

**UNIVERSIDAD RAFAEL LANDÍVAR**

FACULTAD DE INGENIERÍA

VIRTUALIZACIÓN

SECCIÓN 1 VESPERTINA

MGTR. JOSSUE LEONEL SAMAYOA PORTILLO

## **ANSIBLE - HELLO WORLD**

**Julio Anthony Engels Ruiz Coto 1284719**

GUATEMALA DE LA ASUNCIÓN, ABRIL 28 DE 2024

Powered by  
**instruct**

Pantalla completa | Progresos

**ansible-navigator** es una interfaz de usuario textual (TUI) disponible para los suscriptores de Ansible Automation Platform y ha sido introducida con Ansible Automation Platform 2 como la interfaz principal para crear y probar una automatización sensible.

**ansible-navigator** también funciona como reemplazo de la entrega de **ansible-playbook**, entre otros **ansible-\*** utilidades, y es la forma estándar de ejecutar la automatización entrando en Ansible Automation Platform 2.

Los bits están fluyendo...  
Tiempo estimado que queda: 00:55 - Notificarme

Hands-On Lab

The screenshot shows the Ansible Navigator interface within the Instruct platform. On the left, there's a sidebar with 'instruct' branding and navigation links like 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The main area has tabs for 'Welcome' and 'test.yml'. The 'test.yml' tab contains the following YAML code:

```
1 ---  
2 - name: this is just a test  
3   hosts: localhost  
4   gather_facts: true  
5   tasks:  
6     - name: ping test  
7       ping:
```

Below the code editor is a 'TERMINAL' tab showing the command-line interface. It starts with 'rhe1@code-server ~]\$ ansible-navigator run ./test.yml -m stdout', followed by error messages about a missing playbook and a warning about an empty hosts list. Then it shows the playbook execution: 'PLAY [this is just a test] \*\*\*\*', 'TASK [Gathering Facts] \*\*\*\*', 'TASK [ping test] \*\*\*\*', and finally 'PLAY RECAP \*\*\*\*'. The terminal output ends with 'localhost : ok=2 changed=0 unreachable=0 failed=0 rescued=0 ignored=0'. At the bottom of the terminal tab, it says 'in P\_Col 10\_Space 2\_UFT-8\_1F\_Ansible\_Layout-US\_\*'.

To the right of the terminal is a 'Progresos' (Progress) section with a progress bar at 25% completion. Below the progress bar, it says 'gather\_facts: true' and 'tasks:'. Under 'tasks', it lists a single task: '- name: ping test ping:'. Below this, a note says '3. Ahora, hacételo con ansible-navigator:' followed by the command 'ansible-navigator run ./test.yml -m stdout'. A note at the bottom of this section says 'Usted debe estar mirando algo como esto (y parecerse mucho a la salida de "ansible-playbook")'. The entire interface is set against a dark background with orange and white text.

The screenshot shows two vertically stacked terminal windows from a Jupyter Notebook interface, both titled "code-server".

**Top Terminal:**

- Content:** The terminal displays a portion of an Ansible playbook named "testyml". The visible code includes:
 

```

1   ...
2   - name: this is just a test
3     hosts: localhost
4     gather_facts: true
5     tasks:
6       - name: ping test
7         ping:
      
```
- Bottom Content:** The bottom half of the terminal shows the output of the playbook run:
 

```

PLAY NAME: testyml
  OK    CHANGED   UNREACHABLE   FAILED   SKIPPED   IGNORED   IN PROGRESS   TASK COUNT   PROGRESS
0|this is just a test   2        0          0        0        0        0        0        2        100%
      
```

**Right Panel (Instructions):**

- Explore the play that was run by pressing the number corresponding to that line. You should be taken to the next screen that will contain a list of the tasks that ran as a part of that play. Inspect the tasks by pressing the number of the corresponding line.
- Wondering how to select a line with a number greater than 9? Simply prepend the number with a colon. Example: To select line 15, you would issue the command `:15`
- Select the gather\_facts task by pressing `0` and issue the subcommand `:doc`. Using `ansible-navigator` is very helpful for quickly referencing documentation for specific tasks while debugging playbooks.
- Finally, you can pass subcommands directly to `ansible-navigator` on the command line. Try getting docs on with `setup` module:
 

```
ansible-navigator doc setup
```

**Bottom Right Panel (Note):**

Remember, if you ever get stuck in `ansible-navigator`, `Esc` will always bring you to the previous screen. Pressing `Esc` when there are no more screens left will exit `ansible-navigator`.

**Bottom Bar:**

Check

**Bottom Terminal:**

- Content:** The terminal displays a portion of an Ansible playbook named "tmpzezimw4.yml". The visible code includes:
 

```

1   ...
2   - changed: 0
3   - failed: 0
4   - ignored: 0
5   - in_progress: 0
6   - ok: 2
7   play_name: this is just a test
8   progress: COMPLETE
9   skipped: 0
10  task_count: 2
11  unreachable: 0
12
      
```
- Bottom Content:** The bottom half of the terminal shows the output of the playbook run:
 

```

PLAY NAME: tmpzezimw4.yml
  OK    CHANGED   UNREACHABLE   FAILED   SKIPPED   IGNORED   IN PROGRESS   TASK COUNT   PROGRESS
0|this is just a test   2        0          0        0        0        0        0        2        100%
      
```

**Right Panel (Instructions):**

- Let's take a look at some of the common options.
- In the directory `/home/rhel` you should see a file called `ansible-navigator.yaml`. Open it up and check out the contents. Notice that there are currently some execution environment and logging settings already there. Developers love the ability to not only run `ansible-navigator` within the integrated terminal, but I want to be able to pass task output back to my code editor for inspection.
- `ansible-navigator` can configure a preferred editor for the `:open` sub-command. Let's do that now. Open `ansible-navigator.yaml` and set your preferred editor to the code-server instance in your environment. Do this by copying the following settings to the end of the file:
 

```

editor:
  command: code-server {filename}
  console: false
      
```
- Execute your test playbook:
 

```
ansible-navigator run ./test.yml
```

**Bottom Right Panel (Note):**

4. Inspect the play by pressing `0`, inspect the first task by pressing `0` again and issue the subcommand `:open`.

You should now have a new tab within your code editor with the output of that task as the file contents. `:open` works for any page within the navigator TU and can be useful in creating playbooks.

**Bottom Bar:**

Well done!

Loading your next challenge...

The screenshot shows a terminal window titled "instruct" running on a "code-server" instance. The terminal displays the command "ansible-navigator run ./test.yml". The output shows the execution of a test playbook named "test.yml" which contains the following configuration:

```

playbook-artifact:
  save-as: /home/rhel/playbook-artifacts/{playbook_name}-artifact-{version}-{date}
logging:
  level: debug
editor:
  command: code-server {filename}
  console: false

```

The terminal also shows the command "ansible-navigator run ./test.yml" being executed again.

**2. Re-run your test playbook:**

```
ansible-navigator run ./test.yml
```

Notice that `ansible-navigator` knows that it should be using an execution environment but none are currently present. You should see a pull process happening now where an execution environment is being pulled from container registry. `ansible-navigator` can be configured in the same yaml file to pull from your own Private Automation Hub.

**3. Your `test.yml` file should have executed successfully. You can now use `ansible-navigator` to inspect this execution environment by issuing the `:collections` subcommand.**

**4. While inspecting collections, there is a module in the `ansible.utils` collection called `fact_diff`. Locate the author of this module and remember the github handle associated with this person.**

**Well done!**  
Loading your next challenge...

instruct

Progress 8m

Knowledge check

In the previous challenge, you were tasked with finding the github handle of the author of the `fact_diff` module contained in the `ansible.utils` collection. What was that handle?

- @hardcidr
- @ianaboss
- @OSIsupermodel
- @cidrblock

Well done!

You've completed the last challenge.

Powered by  
instruct

Full screen

Progress Completed

Thank you!

Your feedback will be shared with the content creator and used to improve this track for others.

Add more feedback

**La firma de contenidos** es una nueva función centrada en la seguridad disponible en **Ansible Automation Platform 2.2** como una función de **vista previa tecnológica** compatible.

Establece un nuevo marco para una cadena de custodia de contenidos de automatización Ansible, con el objetivo de la firma y distribución de contenidos de extremo a extremo. Comenzar primero con colecciones de firma digital, y luego entornos de ejecución en el futuro, esta característica ayuda a asegurar que la automatización que se está ejecutando en su empresa esté certificada y compatible.

Hands-On Lab

La paciencia es una virtud, y eres un usuario virtuoso.  
Tiempo estimado a la izquierda: 01:56 - Notificarme

The screenshot shows a terminal window titled "automationhub-terminal" and a progress bar window titled "Progresos".

**Terminal Output:**

```
Last failed login: Tue Jun 7 14:43:19 UTC 2022 from 184.147.217.109 on ssh:notty
There was 1 failed login attempt since the last successful login.
[rhel@privatehub-01 ~]$ ls
ansible.cfg                                     community-lab.collection-1.0.0.tar.gz
ansible-test_collection-1.0.0.tar.gz           galaxy_signing_service.asc
ansible-test_core-2.13.0.tar.gz                 config
ansible-test_core-2.13.0.tar.gz                 configured_module_search_path='/home/rhel/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'
ansible_python module location = /usr/lib/python3.9/site-packages/ansible
ansible collection location = /home/rhel/.ansible/collections:/usr/share/ansible/collections
executable location = /usr/bin/ansible
python version = 3.9.7 (default, Sep 13 2021, 08:18:39) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)]
initial version = 3.0.3
libyaml = True
[rhel@privatehub-01 ~]$
```

**Progress Bar:**

La paciencia es una virtud, y eres un usuario virtuoso.  
Tiempo estimado a la izquierda: 01:56 - Notificarme

pestana terminal, encontrará algunos artefactos que se explican a continuación:

1. **ansible.cfg** - Este archivo contiene la configuración de la galaxia para interactuar con el centro de automatización privada instalado, lo que también implica que cualquier ansible-galaxy comando que va de la carpeta actual (`/home/rhel/`) interactuará con este laboratorio.
2. **ansible-test\_collection-1.0.0.tar.gz** - Una colección firmaria que será empaquetada a la instancia de la automatización privada, el espacio de nombres `ansible` ya existe en el centro de automatización privada.
3. **community-lab\_collection-1.0.0.tar.gz** - Una colección firmaria que será empaquetada a la instancia de la automatización privada, el espacio de nombres `community` ya existe en el centro de automatización privada.
4. **galaxy\_signing\_service.asc** - Esta es la clave pública de la pieza clave gpg que se utiliza para configurar la firma en la instancia de centro de automatización privada, esta clave se utilizará para instalar y verificar colecciones de centro de automatización privada.

