

# **Fastcampus Data Science Extension SCHOOL**

**Cloud Service - AWS Lambda, EC2**

# Index

- Introduce to Cloud Computing
- AWS Lambda
- AWS EC2

# Cloud Computing

- 인터넷에 연결된 다른 컴퓨터로 연산을 하는 기술
- 접근성, 주문형 서비스 제공으로 경제적이고 효율적인 컴퓨팅 서비스 제공
- Amazon Web Service(Amazon), Google Cloud Platform(Google), Microsoft Azure(Microsoft), ..
- Virtual Machine, Cloud Storage, Database, Docker Engine 등 다양한 서비스 제공

AWS



- 2006년 Amazon이 시작한 클라우드 컴퓨팅 서비스
- 전세계 20개의 Region, 60개의 Zone을 제공

# Lambda



- <http://aws.amazon.com/lambda>

# AWS Lambda

- Event-driven, serverless computing platform
- Cloud function(GCP), Azure function(MS Azure)

## Lambda can..

- serverless API
- preprocessing
- authorizer
- auto-scaling
- ..

# Hello world with Lambda



# New lambda function

## 시작하기

Lambda 함수를 새로 작성하거나 미리 구성된 예제 중 하나를 선택합니다.

**함수 만들기**

# New lambda function

## 함수 생성

### 새로 작성



간단한 "hello world" 예제로 시작합니다.



### 블루프린트



사전 구성된 템플릿을 Lambda 함수의 시작점으로 선택합니다.



# New lambda function

hello-world-python3



A starter AWS Lambda function.

python3.6

# New lambda function

## 기본 정보 [정보](#)

이름

hello-lambda

역할

함수의 권한을 정의합니다. 생성 후 몇 분 동안 새로운 역할을 사용하지 못할 수 있습니다. Lambda 실행 역할에 대해 [자세히 알아보기](#)

1개 이상의 템플릿에서 새로운 역할을 생성합니다. ▼

선택한 정책 템플릿에서 Lambda가 자동으로 권한이 있는 역할을 생성합니다. 기본 Lambda 권한(예: Amazon CloudWatch에 로깅)이 자동으로 추가됩니다. 함수가 VPC에 액세스하면 필요한 권한도 추가됩니다.

역할 이름

새 역할의 이름을 입력합니다.

