

## Guía para taller Gitlab - Docker

Libreta: Redes

Creado: 01/10/2020 05:28 p. m.

Actualizado: 01/10/2020 06:16 p. m.

Autor: josuedbm7@gmail.com

---

### ☐ Configuraciones e instalación de imagen de Gitlab

1. Conectividad a internet y SSH de la VM
  1. Agregar o cambiar el adaptador de red
  2. Modificar archivo de Netplan
  3. Aplicar cambios
  4. Modificar SSH\*\*\*
  5. Reiniciar servicio SSH\*\*\*
  6. Reboot a la VM
  7. Entrar a VM por SSH
2. Correr los siguientes comandos
  - o Instalar **imagen** de gitlab

docker pull gitlab/gitlab-ee:latest

## CAMBIAR PUERTO Y PERMITIR ACCESO

```
sudo nano /etc/ssh/sshd_config
PermitRootLogin prohibit-password to PermitRootLogin yes
PasswordAuthentication no to PasswordAuthentication yes
```

### Configurar Firewall para dejar pasar el tráfico un puerto por SSH y puerto 8080

#### Instalar openssh

```
sudo apt-get install openssh-server
```

```
sudo apt-get install openssh-client
```

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

```
sudo ufw enable
```

```
sudo ufw allow ssh
```

```
sudo ufw allow 22
```

```
sudo ufw allow 8080
```

### ☐ Explicación de teoría de Docker (presentación)

### ☐ Crear contenedor de gitlab

```
sudo docker run --detach \
--hostname gitlab.example.com \
--publish 10.10.10.220:8080:80 \
--name gitlab-redes \
--restart always \
--volume /srv/gitlab/config:/etc/gitlab \
--volume /srv/gitlab/logs:/var/log/gitlab \
--volume /srv/gitlab/data:/var/opt/gitlab \
gitlab/gitlab-ee:latest
```

☐ Ingresar a Gitlab desde Navegador 10.10.10.220:8080:80 y configurarlo

1. Cambiar contraseña
2. ssh-keygen **(en VM)**
3. cat ~/.ssh/gitlab.pub **(en VM)**
4. Agregar la ssh-keygen a nuestro Gitlab **(en Gitlab)**
5. Crear un proyecto. **(en Gitlab)**
6. Para VSCode crear un access token: zRMf27RkpX1MsW9Q37FE **(en Gitlab)**
7. Abrir VSCode y descargar extensión de Gitlab Workflow **(en VSCode)**
8. Agregar el access token a VSCode: Ctr + shift + p -> gitlab: access token **(en VSCode)**
9. Clonar el proyecto que creamos **(en VSCode)**
10. Hacer un push a nuestro proyecto **(en VSCode)**
11. Verificar en el navegador si se actualizó nuestro Gitlab **(en Gitlab)**
12. Recapitular lo que hemos hecho hasta ahora.
  1. Instalamos nuestra VM en VMWare
  2. Instalamos una interfaz de red en nuestra máquina Host (Windows) para tener un loopback
  3. Conectamos el loopback desde nuestra VM por medio de bridge
  4. Creamos otra interfaz en nuestra VM para tener conectividad a internet
  5. Le asignamos una IP estática a nuestra interfaz de red bridge (loopback) y dejamos con dhcp la que tiene internet
  6. Verificamos si tenemos conectividad de nuestra VM a nuestra máquina host
  7. Instalamos Docker
  8. Instalamos la imagen de Gitlab
  9. Creamos un contenedor de Docker
  10. Abrimos Docker en el navegador (sacamos el tráfico de datos por un puerto desde Docker hacia un puerto de nuestra máquina (VM) y ya que ese puerto está conectado a nuestra máquina host, podemos abrir ese socket en nuestro navegador.
  11. Conectamos VSCode con ese socket
  12. Tenemos conectividad de nuestro host, a nuestra VM. Por lo tanto, podemos visualizar el tráfico de datos de nuestro docker, que está corriendo en VMWare, en una imagen de Ubuntu Server y conecta su puerto virtual a un puerto "real" de la máquina donde está corriendo (ubuntu) y como nosotros conectamos esa interfaz a nuestra máquina host (windows), ¡tenemos conectividad!

Fuente:

<https://www.howtoforge.com/how-to-install-gitlab-server-with-docker-on-ubuntu-1804/>