Las capacidades de verificación de la firma de contenidos sólo están disponibles en `ansible-core` versión=2.13, se puede comprobar la versión de `ansible` utilizando el siguiente comando

```
ansible --version
```

Por favor haga clic en el botón **Siguiente** para continuar aprendiendo.

Bien hecho.  
Cargando tu próximo reto...

Last login: Sat Apr 27 22:41:06 UTC 2024 on pts/0  
rhe1@privatehub-01 ~]\$ ansible-galaxy collection publish ansible-test\_collection-1.0.0.tar.gz -c  
Publishing collection artifact '/home/rhel/ansible-test\_collection-1.0.0.tar.gz' to published\_repo https://privatehub-01/api/galaxy/content/published/  
Collection has been published to the Galaxy server published\_repo https://privatehub-01/api/galaxy/content/published/  
Waiting until Galaxy import task https://privatehub-01/api/galaxy/content/published/v3/import/collections/od2ppf91-0ifa-4049-9114-1568f25b342d/ has completed  
Collection has been imported to the Galaxy server imported\_repo https://privatehub-01/api/galaxy/content/published/  
Collection has been published and imported to the Galaxy server published\_repo https://privatehub-01/api/galaxy/content/published/  
rhe1@privatehub-01 ~]\$ ansible-galaxy collection publish community-lab\_collection-1.0.0.tar.gz -c  
Publishing collection artifact '/home/rhel/community-lab\_collection-1.0.0.tar.gz' to published\_repo https://privatehub-01/api/galaxy/content/published/  
Collection has been published to the Galaxy server published\_repo https://privatehub-01/api/galaxy/content/published/  
Waiting until Galaxy import task https://privatehub-01/api/galaxy/content/published/v3/import/collections/5add33c-9cab-48c-ba6-acb12f91b51d/ has completed  
Collection has been imported to the Galaxy server imported\_repo https://privatehub-01/api/galaxy/content/published/  
rhe1@privatehub-01 ~]\$

Publicación de Ansible Content Collections a instancia de la instancia de centro de automatización privada.

En este desafío, publicaremos las dos colecciones, `ansible-test_collection` y `community-lab_collection` a instancia de centro de automatización privada.

1. Publicar el `ansible-test_collection`. Ejecute el siguiente comando desde la ventana del terminal.

```
ansible-galaxy collection publish ansible-test_collection-1.0.0.tar.gz
```

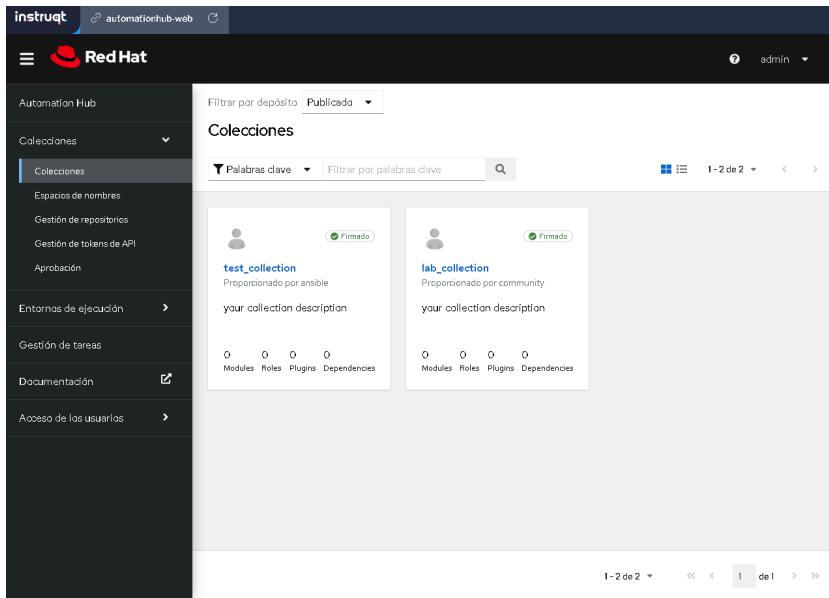
2. Publicar el `community-lab_collection` a instancia de centro de automatización privada. Ejecute el siguiente comando desde la ventana del terminal.

```
ansible-galaxy collection publish community-lab_collection-1.0.0.tar.gz
```

Nota: -c opción se utiliza para ignorar errores de certificado SSL

Haga clic en **Check** para el próximo reto.

Bien hecho.  
Cargando tu próximo reto...



Filtrar por depósito: Publicado

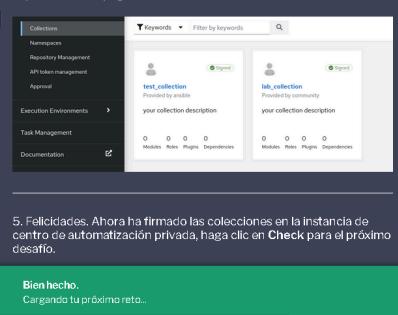
### Colecciones

	test_collection	lab_collection
Preparado por ansible	Preparado por ansible	Preparado por community
your collection description	your collection description	your collection description
0 Modules, 0 Roles, 0 Plugins, 0 Dependencies	0 Modules, 0 Roles, 0 Plugins, 0 Dependencies	0 Modules, 0 Roles, 0 Plugins, 0 Dependencies

Status

Needs Review	Sign and approve
1	1
Needs Review	Sign and approve
1	1

4. Ahora ha firmado y publicado ambas colecciones, ahora se verán en la página **Colecciones** como están firmadas.



5. Felicidades. Ahora ha firmado las colecciones en la instancia de centro de automatización privada, haga clic en **Check** para el próximo desafío.

Bien hecho.  
Cargando tu próximo reto...

instruct@privatehub-01 ~\$ cat ./ansible/collections/ansible\_collections/community.lab\_collection/1.0.0.info/GALAXY.yml

```

community.lab_collection: 1.0.0 was installed successfully
friendly_name: community-lab_collection
friendly_version: 1.0.0
name: Lab Collection
namespace: community
server: https://privaterhub-01/api/galaxy/content/published/
signatures:
  - public_key_fingerprint: 0E450BA4796C33F8A6F8A69BC97AC9643F917EBB
    public_created: '2024-04-27T22:56:22.129629Z'
    signature: '-----BEGIN PGP SIGNATURE-----'

    iQIzBAAABCAdFIEEDKULqnlsM/m+6abyXrJZD+RfrsFAMYtgpYACqkQyXrJZD+R
    frvz+BAg6wsUpL5fI+nyJshW8DM22HFKm+k1w250EKTm+l w8A4N8jBrYOWc
    H4CI8Rh9PEYRLn3TLqvXj90Ms1PBBhn19qrEzwEacc6YKCWhvCLnGV/4Dk0Zeeq
    /B0wC451KSOQa6k3UKODorJ0Slid2d18G0wo+sF9zVQ6A70cBcBV6Q43FXdyK2QlQ
    TnNnbXNjOU0x0c7Kb2zcKdNttZUK7DXx1x1wALfCnOb6M7bt7c14bz3RVdiX12fd
    00hMrSzWlzcuyu8XzJekAQCvOf2yHeKvQjEWmD2rvZp12pRs1lf8its84Z6yBxDx
    lC8lEjwxumbfXvYZSrx5nFlbmChBeN3kZ6e8zWzKxf0mLoTwNif4MR/G1/aAHM
    1K1P1BVj7WNbq2BZRIixyHnOzhuzazKu/1bh9Xw1CcijZ6w2qZtchYMO+UBeqf
    ZMRkurbh4p0fINNzNSBZrd7UP/+UfyfMSDHIIpfbYPWesIwoZfSc/PWzM+AqAf2
    euyVrOBj19G+vN1oquat2Sdo3Gjt6W89wdfJsuDpm12vVvi+UyUK8o0ZMeqipew8
    ogtiZgausUVN9T5wwmDtqnf47XkZxNlSf1/qhZE+qayobr6pRIIOXBH8K0DbmuRZ
    fVdw69jZ08f144KTxWdzh/5we/JI+FRYeEWOMDX47iC/N+prrfBQ=
    =Q5Iq
    ----- END PGP SIGNATURE-----

  signing_service: ansible-default
version: 1.0.0
version_uri: /api/galaxy/content/published/v3/plugin/ansible/content/published/collections/index/community/lab_collection/versions/1.0.0/
friendly_name: community-lab_collection
friendly_version: 1.0.0

```

instruct@privatehub-01 ~\$ cat ./ansible-galaxy collection install community.lab\_collection --keyring

Note: Las últimas tres líneas de la salida de verbose le mostrarán que la verificación de la firma fue exitosa.

```

.
.
.

(exit code 0)

GPG signature verification succeeded for community.lab_collection
community.lab_collection:1.0.0 was installed successfully

```

**4. Nuevos artefactos con instalación de colección.**

Con la colección firmada, ansible-galaxy CLI también crea un nuevo directorio con un archivo llamado `GALAXY.yml` que contiene las firmas de recogida. Ejecute el siguiente comando en el `automatizacionhub-terminal`:

```

cat ./ansible/collections/ansible_collections/community.lab_collection/GALAXY.yml

```

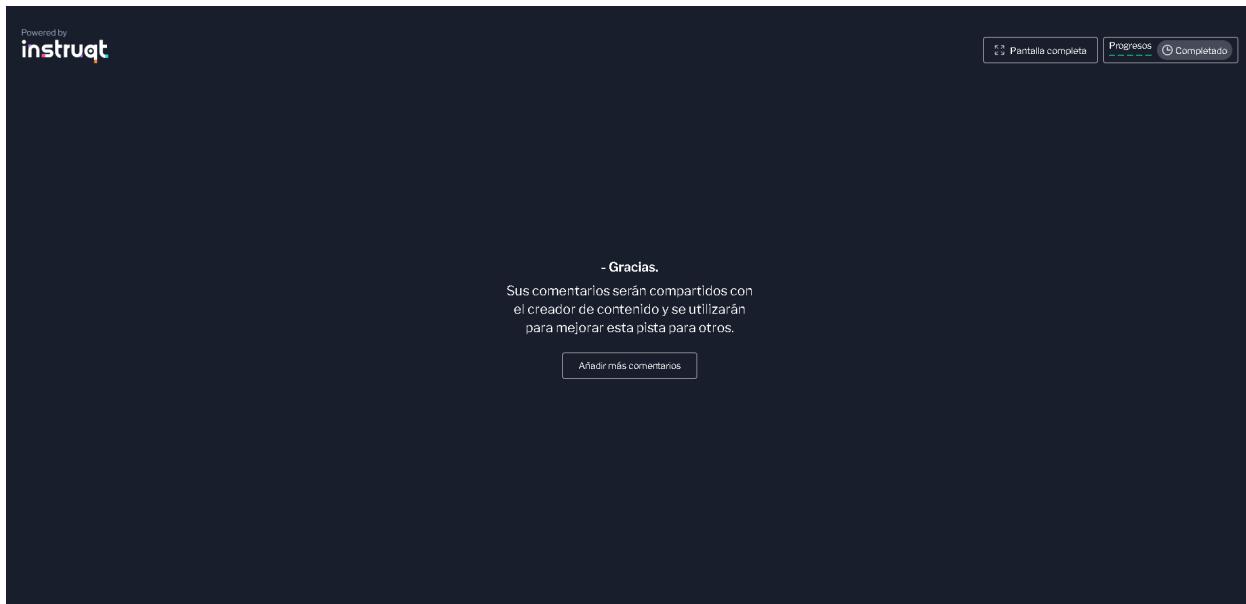
Esto le dará el contenido del archivo que contiene la firma de recogida.