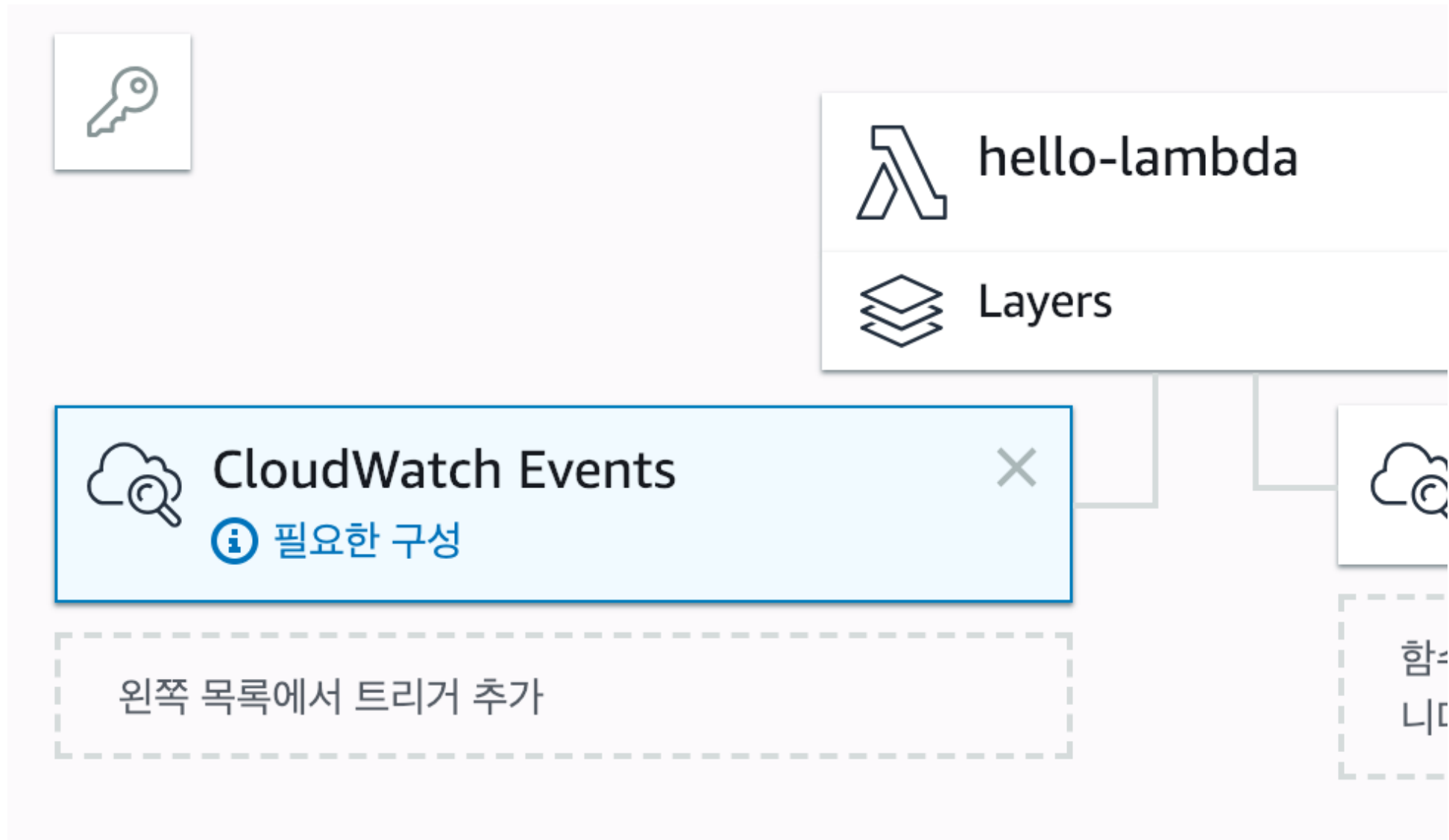
lambda-test

 이 새로운 역할이 현재 함수의 범위로 지정됩니다. 다른 함수와 함께 이를 사용할 수 있도록 IAM 콘솔에서 수정할 수 있습니다.

정책 템플릿

정책 템플릿을 하나 이상 선택합니다. 함수가 생성되기 전에 역할이 생성됩니다. 각 정책 템플릿이 사용자의 역할에 추가할 권한에 대해 [자세히 알아보기](#)

# New lambda function



# 트리거 구성

## 규칙

기존의 규칙을 선택하거나 새로운 규칙을 생성합니다.

새 규칙 생성



새 규칙 선택 또는 생성

## 규칙 이름\*

규칙을 고유하게 식별하려면 이름을 입력합니다.

cron-everymin

## 규칙 설명

규칙에 대한 선택적 설명을 제공합니다.

do some jobs every minute.

## 규칙 유형

이벤트 패턴이나 자동 일정에 따라 대상을 트리거합니다.

☐ 이벤트 패턴

☒ 예약 표현식

## 예약 표현식\*

fastcampus 데이터 사이언스 Extension을 사용하여 자동 일정에 따라 대상을 트리거합니다. cron 표현식은 UTC 기준입니다.

## Schedule with cron

- cron: Unix-like 시스템에서 어떤 일을 주기적으로 실행시키기 위해 사용하는 툴

# cron

```
$ crontab file
```

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7)
# | | | | | OR sun,mon,tue,wed,thu,fri,sat
# * * * * * command
```



## examples

```
0 * * * * /path/to/scrapper.py
```

```
0,20,40 * * * * /path/to/scrapper.py
```

```
*/20 * * * * /path/to/scrapper.py
```

```
*/10 * * * * /path/to/scrapper.py
```

```
* 9-18 * * * /path/to/scrapper.py
```

```
30 * * * 1-5 /path/to/scrapper.py
```

```
*/10 * * * 1-5 /path/to/scrapper.py
```

## in AWS Lambda

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7)
# | | | | | .----- OR sun,mon,tue,wed,thu,fri,sat
# | | | | | .- year (1970-2199)
# | | | | |
# * * * * *
```

