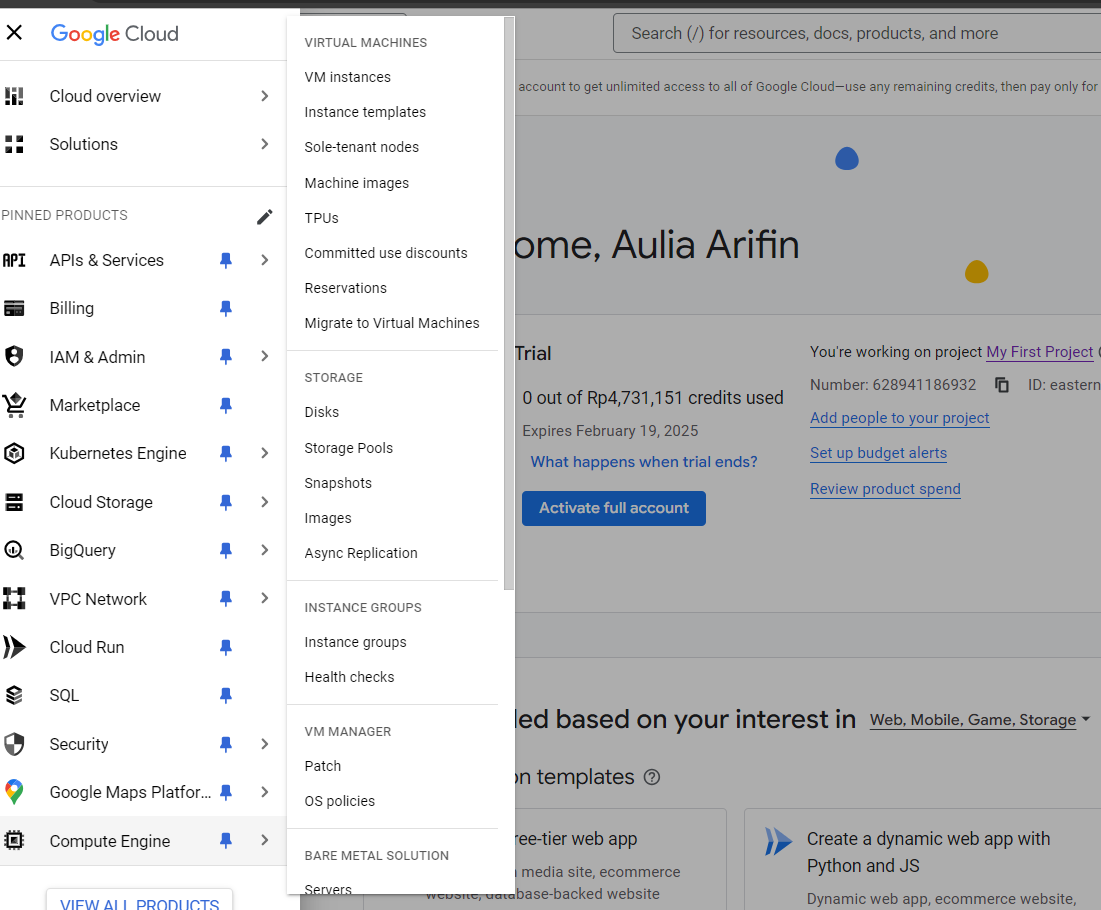
**DOKUMENTASI**

1. **Langkah Langkah Pembuatan Instance di Goole Cloud Platform**

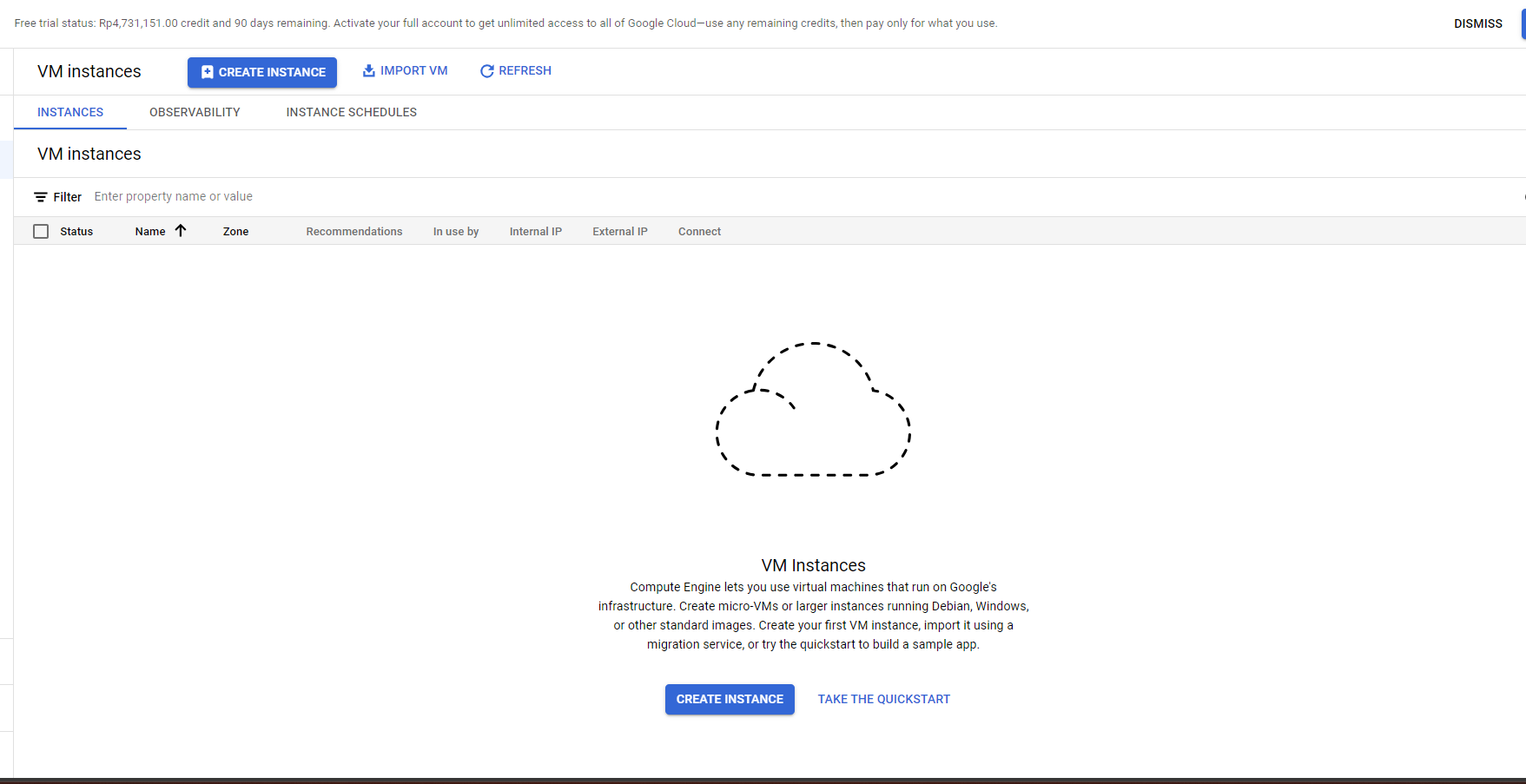
Spesifikasi instance sebagai berikut:

* Region: us-central1-c
* Type: e2-micro
* OS: Ubuntu 22.04 LTS
* Disk: 25 GB

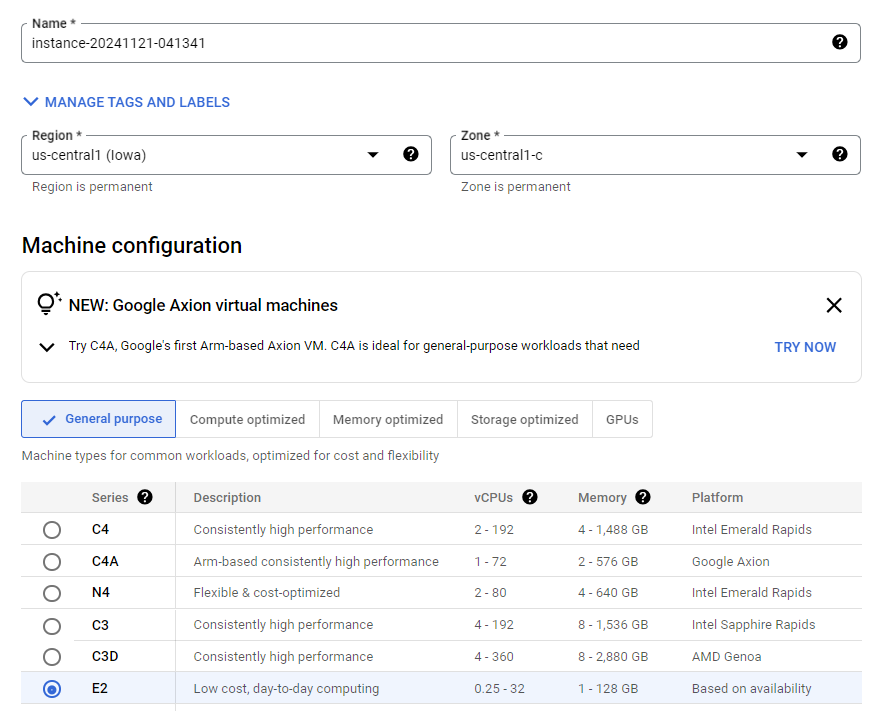
1. Masuk ke halaman console.cloud.google.com
2. Pilih Navigation Menu -> Compute Engine -> VM Instance

