

Desafío # 7

Realizado por: Joselin Teixeira

Fecha de entrega: 29/07/2024

Escenario:

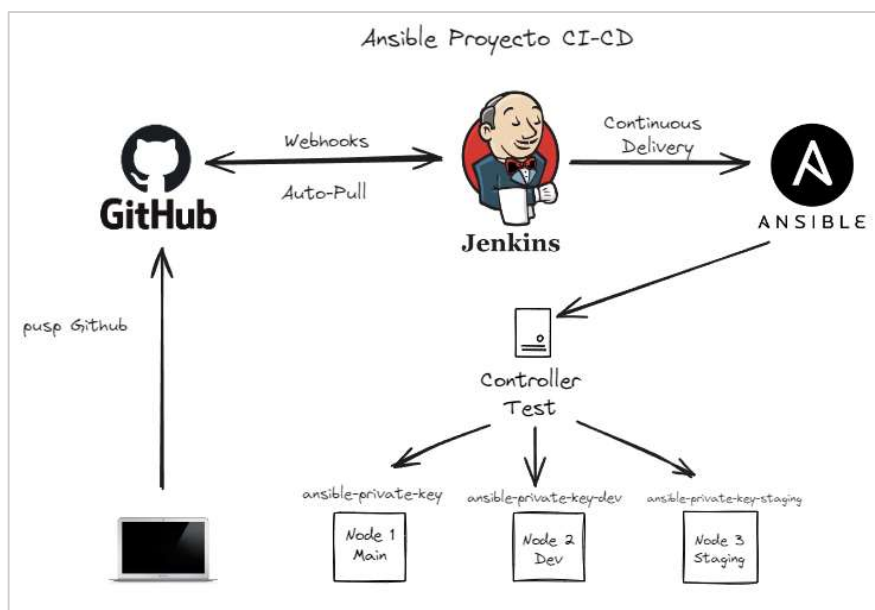
En nuestro sprint anterior trabajamos en modularizar nuestro proyecto de Ansible. Luego del éxito de este cambio el equipo identificó que existen playbook que solo quedan en el entorno local de los desarrolladores y es por esto por lo que necesitamos crear un pipeline CI/CD para forzar que la única manera de ejecutar e interactuar de ejecutar los playbook en los distintos entornos sea desde un código que sea leído de un repo y ejecutado desde un controlador de Jenkins.

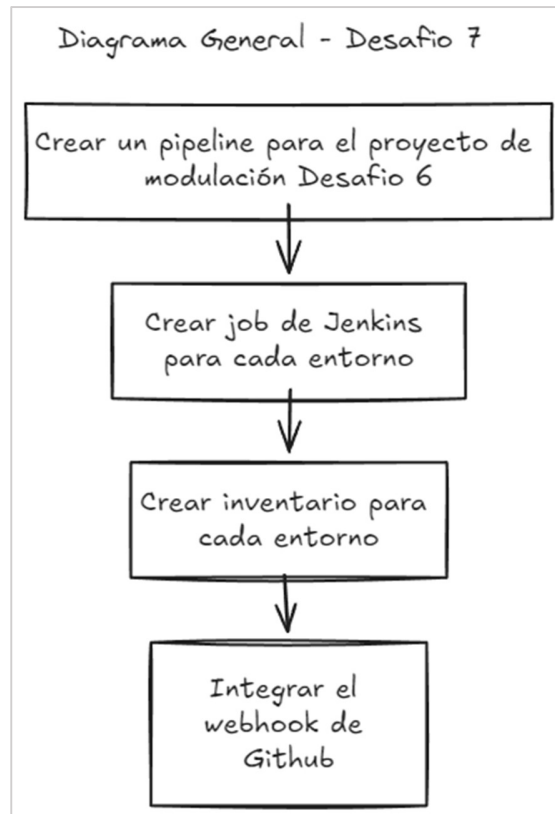
Esta mejora en el proceso busca:

- Aumentar la seguridad ya que las credenciales de acceso solo estarán disponibles en el controlador de Jenkins y no será necesario que un desarrollador disponga de las credenciales de forma local.
- Permite forzar a los desarrolladores a crear sus entornos de trabajo local evitando que el proceso de desarrollo se ejecute en entornos compartidos.
- Todos los cambios podrán ser gestionados como una pieza de software y podrán pasar por un proceso de revisión de PR.
- Mantener la estrategia de branches que vienen utilizando:
 - DEV: entorno de desarrollo, es un entorno donde se llevan cambios frecuentes y los miembros del equipo tienen mayor libertad para ver los cambios en equipos reales.
 - STAGING: es un entorno donde se integran y prueban todos los cambios, los miembros del equipo no cuentan con acceso a los equipos y solo reciben feedback mediante las herramientas CI/CD y los cambios son aplicados solo mediante un PR (pull request).
 - MAIN: es nuestro entorno productivo, no contamos acceso y los cambios

Requisitos:

1. Implementar un pipeline en jenkins para el proyecto visto en el desafío #6.
2. Crear job de jenkins por cada entorno de trabajo y que se ejecute desde su respectiva branch de github [dev,staging,main].
3. Ajustar el inventario para que pueda soportar un conjunto de máquinas para cada entorno.
4. Integrar el webhook de github para que cada vez que se produzca un cambio en el repositorio se ejecute el job del playbook.





