



Thematic Academy

Tema pelatihan:

Pendalaman Microservice
(Pertemuan 15)



BADAN PENELITIAN
DAN PENGEMBANGAN
SUMBER DAYA MANUSIA
Jl. Medan Merdeka Barat No. 9
Jakarta Pusat, 10110

digitalent.kominfo.go.id

📷 [digitalent.kominfo](https://www.instagram.com/digitalent.kominfo)

📘 [digitalent.kominfo](https://www.facebook.com/digitalent.kominfo)

🐦 [DTS_kominfo](https://twitter.com/DTS_kominfo)

📍 [digital talent scholarship 2019](#)



Bagian 1

Deploy Microservices di EC2 AWS

Jumat, 16 Oktober 2020

Setup EC2



aws **Services** muisdong@gmail.com @ dgb-valley46 Oregon Support

★ Favorites

- Resource Groups & Tag Editor

Recently visited

- EC2
- Console Home
- Billing
- EC2 Image Builder

All services

Find services by names, keywords or acronyms.

| | | | |
|----------------------------------|-----------------------------|-------------------------|--------------------------------|
| Compute | Robotics | Machine Learning | AR & VR |
| ★ EC2 | AWS RoboMaker | Amazon SageMaker | Amazon Sumerian |
| Lightsail | | Amazon Augmented AI | |
| Lambda | Customer Enablement | Amazon CodeGuru | Application Integration |
| Batch | AWS IQ | Amazon Comprehend | Step Functions |
| Elastic Beanstalk | Support | Amazon Forecast | Amazon AppFlow |
| Serverless Application Reposi... | Managed Services | Amazon Fraud Detector | Amazon EventBridge |
| AWS Outposts | | Amazon Kendra | Amazon MQ |
| EC2 Image Builder | Blockchain | Amazon Lex | Simple Notification Service |
| | Amazon Managed Blockchain | Amazon Personalize | Simple Queue Service |
| Storage | | Amazon Polly | SWF |
| S3 | Satellite | Amazon Rekognition | |
| EFS | Ground Station | Amazon Textract | AWS Cost Management |
| FSx | | Amazon Transcribe | AWS Cost Explorer |
| S3 Glacier | Quantum Technologies | Amazon Translate | AWS Budgets |
| Storage Gateway | Amazon Braket | AWS DeepComposer | AWS Marketplace Subscriptio... |
| AWS Backup | | AWS DeepLens | |

aws Services ▾

New EC2 Experience [Learn more](#)

EC2 Dashboard

Events
Tags
Limits

INSTANCES

Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations

IMAGES

AMIs
ELASTIC BLOCK STORE

Feedback English (US) ▾

Resources

You are using the following Amazon EBS resources:

- 3 Running Instances
- 0 Dedicated Hosts
- 3 Volumes
- 3 Key Pairs
- 0 Placement Groups

Easily size, configure, and deploy instances for SQL Server. [Learn more](#)

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US West (Oregon) region

Service Health

Step 1: Choose an Amazon Machine Image (AMI)

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.

Search for an AMI by entering a search term e.g. "Windows"

Search by Systems Manager parameter

Quick Start

My AMIs
AWS Marketplace
Community AMIs

☐ Free tier only ⓘ

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01fee56b22f308154

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 ENA Enabled: Yes

Select

64-bit (x86)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0528a5175983e7f28 (64-bit x86) / ami-0476b0c0bd036545d (64-bit)

Select

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 families ▾ Current generation ▾ [Show/Hide Columns](#)

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

| | Family ▾ | Type ▾ | vCPUs ⓘ | Memory (GiB) ▾ | Instance Storage (GB) ⓘ | EBS-Optimized Available ⓘ | Network Performance ⓘ | IPv6 Support ⓘ |
|-------------------------------------|----------|--------------------------------|---------|----------------|-------------------------|---------------------------|-----------------------|----------------|
| <input type="checkbox"/> | t2 | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| <input checked="" type="checkbox"/> | t2 | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | t2 | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | t2 | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |

[Cancel](#) [Previous](#) **Review and Launch** [Next: Configure Instance Details](#)

Feedback English (US) ▾

[Privacy Policy](#) [Terms of Use](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Setup EC2



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

ujicobamicroservice

Download Key Pair



You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

- Buatlah key pair baru, kemudian berikan nama

Setup EC2



Services ▾



muisdong@gmail.com @ dgb-valley46 ▾

Oregon ▾

Support ▾

☐ New EC2 Experience
[Learn more](#)