([https://docs.aws.amazon.com/ko\\_kr/AmazonCloudWatch/latest/events/ScheduledEvents.html#RateExpressions](https://docs.aws.amazon.com/ko_kr/AmazonCloudWatch/latest/events/ScheduledEvents.html#RateExpressions))

# Wildcard

- , : 값을 나열
- - : 범위
- \* : 모든
- / : 간격
- L : 마지막 (day, day of week only)
- W : 평일 (day only)

## warning

1. 월과 요일은 \*를 연속으로 사용할 수 없음  
( \* \* \* \* ? \* == \* \* \* \* \* \* )
2. 최소 간격은 1분( \* \* \* \* ? \* )

## ✔ 실행 결과: 성공 (로그)

### ▼ 세부 정보

아래 섹션은 함수 실행에서 반환된 결과를 보여줍니다.

```
"test"
```

### 요약

코드 SHA-256

jVm0wBFnuJ35SL6sB6H7+f4FtdFB3tT9uiVaqzhBLoQ=

기간

0.45 ms

구성된 리소스

128 MB

요청 ID

ba91e1a1-639d-4

청구 기간

100 ms

사용된 최대 메모리

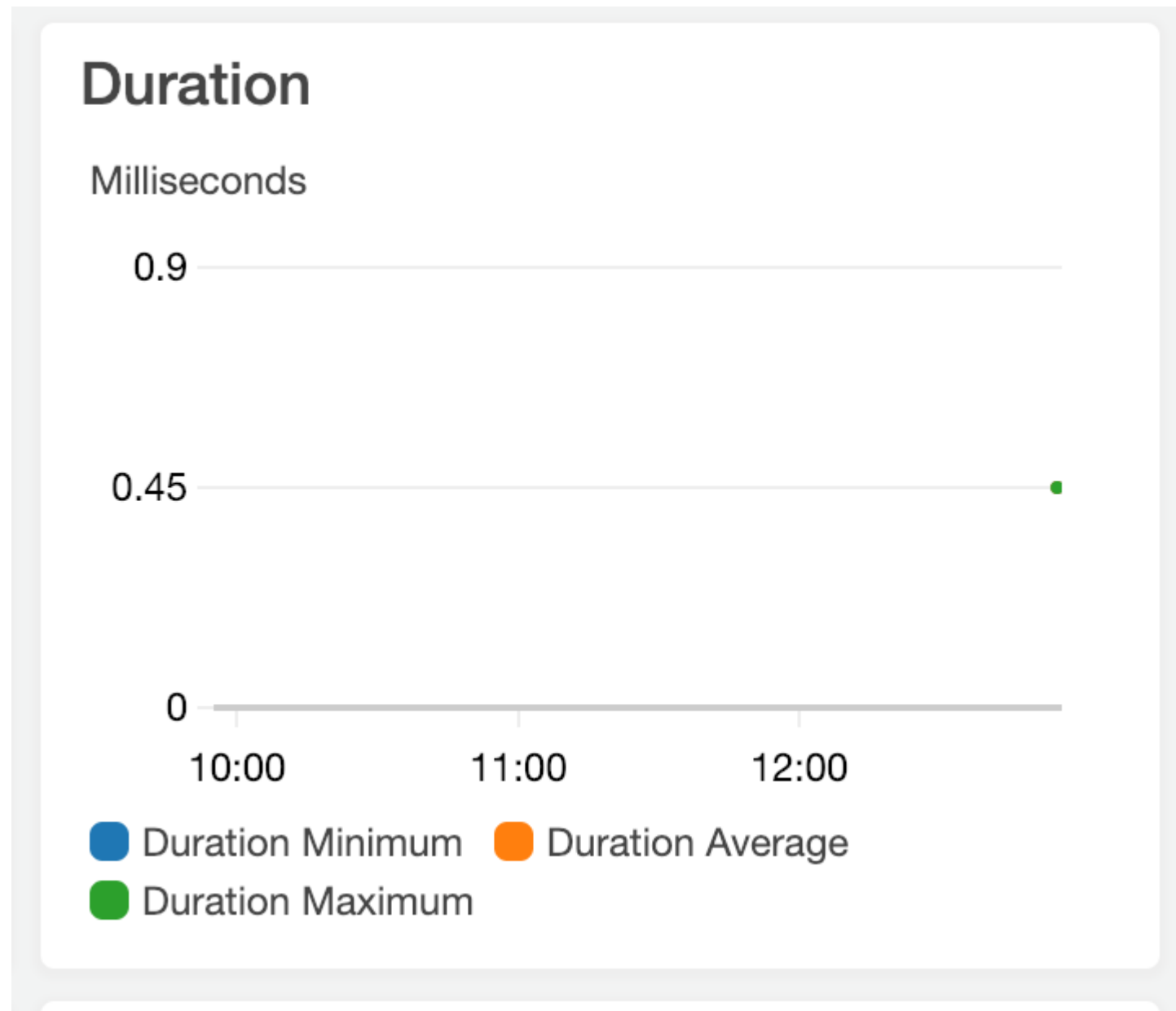
21 MB

### 로그 출력

아래 섹션은 코드의 호출 로깅을 보여줍니다. 이 Lambda 함수에 해당하는 CloudWatch 로그 그룹 내 단일 행과 습니다.

```
START RequestId: ba91e1a1-639d-429a-8c79-4dbd172b0406 Version: $LATEST
value1 = test
value2 = value2
value3 = value3
```

# New lambda function



# schedule scraping with Cloud watch events

**함수 코드** [정보](#)

코드 입력 유형

.zip 파일 업로드 ▼

런타임

Python 3.6 ▼

핸들러

[정보](#)

lambda\_function.lambda\_handler

함수 패키지\*


📁 업로드

function.zip (10.3 MB)

10MB가 넘는 파일의 경우, Amazon S3를 사용한 업로드를 고려하십시오.

# schedule scraping with Cloud watch events

## 함수 코드 정보

 Lambda 함수 "nvscrap"의 배포 패키지가 너무 커서 인라인 코드 편집을 활성화할 수 없습니다. 하지만 여전히 함수를 호출할 수 있습니다.

코드 입력 유형

.zip 파일 업로드 ▼

런타임

Python 3.6 ▼

핸들러 정보

function.lambda\_handler