Inicialmente crearemos un pipeline para trabajar con el entorno modularizado en el **Desafio 6**. Este pipeline nos servirá para verificar que la estructura utilizada anteriormente se ejecute sin inconvenientes y así luego crear en base a este el pipeline multibranch:

Creación Pipeline: desafio7-ansible-jenkins

- Escogeremos SCM: Git
- Colocamos el Repositorio URL: <https://github.com/Joselintp/devops-engineer.git>
- Seleccionamos las credenciales de GitHub.

Panel de Control > desafio7-ansible-jenkins > Configuration

Configure

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

<https://github.com/Joselintp/devops-engineer.git>

Credentials ?

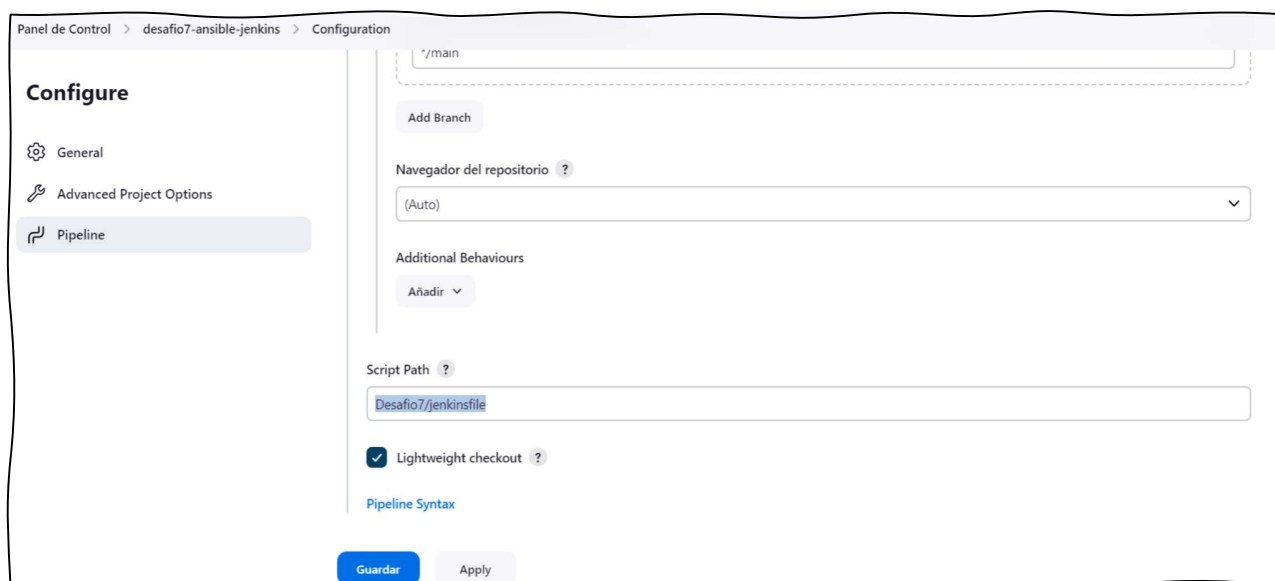
Joselintp/***** (JT-github)

+ Add

Avanzado

Guardar Apply

- Luego editamos Branch Specifier: */main
- Escribimos Script Path: Desafio7/jenkinsfile
- Presionamos: Apply y luego Guardar.



El archivo jenkinsfile se encuentra en: <https://github.com/Joselintp/devops-engineer/blob/main/Desafio7/jenkinsfile>

A continuación, se explica los elementos que componen el pipeline:

```
1  pipeline {
2      agent { label 'ansible-controller' }
3      environment {
4          ANSIBLE_PRIVATE_KEY=credentials('ansible-private-key')
5          ANSIBLE_CONFIG='./Desafio7/ansible/ansible.cfg'
6      }
7      stages {
8          stage('Run Ansible Playbook from Jenkins') {
9              steps {
10                 sh 'ansible-playbook -i ./Desafio7/ansible/inventory.ini --private-key=$ANSIBLE_PRIVATE_KEY ./Desafio7/ansible/main.yml'
11             }
12         }
13     }
14 }
```

agent

- Especifica en qué nodo o agente se ejecutará el pipeline.
- En este caso, se usa el label 'ansible-controller', lo que significa que se ejecutará en un nodo con la etiqueta 'ansible-controller'.

environment

- Define variables de entorno que estarán disponibles durante la ejecución del pipeline.
- ANSIBLE_PRIVATE_KEY: Obtiene la clave privada de SSH desde las credenciales de Jenkins con el ID 'ansible-private-key'. Esto se usa para autenticarse en los hosts de Ansible.
- ANSIBLE_CONFIG: Especifica la ruta al archivo de configuración de Ansible, en este caso './Desafio7/ansible/ansible.cfg'.

stages

- Agrupa los diferentes pasos o etapas del pipeline.

stage ('Run Ansible Playbook from Jenkins')

- Define una etapa llamada "Run Ansible Playbook from Jenkins".