**Bien hecho.**  
Cargando tu próximo reto..

The screenshot shows a terminal window titled "automatizaciónhub-terminal" with the following command history:

```
Last login: Sat Apr 27 23:14:30 UTC 2024  
rhel@privatehub-01 ~]$ touch ~/.ansible/collections/ansible_collections/community/lab_collection/docs/README.md  
rhel@privatehub-01 ~]$ ansible-galaxy collection verify community.lab_collection -keyring keyring.kbx  
Verifying collection 'community.lab_collection' 1.0.0...  
Installed collection found at '/home/rhel/.ansible/collections/ansible_collections/community/lab_collection'  
MANIFEST.json hash: a77953a136fe7d70f0d792f698a05f91dab069c8cdad6da04459e0532c6c1ba  
Collection community.lab_collection contains modified content in the following file(s):  
docs/README.md  
rhel@privatehub-01 ~]$
```

To the right of the terminal, there is a progress bar labeled "Progresos" with a green bar indicating completion. Below the progress bar, a message says "Ejecute el siguiente comando:" followed by a code block containing "touch ~/.ansible/collections/ansible\_collections/community/lab\_collection". A note below says "2. Verificar la colección manipulada". Another "Ejecute el siguiente comando:" section follows, with a code block containing "ansible-galaxy collection verify community.lab\_collection --keyring keyring.kbx -c". A note below says "El comando hará una comparación entre la versión local de la colección y la versión remota en la instancia de centro de automatización privada, la salida mostrará qué archivos han sido cambiados/añadidos/borrados. Salida con verbozidad -vvvv ofrecerá más detalles". A log output block shows the command execution results.



Powered by  
**instruct**

Pantalla completa Progresos

## Bienvenido al controlador de automatización responsable

Bienvenidos a nuestro laboratorio interactivo en el controlador de automatización Ansible, la interfaz de interfaz de interfaz de interfaz de interfaz de Red Hat Ansible Automation Platform.

En este laboratorio, demostraremos lo fácil que es utilizar el controlador de automatización (anex Ansible Tower) ejecutando algunos de los libros de jugadas del laboratorio "Escribir su primer libro de jugadas". No se requiere experiencia de desarrollo de libros de jugadas, aunque ayudará si conoces lo básico.

Exploraremos la interfaz web-UI, crearemos un **Inventario** para administrar nuestros servidores, importar nuestros libros de jugadas a través de **los Proyectos**, añadiremos **plantillas de trabajo** para ejecutar los libros de jugadas y, por último,

Los hámsters están en una paja de té  
Tiempo estimado a la izquierda: 04:47 - Notificarme

Hands-On Lab

Controlador de automatización

Panel de control

Servidores: 1 | Hosts fallidas: 0 | Inventarios: 1 | Errores de sincronización de inventario: 0 | Proyectos: 1 | Errores de sincronización del proyecto: 0

Ejecuciones de trabajo

Tareas recientes: Mes pasado | Todas las tipos de tarea | Todas las tareas

Resumen de ejecuciones de trabajo

Panel de control

Los inventarios en el controlador de automatización tienen varias ventajas sobre los basados en archivos, obtienes todas las funcionalidades de estos últimos, con características adicionales y mejor utilización.

Haga clic en el botón **Inventarios** en el panel de control o en el enlace en el menú de la barra lateral y explore el **inventario de demostración** que viene pre cargado. Verá todos los campos disponibles para usar al crear uno. No te olvides de mirar las pestañas.

Tarea 3 - Explora la sección de proyectos

Los **proyectos** son grupos lógicos de los libros de jugadas Ansible en el controlador de automatización. Estos libros de jugadas generalmente residen en un sistema de control de versión de código fuente como Git (y plataformas como Github o Gitlab). Con **Proyectos** podemos hacer referencia a un repositorio o directorio con uno o varios libros de jugadas, que posteriormente utilizaremos.

Haga clic en el botón **Proyectos** en el panel de Dashboard o el enlace en el menú de la barra lateral y explore el **proyecto Demo** que viene pre cargado. Verá todos los campos disponibles para usar al crear uno. No te olvides de mirar las pestañas.

El próximo desafío

Pulse la **next** botón abajo para ir al siguiente desafío una vez que haya completado las tareas.

- Encuentra un problema?

Bien hecho.  
Cargando tu próximo reto...

<https://control-643c1.firebaseio.com/resources/instruct.com/tutorials/host-summary.firebaseio.html>

**Servidores**

Nombre	Descripción	Actividad	Acciones
node1		On	
node2		On	

**Task 3 - Create the [web] group in Lab-Inventory**

- Once you've added node1 and node2 to the Lab-Inventory
- On the tab bar of Lab-Inventory, click on Groups
- Click on the Add button to create a new group.
- Create a new group named web (notice you don't need to input as [web] as in an inventory file).
- Click Save when you are finished.

**Task 4 - Add node1 and node2 to the [web] group in Lab-Inventory**

- Click on the newly created web group and click on the Hosts tab on the top menu.
- Click Add existing host. A pop up will appear.
- Select both node1 and node2, then click Save.
- Verify node1 and node2 are now added to the web group.

**Well done!**  
Loading your next challenge...

**Proyectos**

Último estado de la tarea	Nombre	Apache playbooks	Organización	Default
Ejecutándose	Apache playbooks	Sincronizar para revisión	https://github.com/legallego/instrukt-wyfp.git	_8_apache_playbooks

**Task 1 - Creation of the Project**

- Click the Projects link in the Resources section of the sidebar
- Click the Add button
- Name the Project as Apache playbooks
- For the Source Control Type dropdown, select Git, this will enable new fields below.
- For the Source Control URL field copy & paste the following URL: https://github.com/legallego/instrukt-wyfp.git
- Leave all the other fields the same and click Save

**Task 2 - Verify the new Project synced successfully**

- Click the Projects link in the Resources section of the sidebar
- Verify that the Status for our Apache playbooks project is green and reads Successful
- Click the Successful and review the output of the sync

**Well done!**  
Loading your next challenge...

**Clave privada SSH**

Arrastra un archivo aquí o navega para cargarlo  
\$encrypted\$

**Certificado SSH firmado**

Arrastra un archivo aquí o navega para cargarlo

Frase de contraseña para la clave privada  Preguntar al ejecutar Método de escalación de privilegios  Usuario para la elevación de privilegios

Contraseña para la elevación de privilegios  Preguntar al ejecutar

**Guardar** **Cancelar**

**Task 2 - Exploring the Machine Type credential**

- Go back to the **Details** tab of the **lab-credentials**
- Click the **Edit** button
- Look at the different fields available.
- Notice how the **SSH Private Key** field only allows us to **Reset**. As we mentioned before, the key is encrypted and not retrievable.
- To exit, click the **Cancel** button to avoid accidental modifications.

**Next Challenge**

Press the **Check** button below to go to the next challenge once you've completed the task.

**Encountered an issue?**

**Well done!**  
Loading your next challenge...

**Plantillas > Install Apache**

**Detalles**

Nombre		Install Apache	Tipo de trabajo	run	Organización	Default
Inventory	Lab-Inventory	Proyecto	Apache playbooks	Entorno de ejecución	Default execution environment	
Playbook	apache.yml	Forks	0	Nivel de detalle	0 (Normal)	
Tiempo de espera	0	Mostrar cambios	Off	Fraccionamiento de trabajos	1	
Creado	27/4/2024, 18:03:54 por admin	Último modificado	27/4/2024, 18:03:54 por admin			
Credenciales	SSH lab-credentials					
Variables	<b>YAML</b> <b>JSON</b>					

**Task 2 - Launch the new Template**

- For the **Job Type** field, leave the default **Run**
- For the **Inventory** field, touch the magnifying glass and select **Lab-Inventory**
- For the **Project** field, touch the magnifying glass and select **Apache playbooks**
- For the **Playbook** dropdown field, you will notice that it's auto populated with all the playbooks available in the **Project** repository we selected. Choose the **apache.yml**.
- For the **Credentials** field, touch the magnifying glass and select **lab-credentials**
- Leave all the other fields the same and click **Save**

**Task 2 - Launch the new Template**

- Click the **Templates** link in the **Resources** section of the sidebar
- You can launch it from this list, by clicking the **Rocket** icon, or
- You can click on the **Install Apache** template and then click the **Launch** button at the bottom.
- Verify the execution was successful

**Next Challenge**

**Encountered an issue?**

**Well done!**  
Loading your next challenge...

**instruct** Automation controller

Vistas

- Panel de control
- Trabajos
- Programaciones
- Flujo de actividad
- Aprobaciones del flujo de trabajo

Recursos

- Plantillas
- Credenciales
- Proyectos**
- Inventarios
- Servidores

Acceso

- Organizaciones
- Usuarios
- Equipos

Administración

- Típos de credencial

Proyectos > Additional playbooks

### Detalles

Último estado de la tarea	Nombre	Additional playbooks	Organización	Default
Correctamente	896964b	https://github.com/legallego/instruct-wyp-additional.git	/var/lib/swy/projects/_10_additional_playbooks	
0 Segundos	Ruta base del proyecto	/var/lib/swy/projects		
27/4/2024, 18:06:03 por admin	Último modificado	27/4/2024, 18:06:03 por admin		

**Detalles** **Editar** **Sincronizar** **ELIMINAR**

Progress 42m

- CLICK THE Projects link in the Resources section of the sidebar
- Click the Add button
- Name the Project as Additional\_playbooks
- For the Source Control Type dropdown, select Git, this will enable new fields below.
- For the Source Control URL field, copy & paste the following URL <https://github.com/legallego/instruct-wyp-additional.git>
- Leave all the other fields the same and click Save

Task 2 - Verify the new Project synced successfully

- After saving, you should be taken to the Details of the project we just created
- The Last Job Status should display Running in blue, give it a few seconds
- If it changed to Successful and is showing in green, the project synced correctly

As an alternative, to see the status of multiple projects, you can:

- Click the Projects link in the Resources section of the sidebar
- Check the Status column for the project is green and reads Successful
- Click the Successful and review the output of the sync

**Well done!**  
Loading your next challenge...

Trabajos > 12 – your first workflow

### Salida

```

graph LR
    INICIAR((INICIAR)) --> APACHE[Install Apache]
    INICIAR --> SERVICES1[Extended services]
    INICIAR --> SERVICES2[Extended services]
    
```

Progress 29m

- Click the Templates link in the Resources section of the sidebar
- You can launch the Workflow from this list by clicking the Rocket icon, or
- You can click on the `your first workflow` name and click the Launch button.
- You will be presented with a smaller Workflow Visualizer that shows the progress of the execution as Running
- Verify the execution was successful by checking all the Job Templates have a green border and a green tick.