EC2 Dashboard

Events

Tags

Limits

INSTANCES

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

IMAGES

AMIs

ELASTIC BLOCK
STORE

Launch Instance ▾

Connect

Actions ▾

Filter by tags and attributes or search by keyword

1 to 4 of 4

| <input type="checkbox"/> | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) | IP |
|-------------------------------------|---------|---------------------|---------------|-------------------|----------------|----------------|--------------|---------------------------|----|
| <input type="checkbox"/> | server1 | i-00c104a475cf8d20f | t2.micro | us-west-2a | running | 2/2 checks ... | None | ec2-54-212-57-210.us-... | 54 |
| <input checked="" type="checkbox"/> | | i-053f4371b82817db4 | t2.micro | us-west-2b | running | Initializing | None | ec2-54-218-252-15.us-... | 54 |
| <input type="checkbox"/> | gate | i-05c104a475cf8d20f | t2.micro | us-west-2a | running | 2/2 checks ... | None | ec2-52-12-160-211.us-... | 52 |
| <input type="checkbox"/> | server2 | i-07f311f4e74ddef1e | t2.micro | us-west-2b | running | 2/2 checks ... | None | ec2-34-221-153-201.us-... | 34 |

klik nanti muncul icon
pencil untuk edit nama

gunakan alamat publik ini untuk login
dengan putty dan private key yang sudah
didownload

Instance: i-053f4371b82817db4 Public DNS: ec2-54-218-252-15.us-west-2.compute.amazonaws.com

Description

Status Checks

Monitoring

Tags

Instance ID i-053f4371b82817db4

Public DNS (IPv4) ec2-54-218-252-15.us-west-2.compute.amazonaws.com

Instance state running

IPv4 Public IP 54.218.252.15

Feedback English (US) ▾

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

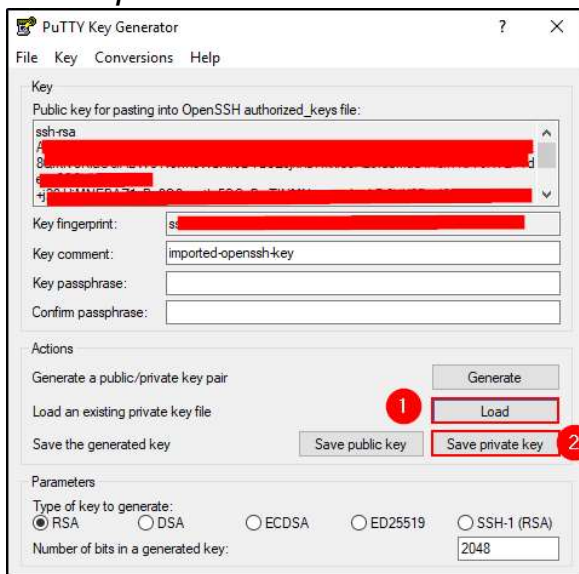
[Privacy Policy](#)

[Terms of Use](#)

Setup Akses EC2 dengan Putty

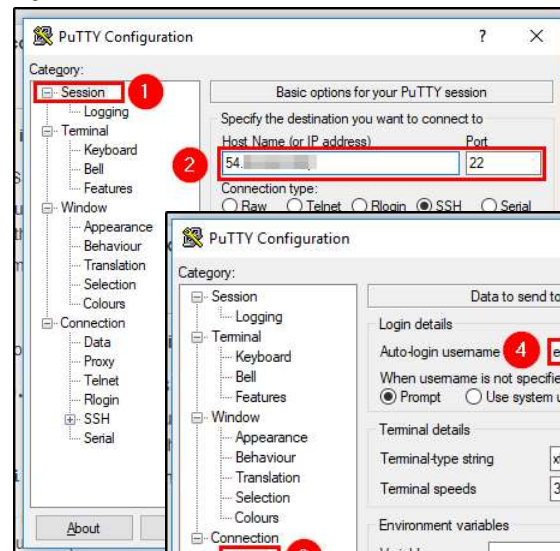


Putty tidak bisa menggunakan *.pem dari aws sehingga harus bikin sendiri
<https://www.edureka.co/community/30960/access-ec2-instance-using-putty>