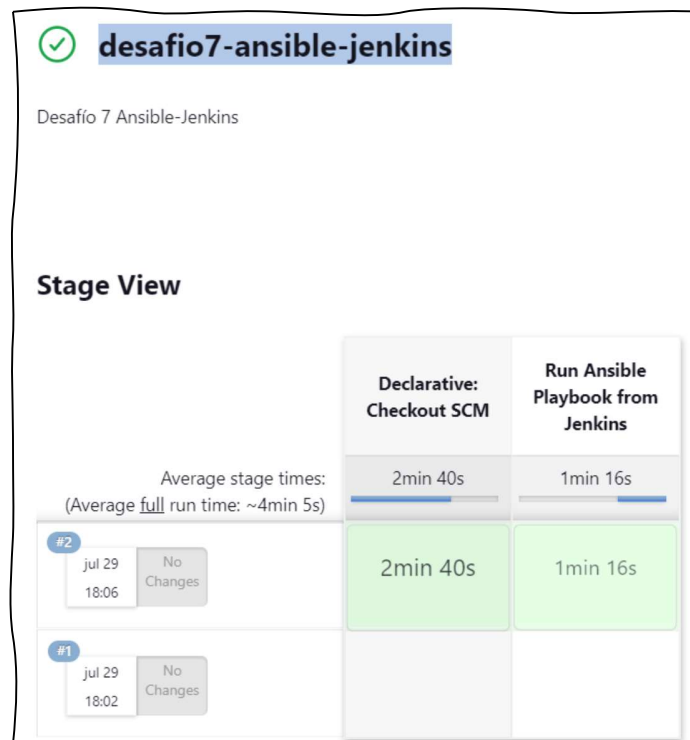
steps

- Especifica los pasos a ejecutar dentro de la etapa.

```
sh 'ansible-playbook...'
```

- Ejecuta un comando de shell (sh) para correr un playbook de Ansible.
- `-i ./Desafio7/ansible/inventory.ini`: Especifica el archivo de inventario de Ansible, que contiene la lista de hosts.
- `--private-key=$ANSIBLE_PRIVATE_KEY`: Usa la clave privada de SSH almacenada en la variable de entorno `ANSIBLE_PRIVATE_KEY` para autenticarse en los hosts.
- `./Desafio7/ansible/main.yml`: Indica el playbook de Ansible principal a ejecutar, ubicado en `./Desafio7/ansible/main.yml`.

Para corroborar que el pipeline se ejecuta correctamente, ejecutamos el mismo como se muestra a continuación:

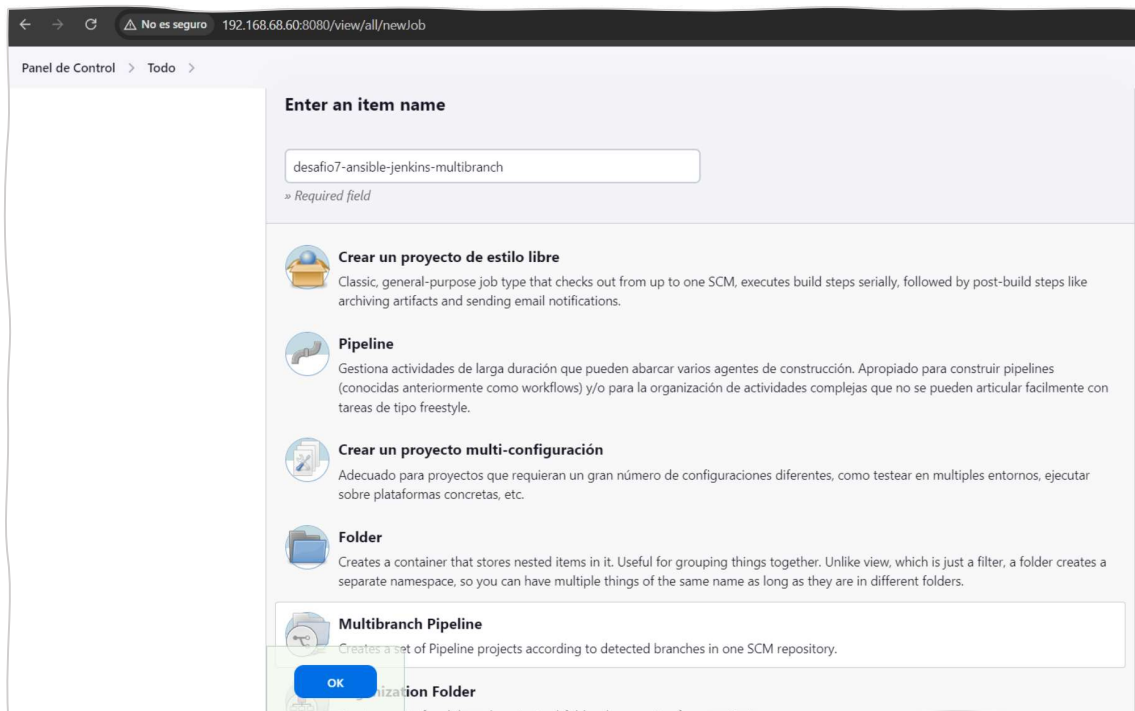


En resumen, este Jenkinsfile configura un pipeline de Jenkins que ejecuta un playbook de Ansible llamado 'main.yml' ubicado en `./Desafio7/ansible/main.yml`. Utiliza la clave privada de SSH almacenada en las credenciales de Jenkins con el ID `'ansible-private-key'` para autenticarse en los hosts especificados en el archivo de inventario `./Desafio7/ansible/inventory.ini`. El pipeline se ejecutará en un nodo con la etiqueta `'ansible-controller'`.

