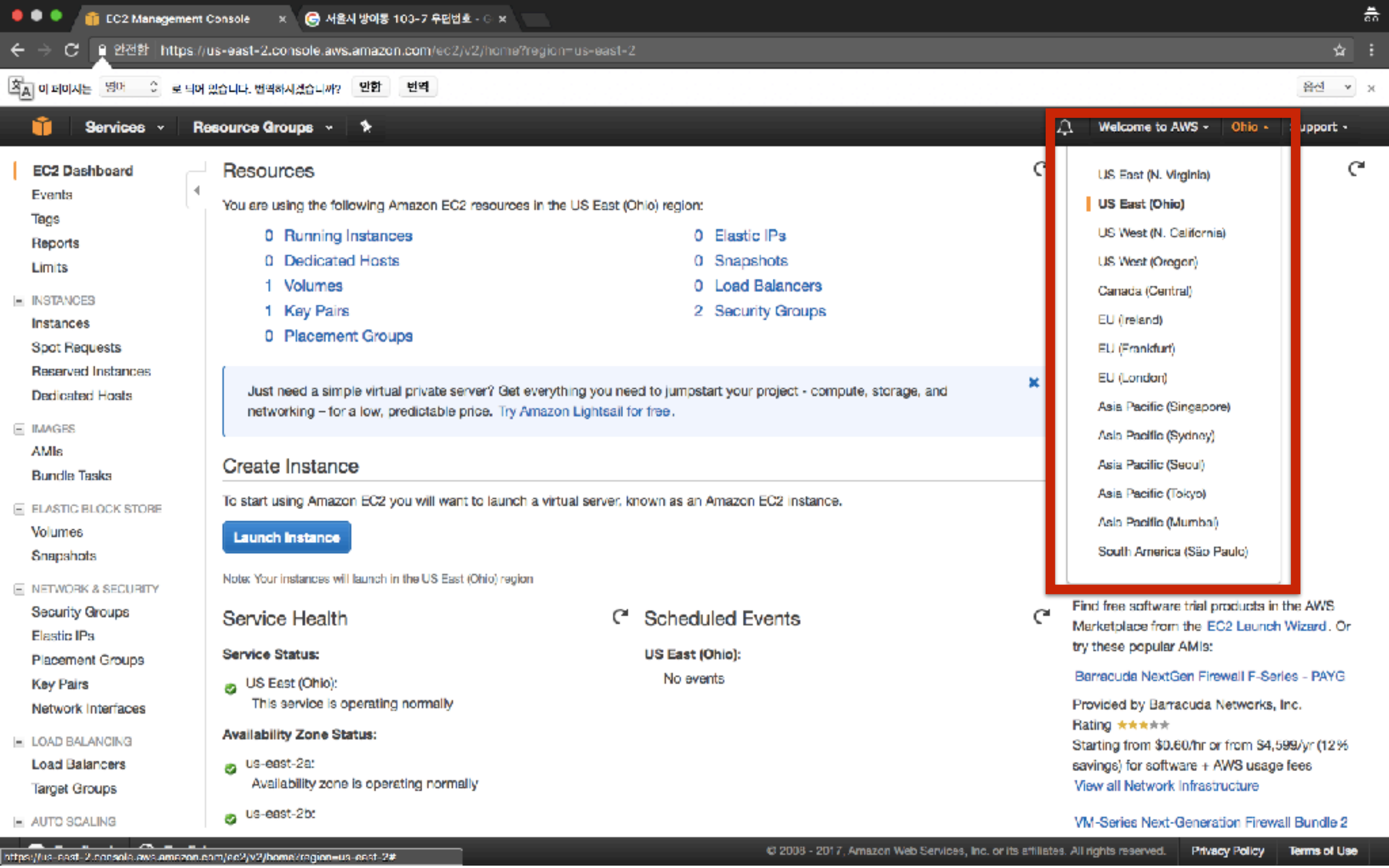
콘솔에 로그인으로 로그인



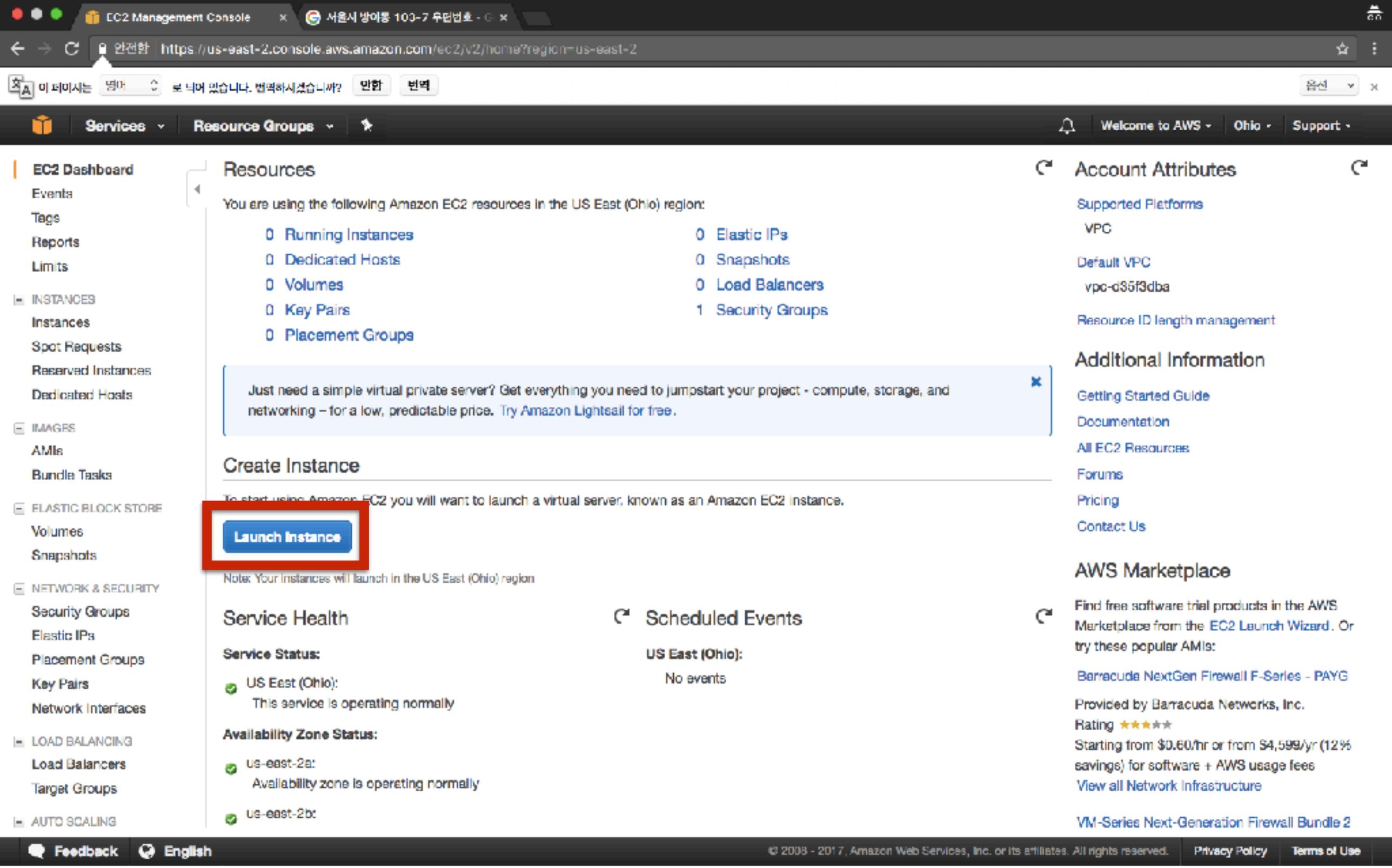
Services의 EC2 선택



Services의 EC2 선택



지역 선택 - Asia Pacific(Seoul)



Launch Instance 선택

EC2 Management Console | 서울시 방이동 103-7 우편번호 - G

완전함 https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

이 페이지는 영어 로 되어 있습니다. 번역하시겠습니까? 번역함 번역 안함 옵션

Services | Resource Groups | Welcome to AWS | Ohio | Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

1 to 30 of 30 AMIs

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux AMI 2017.03.0 (HVM), SSD Volume Type - ami-4191b524

Amazon Linux — Free tier eligible

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

Select

64-bit

Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-0932606c

Red Hat — Free tier eligible

Red Hat Enterprise Linux version 7.3 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