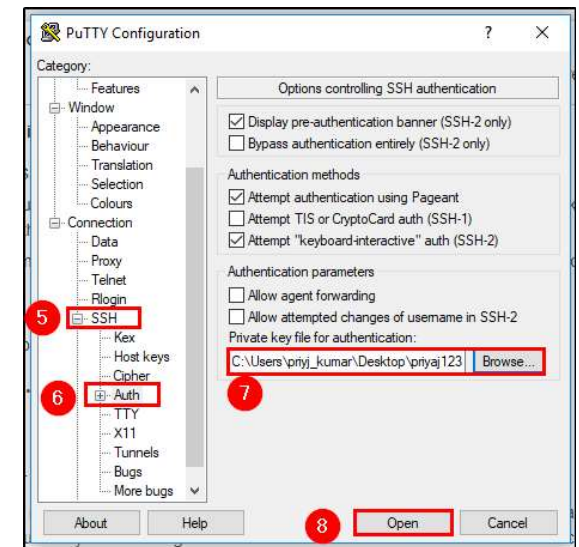
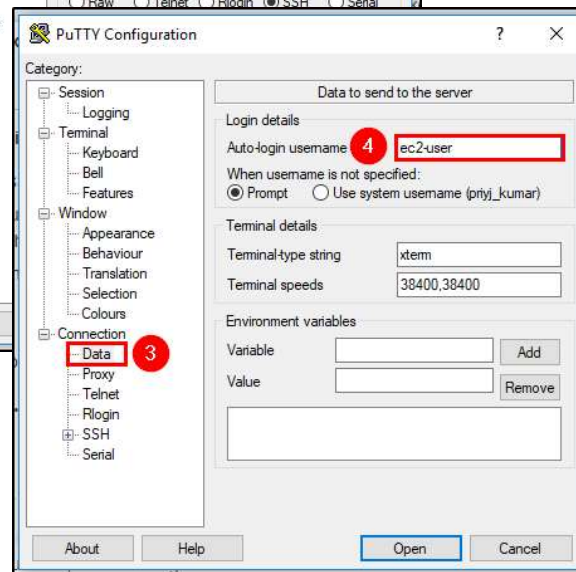


Load your .pem file generated by AWS. Then save the private key (.ppk) file.

*) jangan klik generate



Add ip and
username
ec2-user



Add ppk



Konfigurasi lainnya




Untuk Ubuntu

- `sudo apt update`
- `sudo apt install golang-go`
- `sudo apt install nodejs`
- `sudo apt install npm`

Amazon Linux

- `sudo yum update`
- `sudo yum install golang-go`

<https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html>

- `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash`
 - `. ~/.nvm/nvm.sh`
 - `nvm install node`
- 

Setup Gitlab



The screenshots illustrate the following steps:

- GitLab Project Creation:** A 'New project' button is highlighted. The 'Project name' is set to 'microservice'.
- Repository Status:** A message states 'The repository for this project is empty'.
- Cloning:** The 'Clone' button is highlighted. The 'Clone with SSH' option is selected.
- File Explorer:** A file explorer window shows the 'lathian microservice' directory. A red arrow points to the 'deploy' folder with the text 'klik kanan di folder anda'.
- Git GUI:** The 'Git GUI' window shows the 'Clone Existing Repository' option selected.
- Commit Process:** The 'Commit' button is highlighted. The 'Commit Message' is 'update README.md'.
- File List:** A table of files in the 'deploy' directory is shown:

| Name | Date modified | Type | Size |
|------------|------------------|-----------------|-------|
| bmkp.go | 15/10/2020 16:50 | GO File | 3 KB |
| gate.go | 15/10/2020 12:48 | GO File | 11 KB |
| go.mod | 15/10/2020 6:20 | MOD File | 1 KB |
| go.sum | 15/10/2020 6:20 | SUM File | 8 KB |
| README.md | 16/10/2020 11:18 | MD File | 1 KB |
| server2.js | 15/10/2020 14:27 | JavaScript File | 1 KB |

Additional annotations include 'Copy Paste Program Anda Sebelumnya' and 'Commit to master by'.

Setup Gitlab



MINGW64/d/digitalent/backfront/Materi Ajar (UI)/Materi PPT - Backend/latih...

```
muisd@LAPTOP-QAB9ETZE MINGW64 /d/digitalent/backfront/Materi Ajar (UI)/Materi PPT - Backend/latihan microservice
$ git config --global user.email "muisdong@gmail.com"
muisd@LAPTOP-QAB9ETZE MINGW64 /d/digitalent/backfront/Materi Ajar (UI)/Materi PPT - Backend/latihan microservice
$ git config --global user.name "Abdul Muis"
muisd@LAPTOP-QAB9ETZE MINGW64 /d/digitalent/backfront/Materi Ajar (UI)/Materi PPT - Backend/latihan microservice
$
```

Set user

file baru

Rescan

Stage To Commit

Commit Message: #penambahan data

Push

Push Branches

Source Branches: master

Destination Repository: Remote: origin

Transfer Options: Force overwrite existing branch (may discard changes), Use thin pack (for slow network connections), Include tags

Push

Updating the Git index failed. A rescan will be automatically started to resynchronize git-gui.

