



Pasos a seguir

Ejercicio GitHub – Jenkins – Terraform - AWS



CONTROL DE VERSIONES

Elaborado por: Jonatan Stiven Gutierrez	No. de Versión: 1.0.0
Revisado por:	Fecha de revisión:
Aprobado por:	Fecha de Aprobación:

Historia de Modificaciones

No. de Versión	Fecha de Versión	Autor	Revisado por	Aprobado por	Descripción
1.0.0	05/03/2024	Jonatan Stiven Gutierrez			Documento Original

Lista de distribución

Para	Acción*	Empresa	Firma/Medio de Entrega

* Tipos de acción: Aprobar, Revisar, Informar, Archivar, Complementar, Asistir a junta, Otras (por favor especificar)

Este documento fue elaborado por SETI. Prohibida su reproducción total o parcial sin previa autorización del autor.



Contenido

INTRODUCCION	4
PRERREQUISITOS	4
PASOS A SEGUIR:	5



INTRODUCCION

En este documento, explicaremos el paso a paso a seguir en el un ejercicio práctico de automatización de procesos con GitHub, Jenkins, Terraform, AWS.

PRERREQUISITOS

- Tener una máquina virtual, usar Jenkins mediante un contenedor Docker.
- Tener conocimiento de comandos Linux.



PASOS A SEGUIR:

1. Configurar las credenciales del usuario CLI de AWS en Jenkins.

- Tener disponible las credenciales del usuario CLI.
- Ir a administrar Jenkins.
- Ir a credenciales.
- Deslizamos al final, damos click sobre la palabra global.
- Click en "add credenciales"

Kind	secret text
Scope	Global (Jenkins, nodes, items, all child items, etc...)
Secret	Ingrese el valor
ID	AWS_ACCESS_KEY_ID
Description	AWS_ACCESS_KEY_ID

- Volvemos a dar click en "add credenciales"

Kind	secret text
Scope	Global (Jenkins, nodes, items, all child items, etc...)
Secret	Ingrese el valor
ID	AWS_SECRET_ACCESS_KEY
Description	AWS_SECRET_ACCESS_KEY



2. Verificar si terraform está instalado en el contenedor de Jenkins.

Importante: Tener el contenedor de Jenkins inicializado.

Nota: En el ejemplo se usa como nombre del contenedor jenkins.

- `docker exec jenkins bash`
- `terraform -version`

Nota: Si el comando anterior genera error, entramos con el usuario ROOT en el contenedor.

- `exit`
- `docker exec -u 0 -it jenkins /bin/bash`

Nota: comprobamos si esta creado lo siguiente:

- `cd /var/lib/apt/lists/partial`

Nota: Si no existe, lo creamos y asignamos los permisos.

Nota: Cuando ejecutas el comando `apt-get update` o `apt update`, el gestor de paquetes apt descarga listas de paquetes disponibles en los repositorios configurados en el sistema. Estas listas se almacenan en el directorio `/var/lib/apt/lists/`.

- `mkdir -p /var/lib/apt/lists/partial`
- `chmod 777 /var/lib/apt/lists/partial`
- `ls -ld /var/lib/apt/lists/partial`
- `apt-get update`
- `apt-get install -y wget`
- `wget -version`

Nota: No vayas a descargar el archivo ZIP de Terraform en el directorio raíz `/`, lo cual puede requerir permisos especiales. Te recomendaría cambiar al directorio donde tienes permisos de escritura antes de ejecutar el comando `wget`.

Por ejemplo, puedes cambiar al directorio **/tmp**, que generalmente tiene permisos de escritura para todos los usuarios.

Opcional revisar los permisos y darle `"chmod 777 /tmp"`.

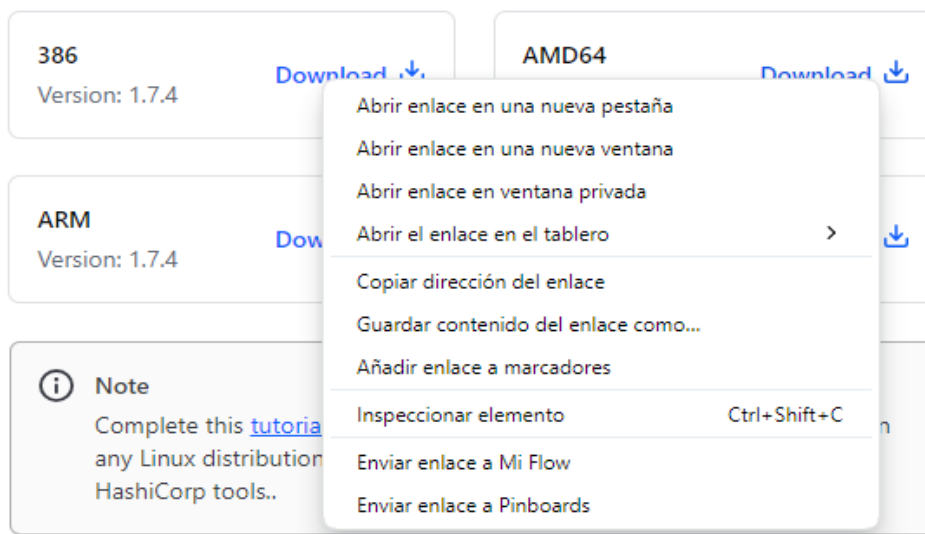
- `cd /tmp`



Nota: Pagina de descarga para Terraform.

Link: <https://developer.hashicorp.com/terraform/install#linux>

Dependiendo la version a descargar damos click derecho y damos click en "Copiar dirección del enlace"



- wget
`https://releases.hashicorp.com/terraform/1.7.4/terraform_1.7.4_linux_amd64.zip`
- unzip terraform_1.7.4_linux_amd64.zip
- mv terraform /usr/local/bin/
- terraform --version
- exit

3. Entramos con nuestro usuario al contenedor y validamos la instalación.

- docker exec -it jenkins bash
- terraform --version