Task 3 - Success or failure? Challenge time!

- If it failed, why did it fail? Click the red Failed and read the output message
- It reports the playbook couldn't find the destination hosts or group in the inventory
- Try to fix it. You need a `database` group in the inventory and to add a `nodes` to it.
- Run the Workflow.
- If you are having trouble solving it, you can Skip and it will be solved for you.

**Well done!**  
Loading your next challenge...

The screenshot shows the Red Hat Ansible Automation Platform interface. On the left, there's a sidebar with sections like 'Vistas', 'Trabajos' (selected), 'Programaciones', 'Flujo de actividad', 'Aprobaciones del flujo de trabajo', 'Recursos', 'Acceso', and 'Administración'. The main area shows a job titled '16 - Install Apache with Survey' under the 'Trabajos' tab. The 'Salida' tab is selected, displaying the command output in a terminal-like interface:

```
Procesamiento de eventos completo. Descargar salida
...
1 ok [node1]
2 ok [node2]
3 ok [node1]
4 ok [node2]
5 ok [node1]
6 ok [node2]
7 changed [node2]
8 changed [node1]
9 PLAY RECAP
10 node1 : ok=4 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
11 node2 : ok=4 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

To the right, a 'Progress' bar indicates the task is at 26%. A detailed list of steps for the challenge is provided:

- Once the Job Template is saved, you should be taken to the Details
- Look for the Survey tab and click on it.
- Click the Add button
- In the Question field, enter what's your name?
- In the Answer variable name, enter student\_name
- In the Default answer type, Skipped
- Press the Save button
- Touch the toggle to Enable the survey

Below this, a 'Task 3 - Launch the new Template' section contains further instructions:

- Click the Templates link in the Resources section of the sidebar
- You can launch it from this list, by clicking the Rocket icon, or
- You can click on the Install Apache with Survey template and then click the Launch button at the bottom.
- Enter your name at the pop-up.
- Verify the execution was successful by visiting the Node1 Web tab in the top left of the screen, you should see the Apache message with your name.

At the bottom, there are status indicators: 'Next Challenge' (checked), 'Encountered an issue?' (unchecked), and a green 'Well done!' message: 'Loading your next challenge...'.

The screenshot shows the Red Hat Automation Controller Node1 Web interface. The top navigation bar includes the Instruct logo, Automation controller, and Node1 Web tabs. The left sidebar contains sections for Vistas (Panel de control, Trabajos, Programaciones, Flujo de actividad, Aprobaciones del flujo de trabajo), Recursos (Plantillas, Credenciales, Proyectos, Inventarios, Servidores), Acceso (Organizaciones, Usuarios, Equipos), and Administración (Tipos de credencial). The main content area displays the "Panel de control" dashboard with metrics: 3 Servidores, 0 Hosts fallidos, 2 Inventarios, 0 Errores de sincronización de inventario, 3 Proyectos, and 0 Errores de sincronización del proyecto. Below the dashboard is a chart titled "Estado de la tarea" showing "Ejecuciones de trabajo" over time. The right side features a "Progress" bar at 26%, a "Workflow Approval" section listing recent tasks, and a "Success!" summary message:

Congratulations on finishing the **Introduction to automation controller** laboratory.

You are welcome to stay and play around. Try other combinations for the Workflow, or explore the rest of the sections to get to know automation controller a bit more for example!

When you are done, we invite you to find more information about Ansible at [Ansible.com](#)

Did you like Ansible automation controller? [Try the full Ansible Automation Platform today!](#)

Encountered an issue?

Well done! You've completed the last challenge.

Powered by **Instruct**

Full screen | Progress (Completed)

Powered by  
**instruct**

Full screen Progress

## Challenge summary

In this challenge, we'll look at the tooling ACME Corp use to deploy the *Let's Quiz!* application.

The diagram illustrates the ACME Corp CI/CD environment. It shows two main teams: Development and Operations, connected by a central Jenkins icon. Below them are various tools and components: VS Code, GitLab, Jenkins, Automation controller, and Web server(s).

Hands-On Lab

Loading... now is a good time to grab a cup of coffee  
Estimated time left: 01:59 ▲ Notify me

VS Code GitLab Jenkins Controller

File Edit Selection View Go Run Terminal Help Welcome - acme.corp - code-server

Explorer

- ACME CORP
  - app
  - > lets\_quiz
  - version.py
  - .flake8
  - .gitignore
  - LICENSE
  - pipfile
  - README.md
  - setup.py
- playbooks
  - files
  - group\_vars
  - templates
  - app\_release.yml
  - configure\_jenkins\_job.yml
  - configure\_webserver.yml
  - deploy\_acme\_app.yml
  - restart\_workflow.yml
  - .gitignore

Progress 53m

Resources

- Templates  DevOps Workflow
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Resources

- Templates  DevOps Workflow
- Credentials
- Projects
- Inventories
- Hosts

Access

- Organizations
- Users
- Teams

Well done!  
Loading your next challenge...

Select the DevOps Workflow job template by clicking on the checkbox next to it.  
Click on the visualizer button located at the right-hand side of the DevOps Workflow job template.

**Jenkins**

Panel de Control > ACMECorp >

### Pipeline ACMECorp

**Stage View**

	SCM Get Code	Installing packages	Static Code Checking	Build and Tag
Average stage times:	1s	161ms	25ms	35ms
1s	aborted	161ms	25ms	35ms
1454	No Changes	aborted	aborted	aborted

**Enlaces permanentes**

- Última ejecución (#1) hace 3 Min 15 Seg\*
- Última ejecución fallida (#1) hace 3 Min 15 Seg\*
- Última ejecución exitosa (#1) hace 3 Min 15 Seg\*

**ATTEMPTS**

instruct

File Edit Selection View Go Run Terminal Help

SOURCE CONTROL

Message (Ctrl+Enter to commit on "main")

Changes

home.html app/lets\_quiz/templates/quiz M

```
app > lets_quiz > templates > quiz > home.html > ...
1  % extends 'base.html' %
2  % block title %} Let's Quiz | Home {%- endblock title %}
3  % block content %
4
5    <section id="section-jumbotron" class="jumbotron jumbotron-fluid d-flex justify-content-center align-items-center">
6      <div class="container text-center">
7        <h1 class="display-1 text-info">Let's Quiz</h1>
8
9        <p class="display-4 d-none d-sm-block">The latest and greatest version of the app deployed successfully</p>
10       <p class="display-4 d-none d-sm-block">Test your skills and become a master.</p>
11
12       (% if not request.user.is_active %)
13       <p class="text">Sign up if you haven't already and get access to millions of quizzes on the topic of your choice!</p>
14       <p><strong>Start Your Journey Here!</strong></p>
15       (% endif %)
16     </div>
17   </section>
18
19  % endblock content %]
```

REST API Jenkins 2.387.1

**Progress**

Task 2 - Controller and Jenkins integration

The Configure Jenkins Job controller job template adds a step to the current pipeline that calls controller's API to start a controller workflow.

Run the configure Jenkins Job job template:

- Click on the Controller tab at the top of the browser window.
- If not already logged in, log in using the provided credentials.
- Under Resources, click on Templates. This action displays the current Job Templates configured in controller.
- Run Configure Jenkins Job by clicking on the icon on the right-hand side.

**Progress**

Task 4 - Check the pipeline status

Check that the Gitea webhook started the pipeline.

- Click on the Jenkins tab at the top of the browser window.
- In the Pipeline ACMECorp window, you'll see that the Gitea webhook started the pipeline.
- Click on the number icon to open the Jenkins build. In the screenshot below, the build number is 1.

**Stage View**

	SCM Get Code	Installing packages	Static Code Checking	Build and Tag
Average stage times:	300ms	8s	0	0

Skip Check

**instruct** Controller Git Jenkins Let's Quiz

Vistas Panel de control Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos Administración

Trabajos > 4 - DevOps Workflow Salida

Salida

Detalles Salida Salida

DevOps Workflow Correctamente Nodos totales 4

INICIAR Deploy to Prod? Create App Release Config Webservers Deploy ACME App

Dashboard Schedules Active Stream Monitor Approvals Resources

Progress 29m

This action opens up the DevOps Workflow Output interface. The green indicates that all the tasks completed successfully.

We can also look more in-depth at the execution outcomes and details in the DevOps Workflow Output interface.

- Click on the Create App Release node.

Create App Release 00:00:08 Config Webservers 00:00:06 Deploy ACME App 00:00:10

play to Prod? 00:00:56 play to Prod? 00:00:08

This action displays the Create App Release controller job details.

Let's look at the Create App Release controller job in more detail.

Skip Check

instruct Controller VS Code Git Jenkins Let's Quiz

Vistas Panel de control Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos Administración Tipos de credencial

Panel de control

4 Servidores	0 Hosts fallidos	1 Inventarios	0 Errores de sincronización de inventario	0 Proyectos	0 Errores de sincronización del proyecto
--------------	------------------	---------------	---	-------------	--

Estado de la tarea Tareas redientes Plantillas redientes

Mes pasado Todas las tipos de tarea Todas las tareas

Ejecuciones de trabajo

Password: learn\_ansible

Introduction

In this lab, we've used automation controller's enterprise features to enhance how the ACME Corp development and operations teams integrate and automate together. Now it's your turn try!

Where to go next

Whether you are beginning your automation journey or a seasoned veteran, there are a variety of resources to enhance your automation knowledge:

- Self-paced exercises - Explore our all self-paced labs
- Trial subscription - Are you ready to install in your environment? Get your trial subscription for unlimited access to all the components of Ansible Automation Platform.
- Subscribe to the Red Hat Ansible Automation Platform YouTube channel.

Encountered an issue?

Well done! You've completed the last challenge.

**Thank you!**

Your feedback will be shared with the content creator and used to improve this track for others.

[Add more feedback](#)

Please wait while we set up the challenge.

**Get Started with VS Code in the Web**

Discover the best customizations to make VS Code in the Web yours.

**Choose the look you want**

The right color palette helps your focus on your code, is easy on your eyes, and is simply more fun to use.

Browse Color Themes: [Tip: Use keyboard shortcut \(Ctrl+K Ctrl+T\)](#)

One shortcut to access everything

Just the right amount of UI

Rich support for all your languages

**PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPITER**

```
[rhel@ansible ~]$ ansible-builder --help
usage: ansible-builder [-h] [-version] [create,build,introspect] ...