Select

64-bit

SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-61a7fd04

SUSE Linux — Free tier eligible

SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

Select

64-bit

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-618fab04

Free tier eligible

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm

Select

64-bit

Feedback | English

Privacy Policy | Terms of Use

Amazon Linux 선택

EC2 Management Console | 서울시 방이동 103-7 우편번호 - G

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

이 페이지는 영어 로 되어 있습니다. 번역하시겠습니까? 번역 안함

Services | Resource Groups | Welcome to AWS | Ohio | Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

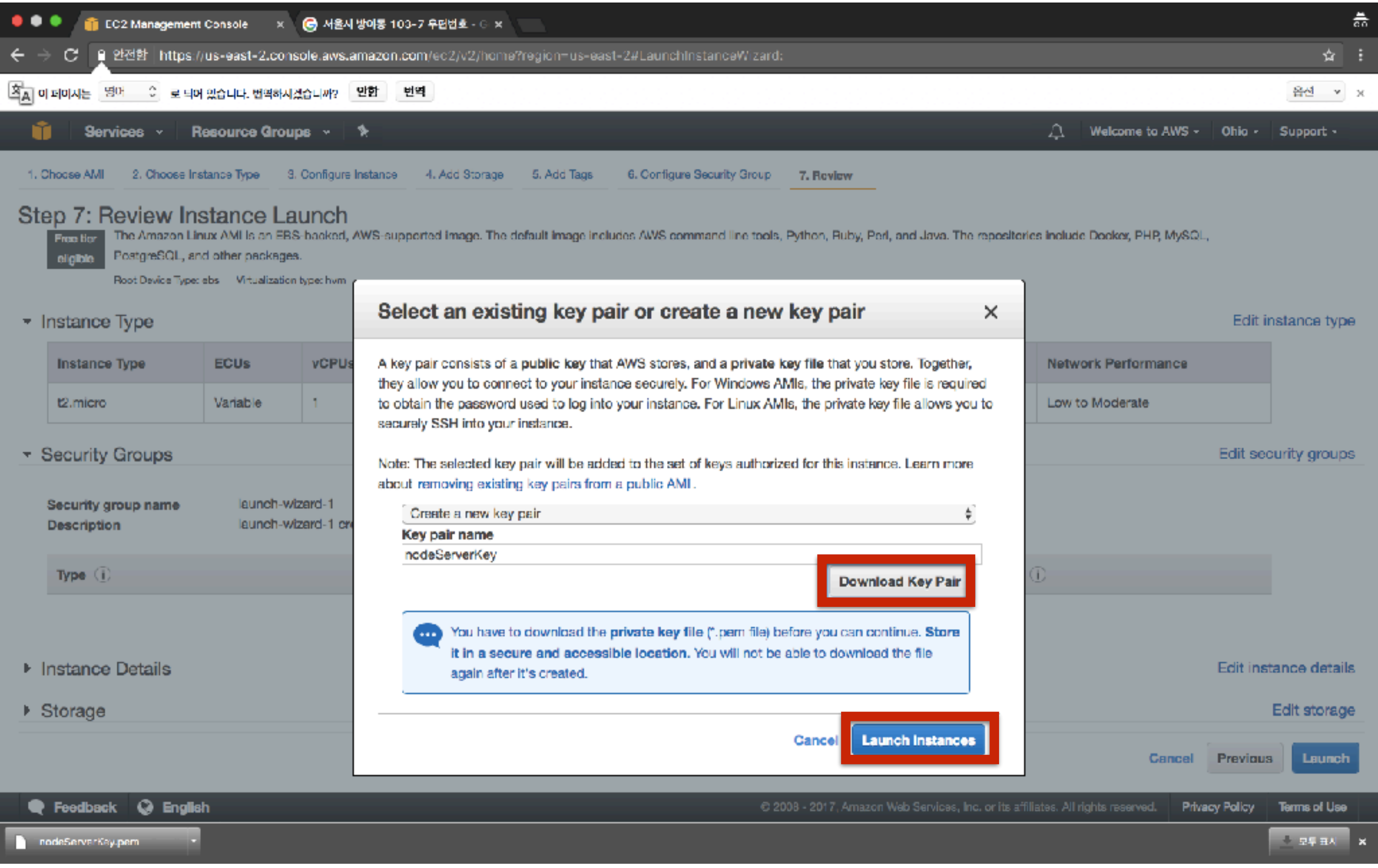
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes
☐	General purpose	m4.xlarge	4	16	EBS only	Yes	High	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details