함수 패키지\*

 업로드

10MB가 넘는 파일의 경우, Amazon S3를 사용한 업로드를 고려하십시오.

## Do it yourself

- function.zip의 압축을 풀어 function.py에 대하여 지난시간 활성화한 mlab database에 검색어를 저장하도록 하는 코드를 완성하세요  
pymongo 는 requirements에 존재합니다.



# EC2



- <https://aws.amazon.com/ec2/>

## AWS EC2

- Amazon Elastic Computer Cloud
- rent virtual computers
- Google Compute Engine, Azure Virtual Machines

## Install AWS CLI for windows

- 64bit: <https://s3.amazonaws.com/aws-cli/AWSCLI64.msi>
- 32bit: <https://s3.amazonaws.com/aws-cli/AWSCLI32.msi>

## windows10 built-in SSH

- <https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/>

## MacOS, Linux

```
$ pip install awscli
```

or

```
$ pip3 install awscli
```

## 인스턴스 생성

---

Amazon EC2 사용을 시작하려면 Amazon I



미국 동부 (뉴저지주 뉴저지)

미국 동부 (오하이오)

미국 서부 (캘리포니아 북부)

미국 서부 (오레곤)

아시아 태평양 (뭄바이)

아시아 태평양 (서울)

**아시아 태평양 (싱가포르)**

아시아 태평양 (시드니)

아시아 태평양 (도쿄)

캐나다 (중부)

EU (프랑크푸르트)

EU (아일랜드)

EU (런던)

EU (파리)




## Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-06c43a7df16e8213c

프리 티어 사용 가능

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

루트 디바이스 유형: ebs      가상화 유형: hvm

	<u>General purpose</u>	<u>t2.micro</u> 프리 티어 사용 가능	1	1
---	------------------------	--------------------------------	---	---

## 기존 키 페어 선택 또는 새 키 페어 생성



키 페어는 AWS에 저장하는 **퍼블릭 키**와 사용자가 저장하는 **프라이빗 키 파일**로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 추가됩니다. [퍼블릭 AMI에서 기존 키 페어 제거](#)에 대해 자세히 알아보십시오.