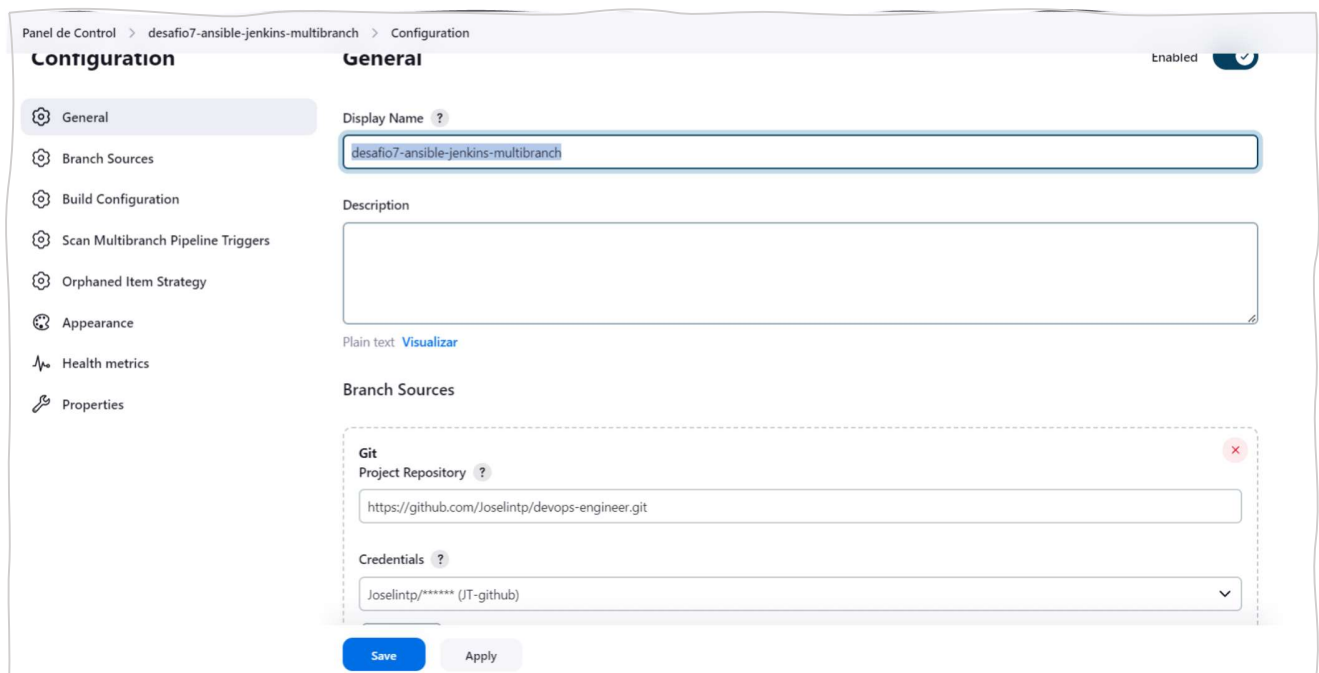
Ahora crearemos el pipeline que nos permitirá trabajar en los entornos: Dev, Staging y Main

Creación Pipeline: desafio7-ansible-jenkins-multibranch

- En Panel de Control seleccionamos +Nueva Tarea
- Indicamos el nombre del pipeline: desafio7-ansible-jenkins-multibranch
- Seleccionamos Multibranch Pipeline
- Presionamos: OK



- Colocamos el Repositorio URL: <https://github.com/Joselintp/devops-engineer.git>
- Seleccionamos las credenciales de GitHub.



- Mode: by Jenkinsfile
- Escribimos Script Path: Desafio7/jenkinsfilebybranch
- Presionamos: Apply, luego Save

Panel de Control > desafio7-ansible-jenkins-multibranch > Configuration

Build Configuration

Mode
by Jenkinsfile

Script Path ?
Desafio7/jenkinsfilebybranch

Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run ?

Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

☐ Abort builds ?

Save Apply

El archivo jenkinsfile se encuentra en: <https://github.com/Joselintp/devops-engineer/blob/main/Desafio7/jenkinsfilebybranch>