Tooling to help build container images for running Ansible content. Get started by looking at the help text for one of the subcommands.

positional arguments:
  (create,build,introspect)
    create           The command to invoke.
    build           Creates a build context, which can be used by podman to build an image.
    introspect      Builds a container image.
    introspect      Introspects collections in folder.

optional arguments:
  -h, --help        show this help message and exit
  -version         Print ansible-builder version and exit.
[rhel@ansible ~]$
```

**ANSIBLE-BUILDER**

**Execution Environment**

**execution-environment.yml**

```
version: 3
images:
  base_image:
    name: registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest
dependencies:
  galaxy:
    collections:
      - ansible.netcommon
options:
  package_manager_path: /usr/bin/microdnf
```

**PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPITER**

```
<none>          f067550ddab  6 minutes ago
<none>          1b2f717c6ica  6 minutes ago
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8 latest 4740c8b8adcb  10 months ago
registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8   latest 1f382a747255  10 months ago
325 MB
```

```
[rhel@ansible ~]$ ansible-builder build -t local/netcommon-ee
Running command:
podman build -t context/contextfile -c local/netcommon-ee context
Context file buildfile can be found at: /home/rhel/minimal-downstream-with-galaxy/context
[rhel@ansible ~]$ podman tag localhost/ansible-execution-env localhost/netcommon-ee
[rhel@ansible ~]$ podman run -it localhost/ansible-execution-env bash
bash-4.4$
```

```
# /usr/share/ansible/collections/ansible_collections
Collection          Version
-----
ansible.netcommon  6.1.1
ansible.dnsmasq  4.1.0
bash-4.4$
```

**ANSIBLE-BUILDER**

**Progress**

or write these commands in the terminal

```
ansible-builder --version
```

Above command gives you the version of `ansible-builder` installed on the lab machine. For this lab that is 3.0.0 which should also be visible in the command output

```
ansible-builder --help
```

This command will give you help output of the `ansible-builder`; from the output you can see three subcommands available. For this lab, as we are focusing mainly on building execution environments, the `build` subcommand is important.

```
ansible-builder build --help
```

This command gives you the help section on the `build` subcommand of `ansible-builder`; please briefly take a look at these different options as they will be used throughout this lab.

**Next Challenge**

**Additional resources**

**Encountered an issue?**

**Well done!**

Loading your next challenge...

**Automation Hub**

**Entornos de ejecución**

Filtrar por ... Agregar entorno de ejecución Enviar imágenes de contenedores

Nombre del repositorio de con...	Descripción	Creado	Última modificación	Tipo de r...
ansible-builder-rhel8		hace 10 meses	hace 10 meses	Local
ee-29-rhel8		hace 10 meses	hace 10 meses	Local
ee-minimal-rhel8		hace 10 meses	hace 10 meses	Local
ee-supported-rhel8		hace 10 meses	hace 10 meses	Local
netcommon-hub-ee		hace 2 minutos	hace 2 minutos	Local

1-5 of 5

Task - Let's check what's built

```
podman images
```

You will see that it builds an execution environment called `localhost/ansible-execution-env` by default, this is our execution environment built with the collection we need. You can tag the image post build if you want using the podman CLI using the following command, please run this command.

```
podman tag localhost/ansible-execution-env localhost/netcommon-hub-ee
```

Task - Let's push this EE to Automation Hub

Automation hub also acts as a registry to store your custom execution environments. To push this execution environment to Automation hub, you need to first tag the EE with a new name and address of the automation hub, please run the following commands.

Login to automation hub with podman

```
podman login privatehub-01.$INSTROQT_PARTICIPANT_ID.instrukt.io
```

Check

**VSCode**

**Automation Hub**

File Edit Selection View Go Run Terminal Help

execution-environment.yml - minimal-downstream-with-hub-certs - code-server

MINIMAL-DOWNSTREAM-WITH-HUB-CERTS

execution-environment.yml

```
version: '1'
build:
  context: .
  dockerfile: Dockerfile
  args:
    - HUB_CERTS_DIR=/etc/pki/ca-trust/source/anchors
    - HUB_CLOUD_ID=privatehub-01.$INSTROQT_PARTICIPANT_ID.instrukt.io
  options:
    package_manager_path: /usr/bin/microdnf
  additional_build_files:
    - src: files
      dest: configs
  additional_build_steps:
    prepended_galaxy:
      - COPY _build/configs/ansible.cfg /etc/ansible/ansible.cfg
    prepended_base:
      - COPY _build/configs/cert.pem /etc/pki/ca-trust/source/anchors/
      - RUN update-ca-trust
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

Copying blob 58c6abb1c191 done  
Copying blob 5dd1d888fc done  
Copying blob 536270d66cc6 done  
Copying blob aa083d408fae done  
Copying blob c234588c3916 skipped: already exists  
Copying blob bb46993144ab skipped: already exists  
Copying blob 33986316e89f done  
Copying blob 371f2ebdd1a done  
Copying blob afdfa955e218 done  
Copying blob 32597689953f done  
Copying config 59f7636276 done  
Writing manifest to image destination  
Storing signatures

(rhel@ansible) \$

1 M 26 Col 1 Space 2 UTF-8 LF YAML Layout US Ansible Execution Environment

Well done!

Encountered an issue?

Next Challenge

Username: admin  
Password: ansible123!

Let's verify if the image is pushed to Automation Hub,  

- Go to Automation Hub tab
- Login using the below credentials

In the next challenge, we will try to pull the collection without the token inside the ansible.cfg file.

Well done!

Loading your next challenge...

**MINIMAL-DOWNSTREAM-WITH-HUB-CERTS**

```

12
13   options:
14     | package_manager_path: /usr/bin/microdnf
15
16   additional_build_steps:
17     | src_files:
18       | dest: configs
19
20   additional_build_steps:
21     prepend_galaxy:
22       | - ARG ANSIBLE_GALAXY_SERVER_RH_CERTIFIED_REPO_TOKEN
23     prepend_base:
24       | OSPL_Bndl/configs/cert.pem /etc/pki/ca-trust/source/anchors/
25       | rpm update-ca-trust
26
27
28
29
30

```

**PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    JUPYTER**

**TERMINAL**

```

[4/4] STEP 15/20: RUN rm -rf /output
--> 5f23e6970d6
[4/4] STEP 17/20: LABEL ansible-execution-environment=true
--> 6ad9f86bc6e
[4/4] STEP 18/20: USER 1000
--> c9093108c4
[4/4] STEP 19/20: ENTRYPOINT ["/opt/builder/bin/entrypoint", "dumb-init"]
--> 770c25f6e
[4/4] STEP 20/20: CM ["Bash"]
[4/4] STEP 21/20: ansible-execution-env:latest
--> 11180974680
Successfully tagged localhost/ansible-execution-env:latest
11180974833a5bdeaa02881832b6409302ee3787287c4bd7d24af45977212f

```

**Completion! The build context can be found at: /home/rhel/minimal-downstream-with-hub-certs/context**

**Powered by** **instruct**

**Progress** **2m**

**Documentation** **User Access**

Now come back to the VSCode tab and click on Terminal, to open a new terminal. Once done, type the following command to populate the token locally

```
copy
export ANSIBLE_GALAXY_SERVER_RH_CERTIFIED_REPO_TOKEN=paste_token_here
```

**NOTE** Make sure to paste the token after the "=" sign in the above command.

**Task - Let's build the EE**

Now that you have populated the token locally, let's try rebuilding our execution environment. Run the following command

```
copy
ansible-builder build -v 3 --build-arg ANSIBLE_GALAXY_SERVER_RH_CERTIFI
```

The above command will pass the local token to the definition file and make the token available to the build process.

Hopefully, if you have done everything correctly you should have your new execution environment, this way you didn't have to save the token inside your ansible.cfg file which can be advantageous when you have to save the EE definition to source control systems like git, as you won't be saving token there for anyone to have access to your Automation hub.

**Well done!**  
You've completed the last challenge.

**Full screen** **Progress** **Completed**

**Thank you!**  
Your feedback will be shared with the content creator and used to improve this track for others.

[Add more feedback](#)

## Managing user access and content upload policies with private automation hub.

We will focus on two main personas in this lab:

- Automation Curator - Someone who will manage content in the private automation hub for your organization. Curator is the manager/overseer of all automation content.
- Automation Developer - An automation developer who is associated with a specific group in your organization for e.g a network, cloud or storage automation developer whose role is write Ansible automation for their group.

For this lab, curator wants to give access to cloud and network automation developers the permissions to push content collections and automation execution environments to the private automation hub instance.

Hands-On Lab

Loading... now is a good time to grab a cup of coffee  
Estimated time left: 02:18 - [Notify me](#)

The screenshot shows a hands-on lab interface. On the left, there's a terminal window titled 'automationhub-web' with a sub-tab 'automationhub-terminal'. The terminal displays a command-line session where a configuration file 'galaxy-importer.cfg' is being viewed. The file contains settings like 'RUN\_ANSIBLE\_TEST = False' and 'CHECK\_REQUIRED\_TAGS = True'. On the right, there's a 'Progress' bar at 42m, a note about the 'backend' tab, and a list of tasks: 'automationhub-terminal tab is open, click on the tab and go through the next steps to check how upload policies have been setup.' and 'Run the following command on the terminal'. Below this is a code block with the command 'cat /etc/galaxy-importer/galaxy-importer.cfg'. Further down, there's a note about upload policies, a 'NOTE' about content approval, and a summary section with three items: 'Summary', 'Next Challenge', and 'Encountered an issue?'. The 'Well done!' section at the bottom says 'Loading your next challenge...'.

**Thank you!**

Your feedback will be shared with the content creator and used to improve this track for others.

[Add more feedback](#)

## 💡 Descripción general del entorno de ACME Corp



Laboratorio práctico

Empezar

**Vista de topología**

**Leyenda**

- Nodo de control (C)
- Nodo de ejecución (Ex)
- Nodo híbrido (Hy)
- Nodo satélite (h)

**Detalles**

Haga clic en el ícono de un nodo para mostrar los detalles.

**Topology View**

Legend

- Nodo de control
- Nodo de ejecución
- Nodo híbrido
- Nodo satélite

Estado:

- Activo
- Inactivo
- Aproximación
- Desaproximación
- Error
- Habilitado
- Deshabilitado

Tipos de enlace:

- Activo
- Alavancando
- Dividiendo

El visor de topologías es una forma sencilla de ver el estado de mantenimiento de las instancias. Puede ver el estado de las instancias y el tipo de nodo de malla en la esquina superior derecha.

También puede profundizar en la configuración del nodo haciendo clic en el nombre de la instancia.

Próximo reto

Has encontrado algún problema?

Bien hecho!

Cargando tu próximo reto...