Unlock Index Continue

Setup CI/CD autodeploy



1. CI/CD di gitlab

Setting (menu paling bawah) > CI/CD > Variables

Tambahkan SSH_PRIVATE_KEY diisi dengan isi .pem terakhir

Tambahkan DEPLOY_SERVER diisi dengan ip atau domain-name instance AWS kita

2. Buat .gitlab-ci.yml

production:

image: golang

stage: deploy

before_script:

- mkdir -p ~/.ssh

- echo -e "\$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa

- chmod 600 ~/.ssh/id_rsa

- '[[-f /.dockerenv]] && echo -e "Host *\n\tStrictHostKeyChecking no\n\n" >

~/.ssh/config'

script:

#- go mod vendor # tidak usah kalo folder vendor ikut dicommit

- go build -o main.exe main.go

- scp main.exe ubuntu@\$DEPLOY_SERVER:/home/ubuntu/

environment:

name: production

3. Ssh ke server, lalu ls -al untuk melihat hasil exe-nya apakah benar terupload

./main.exe # untuk menjalankan binary-nya di server

4. Kalau nodejs pakai cara yg di slide saja (ganti "stage: production" menjadi "stage: deploy", juga hapus "manual" kalau ingin tiap ada yg commit ditrigger CI/CD-nya), karena butuh source code ada di server production kalau nodejs

Setup CI/CD autodeploy



The screenshot shows the GitLab interface with the following elements and annotations:

- Left Sidebar:** The **Settings** menu is open, and the **CI / CD** option is selected. Red circles highlight the **Settings** and **CI / CD** items.
- Main Content Area:** The **Variables** section is active. It explains that environment variables are applied to environments via the Runner. Red circles highlight the **Variables** header and the **Add Variable** button.
- Add variable dialog:** A modal window is open for adding a new variable.
 - Key:** The text **SSH_PRIVATE_KEY** is entered. A red circle highlights this field, with a red arrow pointing to it from the text "diambil dari file privatekey.pem saat anda membuat EC2 instance".
 - Value:** A long alphanumeric string representing the private key is pasted into the text area. A red circle highlights this area, with a red arrow pointing to it from the same text.
 - Type:** The dropdown is set to **Variable**.
 - Environment scope:** The dropdown is set to **All (default)**.
 - Flags:** The **Protect variable** checkbox is checked. A red circle highlights this checkbox, with a red arrow pointing to it from the text "diambil dari file privatekey.pem saat anda membuat EC2 instance".
 - Buttons:** At the bottom right, the **Add variable** button is highlighted with a red circle and a red arrow.

Setup CI/CD autodeploy



```
D:\digital\backfront\Materi Ajar (UI)\Materi PPT - Backend\latihan microservice\deploy\gitlab-ci.yml - Sublime
File Edit Selection Find View Goto Tools Project
.gitlab-deploy.sh x muis-oregon.pub x .gitlab-ci.yml x muis-oregon.pe
1 #Production stage
2 production:
3   image: golang
4   stage: deploy
5   before_script:
6     - mkdir -p ~/.ssh
7     - echo -e "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa
8     - chmod 600 ~/.ssh/id_rsa
9     - '[[ -f /.dockerenv ]] && echo -e "Host *\n\tStrictHostKeyCheck
10   script:
11     #- go mod vendor # tidak usah kalo folder vendor ikut dicommit
12     - go build -o main.exe main.go
13     - scp main.exe ec2-user@$GATE_SERVER:/home/ec2-user/
14     - scp main.exe ec2-user@$SERVER1:/home/ec2-user/
15     - scp main.exe ec2-user@$SERVER2:/home/ec2-user/
16   environment:
17     name: production
Line 18: Column 1 Spaces: 2 YAMLL
```

GitLab Projects Groups More Search or jump to...