A continuación, se explica los elementos que componen el pipeline:

```

Code Blame 41 lines (41 loc) · 1.46 KB Code 55% faster with GitHub Copilot
1 pipeline {
2   agent { label 'ansible-controller' }
3   environment {
4     ANSIBLE_CONFIG = './Desafio7/ansible/ansible.cfg'
5   }
6   stages {
7     stage('Run Ansible Playbook from Jenkins for development') {
8       when {
9         branch 'dev'
10      }
11      environment {
12        ANSIBLE_PRIVATE_KEY = credentials('ansible-private-key-dev')
13      }
14      steps {
15        sh 'ansible-playbook -i ./Desafio7/ansible/inventory.ini --private-key=$ANSIBLE_PRIVATE_KEY ./Desafio7/ansible/main.yml --limit development'
16      }
17    }
18    stage('Run Ansible Playbook from Jenkins for staging') {
19      when {
20        branch 'staging'
21      }
22      environment {
23        ANSIBLE_PRIVATE_KEY = credentials('ansible-private-key-staging')
24      }
25      steps {
26        sh 'ansible-playbook -i ./Desafio7/ansible/inventory.ini --private-key=$ANSIBLE_PRIVATE_KEY ./Desafio7/ansible/main.yml --limit staging'
27      }
28    }
29    stage('Run Ansible Playbook from Jenkins for production') {
30      when {
31        branch 'main'
32      }
33      environment {
34        ANSIBLE_PRIVATE_KEY = credentials('ansible-private-key')
35      }
36      steps {
37        sh 'ansible-playbook -i ./Desafio7/ansible/inventory.ini --private-key=$ANSIBLE_PRIVATE_KEY ./Desafio7/ansible/main.yml --limit production'
38      }
39    }
40  }
41 }

```

pipeline

- define un pipeline de Jenkins, que es una secuencia de pasos para automatizar un proceso.

agent

- Especifica en qué nodo o agente se ejecutará el pipeline.
- En este caso, se usa el label 'ansible-controller', lo que significa que se ejecutará en un nodo con la etiqueta 'ansible-controller'.

environment

- Define variables de entorno que estarán disponibles durante la ejecución del pipeline.
- ANSIBLE_CONFIG: Especifica la ruta al archivo de configuración de Ansible, en este caso './Desafio7/ansible/ansible.cfg'.

stages

- Agrupa los diferentes pasos o etapas del pipeline.

stage ('Run Ansible Playbook from Jenkins for development')

- Define una etapa llamada "Run Ansible Playbook from Jenkins for development".

when

- Especifica una condición para ejecutar la etapa.
- En este caso, la etapa se ejecutará solo cuando la rama actual sea 'dev'.

environment

- Define variables de entorno específicas para esta etapa.
- ANSIBLE_PRIVATE_KEY: Obtiene la clave privada de SSH desde las credenciales de Jenkins con el ID 'ansible-private-key-dev'. Esto se usa para autenticarse en los hosts de Ansible.

steps

- Especifica los pasos a ejecutar dentro de la etapa.


sh 'ansible-playbook...'

- Ejecuta un comando de shell (sh) para correr un playbook de Ansible.
- **-i ./Desafio7/ansible/inventory.ini**: Especifica el archivo de inventario de Ansible, que contiene la lista de hosts.
- **--private-key=\$ANSIBLE_PRIVATE_KEY**: Usa la clave privada de SSH almacenada en la variable de entorno ANSIBLE_PRIVATE_KEY para autenticarse en los hosts.
- **./Desafio7/ansible/main.yml**: Indica el playbook de Ansible principal a ejecutar, ubicado en './Desafio7/ansible/main.yml'.
- **--limit development**: Limita la ejecución del playbook solo a los hosts etiquetados como 'development' en el inventario.

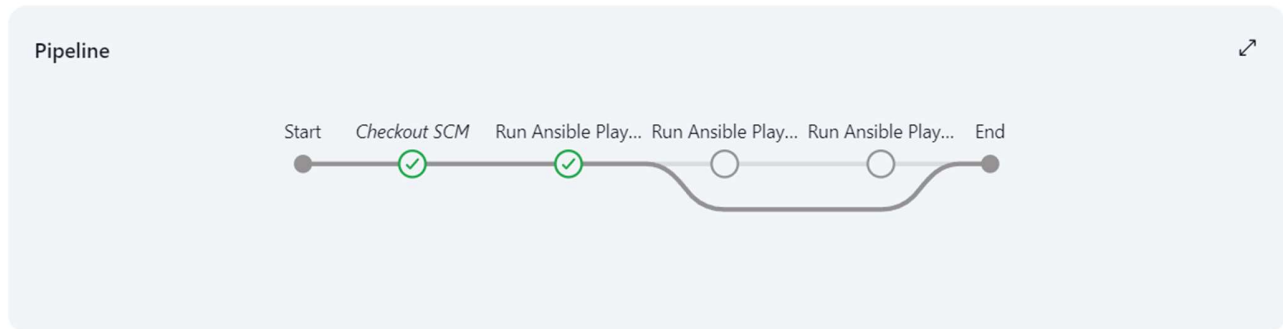
Las siguientes etapas, "Run Ansible Playbook from Jenkins for staging" y "Run Ansible Playbook from Jenkins for production", siguen un patrón similar:

- Se ejecutan cuando la rama actual es 'staging' o 'main', respectivamente.
- Utilizan diferentes credenciales de clave privada de SSH para autenticarse en los hosts.
- Limitan la ejecución del playbook a los hosts etiquetados como 'staging' o 'production' en el inventario.