**Instruct** Controlador

Red Hat Enterprise Automation Platform

Plantillas Credenciales Proyectos Inventarios Servidores

Acceso Organizaciones Usuarios Equipos

Administración Tipos de credencial Notificación Trabajos de gestión Grupos de instancias

Instancias Aplicaciones Entornos de ejecución Vista de topología Ajustes

Grupos de instancias > Johannesburg data center

Instancias

Valores de grupo de instancias

Nombre Estado Tipo de nodo Ajuste de la capacidad Capacidad usada Acción

jhb-exec Listo execution CPU 8 forks 56 RAM 0 Capacidad usada 0% 0

1-1 de 1 elementos 1 de 1 página

Detalles

Host Name: jhb-exec Status: Healthy Policy Type: Auto Instance Groups: default, Johannesburg data c. Last Health Check: 19/01/2023, 00:20:28 Capacity Adjustment: CPU 8 forks 56 RAM 56 Ran health check: Enabled

Ran health check: Enabled

Anota la fecha y hora de la última comprobación de estado.

Proximo reto:  ¿Has encontrado algún problema?

Bien hecho! Cargando tu próximo reto.

---

instruct Controlador

Panel de control Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo

Recursos Plantillas Credenciales Proyectos Inventarios Servidores

Acceso Organizaciones Usuarios Equipos

Administración Tipos de credencial

Vistas

Trabajos > II - Debug info

Salida

Debug info Correctamente Jugadas: 1 Tareas: 6 Servidores: 1 Tiempo transcurrido: 00:00:09

Stdout

Procesamiento de eventos completo. Descargar salida

```

6 TASK [Gather top CPU hogs] *****
7 ok: [web.jhb.example.com]
8
9
10 TASK [Gather top memory hogs] *****
11 ok: [web.jhb.example.com]
12
13 TASK [Swap + wait states] *****
14 ok: [web.jhb.example.com] => {
15   "vmlinstdout_lines": [
16     "procs -----memory----- swap----io---system---cpu---",
17     "r b swpd free buff cache si so bi bo in cs us syid wa st",
18     "1 0 0 2177708 3204 1185804 0 0 343 325 201 155 8 190 0 0",
19     "0 0 0 2177768 3204 1185852 0 0 0 16 1688 2500 3 2 96 0 0",
20     "0 0 0 2177548 3204 1185852 0 0 0 16 1781 2763 3 1 97 0 0",
21   ]
}

```

ACME Corp IT recibió informes de tiempos de respuesta lentos en el sitio web. Deben recopilar información del sistema de las instancias de Red Hat Enterprise Linux (RHEL) en la oficina remota de Johannesburg y en el centro de datos de Raleigh.

Usarán la plantilla de trabajo creada previamente para recopilar información y encontrar errores. **Debug info**

Vamos a ejecutar la plantilla de trabajo en el centro de datos de Raleigh. **Debug info**

- En la navegación lateral de la sección Recursos, haga clic en **Plantillas**. Haga clic en el ícono situado debajo de la columna **Acciones** en la fila de la plantilla de trabajo. Esto abrirá una nueva ventana **Launch | Debug info**.

Launch | Debug info

Selected: Test Job

Name:  Johanesburg DC

Raleigh DC

Launch Cancel

Bien hecho! Cargando tu próximo reto..

**instruct** Controlador

Vistas Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos Administración Tipos de credencial Notificación Trabajos de gestión 1 INTENTOS

Trabajos > 18 - Mesh route info Salida

Mesh route info Ejecutándose Tiempo transcurrido 00:00:01

Stdout Seguir

Esperando la salida de la tarea...

15m

Tenga en cuenta el resultado de la plantilla de trabajo. Mesh route info

Error no route to node from releigh-controller

Tarea: configurar el nodo dublin-hop

ACME Corp debe permitir que el nodo de salto entre el tráfico de automatización entre Raleigh y Johannesburg. Han creado la plantilla de trabajo y un nuevo inventario en el controlador de automatización para realizar la configuración. **Setup Dublin hop node Dublin DC**

Nota La plantilla de trabajo es un ejemplo sencillo para demostrar la resistencia de la malla de automatización. Visite la documentación oficial para conocer las arquitecturas compatibles y las prácticas recomendadas. **Setup Dublin hop node**

Vamos a ejecutar la plantilla de trabajo para habilitar el nodo. **Setup Dublin hop node dublin-hop**

- En la navegación lateral de la sección Recursos, haga clic en.

Bien hecho! Cargando tu próximo reto..

**instruct** Controlador

Vistas Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos

Panel de control

3 Servidores 0 Hosts fallidos 3 Inventarios 0 Errores de sincronización de inventario 1 Proyectos 0 Errores de sincronización del proyecto

Estado de la tarea Tareas recientes Plantillas recientes

Mes pasado Todos los tipos de tarea Todas las tareas

Ejecuciones de trabajo

Progreso 9m

Configure el inventario para que utilice el grupo de instancias. Johannesburg DC Johannesburg data center

Tarea: configurar el nodo dublin-hop

Ejecute plantillas de trabajo para configurar y probar el nodo de salto de Dublin.

Ejecute la plantilla de trabajo para habilitar el nodo. **Setup Dublin hop node dublin-hop**  
Ejecutar la plantilla de trabajo con el inventario **Debug info Johannesburg DC**

Si la plantilla de trabajo se ejecuta correctamente, ¡ha completado su misión! **Debug info**

Nota La plantilla de trabajo puede tardar unos segundos en iniciarse mientras el nodo está disponible. **Debug info jhb-exec**

Felicitaciones por completar el laboratorio!

Cierre del laboratorio

Has encontrado algún problema?

Bien hecho! Has completado el último desafío.

**¡Gracias!**

Sus comentarios se compartirán con el creador de contenido y se utilizarán para mejorar esta pista para otros.

[Añade más comentarios](#)

Powered by **instruct**

Pantalla completa

# Introducción a la automatización de ServiceNow

Tiempo límite: 55 minutos

Laboratorio práctico

Lanzar →

The screenshot shows a web-based automation lab interface. At the top, there's a header with the 'instruct' logo and a 'Pantalla completa' button. Below the title 'Introducción a la automatización de ServiceNow', it says 'Tiempo límite: 55 minutos'. A 'Lanzar →' button is visible on the right. The main area is a dark-themed workspace divided into several sections:

- Code Editor:** Shows two files: 'readme.md' and 'main.yml'. The 'main.yml' file contains the following Ansible code:

```
1 # Environment credentials
2
3 ## Automation Controller
4 - username: student
5 - password: learn_ansible
6
7 ## ServiceNow
8 - username: yte9pokfbs1
9 - password: yte9pokfbs1
10
11
```
- Progress Bar:** Shows a progress bar at 52m.
- Credential Table:** A table titled 'Progreso' with two rows:

llave	valor
nombre de usuario	YTE9POKFBS1
contraseña	YTE9POKFBS1
- ServiceNow Navigation:** A sidebar with 'servicenow Service Automation' and links for 'Home', 'Self-Service - Incidents', and 'Problem - Assigned to me'.
- Summary Message:** A green box at the bottom right says '¡Deberías ver que se ha creado un nuevo incidente! Para verificar que lo ha hecho usted, acceda al incidente recién creado con la víñeta y la captura de pantalla anteriores. Su nombre de usuario único para este entorno debe mostrarse en la descripción del incidente. El número de incidente también debe coincidir con el número de incidente que se muestra en la salida de ejecución del trabajo dentro de Automation Controller.'
- Feedback Section:** A green box asking '¿Has encontrado algún problema?' with a 'Sí' button.
- Status Bar:** At the bottom, it says '¡Bien hecho! Cargando tu próximo reto...'.

**Trabajos > 5 - 2 - Attach problem (problem-attach.yml)**

### Salida

2 - Attach problem (problem-attach.yml) Correctamente Jugadas 1 Tareas 7 Servidores 1 Tiempo transcurrido 00:00:14

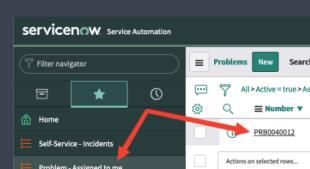
```
Stdout ▾ 🔍 Procesamiento de eventos completo. Descargar salida
1 al_request": "", "short_description": "User created a new incident using Ansible Automation Platform", "close_code": 0, "closed_by": "self-service", "display": "work_start", "first_contact": "2024-05-02T01:06:07Z", "last_update": "2024-05-02T01:06:07Z", "location": "localhost", "log": "INFO", "log_level": "INFO", "opened_at": "2024-05-02T01:06:07Z", "parent_incident": "sys_id": "8c0375fb7b1c2505de0da0a93bb35ee", "contact_type": "email", "reopened_by": "self-service", "incident_state": "1", "urgency": "low", "problem_id": "company", "reassignment_count": "0", "activity_due": "2024-05-02T01:06:07Z", "assigned_to": "severin", "severity": "3", "comments": "", "approval": "not requested", "sla_due": "", "comments_and_work_notes": "", "due_date": "2024-05-02T01:06:07Z", "sys_mod_count": "0", "reopen_count": "0", "sys_tags": "", "escalation": "", "upon_approval": "proceed", "correlation_id": "", "location": "", "category": "inquiry", "attachments": []}
11
12 TASK [Create a problem from incident] *****
13 [WARNING]: Encountered unknown value 1 while mapping field state.
14 [WARNING]: Encountered unknown value 1 while mapping field problem_state.
15 changed: [localhost]
16
17 TASK [Assign a problem to a user for assessment] *****
18 changed: [localhost]
19
20 TASK [Update incident status now that problem has been created] *****
21
22 PLAY RECAP *****
23 localhost : ok=5 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

**Progreso** 50m

A continuación, navegue y ejecute la nueva plantilla de trabajo desde el controlador de automatización presionando el icono de cohete junto a la plantilla. Resources > Templates 2 - Attach problem (problem-attach.yml)

● Inspeccionar los resultados

¡Eso debería haberse completado con éxito! Vuelve a ServiceNow y comprueba si hay un nuevo número de problema en los proveedores asignados:



serviceNow Service Automation

Filter navigator

Problems New Search

Home

All > Active = true > Asigned

Self-Service - Incidents

Number PB80040012

Actions on selected rows...

También puede comprobar si el estado del incidente se ha actualizado seleccionando su número de incidente en. Ahora también debería haber un problema al que se hace referencia en la página del incidente. Self-service - Incidente

Pase a la siguiente sección seleccionando el botón verde "Siguiente" a continuación.

¿Has encontrado algún problema?

¡Bien hecho!

Cargando tu próximo reto...

**Trabajos > 6 - 3 - Attach change request (change-attach.yml)**

### Salida

3 - Attach change request (change-attach.yml) Correctamente Jugadas 1 Tareas 5 Servidores 1 Tiempo transcurrido 00:00:11

```
Stdout ▾ 🔍 Procesamiento de eventos completo. Descargar salida
0
1 PLAY [Automate SNOW] *****
2
3 TASK [Gathering Facts] *****
4
5 TASK [Assign problem for assessment] *****
6 changed: [localhost] => (item={number: 'PRB0041645', opened_at: '2024-05-02 01:06:07'})
7 [WARNING]: Encountered unknown value 3 while mapping field state.
8 [WARNING]: Encountered unknown value 3 while mapping field problem_state.
9
10 TASK [Create change request for resolving a problem] *****
11 changed: [localhost] => (item={number: 'PRB0041645', opened_at: '2024-05-02 01:06:07'})
12
13 PLAY RECAP *****
14 localhost : ok=5 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

**Progreso** 49m

elimine o modifique para abordar un problema que puede estar relacionado con un incidente pasado o en curso.