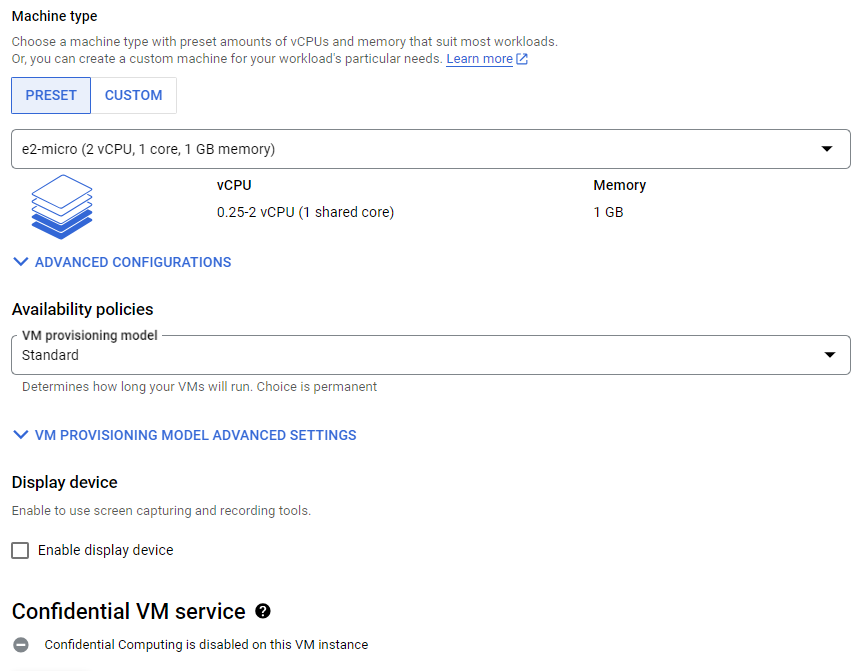


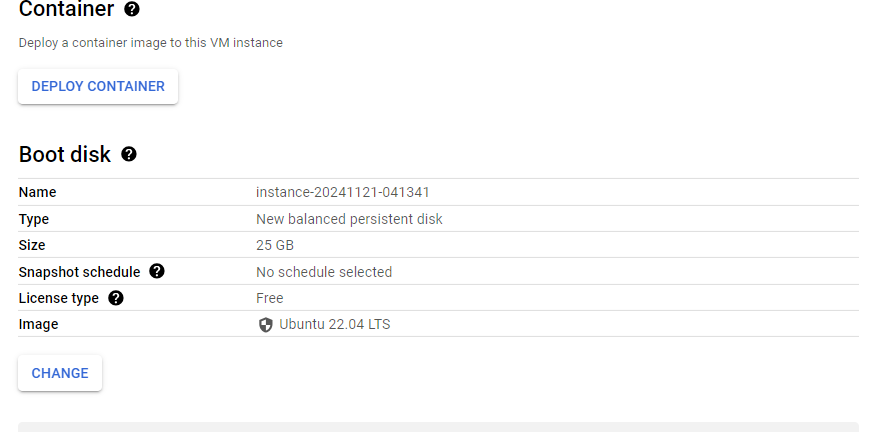
1. Pilih menu CREATE INSTANCE

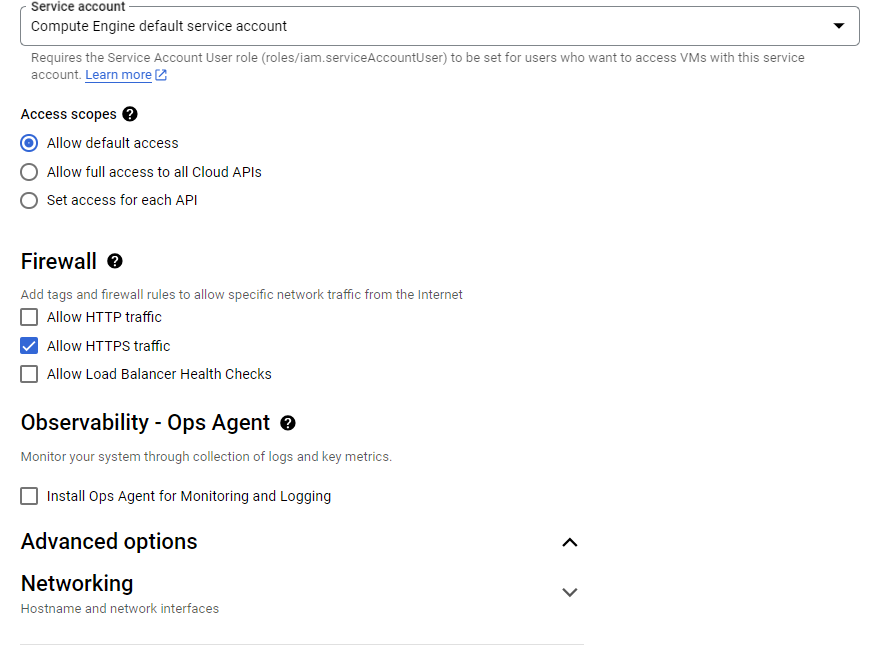


1. Membuat instance sesuai dengan spesifikasi yang ditentukan

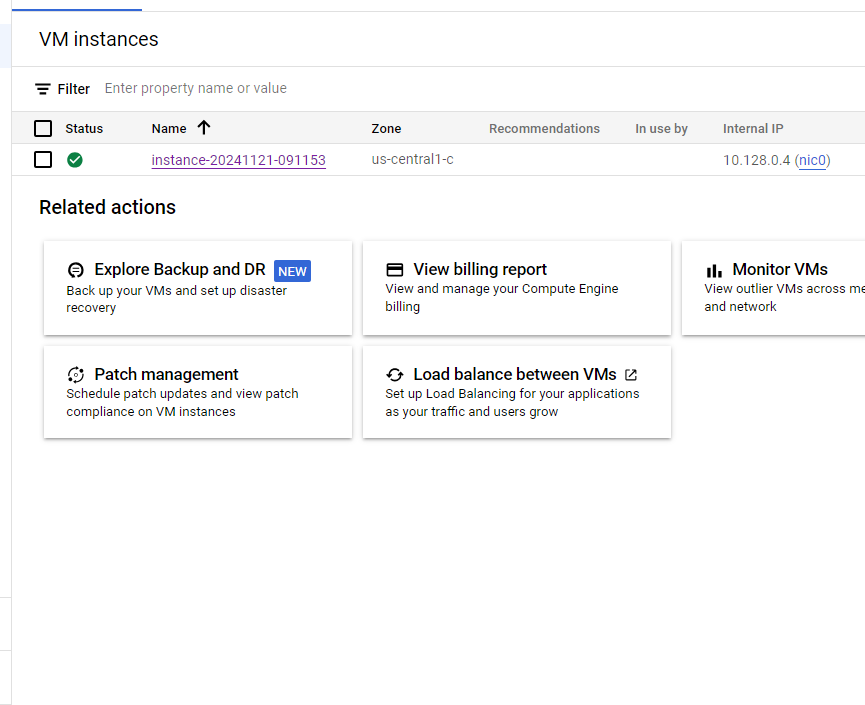








1. Kemudian klik create
2. Instance siap digunakan