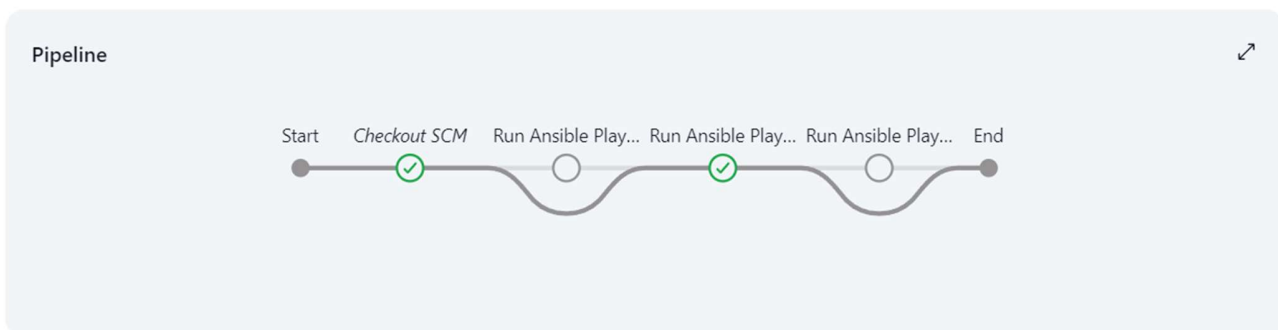
Para corroborar que el pipeline se ejecuta correctamente, ejecutamos el mismo para cada rama como se muestra a continuación:

desafio7-ansible-jenkins-multibranch						
 desafio7-ansible-jenkins-multibranch Disable Multibranch Pipeline						
Branches (3)						
S	W	Name ↓	Último Éxito	Último Fallo	Última Duración	
✓	☀	dev	20 Min #3	3 Hor 46 Min #2	1 Min 13 Seg	▶
✓	☁	main	17 Min #4	3 Hor 46 Min #2	1 Min 9 Seg	▶
✓	☁	staging	3 Min 42 Seg #7	7 Min 29 Seg #4	1 Min 3 Seg	▶

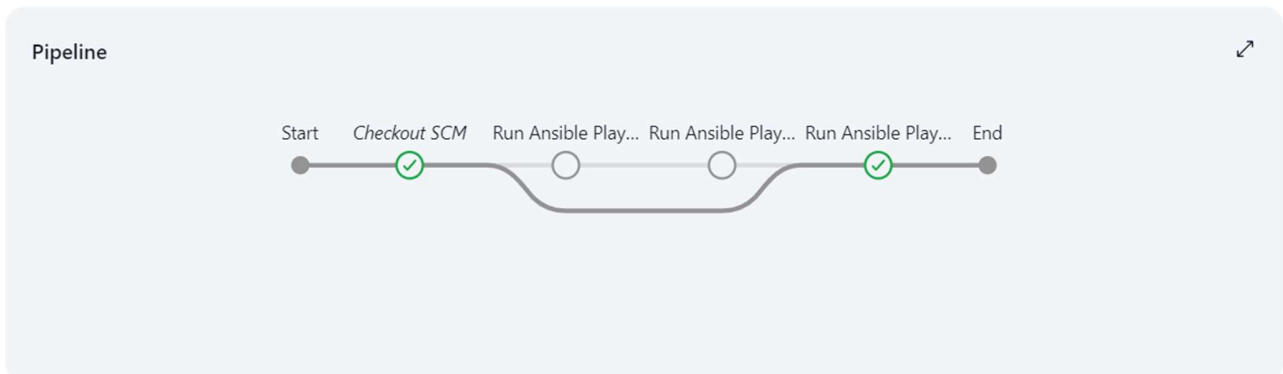
Pipeline Overview 'Run Ansible Playbook from Jenkins for development'



Pipeline Overview 'Run Ansible Playbook from Jenkins for staging'