Environment variables are configured by your administrator to be protected by default

| Type | Key | Value | Protected | Masked | Environments |
|----------|-----------------|-------|-----------|--------|---------------|
| Variable | GATE_SERVER | ***** | ✓ | ✗ | All (default) |
| Variable | SERVER1 | ***** | ✓ | ✗ | All (default) |
| Variable | SERVER2 | ***** | ✓ | ✗ | All (default) |
| Variable | SSH_PRIVATE_KEY | ***** | ✓ | ✗ | All (default) |

Reveal values Add Variable

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

| Public DNS (IPv4) | IPv4 Public IP | IPv6 IPs | Key Name |
|---------------------------|----------------|----------|-------------------|
| ec2-54-212-57-210.us-... | 54.212.57.210 | - | muiss-oregon-s... |
| ec2-54-218-252-15.us-... | 54.218.252.15 | - | fachrisiddiqakbar |
| ec2-52-12-160-211.us-... | 52.12.160.211 | - | muiss-oregon |
| ec2-34-221-153-201.us-... | 34.221.153.201 | - | muiss-oregon-s... |



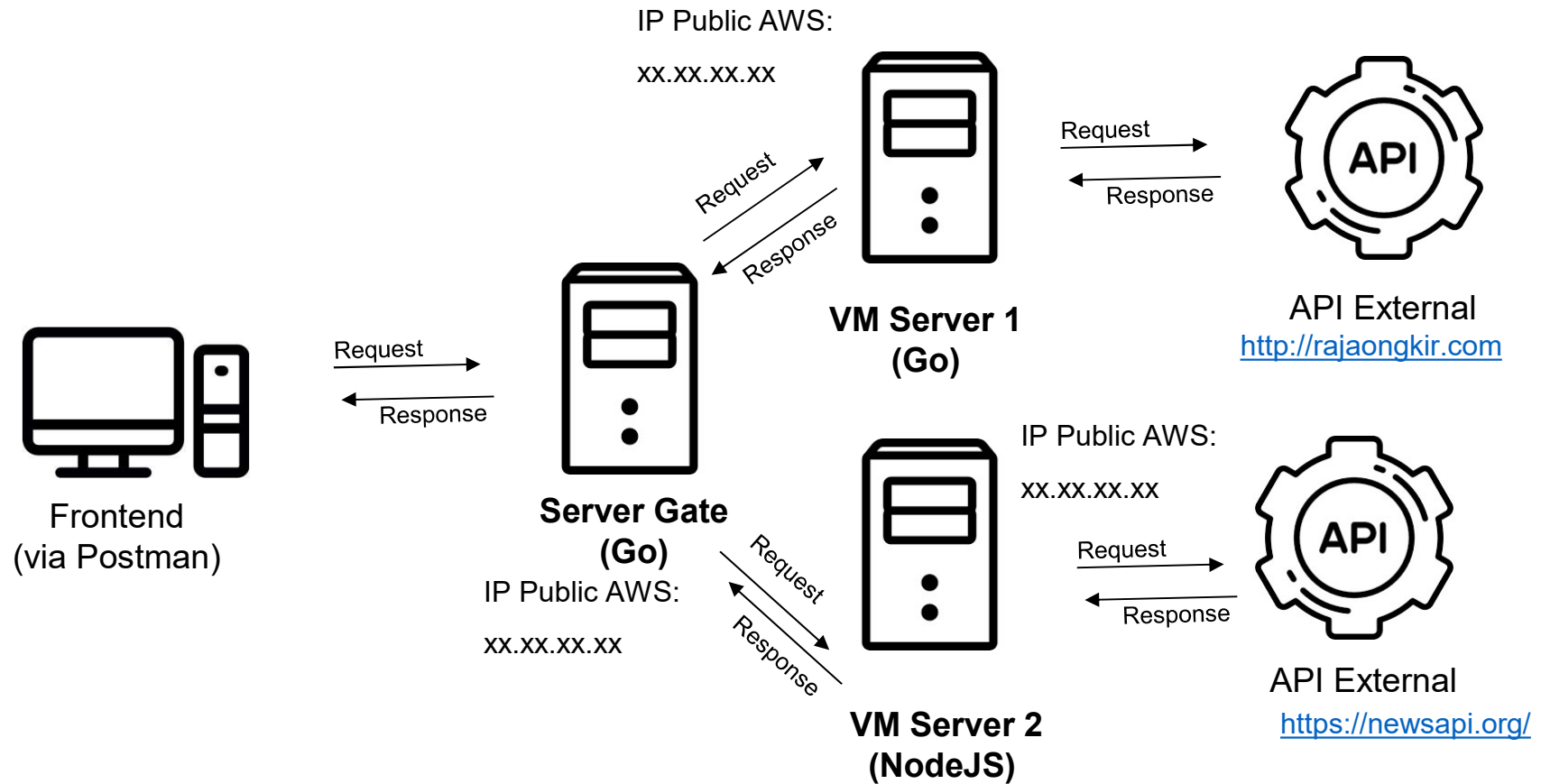
Konfigurasi lainnya



- Setup inbound dan outbound AWS agar dapat diakses melalui luar
- IP Elastic
- (optional) Load Balancer