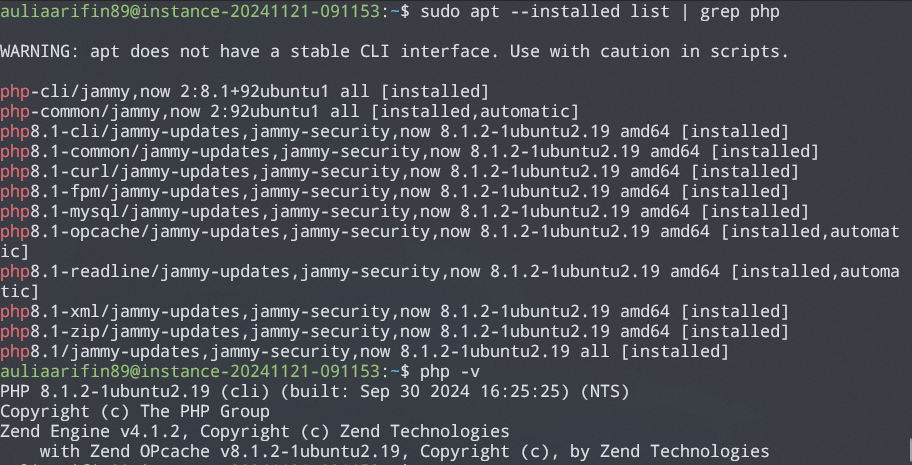


1. **Informasi Stack dan Environment**

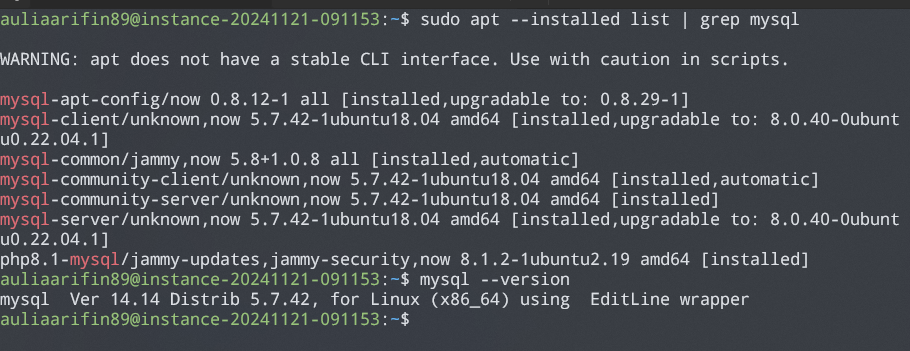
* Nginx



* PHP 8.1



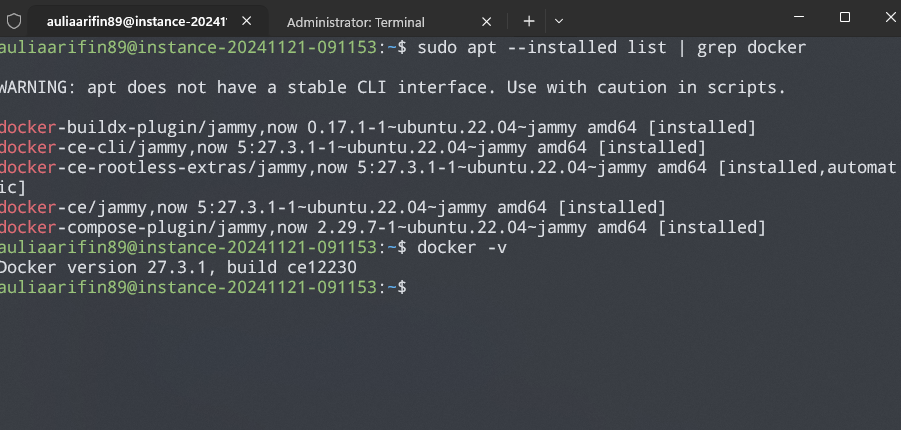
* MYSQL 5.7



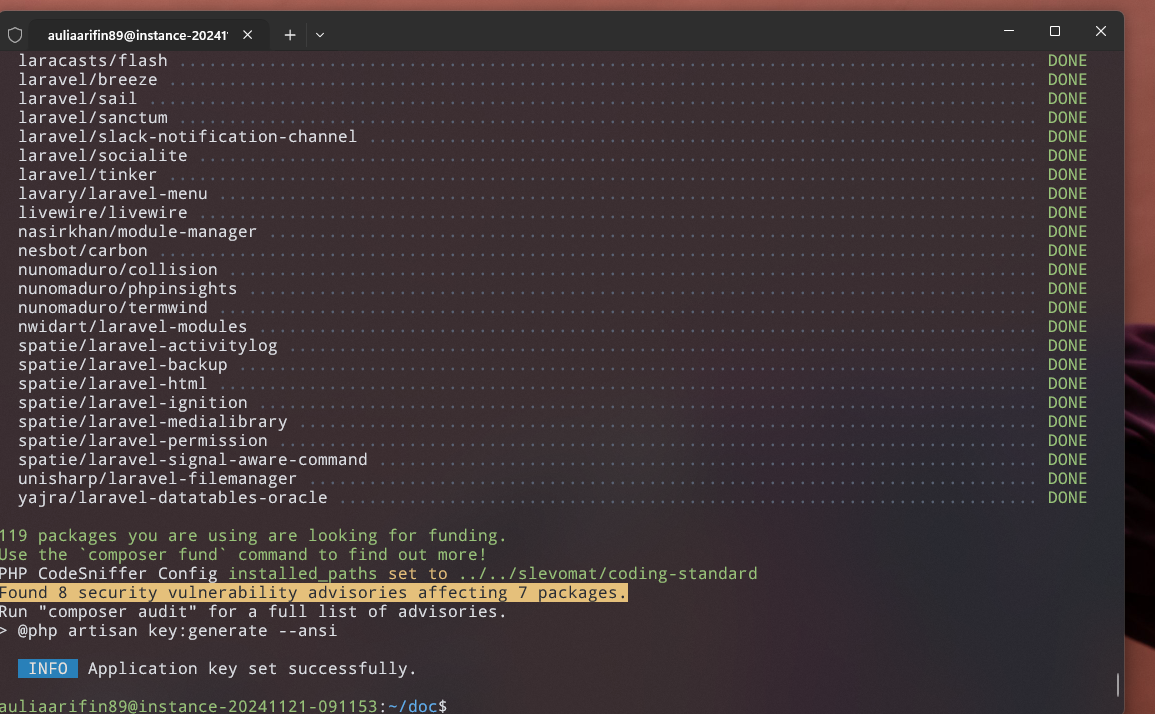
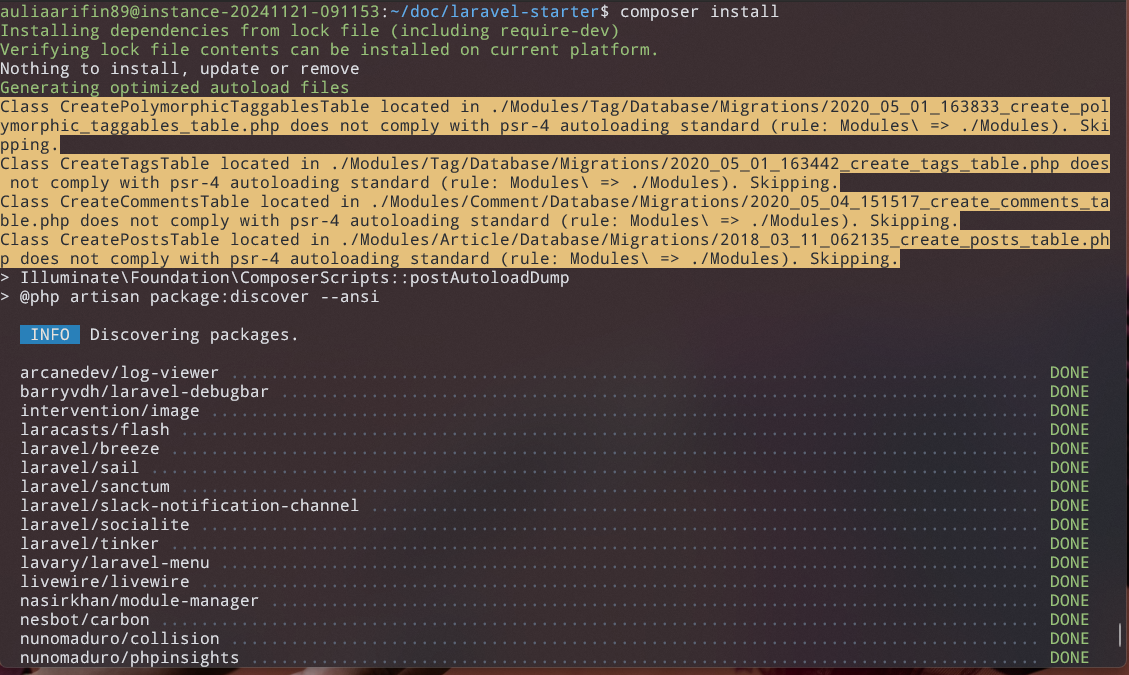
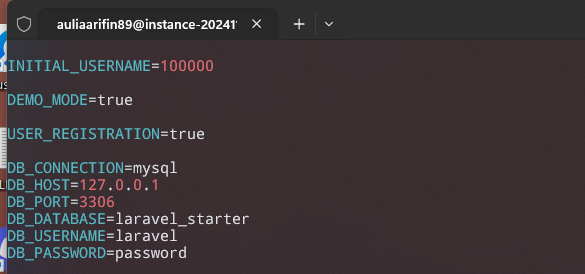
* Composer