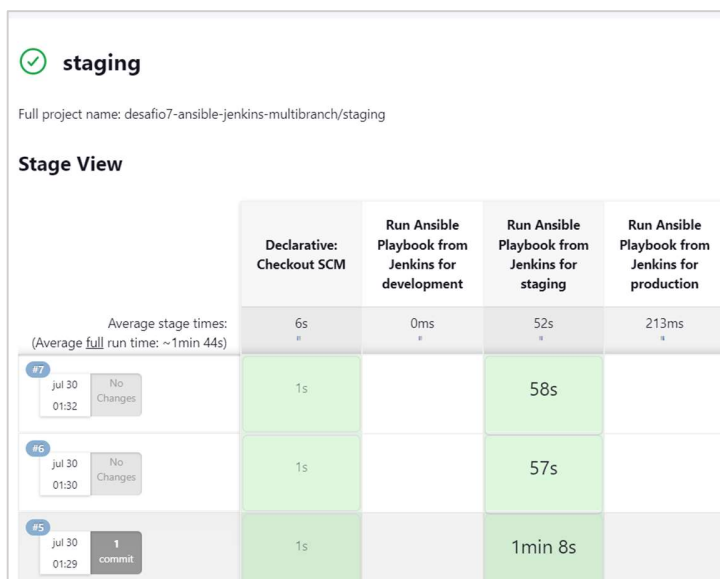
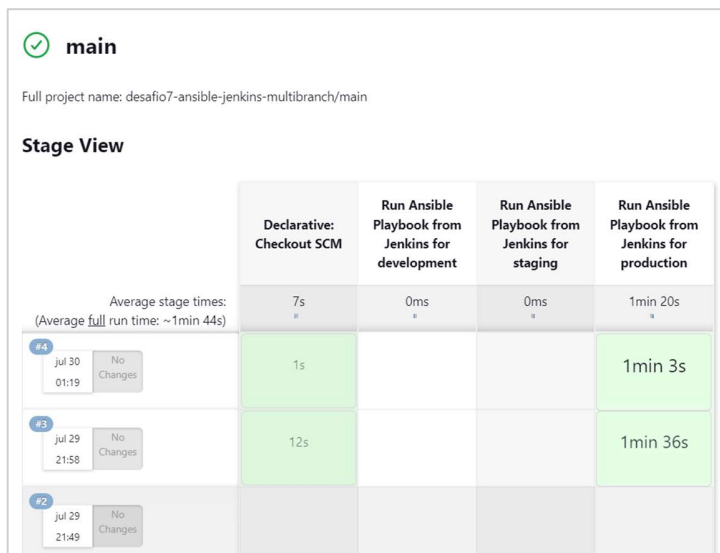
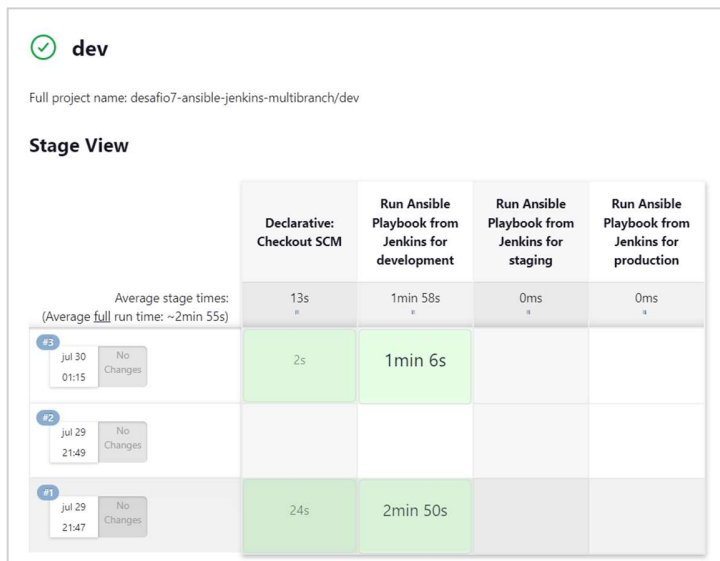


Pipeline Overview 'Run Ansible Playbook from Jenkins for production'



En resumen, este Jenkinsfile configura un pipeline de Jenkins que ejecuta un playbook de Ansible llamado `'main.yml'` ubicado en `'./Desafio7/ansible/main.yml'`. Utiliza diferentes claves privadas de SSH almacenadas en las credenciales de Jenkins para autenticarse en los hosts especificados en el archivo de inventario `'./Desafio7/ansible/inventory.ini'`. El pipeline se ejecutará en un nodo con la etiqueta `'ansible-controller'`. Las etapas se ejecutan en función de la rama actual (`'dev'`, `'staging'` o `'main'`) y limitan la ejecución del playbook a los hosts específicos de cada entorno (`'development'`, `'staging'` o `'production'`).

