**Hands-On Lab**

Jenkins Pipelines (CI/CD)

**DevSecOps Theory to Practice**

**Contenido**

[1. Objetivo 3](#_Toc75297655)

[2. Instrucciones para levantar el entorno 3](#_Toc75297656)

[3. Codigo fuente de los laboratorios 5](#_Toc75297657)

[4. Ejercicio GitLab CI/CD Básico 5](#_Toc75297658)

# Objetivo

El objetivo de esta sección es indicar los pasos para llevar a cabo el Hands-on de Jenkins en lo que respecta a un CI/CD básico, el objetivo es entender la sintaxis de configuración del pipeline y sus posibilidades.

# Instrucciones para levantar el entorno

Para iniciar el entorno deberás ingresar al portal de Microsoft Azure (<https://portal.azure.com>) con las credenciales que se te proporcionaron, tene en cuenta que la primera vez que inicies sesión se te solicitara un cambio de contraseña por razones de Seguridad.

Una vez dentro del portal veras una imagen como la siguiente:

A screenshot of a computer

Description automatically generated with medium confidence

Deberás ir al icono que se encuentra al lado de la barra de búsqueda:

A picture containing graphical user interface

Description automatically generated

Este icono te permite abrir una sesión de Cloud Shell, esto es una línea de comandos integrada 100 % en el navegador, lo que da como beneficio que no tengas que instalar absolutamente nada en tu computadora. Por ser la primera vez te va a aparecer una pantalla similar a la siguiente:

Graphical user interface, text, application

Description automatically generated

Le indican “Create Storage” y en cuestión de 2 a 3 minutos tendrán la consola lista para ejecutar comandos:

Text

Description automatically generated

Esta terminal la podrán iniciar las veces que sea necesario indicando el botón que se encuentra al lado del cuadro de búsqueda dentro del portal de Microsoft Azure.

# Codigo fuente de los laboratorios

Para facilitar la distribución de los laboratorios autoguiados los mismos han sido subidos a GitLab, con lo cual podrás clonarlos de ahí y hacerlos a medida que se vayan revisando a lo largo del curso. La URL del repositorio es:

<https://github.com/Cloud-Legion/DevSecOps-Theory-To-Practice.git>

La estructura es similar a la siguiente:

Graphical user interface, text, application, email

Description automatically generated

No te preocupes que los instructores te irán guiando con relación a que laboratorios hacer y estarán atentos a cualquier duda que tengas durante el desarrollo de los mismos😊

# Ejercicio Jenkins CI/CD Básico

Llego la hora del ejercicio, habiendo hecho lo anterior estas listo para empezar a meter mano, para esto abrí una sesión de Cloud Shell y espera que se abra la ventana:

Graphical user interface, text, application

Description automatically generated

Dentro de la terminal ejecuta el siguiente comando para clonar el repositorio desde GitHub a tu espacio privado dentro de la terminal anteriormente configurada:

**git clone https://github.com/Cloud-Legion/DevSecOps-Theory-To-Practice.git**

Esto te lo descargara a tu espacio privado y con el comando **ls** podrás visualizar la carpeta de manera local:

Text

Description automatically generated

Hecho esto debemos ingresar a la carpeta indicando la orden **cd DevSecOps-Theory-To-Practice**, podes tabular también en la primera palabra para que la terminal te lo autocomplete 😊, hecho eso veras algo como lo siguiente (que es lo mismo que verías en GitHub)



Deberás ir a la carpeta **/DevSecOps-Theory-To-Practice/Semana 1/Dia 2/Laboratorio Jenkins/Terraform** con la orden **cd**, una vez ahí indica el comando ls para ver el contenido de la carpeta, deberías ver 2 archivos como los siguientes:

****

Estos son manifestos de Terraform y te permiten levantar cualquier tipo de infraestructura sin tener que configurar nada manualmente, para este caso puntual Terraform va a levantar una instancia 100 % configurada de Jenkins para que puedas desarrollar el laboratorio sin tener que hacer configuraciones de instalación, dentro de esta carpeta ejecuta el comando **terraform init**, al terminar la ejecucion deberías ver una pantalla como la siguiente indicando que se inicializo correctamente:

Text

Description automatically generated

Concluido esto ejecuta **terraform apply --auto-approve**, esto hará que se despliegue una máquina virtual dentro de tu misma suscripción de Azure con Jenkins instalado, a esta maquina virtual vas a poder acceder a través de Internet con una IP publica provista automáticamente.