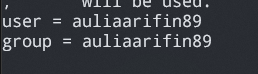


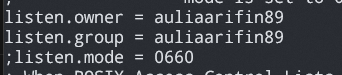
* Docker

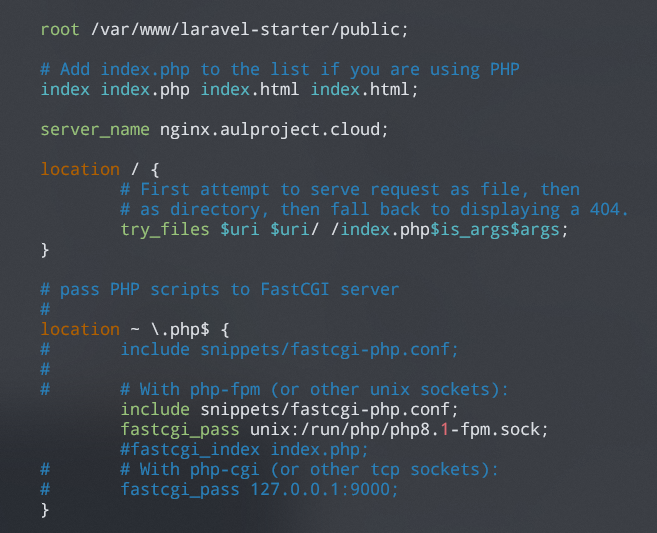


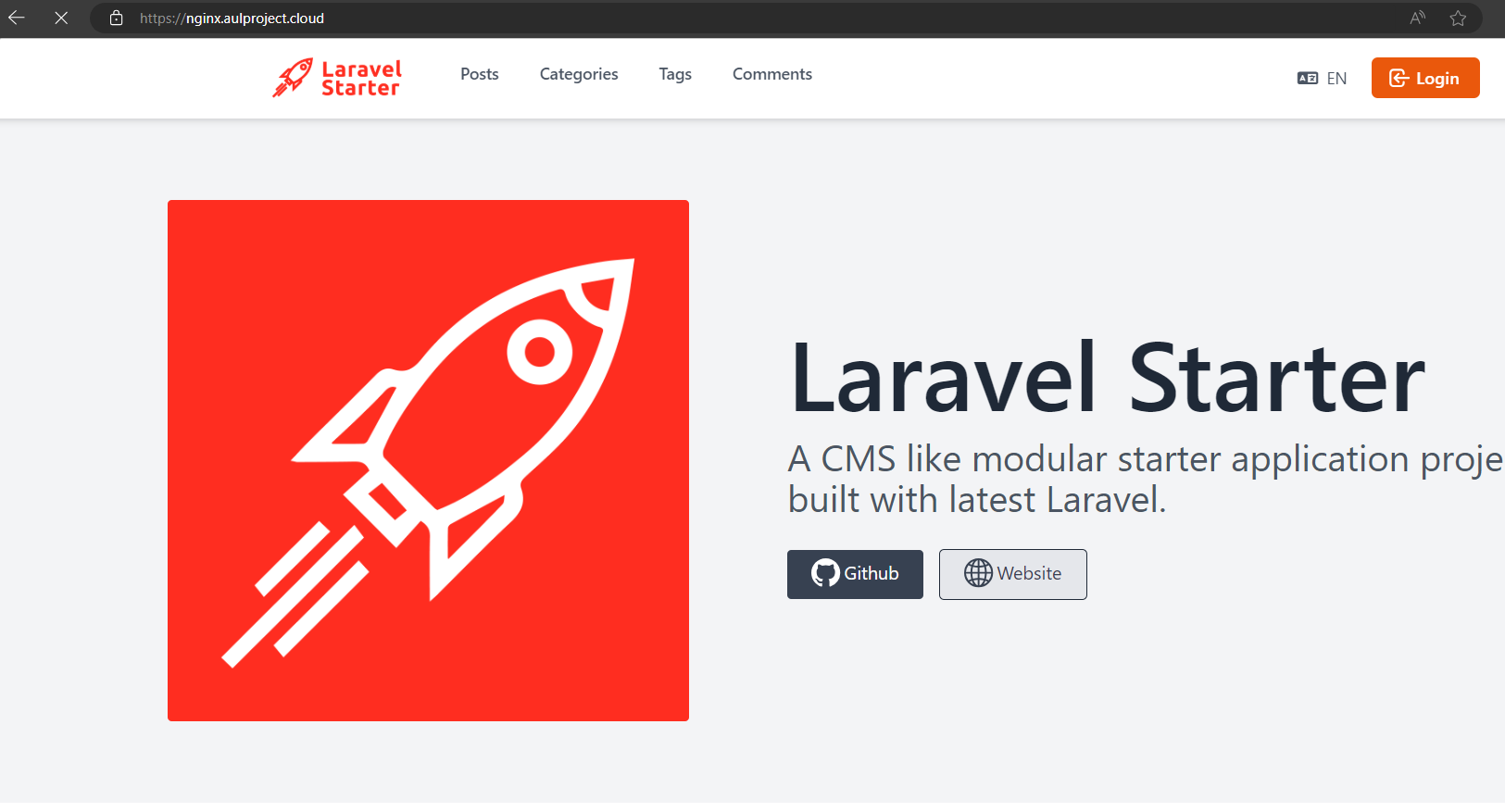
1. **Deploy Aplikasi A dan B**
2. **Deploy aplikasi A**

* Untuk download dan install Laravel-starter lakukan dengan perintah “composer create-project nasirkhan/laravel-starter”
* 
* Setelah proses download Laravel-starter selesai, masuk folder Laravel-starter dengan perintah “cd laraver-starter”
* Setelah masuk kedalam folder laravel-starter lakukan perintah “composer install” untuk melakukan installasi dependencies yang diperlukan pada Web App yang akan digunakan
* 
* Setelah proses installasi dependencies selesai, lakukan setting database untuk laravel-starter
* Masuk mysql dengan perintah “mysql -u root -p”
* Setelah masuk buat user untuk Web App laravel dengan perintah “CREATE USER ‘laravel’@’%’ IDENTIFIED BY ‘password’;”
* Kemudian buat database sama seperti di folder .env dengan perintah CREATE DATABASE laravel\_starter;
* Setelah database dibuat, lakukan pemberian hak akses pada user mysql telah dibuat tadi dengan perintah “GRANT ALL PRIVILEGES ON laravel\_starter.\* TO 'laravel'@'%' IDENTIFIED BY 'password';
* Lakukan perintah FLUSH PRIVILEGES; untuk membersihkan cache dan membuat system membaca ulang tanpa harus merestart server. Setelah itu keluar mysql dengan perintah exit; atau bisa juga dengan shortcut ctrl+d.
* Kemudian setting file .env dengan database dan user yang telah dibuat tadi dengan perintah “nano .env”.
* Sesuaikan di bagian database dengan data yang telah dibuat tadi.
* 
* Setelah itu lakukan perintah “php artisan migrate” untuk mengimpor table dari laravel-starter ke database server.
* Kemudian copy folder laravel-starter ke dalam directory /var/www menggunakan perintah “sudo cp -R ~/laravel-starter /var/www”
* Kemudian ubah ownerhip folder laravel-starter dengan perintah “sudo chown -R auliaarifin89:www-data /var/www/laravel-starter”
* Setelah itu ubah user di nginx.conf mendi user regular dengan perintah “vim /etc/nginx/nginx.conf” ubah bagian user menjadi user reguler
* 
* Kemudian ganti juga di bagian /etc/php8.1/fpm/pool.d/www/conf
* Ubah bagian dibagian user, listen.owner, listen.group





* Kemudian restart nginx dan php fpm dengan perintah
* “sudo systemctl restart nginx”
* “sudo systemctl restart php-fpm.service”
* Setelah itu konfigurasi nginx di bagian /etc/nginx/sites-available/default
* Ubah bagian root directory menjadi /var/www/laravel-starter/public
* Untuk index tambahkan index.php
* Ubah bagian server name menjadi nama domain / alamat ip, untuk ini saya menggunakan domain yang telah saya pointing menggunakan cloudflare
* Lengkap konfigurasinya sebagai berikut
* 
* Dibagian “fastcgi\_pass unix:/run/php/php8.1-fpm.sock” sesuai dengan versi php-fpm yang telah di install di awal tadi
* Kemudian restart nginx dengan perintah “sudo systemctl restart nginx”
* Untuk mengaktifkan https, install certbot untuk installasi sslnya dengan perintah “sudo apt install certbot python3-certbot-nginx”
* Setelah itu lakukan installasi ssl dengan perintah “sudo certbot –nginx -d nama-domain”
* Installasi ssl selesai



1. **Deploy aplikasi B**
2. **Shell script Backup**

#!/bin/bash

**#Configurasi variable**

DATE=$(date +%Y-%m-%d)

BACKUP\_DIR="/home/auliaarifin89/backup"

SOURCE="/var/www/laravel-starter"

GS\_URI="gs://laravel-starter/backup-laravel-starter"

RETENTION\_DAYS=7

**#Melakukan Proses backup database menggunakan mysqldump hasil dari mysqldump di arahkan ke folder /home/auliaarifin89/backup/database**

echo "====>> Backup Database <===="

mysqldump -u laravel -p --verbose laravel\_starter > $BACKUP\_DIR/database/$DATE-laravel\_starter.sql

sleep 2

**#Melakukan Proses backup source-code**

**#perintah tar perintah {c} digunakan untuk membuat archive,**

**#perintah {v} digunakan untuk menampilkan proses pembuatan archive,**

**#perintah {f} digunakan untuk menunjuk kemana file yang ingin di arsip,**

**#hasil dari perintah tar -cvf di arahkan ke folder/home/auliaarifin89/backup/source-code**

echo "====> Source Code Backup <===="

tar -cvf $BACKUP\_DIR/source-code/$DATE-larvel-starter.tar.gz $SOURCE

sleep 3

**#gsutil merupakan perintah untuk mengakses dan mengelola google cloud storage**

**# perintah dibawah ini copy file .sql dari folder /home/auliaarifin89/backup/database/ ke dalam bucket GCS**

echo "====> Upload to Google Cloud Storage <===="

echo "====> DATABASE <===="

gsutil cp $BACKUP\_DIR/database/$DATE-laravel\_starter.sql $GS\_URI/database

sleep 3

**# perintah dibawah ini copy file .tar.gz dari folder /home/auliaarifin89/backup/sourcode/ ke dalamb bucket GCS**

echo "====> Source code <===="

gsutil cp $BACKUP\_DIR/source-code/$DATE-larvel-starter.tar.gz $GS\_URI/source-code