Stage View por Branch

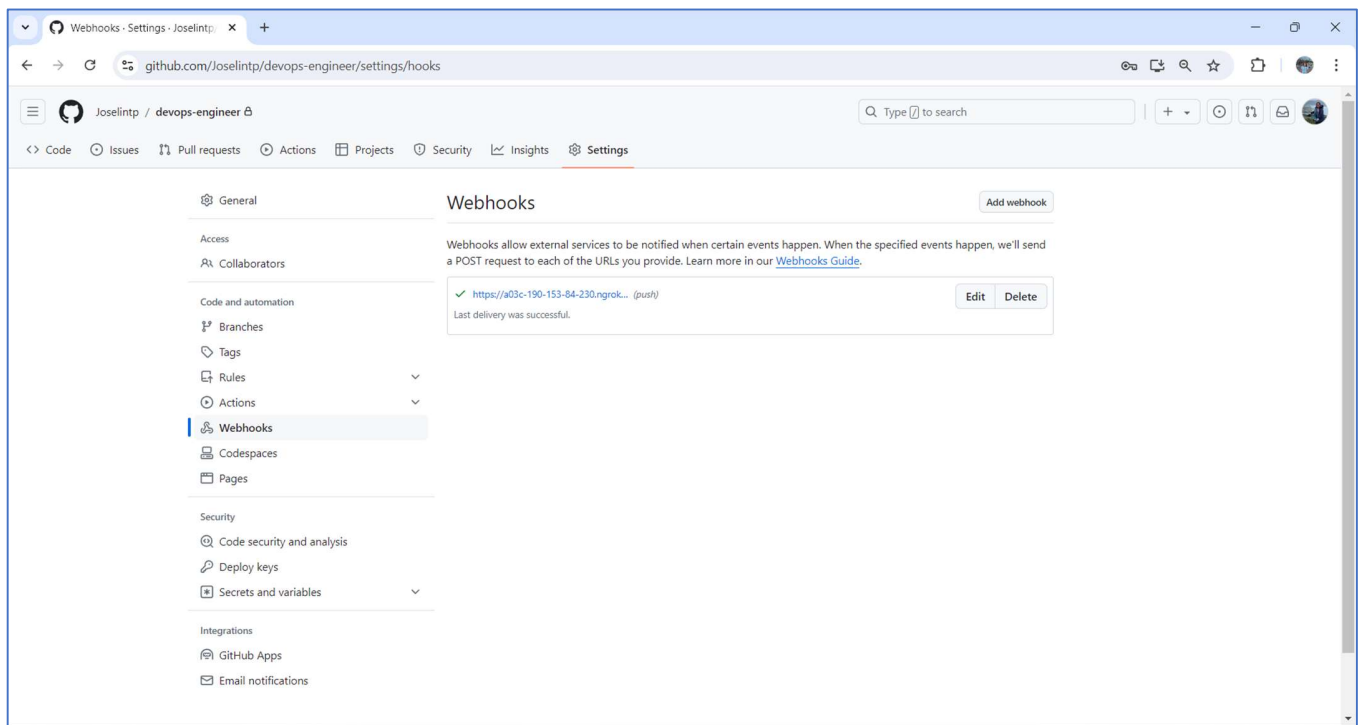


Referencias:

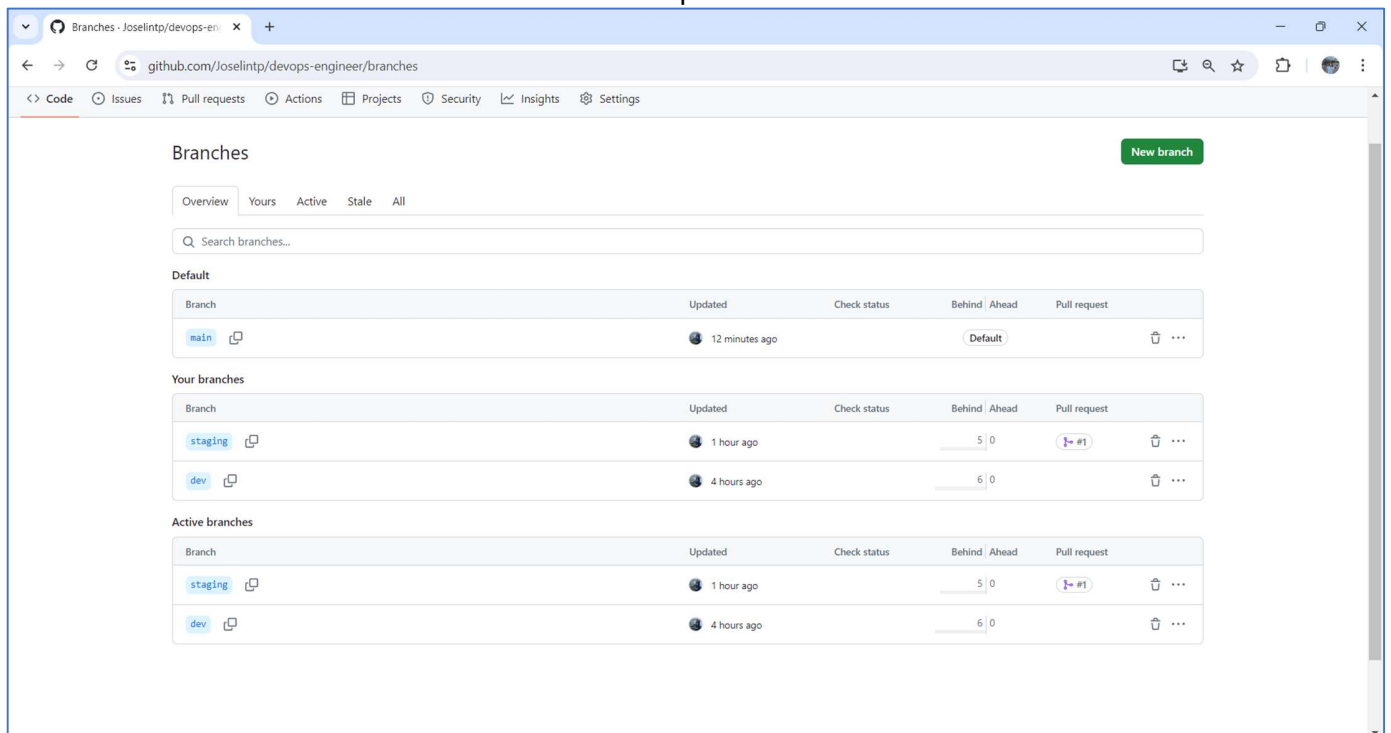
[Jenkins Multibranch pipeline](#)
[Using Git branch](#)
[Working with ansible inventories](#)

Prerrequisitos:

Debemos tener configurado el Webhooks en nuestro repositorio.



Debemos tener las 3 branches creadas en nuestro repositorio:



Debemos tener las ansible-private-key para cada entorno en Jenkins y para Github:

<http://192.168.68.60:8080/manage/credentials/>

		System	(global)	JT-github	Joselintp/***** (JT-github)
		System	(global)	ansible-private-key	ubuntu (ansible-private-key)
		System	(global)	ansible-private-key-dev	ubuntu (ansible-private-key-dev)
		System	(global)	ansible-private-key-staging	ubuntu (ansible-private-key-staging)