Feedback | English | © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

t2.micro로 인스턴스 생성



key pair 새로 생성하여 사용 (create a new key pair)

EC2 Management Console | 서울시 방이동 103-7 우편번호 - G

https://ap-northeast-2.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-2#Instances:sort=instanceId

이 페이지는 영어 로 되어 있습니다. 번역하시겠습니까? 번역함 번역 안함 옵션

Services | Resource Groups | Welcome to AWS | Seoul | Support

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits

INSTANCES
- Instances
- Spot Requests
- Reserved Instances
- Dedicated Hosts

IMAGES
- AMIs
- Bundle Tasks

ELASTIC BLOCK STORE
- Volumes
- Snapshots

NETWORK & SECURITY
- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

LOAD BALANCING
- Load Balancers
- Target Groups

AUTO SCALING

Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword | 1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-038af8286c9f96a3a	t2.micro	ap-northeast-2a	running	Initializing	None	ec2-52-78-159-87.ap-no...	52.78.159.87

Instance: i-038af8286c9f96a3a Public DNS: ec2-52-78-159-87.ap-northeast-2.compute.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID	i-038af8286c9f96a3a	Public DNS (IPv4)	ec2-52-78-159-87.ap-northeast-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	52.78.159.87
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-9-245.ap-northeast-2.compute.internal

Feedback | English | © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use

Public IP로 접속

mac은 기본 터미널 사용
윈도우는 putty 사용

```
➜  Server chmod 400 nodeServerKey.pem
➜  Server ssh -i nodeServerKey.pem ec2-user@52.78.159.87

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
8 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-9-245 ~]$
```

ssh를 이용해서 접속 (mac의 경우 nodeServerKey.pem에 권한을 줘야함)

ssh -i nodeServerKey.pem ec2-user@52.78.159.87

로그인 관련 AWS 레퍼런스

http://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html

리눅스 명령어 모음 블로그

http://www.emh.co.kr/content.pl?linux_basic_commands

Nodejs 설치

Run as root on RHEL, CentOS or Fedora, for Node.js v6 LTS:

```
curl --silent --location https://rpm.nodesource.com/setup_6.x | bash -
```

Then install, as root:

```
yum -y install nodejs
```

curl --silent --location https://rpm.nodesource.com/setup_6.x | bash -

yum -y install nodejs

npm init

Package.json

1. root@ip-172-31-9-245:/home/ec2-user/study/study-prepare (ssh)

```
{
  "name": "study-prepare",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
~
~
~
~
~
~
~
~
~
~
~
~
~
"package.json" 11L, 209C                                    1,1          모두
```

```
npm install -save express
npm install -save body-parser
```

pacakge.json

1. root@ip-172-31-9-245:/home/ec2-user/study/study-prepare (ssh)

```
{
  "name": "study-prepare",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.17.2",
    "express": "^4.15.3"
  }
}
~
~
~
~
~
~
~
~
~
~
~
~
"package.json" 15L, 289C                                1,1           모두
```

vim index.js

```
var http = require('http');
var express = require('express');
var bodyParser = require('body-parser');
var app = express();

app.set('port', process.env.PORT || 3000);

app.get('/', function(req, res) {
    console.log('test');
    res.end('test');
});
```

```
app.use(function(req, res, next) {
    res.status(404).send('Not Found...' + '(' + req.originalUrl + ')');
});

app.use(function(err, req, res) {
    res.status(err.status || 500);
    res.send('Error occured...' + '(' + req.originalUrl + ')');
});

app.listen(app.get('port'));
```

npm install -g forever

forever start index.js

```
app.use(bodyParser.json({ limit: '10mb' }));
app.use(bodyParser.urlencoded({ extended: false }));

app.post('/', function(req, res){
    console.log(req.body);
    res.json(req.body);
});
```

Mysql install

```
sudo yum install mysql-server
```

```
sudo /sbin/service mysqld start
```

```
sudo /usr/bin/mysql_secure_installation
```

```
npm install -save mysql
```

https://www.w3schools.com/nodejs/nodejs_mysql.asp