**#Melakukan hapus data yang lebih dari 7 hari di local VM**

#Cleanup local backup

echo "====>>Clean up local directory database<<===="

find $BACKUP\_DIR/database -type f -name "$DATE-laravel\_starter\*" -mtime +$RETENTION\_DAYS -exec rm {} \;

echo "====>>Clean up local directory source-code<<===="

find $BACKUP\_DIR/source-code -type f -name "$DATE-laravel\_starter\*" -mtime +$RETENTION\_DAYS -exec rm {} \;

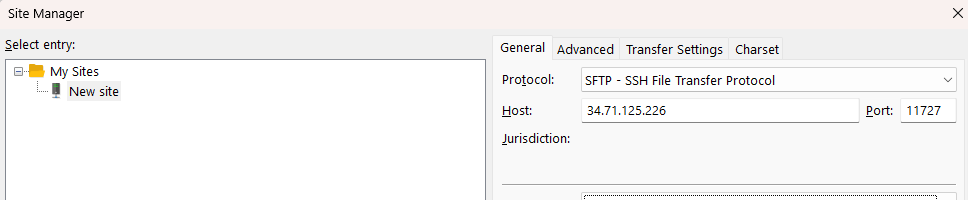
echo "Backup completed and uploaded to GCS. Local backups older than $RETENTION\_DAYS days are deleted."

1. **Developer Access**
2. Akses file menggunakan FileZila

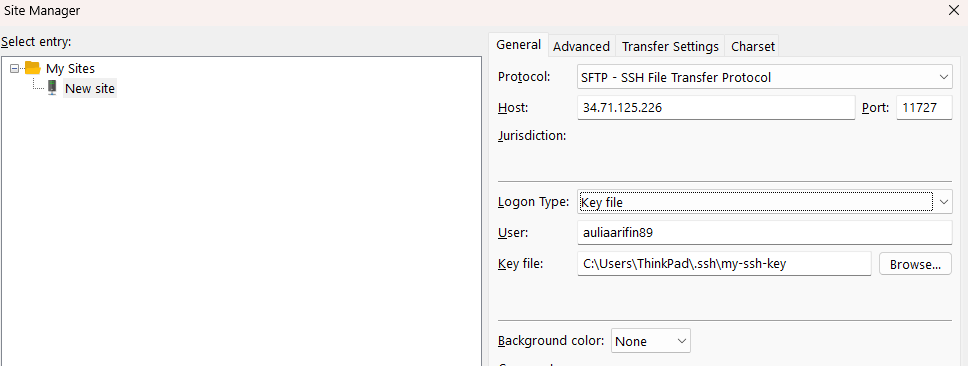
* Buka FileZila

****

* Klik File -> Site Manager
* Pilih protocol SFTP
* Isi Host dengan IP VM
* isi port ssh

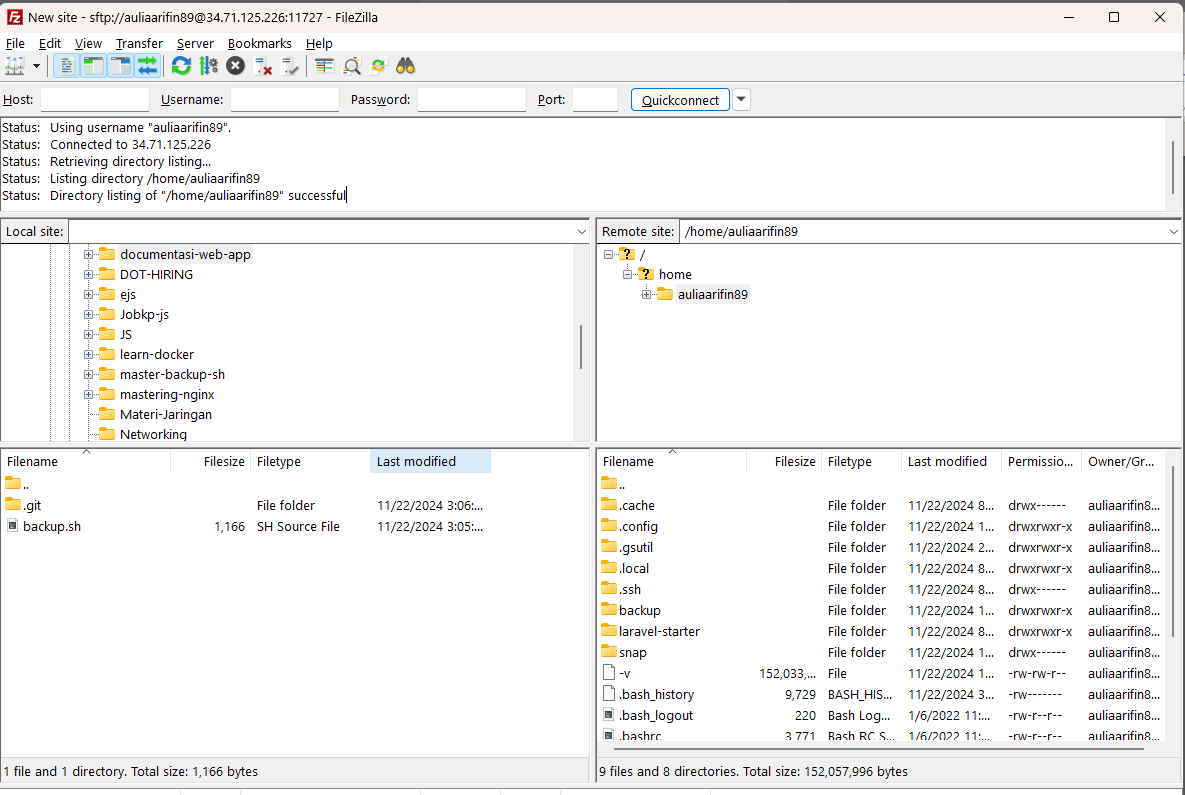
****

* pilih logon type “keyfile”