Deploy di AWS



Penjelasan Detail



- Terdapat 3 Server: Gate (Golang), Server 1 (Golang) dan Server 2 (NodeJS). Masing – masing memiliki tugas:
 - Gate: Sebagai gerbang server yang menghubungkan antara Frontend dan Backend dan AUTH
 - Server1 : Sebagai server yang menghubungkan dengan API External
 - Server2: Sebagai server yang menghubungkan dengan database internal
- **Gate**, menerima input dari Frontend, untuk kemudian langkah awal adalah menerima username dan password untuk divalidasi. Jika valid maka session akan aktif dan Frontend dapat request data dari server 1 dan server 2
- **Server1**, menerima request dari Gate untuk mengakses external API seperti dari <http://rajaongkir.com> untuk kemudian data nya akan dikirimkan berupa response ke Gate guna diteruskan ke Frontend
- **Server2**, menerima request dari Gate untuk mengakses external API seperti dari <http://newsapi.org/s/indonesia-news-api> untuk kemudian data nya akan dikirimkan berupa response ke Gate guna diteruskan ke Frontend



Konfigurasi VM



- **Buatlah 3 EC VM** untuk server gate, server 1 dan server 2 masing – masing dengan konfigurasi yang sama
- **Konfigurasi dan lakukan instalasi untuk library yang dibutuhkan: NodeJS/Go**





Konfigurasi Server 1



- `sudo apt-get update`
- `sudo apt-get install golang-go`
- Buatlah program `server1.go` sesuai requirement tersebut, jalankan pada IP anda: `xx.xx.xx.xx:1234`





Konfigurasi Server 2



- `sudo apt-get update`
- `sudo apt install nodejs`
- `sudo apt install npm`
- Buatlah program `server2.js` sesuai requirement tersebut, jalankan pada IP anda: `xx.xx.xx.xx:4321`

<https://www.hostinger.co.id/tutorial/install-node-js-ubuntu/>





Konfigurasi Gate



DIGITAL
TALENT
SCHOLARSHIP

- `sudo apt-get update`
- `sudo apt-get install golang-go`
- Buatlah program `server_gate.go` sesuai requirement tersebut, jalankan pada IP anda:
`xx.xx.xx.xx:5000`



Daftar API



Server1

| Method | URL | Deskripsi |
|--------|---|---|
| GET | https://xx.xx.xx.xx:1234/getKota | Mendapatkan data kota, provinsi dan ID yang akan digunakan untuk memeriksa ongkos kirim |
| GET | https://xx.xx.xx.xx:1234/getOngkir | Mendapatkan data transaksi given ID Kota |

Server2

| Method | URL | Deskripsi |
|--------|---|---|
| GET | https://xx.xx.xx.xx:4321/getTopHeadline | Mendapatkan top headline dari Indonesia |
| | | |



Daftar API



Server Gate

| Method | URL | Deskripsi |
|--------|---|--|
| GET | https://xx.xx.xx.xx:1234/getKota | Mendapatkan data kota, provinsi dan ID yang akan digunakan untuk memeriksa ongkos kirim (data actual didapatkan dari server 1) |
| GET | https://xx.xx.xx.xx:1234/getOngkir | Mendapatkan data transaksi given ID Kota (data actual didapatkan dari server 1) |
| GET | https://xx.xx.xx.xx:4321/getTopHeadline | Mendapatkan top headline dari Indonesia (data actual didapatkan dari server 2) |





#DIGITALINAJA

TERIMA KASIH



BADAN PENELITIAN
DAN PENGEMBANGAN
SUMBER DAYA MANUSIA
Jl. Medan Merdeka Barat No. 9
Jakarta Pusat, 10110

digitalent.kominfo.go.id

digitalent.kominfo

digitalent.kominfo

DTS_kominfo

digital talent scholarship 2019