Se ha creado un nuevo cuaderno de estrategias en la pestaña vs code

Revise el nuevo cuaderno de estrategias que se llama como antes change-attach.yml

● Crear una solicitud de cambio

Navegue hasta el controlador de automatización e inicie la plantilla de trabajo Resources > Job Templates 3 - Attach change request (change-attach.yml)

- Esté atento a la finalización del trabajo en el controlador de automatización

● Inspeccionar los resultados

Vuelve a ServiceNow, selecciona debajo de tus favoritos. Esto debería enumerar todas las solicitudes de cambio abiertas. Deberá ver que se ha creado una nueva solicitud de cambio con el título. Observe cómo también se han actualizado otros campos relevantes (como "Descripción" y "Motivo de espera"). Change - Open "Reboot the webserver"

Pase a la siguiente sección seleccionando el botón verde "Siguiente" a continuación.

¿Has encontrado algún problema?

¡Bien hecho!

Cargando tu próximo reto...

**Controlador de automatización**

```

1  --- name: Autonome SNOW
2  hosts: localhost
3  connection: local
4  collections:
5  - servicenow_items
6  vars:
7  | demo_username: "{{ lookup('env', 'SN_USERNAME') }}"
8
9  tasks:
10
11  - name: Create or update configuration item
12    servicenow_item_configuration_item:
13      name: "{{ item.hostname }}-{{ demo_username }}"
14      assigned_to: "{{ demo_username }}"
15      ip_address: "{{ item.default_ip }}"
16      mac_address: "{{ item.default_mac }}"
17      environment: test
18      others:
19        sys_class_name: cmdb_ci_linux_server
20        loop: "{{ node_info }}"
21        register: configuration_item
22
23
24  - name: debug
25    debug:
26      msg: "{{ configuration_item }}"
27
28
29
30

```

**Progreso**

Los dos nuevos cuadros de estrategias se han combinado en una única plantilla de trabajo de flujo de trabajo en el controlador de automatización denominada: 4.0 - Query node info and update CROR (multiple job templates)

- Esta nueva línea de plantilla de trabajo de flujo de trabajo también tiene un nuevo botón que, cuando se hace clic en él, muestra una visualización de las tareas que se están realizando que se muestran como nodos en un mapa.
- Iniciar la nueva plantilla de trabajo de flujo de trabajo
- A medida que se ejecuta el flujo de trabajo, al hacer clic en cualquiera de los nodos debería acceder al trabajo correspondiente que se inició para cada uno de los nodos del visualizador.

**Inspeccionar los resultados**

Para ver el grupo actualizado de elementos de configuración de Linux, escriba "Linux" en el cuadro del navegador de filtros de ServiceNow y haga clic en para ver una lista de elementos de configuración en el grupo de servidores de Linux. Configuration -> Servers -> Linux

- Observe que se han creado dos nuevos elementos de configuración y que los campos de dirección IP se han adjuntado a los elementos de configuración

Pase a la siguiente sección seleccionando el botón verde "Siguiente" a continuación.

Has encontrado algún problema?

Bien hecho!

Cargando tu próximo reto...

**Controlador de automatización**

**Trabajos > 7 - 5 - Query and close records by user (close-records-by-user.yml)**

**Salida**

Volver a Tareas Detalles Salida

5 - Query and close records by user (close-records-by-user.yml) Ejecutándose Tiempo transcurrido 00:00:00

Stdout Seguir

```

5
6 TASK [find user created problems] *****
7 [WARNING]: Encountered unknown value 3 while mapping field state.
8 [WARNING]: Encountered unknown value 3 while mapping field problem_state.
9 ok: [localhost]
10
11 TASK [find user created change requests] *****
12 ok: [localhost]
13
14 TASK [find user created configuration items] *****
15 ok: [localhost]
16
17 TASK [query incident number and creation time] *****
18 ok: [localhost] => {item={'parent': '', 'made_sla': 'true', 'caused_by': '', 'watch_list': '', 'upon_reject': 'cancel', 'sys_updated_on': '2024-05-02 01:06:13', 'child_incidents': '0', 'origin_table': '', 'task_effective_number': 'INC 0013966', 'approval_history': '', 'skills': '', 'number': 'INC0013966', 'resolved_by': '', 'sys_updated_by': 'VALUE _SPECIFIED_IN_NO_LOG_PARAMETER', 'opened_by': '102cacfdb3d4250b76a0d0f640cb3', 'user_input': '', 'sys_created_on': '2024-05-02 01:03:11', 'sys_domain': 'global', 'state': 'in_progress', 'route_reason': '', 'sys_crated_at': '2024-05-02 01:03:11', 'knowledg': 'false', 'order': '1', 'calendar_start': '', 'business_impact': 'VALUE _SPECIFIED_IN_NO_LOG_PARAMETER', 'priority': '5', 'sys_domain_path': '/', 'rfe': '', 'time_worked': '', 'expected_start': '', 'opened'

```

**Progreso**

de usuario. A continuación, Ansible transforma estos registros devueltos en listas simples de objetos y pasa las listas a sus respectivos módulos para actualizar/cerrar los registros. Para cualquier campo no implementado por el propio módulo, hay un parámetro de módulo llamado que se puede usar para especificar cualquier otro campo o campos personalizados para esa tabla en particular. close-records-by-user.yml \*\_info other

**Consultar y cerrar registros**

Para realizar esta automatización:

Navegue e inicie la plantilla de trabajo en el controlador de automatización Resources > Templates 5 - Query and close records by user (close-records-by-user.yml)

- Supervise la salida de la plantilla de trabajo y espere a que se complete el trabajo

**Inspeccionar los resultados**

Una vez que se complete el trabajo, todos los registros que había creado en desafíos anteriores deben cerrarse o eliminarse. La salida del trabajo debe mostrar todos los números de registro relevantes que se limpian. Actualice sus vistas en ServiceNow para ver las actualizaciones.

Ahora que todo se ha limpiado, pase al siguiente desafío! Haga clic en el botón verde "Siguiente" a continuación.

Has encontrado algún problema?

Bien hecho!

Cargando tu próximo reto...

instruct  Código VS  ServiceNow

student

Panel de control

Servidores: 0 | Hosts fallidos: 0 | Inventarios: 1 | Errores de sincronización de inventario: 0 | Proyectos: 0 | Errores de sincronización del proyecto: 0

Estado de la tarea: Mes pasado, Todos los tipos de tarea, Todas las tareas

Ejecuciones de trabajo: Fecha: 4/2, 4/4, 4/6, 4/8, 4/10, 4/12, 4/14, 4/16, 4/18, 4/20, 4/22, 4/24, 4/26, 4/28, 4/30, 5/2

Este inventario se puede dividir de muchas maneras diferentes. A continuación se muestra la consulta de inventario proporcionada para este inventario.

```
# Group hosts automatically, according to values of manufacturer
# Include only records with the specified operating system.
# Groups will most likely overlap.
plugin: servicenow.item.inventory
group_by:
  manufacturer:
    os:
      includes:
        - Linux Red Hat
        - Windows XP
```

El inventario anterior consulta la CMDB para los sistemas operativos Linux y Windows XP y agrupa los resultados por fabricante.

¡Todo terminado!

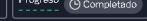
¿Has encontrado algún problema?

¡Bien hecho!

Has completado el último desafío.

https://aplicaciones0730.itsit.itsit.es/play/instruct.com

Powered by 

Pantalla completa 

**¡Gracias!**  
Sus comentarios se compartirán con el creador de contenido y se utilizarán para mejorar esta pista para otros.

Añade más comentarios

Last failed login: Fri Oct 28 05:01:07 UTC 2022 from 70.49.106.211 on ssh:notty  
There was 1 failed login attempt since the last successful login.  
[root@controller ~]\$

## Understanding the lab environment

It's important to understand the lab environment first before starting the challenges, in this lab you will see three tabs:

- Terminal** - We will use this to run commands to setup the Ansible project and sign it using `ansible-sign`
- Automation-Controller** - This is the WebUI for automation controller and will be used to verify content. You can login to the WebUI using the below username/password combination.
- Gitea** - Gitea is a community managed lightweight code hosting solution written in Go. This will be used to host the Ansible project and the project URL from gitea will be used in automation controller. You can login using to gitea using the below credentials.

Username: `admin`  
Password: `ansible123!`

Gitea - Gitea is a community managed lightweight code hosting solution written in Go. This will be used to host the Ansible project and the project URL from gitea will be used in automation controller. You can login using to gitea using the below credentials.

Username: `student`  
Password: `learn_ansible`

This system already has the CLI tool installed from the Ansible Automation Platform 2.3 installer. If you run the command on the terminal tab, you will find a few artifacts which are explained below: `ansible-sign ls`

`ansible-sign-demo` - This is an empty repo that will hold our playbooks and the contents of this repo will be signed using the tool in later challenges. `ansible-sign`

We will create more artifacts as we go and explain them as we go along.

Please click the **Next** button to continue learning.

**Próximo**

Last failed login: Fri Oct 28 05:01:07 UTC 2022 from 70.49.106.211 on ssh:notty  
There was 1 failed login attempt since the last successful login.  
[root@controller ~]\$

## Comprendiendo el entorno del laboratorio

Es importante comprender el entorno del laboratorio antes de comenzar los desafíos, en este laboratorio verás tres prestarias:

- Terminal** - Usaremos esto para ejecutar comandos para configurar el proyecto de Ansible y firmarlo usando `ansible-sign`
- Controlador de automatización** - Esta es la interfaz de usuario web para el controlador de automatización y se usará para verificar el contenido. Puede iniciar sesión en la interfaz de usuario web utilizando la siguiente combinación de nombre de usuario y contraseña.
- Gitea** - Gitea es una solución de alojamiento de código ligero administrada por la comunidad escrita en Go. Esto se usará para alojar el proyecto de Ansible y la URL del proyecto de gitea se usará en el controlador de automatización. Puede iniciar sesión con gitea con las siguientes credenciales.

