

DIG ONLINE

<https://www.diggui.com/>

GITHUB

<https://github.com/Z3nTep/Configuraciones>

<https://github.com/karura5/CATSKILLS/>

[Subdominis a AWS - HackMD](#)

[Joan Iznardo - HackMD](#)

[cuellarcarla/terraform](#)

TEMPLATES

[HTML5 UP! Responsive HTML5 and CSS3 Site Templates](#)

TERRAFORM

terraform init

terraform fmt

terraform validate

terraform plan -out aesp.tfplan

terraform apply ...

terraform output

terraform destroy

terraform init → Inicializa el directorio de trabajo y descarga los proveedores necesarios.

terraform plan → Muestra los cambios que se aplicarán en la infraestructura antes de ejecutarlos.

terraform apply → Aplica los cambios definidos en los archivos de configuración.

terraform destroy → Elimina la infraestructura creada con Terraform.

terraform validate → Verifica que la configuración es válida.

terraform fmt → Formatea los archivos de configuración con el estilo estándar.

terraform output → Muestra los valores de salida definidos en la configuración.

terraform state → Gestiona el estado de Terraform (mover, eliminar, visualizar recursos).

terraform taint → Marca un recurso para que se recree en la próxima ejecución de apply.

Alias:

```
sudo vi .bash_aliases
```

## CREDENCIALES

```
tee /tmp/setup-ec2.sh << EOT
sudo snap install aws-cli --classic
sudo snap install terraform --classic
terraform -install-autocomplete
export TF_PLUGIN_CACHE_DIR="$HOME/.terraform.d/plugin-cache"
grep TF_PLUGIN_CACHE_DIR ~/.bashrc || echo 'export
TF_PLUGIN_CACHE_DIR="$HOME/.terraform.d/plugin-cache"' >> ~/.bashrc
EOT
source /tmp/setup-ec2.sh
```

-----

```
tee /tmp/setup-creds.sh << EOT
aws configure set region us-east-1 --profile default
aws configure set output json --profile default
aws configure set aws_access_key_id CHANGEME --profile default
aws configure set aws_secret_access_key CHANGEMETOO --profile default
aws configure set cli_pager "" --profile default
aws configure set cli_history enabled
EOT
source /tmp/setup-creds.sh
```

---

```
sudo nano ~/.aws/credentials
```

---

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s  
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
kubectl version --client
```

Si eso no funciona, pues el de santos pardos.

<https://github.com/santos-pardos/K8s-Eks>

```
git clone https://github.com/cuellarcarla/terraform.git
```

## COMPROBACIONES

```
aws --version
```

```
terraform --version
```

```
docker --version
```

```
docker-compose --version
```

```
kubectl version --client
```

ami-084568db4383264d4

Para archivos .tar.gz o .tgz:

```
bash
```

```
tar -xvzf archivo.tar.gz
```

Esto extrae el contenido del archivo al directorio actual.

Para archivos .zip:

```
bash
```

```
unzip archivo.zip
```

Este comando descomprime el archivo en el directorio actual.

Para archivos .rar (asegúrate de tener instalado unrar):

```
bash
```

```
unrar x archivo.rar
```

Esto descomprimirá el archivo con su estructura completa.

Para archivos .7z:

```
bash
```

```
7z x archivo.7z
```

Es necesario tener el programa p7zip instalado para este tipo de archivos.

```
eval "$(ssh-agent -s)"
```

```
chmod 400 labs..
```

```
Passw0rdRootAcc0unt
```

```
sudo apt install zip -y
```

```
zip -r backup.zip /var/www/html
```

```
tar -czvf backup.tar.gz /var/www/html
```

```
sudo tar -czvf backup.tar.gz /var/www/html
```

```
sudo apt-get install zip -y # Instala la herramienta zip si no la tienes aún
```

```
zip -r nombre_archivo.zip /ruta/a/la/carpeta
```

```
tar -cvf backup.tar /var/www/html
```

```
zip backup.zip backup.tar
```

```
sudo apt update
```

```
sudo apt install p7zip-full -y
```

```
7z a backup.zip /var/www/html
```

```
7z a backup.zip /var/www/html
```

```
sudo mysqldump -u admin -p -h
```

```
wordpress-db.c5y99z4gvpom.us-east-1.rds.amazonaws.com > wordpressdb.sql
```

rtPsvlOgO7Ar8KV4oYy

mysql -u admin -p -h wordpress-db.c5y99z4gvpom.us-east-1.rds.amazonaws.com

mysqldump -u admin -p wordpress -prtPsvlOgO7Ar8KV4oYy -h

wordpress-db.c5y99z4gvpom.us-east-1.rds.amazonaws.com > wordpressdb.sql

aws s3 cp wordpressdb.sql s3://aws-butcket-01/

aws s3 ls s3://aws-butcket-01/

TTvZlBoY9zT21Q28RbwH

fs-0ecb6132e787d4256.efs.us-east-1.amazonaws.com

mysql -u admin -pTTvZlBoY9zT21Q28RbwH -h

wordpress-db.cspfe1inrs2v.us-east-1.rds.amazonaws.com

cat /var/www/html/wordpress/wp-config.php

vi /var/www/html/wordpress/wp-config.php

ls -ld /var/www/html/wordpress

cd /etc/apache2/sites-enabled

systemctl status apache2

systemctl reload apache2