****

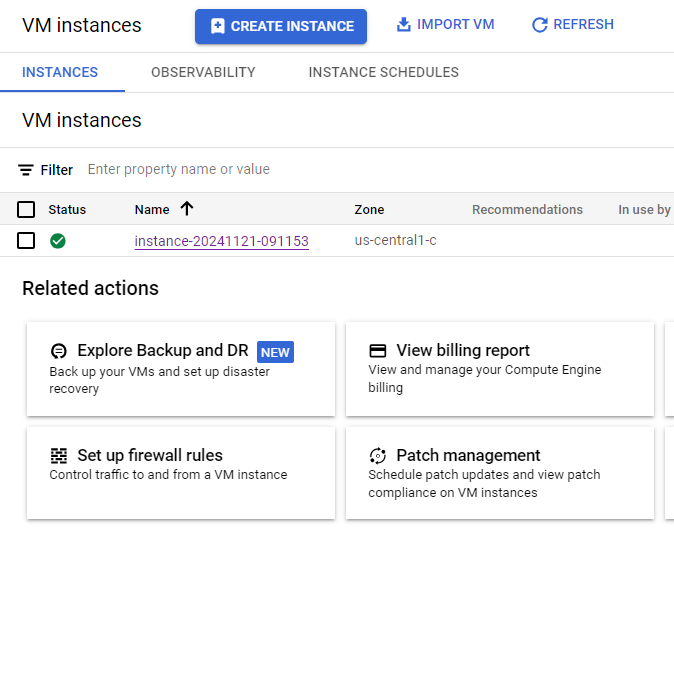


* Browse keyfile
* Kemudian klik Connect

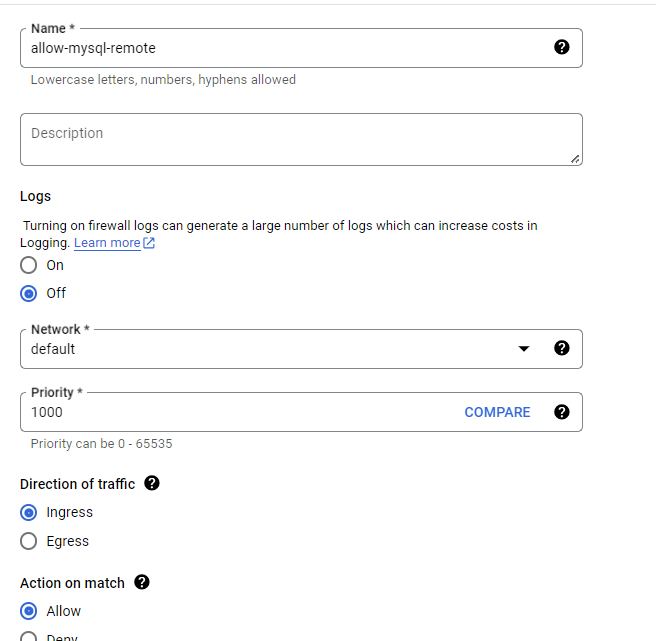
****

1. Akses database dari SQL client via remote

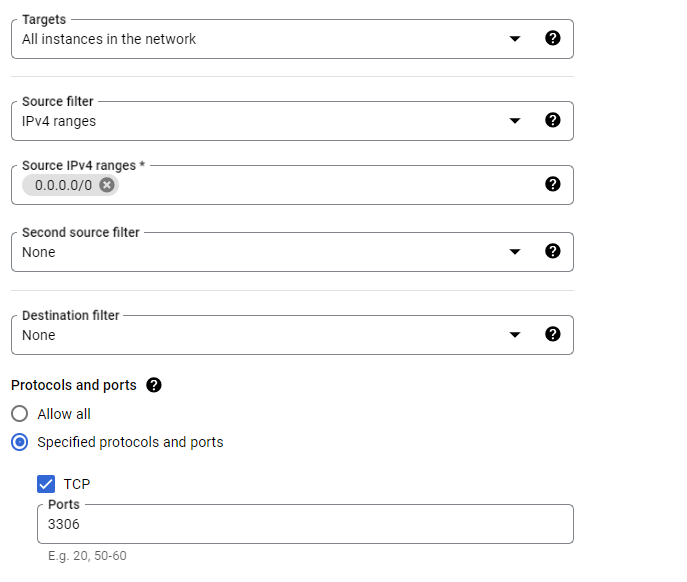
* Lakukan konfigurasi pada firewall instance
* Masuk ke memu Compute Engine -> VM Instance -> Setup Firewall rules

****

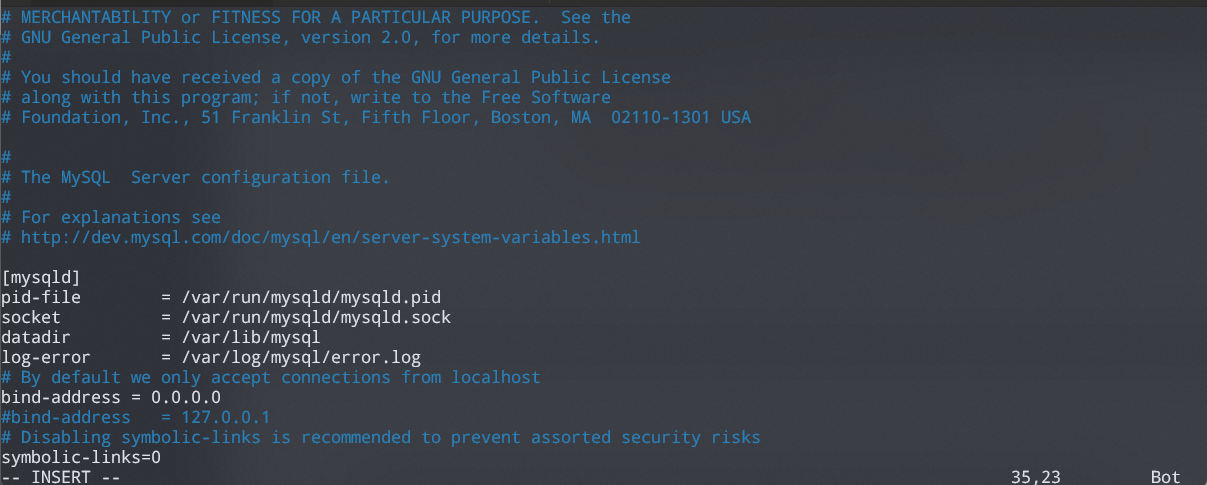
* Klik Create Firewall Rule
* Masukan nama untuk firewall

****

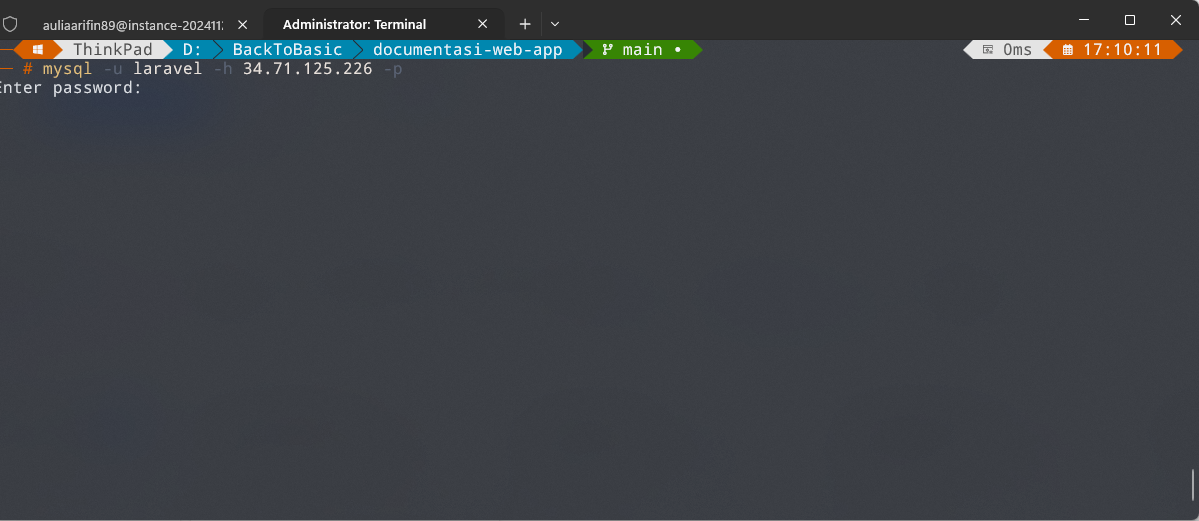
* Untuk menu lain biarkan default saja
* Ganti target ke All instance in the network
* Untuk source IPV4 range masukan 0.0.0.0/0 hal ini bertujuan agar bisa di akses secara public
* Cheklist protocol TCP dan masukan port mysql, default portnya 3306

****

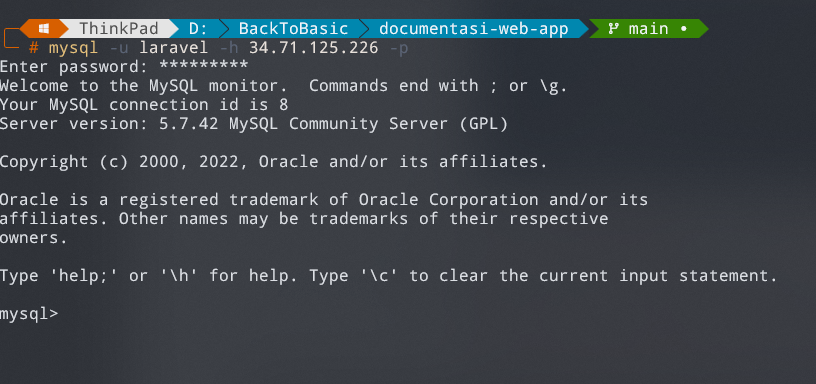
* Kemudian klik Create.
* Setelah Firewall Rules dibuat buka terminal
* Kemudian konfigurasi dibagian mysqld.cnf di vm
* Perintah “nano /etc/mysql/mysql.conf.d/mysqld.cnf”
* Ubah bind-address menjadi 0.0.0.0
* Setelah selsai simpan dengan cara ctrl+x -> y -> enter

****

* Setelah dibuah lakukan restart mysql dengan perintah “sudo systemctl restart mysql”
* Buka terminal di client windows / linux
* Masuk dengan perintah mysql -u “username” -h “IP VM” -p
* Kemudian enter, setelah itu akan muncul menu memasukan password, masukan password dari user mysql tersebut