La provision de esta maquina virtual tardara de 5 a 7 minutos debido a configuraciones que deben realizarse, deberás ver una pantalla como la siguiente a lo largo del proceso:

Text

Description automatically generated

Terminada la ejecucion es probable que se sigan ejecutando tareas en segundo plano, pero en este punto deberías tener un output con la IP publica asignada:

Diagram

Description automatically generated with medium confidence

Esta IP publica luego de 5 minutos podrás usarla para iniciar sesión en Jenkins a través de un navegador como Chrome, Firefox o Edge indicando en este caso el puerto 8080.

Graphical user interface, text, application

Description automatically generated

Para iniciar el proceso de configuración debemos entrar por SSH a la instancia:

**ssh azureuser@xxx.xxx.xxx.xxx** (\*) Donde xxx.xxx.xxx.xxx es la IP Publica de tu Jenkins.

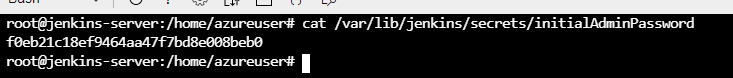
Ingresar con los siguientes datos:

Usuario: **azureuser**

Password: **Password1234!**

Una vez dentro ejecutamos el siguiente comando para obtener la password inicial:

**cat /var/lib/jenkins/secrets/initialAdminPassword**



Con esta password podremos configurar Jenkins:

Graphical user interface, text, application

Description automatically generated

Esto instalara los plugines base de la solucion y no debería tardar más de 5 min.

Graphical user interface

Description automatically generated with medium confidence

Indicar siguiente

Graphical user interface, text, application, email

Description automatically generated

Hecho lo anterior deberá aparecer la pantalla principal de la herramienta:

Graphical user interface, application, Word

Description automatically generated

En este punto estamos listos para crear un nuevo pipeline (ítem) para probar un pipeline en Jenkins, para esto creamos un nuevo ítem con un proyecto libre que llamaremos “demo”:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

En la sección de Source code management seleccionar git como opcion e indicar el siguiente repositorio: <https://github.com/himmAllRight/himmAllRight-source>

Graphical user interface, text, application, email

Description automatically generated

En la sección de **build triggers**, selecciona **Poll SCM** Sin añadir ningún parámetro de programación, se activará cada vez que se detecte una nueva confirmación. Esto es lo que necesitamos.

Graphical user interface, text, application, email

Description automatically generated

En la sección **Build**, hace clic en **Add build step**, y selecciona **Execute shell**. Aquí es donde podemos añadir los comandos del shell para construir el sitio web con hugo. Añadí el siguiente comando:

**hugo -D -F -b "http://127.0.0.1" -d public**

**Graphical user interface, text, application

Description automatically generated**

Para el despliegue, utiliza rsync para copiar los archivos de build al servidor web nginx. Este paso será otro comando de shell, así que lo podés añadir como otro paso de "build". Añadí otro shell Execute y pega el siguiente comando dentro del cuadro de texto:

**sudo cp -R "$WORKSPACE/public/" /var/www/html**

**Graphical user interface, text, application, email

Description automatically generated**

Concluida la configuración iremos a la pantalla principal previo guardado de los cambios, y seleccionaremos “Build”:

A screenshot of a computer

Description automatically generated

Concluido podremos ir a la siguiente opcion para ver el resultado de la ejecucion y todos los detalles pertinentes:

Graphical user interface, application

Description automatically generated

En este caso la ejecucion fue un éxito:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Si vamos a la URL de nuestro Jenkins pero indicando :80/public podremos ver un sitio web levantado con el framework Hugo como parte del laboratorio, esto obviamente es meramente para demostración, el poder que se tiene a la hora de armar pipelines:

Graphical user interface, text, application, email

Description automatically generated

Concluido el laboratorio ejecuta dentro del Shell de Azure la instrucción terraform **destroy –auto-approve** para borrar la maquina virtual usada en este laboratorio, esto debe realizarse con todas a excepción de la de GitLab, la cual será utilizada en otros laboratorios.

Text

Description automatically generated