

Desafío 1 - Bootcamp Devops Engineer

Alumno: Ignacio Peretti

Este desafío tiene como objetivo desarrollar un pipeline declarativo en Jenkins que nos permite crear usuarios dentro de un sistema Linux.

Escenario:

El área de seguridad cuenta con un equipo que tiene la responsabilidad de gestionar la creación de usuarios y baja de usuarios en los sistemas de la empresa.

Debido a que en el último tiempo hay muchos ingresos algunos operadores cometieron algunos errores y debido a esto nos encomendaron generar un job en jenkins que permita generar estos usuarios.

Requisitos:

1. El operador de seguridad desea que al momento de crear el usuario se tomen estos datos de entrada:

Dato	Descripción
Login	Es un identificador único que se compone del nombre y el apellido
Nombre y Apellido	Nombre y Apellido del usuario
Departamento	Este es el grupo que corresponde al área del usuario. Estos grupos son contabilidad,
	finanzas, tecnología.

2. Requiere que la automatización genere un password temporal que sea asignado al usuario, que luego el usuario final deberá cambiar en el primer inicio de sesión.

3. El operador requiere que pueda obtener la password temporal para copiarla y enviarla por email al usuario final.

Resolución paso número 1

Empezamos definiendo LOGIN y PASSWORD.

Seguido iniciamos el pipeline y le pasamos los parametros “name” y “surname” que van a formar

nuestro LOGIN.

seguido, nuestro último parámetro será choice que nos dará la opción de elegir el departamento

donde ubicamos a nuestro usuario.

```
def LOGIN
def PASSWORD
pipeline {
  agent any
  parameters {
    string(name: 'name', defaultValue: '', description: 'Nombre')
    string(name: 'surname', defaultValue: '', description: 'Apellido')
    choice(name: 'departament', choices: ['contabilidad', 'finanzas', 'tecnologia'], description: 'Selecciona el departamento')
  }
}
```

Comenzamos con los STAGES siendo el primero para la creación del usuario y asignarle un grupo.

En este paso se concatenan los parámetros NAME y SURNAME para formar el LOGIN.

la siguiente ejecución será la creación del usuario, la cual se realiza mediante el comando “**sudo useradd -m -c**” y le pasamos los parámetros NAME y SURNAME seguido ubicamos al usuario en un departamento con el parámetro DEPARTAMENT más nuestro LOGIN.

Por último en este paso asignamos el usuario a un departamento utilizando el comando “**sudo usermod -aG**” junto con el parámetro DEPARTAMENT y nuestro LOGIN ya formado.

```
stages {
  stage("Creación de usuario y asignamiento de grupo") {
    steps {
      script {
        //Se concatena el nombre y apellido para formar el login
        LOGIN = "${params.name}${params.surname}"
      }

      //Creación del usuario
      sh "sudo useradd -m -c '${params.name} ${params.surname}, Departamento de ${params.departament}' ${LOGIN}"

      //Se asigna el usuario a un grupo (departamento)
      sh "sudo usermod -aG ${params.departament} ${LOGIN}"
    }
  }
}
```

Resolución paso número 2

Comenzamos con el paso número 2 la generación de una contraseña.

A lo anteriormente definido como PASSWORD le voy a pasar el script para que ejecute en la consola el comando “**openssl rand -base64 12**” que lo que nos dará va a ser una contraseña aleatoria.

Siguiente la asignamos la contraseña aleatoria-temporal al usuario con los siguientes comando a ejecutar en consola:

"echo '\${LOGIN}:\${PASSWORD}' | sudo chpasswd"

"sudo passwd -e \${LOGIN}"

Con esto ya asignamos la contraseña a nuestro usuario.

```
stage("Generación de password") {
  steps {
    script {
      //Se genera una password aleatoria
      PASSWORD = sh(script: 'openssl rand -base64 12', returnStdout: true).trim()
    }

    //Se asigna la password aleatoria-temporal al usuario
    sh "echo '${LOGIN}:${PASSWORD}' | sudo chpasswd"
    sh "sudo passwd -e ${LOGIN}"
  }
}
```

Resolución paso número 3

Este es el último paso, la visualización de la contraseña que le asignaremos al usuario.

Lo lograremos haciendo un ECHO pasándole nuestro LOGIN que mostrará el nombre y apellido del usuario junto con PASSWORD que nos enseñará la contraseña aleatoria.

```
stage("Visualización de password temporal") {
  steps {
    //Se muestra la contraseña que será enviada al usuario
    echo "La contraseña del usuario ${LOGIN} es ${PASSWORD}"
  }
}
```


Así finalizamos el desafío número 1.

1 - Creamos un usuario y tomamos sus datos de entrada (Nombre, Apellido, Login y Departamento)

2 - Automatizamos y generamos una password aleatoria, temporal que asignamos a un usuario.

3 - Visualizamos la password temporal para que el operador pueda copiarla y entregarla al usuario final.

Imágenes del correcto funcionamiento del script.

 **Salida de consola**

```
Started by user Ignacio Peretti
Obtained Jenkinsfile-desafio from git https://github.com/IgnacioPeretti/educacionitdevops.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/desafio1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential githubdemo
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/desafio1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/IgnacioPeretti/educacionitdevops.git # timeout=10
Fetching upstream changes from https://github.com/IgnacioPeretti/educacionitdevops.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials Token beta educacionIT
> git fetch --tags --force --progress -- https://github.com/IgnacioPeretti/educacionitdevops.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision fd971d1fc06b8310c0ee0f9ecb116c1d85525966 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f fd971d1fc06b8310c0ee0f9ecb116c1d85525966 # timeout=10
Commit message: "Desafio listo"
> git rev-list --no-walk d36d4e6a10099d554a6f8b6f47fc8cabb05522e0 # timeout=10
```

```
> git rev-list --no-walk d36d4e6a10099d554a6f8b6f47fc8cabb05522e0 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Creaci3n de usuario y asignamiento de grupo)
[Pipeline] script
[Pipeline] {
[Pipeline] }
[Pipeline] // script
[Pipeline] sh
+ sudo useradd -m -c Ignacio Peretti, Departamento de contabilidad IgnacioPeretti Primer Paso
[Pipeline] sh
+ sudo usermod -aG contabilidad IgnacioPeretti Primer Paso
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Generaci3n de password)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ openssl rand -base64 12 Segundo Paso
[Pipeline] }
[Pipeline] // script
[Pipeline] sh
+ sudo chpasswd
+ echo IgnacioPeretti:V6Pvmg70i/Cxgs4d Segundo Paso
[Pipeline] sh
+ sudo passwd -e IgnacioPeretti Segundo Paso
```

```
[Pipeline] sh
+ sudo chpasswd
+ echo IgnacioPeretti:V6Pvmg70i/Cxgs4d
[Pipeline] sh
+ sudo passwd -e IgnacioPeretti
passwd: password expiry information changed.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Visualización de password temporal)
[Pipeline] echo
La contraseña del usuario IgnacioPeretti es V6Pvmg70i/Cxgs4d
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Segundo Paso

Tercer Paso