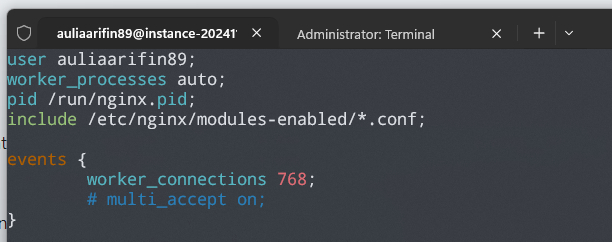
****

* Berhasil masuk mysql

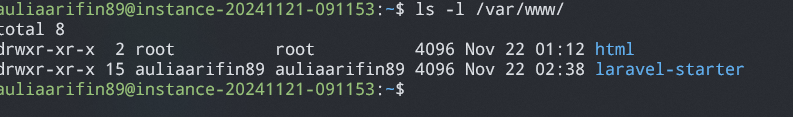


1. **Security Deploy menggunakan user reguler**
2. Aplikasi a

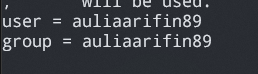
* Nginx.conf

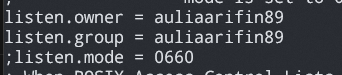


* File ownership /var/www/laravel-starter



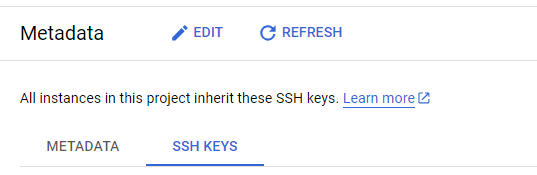
* Php-fpm





1. Aplikasi b
2. Remote SSH menggunakan keyfile

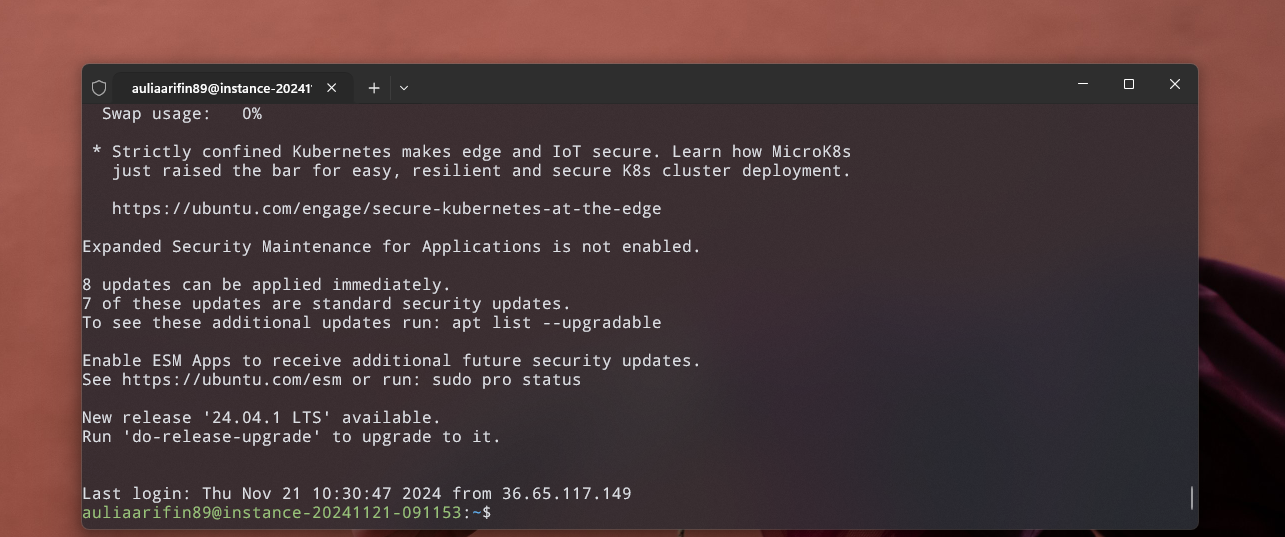
* Buka terminal
* Generate SSH keygen menggunakan email yang telah terdaftar untuk gcp “ssh-keygen -t rsa -f [lokasi untuk menyimpan hasil generate ssh] -C email
* Setelah itu copy isi dari file .pub yang telah di generate
* Masuk ke console.cloud.google.com
* Masuk menu navigasi -> Compute Engine -> Meta Data
* Masuk ke menu SSH keys -> klik Edit di bagian navbar atas



* Masukan isi ike dari ssh.pub tadi lalu klik save

****

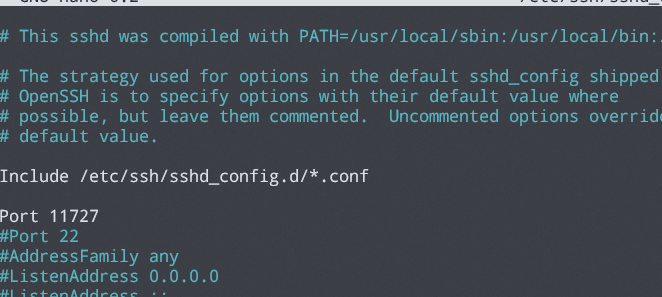
* Melakukan remote ssh via terminal windows dengan cara “ssh -I {ssh\_private\_key} username@ip-instance”



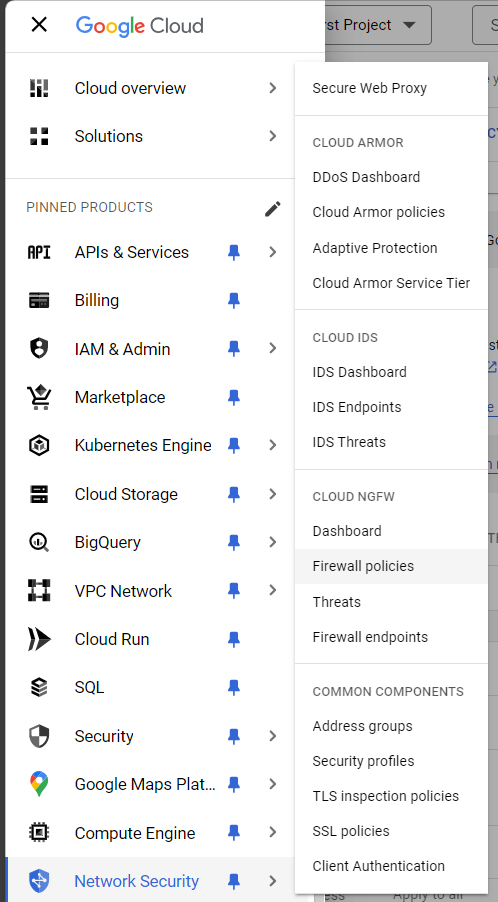
1. Buka akses untuk HTTPS untuk aplikasi A, aplikasi B, dan SSH service

**Pertama membuat rule untuk ssh**

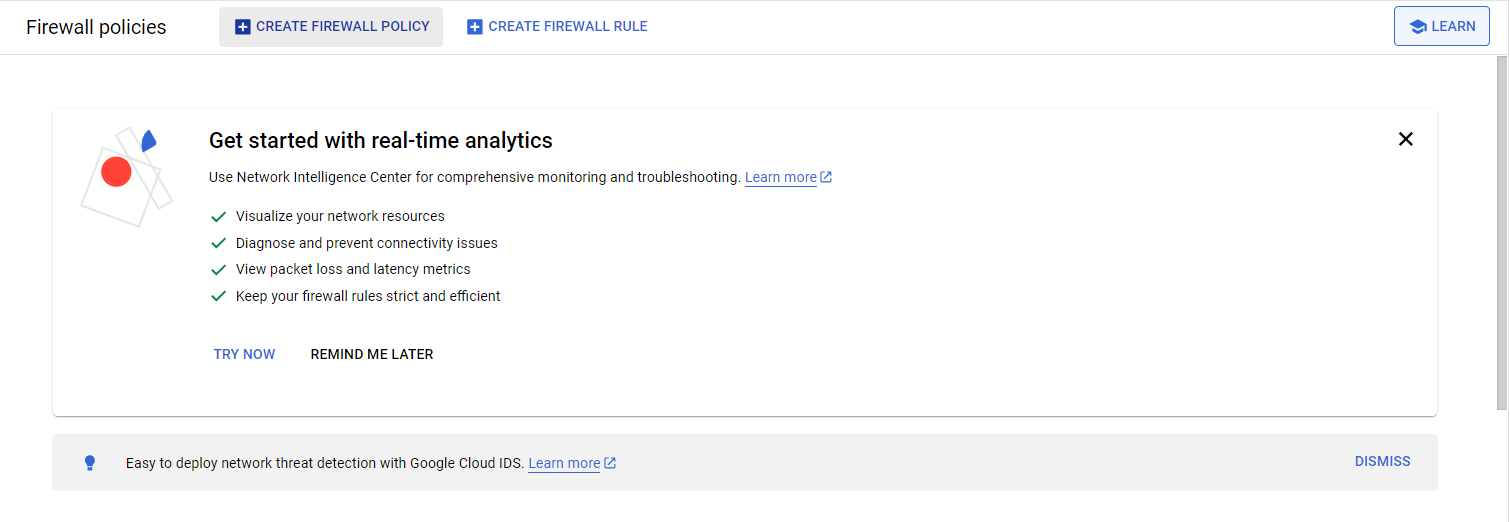
* Ubah settingan port ssh pada instance dengan port 11727 pada file sshd\_config dengan perintah “sudo nano /etc/ssh/sshd\_config”