기존 키 페어 선택



키 페어를 선택하십시오

키 페어 없음



### 키 페어 없음

키 페어가 없습니다. 계속하려면 위에서 **[새 키 페어 생성]** 옵션을 선택하여 새 키 페어를 작성하십시오.

[취소](#)

[인스턴스 시작](#)



새 키 페어 생성

키 페어 이름

dses-ec2t4m

키 페어 다운로드

로드밸런서

대상 그룹



의견



한국어



dses-ec2t4m.pem



## 시작 상태



지금 인스턴스를 시작 중입니다.

다음 인스턴스 시작 개시: [i-0e0c66a6e6ede2fe7](#) [시작 로그 보기](#)



예상 요금 알림 받기

[결제 알림 생성](#) AWS 결제 예상 요금이 사용자가 정의한 금액을 초과하는 경우(예를 들면 프리 티어를

## 인스턴스에 연결하는 방법

인스턴스 시작



연결

작업 ▼



태그 및 속성별 필터 또는 키워드별 검색



Name



인스턴스 ID



인스턴스 유형



가용 영역



인스턴스 상태



i-0e0c66a6e6ede2fe7

t2.micro

ap-northeast-1a




pending

가용 영역

인스턴스 상태

ap-northeast-1a

 running

인스턴스: **i-0e0c66a6e6ede2fe7** 퍼블릭 DNS: **ec2-13-115-82-39.ap-northeast-1.compute.amazonaws.com**


설명	상태 검사	모니터링	태그
인스턴스 ID	i-0e0c66a6e6ede2fe7 		
인스턴스 상태	running		
인스턴스 유형	t2.micro		
탄력적 IP			
가용 영역	ap-northeast-1a		
보안 그룹	<a href="#">launch-wizard-1</a> . <a href="#">인바운드 규칙 보기</a> . <a href="#">view outbound rules</a>		
예약된 이벤트	예약된 이벤트 없음		
AMI ID	<a href="#">ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180912 (ami-06c43a7df16e8213c)</a>		
퍼블릭 DNS(IPv4)	ec2-13-115-82-39.ap-northeast-1.compute.amazonaws.com		
IPv4 퍼블릭 IP	13.115.82.39		
IPv6 IP	-		
프라이빗 DNS	ip-172-31-36-135.ap-northeast-1.compute.internal		
프라이빗 IP	172.31.36.135		
보조 프라이빗 IP			
VPC ID	vpc-8e3e1be9		
서브넷 ID	subnet-884bacc0		


---


**pute.amazonaws.com**

클립보드에 복사


---

퍼블릭 DNS(IPv4)	ec2-	
	1.compute.amazonaws.com	
IPv4 퍼블릭 IP	13.115.82.39	
IPv6 IP	-	

 dses-ec2t4m.pem

 function.py

 function.zip

 package ▶

 requirements.txt



dses-ec2t4m.pem  
printable encoded archive - 2 KB

## 인스턴스에 연결



- 다음에 연결:
- ☒ 독립 실행형 SSH 클라이언트 ⓘ
  - ☐ 현재 웹 브라우저에서 Java SSH 클라이언트에 직접(Java 필요) ⓘ

### 인스턴스 액세스 방법:

1. SSH 클라이언트를 개방하십시오. ([PuTTY를 사용하여 연결](#) 방법 알아보기)
2. 프라이빗 키 파일(dses-ec2t4m.pem)을 찾습니다. 마법사가 인스턴스를 시작하는 데 사용되는 키를 자동으로 검색합니다.
3. SSH가 작동하려면 키가 공개적으로 표시되지 않아야 합니다. 필요할 경우 이 명령을 사용합니다.

```
chmod 400 dses-ec2t4m.pem
```

4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:

```
ec2-13-115-82-39.ap-northeast-1.compute.amazonaws.com
```

예:

```
ssh -i "dses-ec2t4m.pem" ubuntu@ec2-13-115-82-39.ap-northeast-1.compute.amazonaws.com
```

대부분의 경우 위의 사용자 이름이 맞지만, AMI 사용 지침을 숙지하여 AMI 소유자가 기본 AMI 사용자 이름을 변경하지 않도록 하십시오.