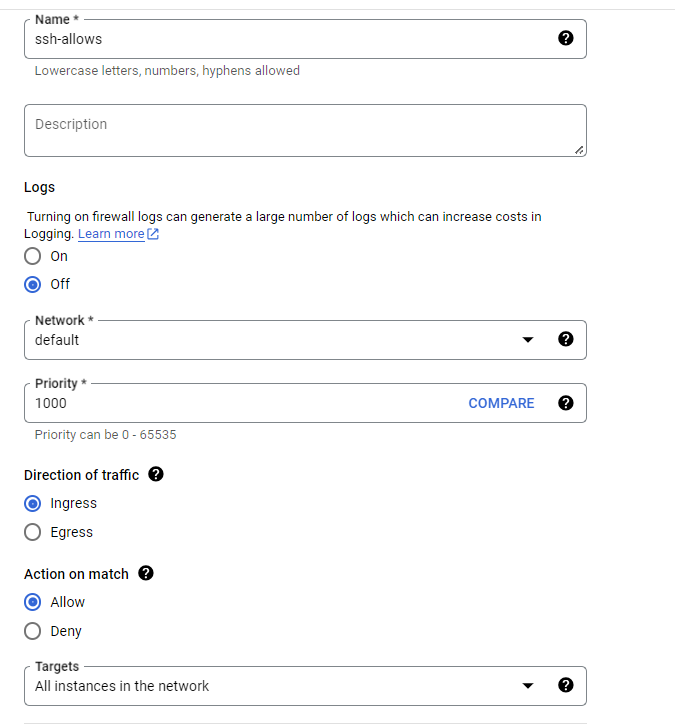
* Setelah itu save dengan cara tekan ctrl+x lalu tekan Y dan enter
* Setelah itu restart ssh-server dengan perintah “sudo systemctl restart ssh”
* Kemudian lakukan konfigurasi di bagian firewall pada instance
* Masuk ke menu Network security -> Firewall Policies



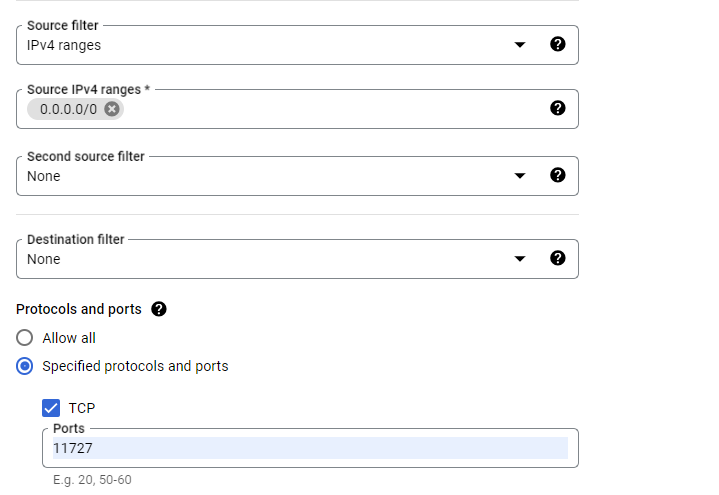
* Create Firewall Rule



* Masukan nama firewall ssh-allows
* Menu Targets pilih All instance in the network ( untuk konfigurasi ini berlaku ke semua instance yang berada pada network defaults)



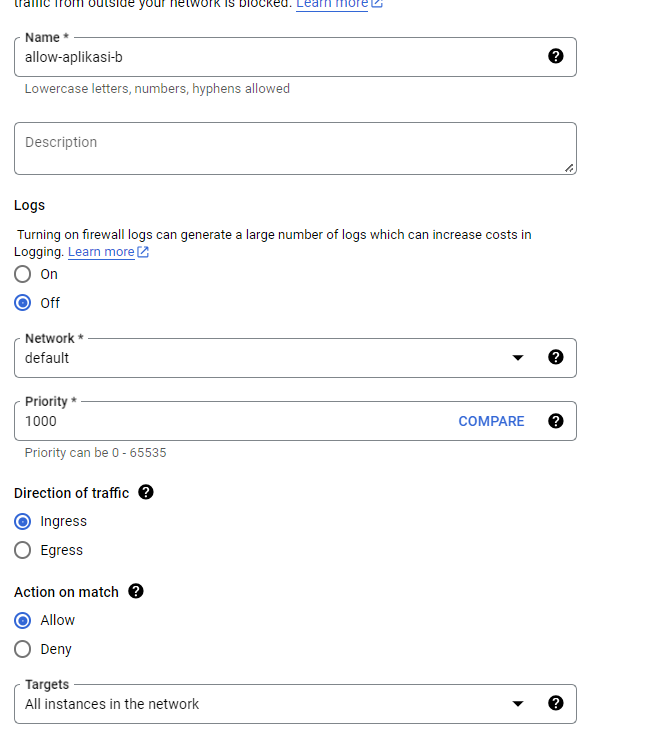
* Isi Source IPv4 ranges menjadi 0.0.0.0/0 ( 0.0.0.0/0 ini bermaksud untuk mengizinkan untuk bisa di akses secara public)
* Protocol and ports checklist di Specified protocols and port
* Lalu cheklist di TCP dan isikan port 11727 untuk port ssh



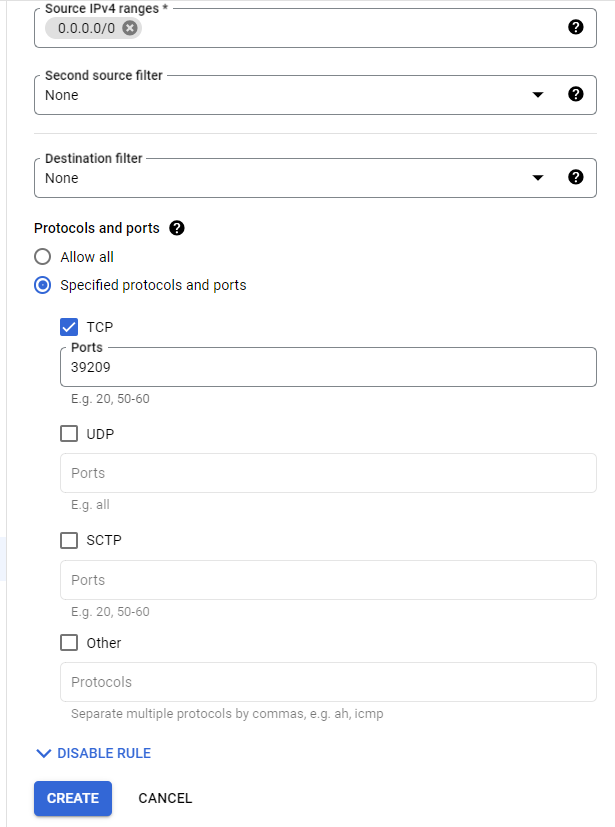
* Kemudian Create

Membuat Firewall Rule untuk aplikasi B port 39209

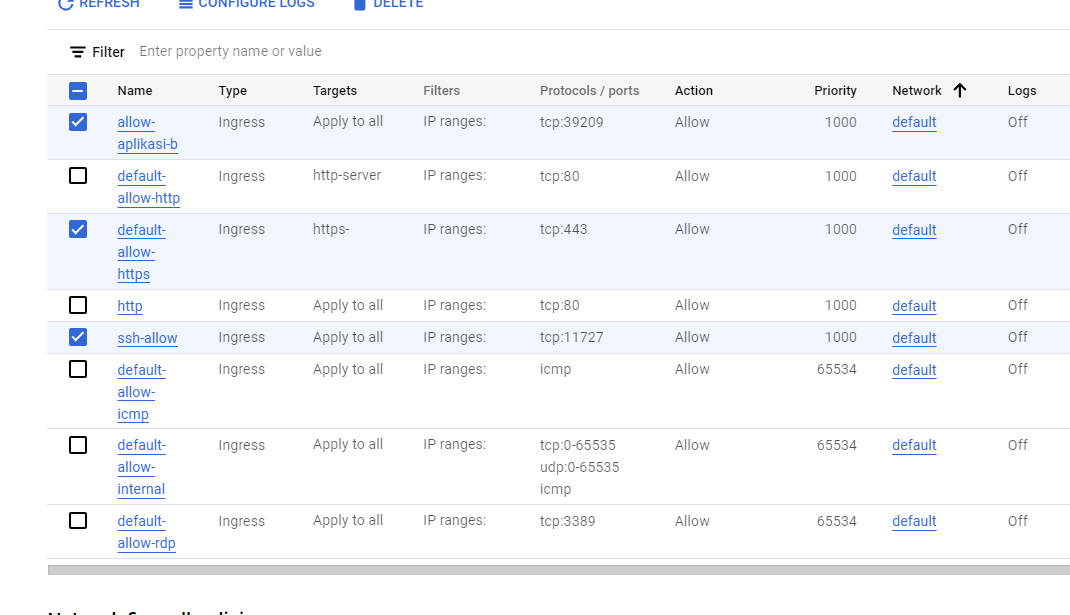
* Secara default Ketika proses pembuatan instance melakukan check list pada Allow Https maka instance otomatis akan membuat rule untuk port https yaitu port 443 jadi untuk aplikasi A sudah otomatis ada Firewall Rule dengan mengizinkan akses ke port 443
* Masuk ke menu Network security -> Firewall Policies
* Create Firewall Rule
* Masukan nama “allow-aplikasi-b”
* Menu Targets pilih All instance in the network ( untuk konfigurasi ini berlaku ke semua instance yang berada pada network defaults)



* Isi Source IPv4 ranges menjadi 0.0.0.0/0 ( 0.0.0.0/0 ini bermaksud untuk mengizinkan untuk bisa di akses secara public)
* Protocol and ports checklist di Specified protocols and port
* Lalu cheklist di TCP dan isikan port 39209 untuk port aplikasi b



* Kemudian Create
* Firewall Rule telah berhasil dibuat



1. **CI/CD**