인스턴스에 연결하는 데 도움이 필요한 경우 [연결 설명서](#)을(를) 참조하십시오.

닫기

```
$ ls  
dses-ec2t4m.pem  
( dses364 )
```

```
$ chmod 400 dses-ec2t4m.pem
```

```
-r-----@
```

```
$ ssh -i "dses-ec2t4m.pem" ubuntu@<your-public-domain>
```

Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1067-aws x86\_64)

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

Get cloud support with Ubuntu Advantage Cloud Guest:  
<http://www.ubuntu.com/business/services/cloud>

0 packages can be updated.  
0 updates are security updates.

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.

```
ubuntu@ip-172-31-36-135:~$ clear
ubuntu@ip-172-31-36-135:~$
```



```
Resolving repo.continuum.io (repo.continuum.io)... 104.16
Connecting to repo.continuum.io (repo.continuum.io)|104.1
HTTP request sent, awaiting response... 200 OK
Length: 523283080 (499M) [application/x-sh]
Saving to: 'Anaconda3-4.4.0-Linux-x86_64.sh'

Anaconda3-4.4.0-Linux-x86_64.  51%[=====
```

```
$ wget https://repo.continuum.io/archive/Anaconda3-4.4.0-Linux-
x86_64.sh
```

```
$ bash Anaconda3-4.4.0-Linux-x86_64.sh
```

```
$ vi ~/.bashrc
```

```
export PATH="/home/ec2-user/anaconda3/bin:$PATH" <= add
```

```
ubuntu@ip-172-31-36-135:~$ vi .bashrc
ubuntu@ip-172-31-36-135:~$ source ~/.bashrc
ubuntu@ip-172-31-36-135:~$ which python
/home/ubuntu/anaconda3/bin/python
ubuntu@ip-172-31-36-135:~$
```

```
ubuntu@ip-172-31-36-135:~$ python
Python 3.6.1 |Anaconda 4.4.0 (64-bit
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1
Type "help", "copyright", "credits"
>>> █
```

```
$ ipython
```

```
In [1]: from IPython.lib import passwd
```

```
In [2]: passwd()
```

```
Enter password:
```

```
Verify password: 
```

copy **SHAstring!!!**

```
$ jupyter notebook --generate-config
```

```
$ mkdir certs && cd certs
```

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -  
keyout mycert.pem -out mycert.pem
```

```
Generating a 1024 bit RSA private key  
.++++++  
.....++++++  
writing new private key to 'mycert.pem'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [AU]:KR  
State or Province Name (full name) [Some-State]:Seoul  
Locality Name (eg, city) []:Seoul  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NONE  
Organizational Unit Name (eg, section) []:NONE  
Common Name (e.g. server FQDN or YOUR name) []:dses  
Email Address []:email  
ubuntu@ip-172-31-36-135:~/certs$
```

```
$ vi .jupyter/jupyter_notebook_config.py
```

```
c = get_config()

# Kernel config
c.IPKernelApp.pylab = 'inline'

# Notebook config
c.NotebookApp.certfile = u'/home/ubuntu/certs/mycert.pem' #location of cert
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.password = u'sha1:...'

c.NotebookApp.port = 8888
```

```
:wq
```

```
$ mkdir dev/pynbs && cd dev/pynbs
```

```
$ jupyter notebook
```

```
https://<your-public-domain>:8888/
```



/ter

Password:

Upload

New ▾

↺

Notebook:

Python 3

↑

Other:

Text File

Folder

Terminal