Nombre de usuario: `admin`  
Contraseña: `ansible123!`

Gitea - Gitea es una solución de alojamiento de código ligero administrada por la comunidad escrita en Go. Esto se usará para alojar el proyecto de Ansible y la URL del proyecto de gitea se usará en el controlador de automatización. Puede iniciar sesión con gitea con las siguientes credenciales.

Nombre de usuario: `student`  
Contraseña: `learn ansible`

Este sistema ya tiene instalada la herramienta CLI desde el instalador de Ansible Automation Platform 2.3. Si ejecuta el comando en la pestaría de terminal, encontrará algunos artefactos que se explican a continuación: `ansible-sign ls`

`ansible-sign-demo` - Este es un repositorio vacío que contendrá nuestros libros de jugadas y el contenido de este repositorio se firmará con la herramienta en desafíos posteriores. `ansible-sign`

Crearímos más artefactos a medida que avancemos y los explicaremos a medida que avancemos.

Haga clic en el botón **Siguiente** para continuar aprendiendo.

**Bien hecho!**  
Cargando tu próximo reto...

instruct@controller ~ Terminal ↵ Controlador de automatización

```
...[long command output showing gpg key generation and project creation]...
cat ~/signing_demo.asc
```

**Utilizaremos todo el contenido de este archivo en los próximos desafíos cuando mostraremos la verificación de proyectos en el controlador de automatización.**

**Tarea 3: Verificar si el par de claves creado se muestra en el sistema**

Ejecute el siguiente comando para enumerar las claves gpg que existen en el sistema, esto debe tener solo un par de claves, el que se acaba de crear.

```
gpg --list-keys
```

**Resumen**

- Creamos un par de claves gpg en este almacén de claves del sistema predeterminado, la clave privada se utilizará para firmar el proyecto de Ansible localmente.
- Exportamos la clave pública a un archivo y usaremos el contenido del archivo en el controlador de automatización en los próximos desafíos para verificar el proyecto.

**Próximo reto**

**Has encontrado algún problema?**

**Bien hecho!**  
Cambiando tu próximo reto...

instruct@controller ~ Terminal ↵ Controlador de automatización ↵ Trabajos

**Vistas**

- Panel de control
- Trabajos
- Programaciones
- Flujo de actividad
- Aprobaciones del flujo de trabajo

**Recursos**

- Plantillas
- Credenciales
- Proyectos
- Inventarios
- Servidores

**Acceso**

- Organizaciones
- INTENTOS**

**⚠ Incorrect solution 10/10**  
El proyecto firmado no se creó en el controlador de automatización

**⚠ Incorrect solution 9/10**  
El proyecto firmado no se creó en el controlador de automatización

**⚠ Incorrect solution 8/10**

**Trabajos**

Nombre	Estado	Tipo	Hora de inicio	Hora de finalización
9 - Proyecto firmado	Correctamente	Actualización de fuente de control	1/5/2024, 19:44:41	1/5/2024, 19:44:45
8 - Proyecto firmado	Correctamente	Actualización de fuente de control	1/5/2024, 19:44:14	1/5/2024, 19:44:19
7 - Proyecto firmado	Correctamente	Actualización de fuentes de control	1/5/2024, 19:42:35	1/5/2024, 19:42:40

**Progreso**

```
PLAY [Perform project signature/checksum verification] *****
TASK [Verify project content using GPG signature] *****
ok: [localhost]
TASK [Verify project content against checksum manifest] *****
ok: [localhost]
```

Las tareas anteriores indican que la validación de la firma se realizó correctamente. En el próximo reto añadiremos más archivos a nuestro proyecto y veremos qué pasa si no firmamos esos archivos.

**Comprobar**

**;Gracias!**

Sus comentarios se compartirán con el creador de contenido y se utilizarán para mejorar esta pista para otros.

[Añade más comentarios](#)

Powered by  
**instruqt**

Pantalla completa

## DOD Automated Compliance

Explore Compliance as Code for Linux, Windows, Cloud, and Networking all in one place.

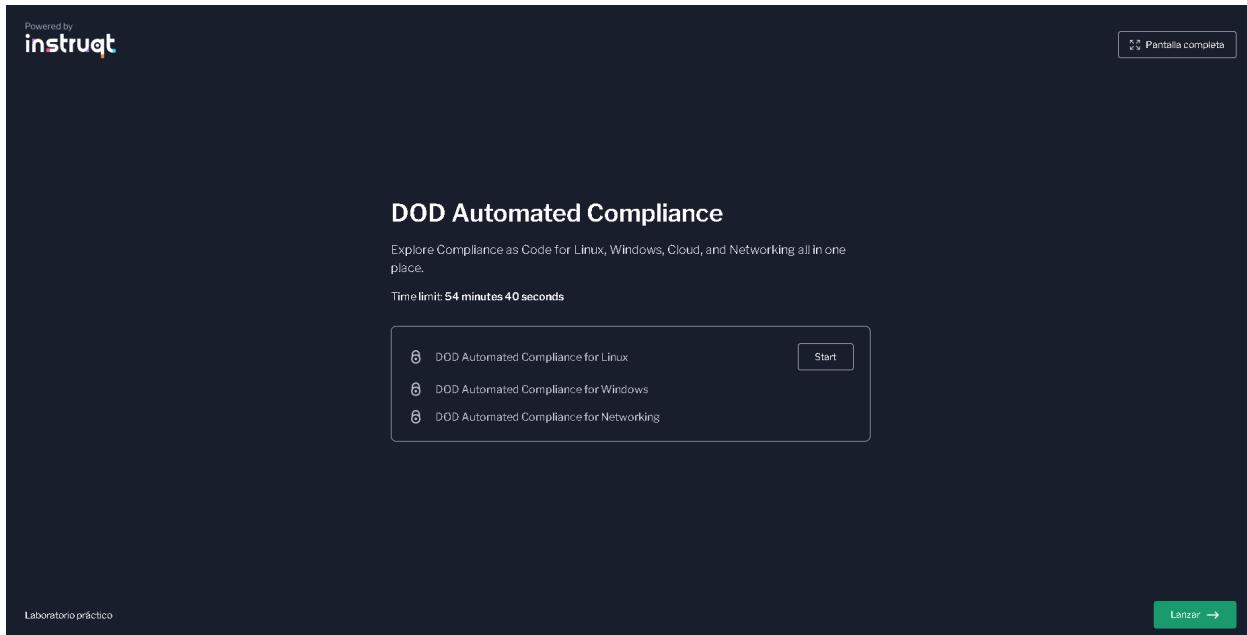
Time limit: 54 minutes 40 seconds

Start

⑥ DOD Automated Compliance for Linux  
⑥ DOD Automated Compliance for Windows  
⑥ DOD Automated Compliance for Networking

Laboratorio práctico

Lanzar →



instruqt Interfaz de usuario del controlador Código VS Nodo 1 Web web win1 admin

Vistas Panel de control Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos Administración Tipos de credencial Notificación

Trabajos > 5 - LINUX / Fact Scan Salida

« Volver a Tareas Detalles Salida

LINUX / Fact Scan Ejecutándose Tiempo transcurrido 00:00:01

Stdout Seguir

2	PLAY [Scan host for facts] *****	20:04:5
3	3	
4	TASK [Gathering Facts] *****	20:0
5	4:58	
6	ok: [node1]	
7	TASK [Get packages] *****	20:0
8	4:59	

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

Localice el trabajo **LINUX/DISA STIG** y haga clic en el botón . Ingrese al campo de límite y Lanzar `os_linux`

En esta plantilla de trabajo, Ansible se conecta a dos servidores RHEL, que se ejecutan en la nube, a través de SSH y luego ejecutan tareas para cada elemento STIG (la mayoría requiere una sola tarea).

La conexión SSH es totalmente segura, ya que utiliza una clave SSH cifrada para realizar la autenticación. La clave es almacenada por Ansible pero, después de que se agrega por primera vez, es ilegible permanentemente, lo que garantiza que nadie pueda verla.

Tarea 3 - (opcional) Ejecutar trabajos adicionales

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

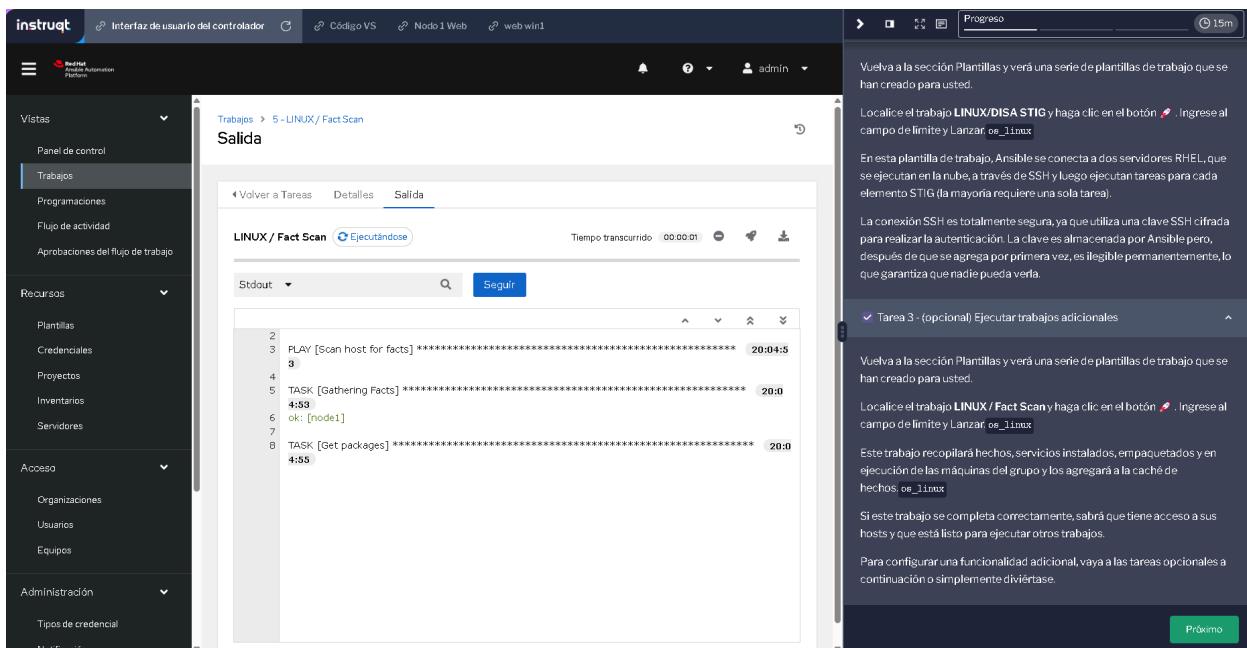
Localice el trabajo **LINUX / Fact Scan** y haga clic en el botón . Ingrese al campo de límite y Lanzar `os_linux`

Este trabajo recopilará hechos, servicios instalados, empaquetados y en ejecución de las máquinas del grupo y los agregará a la caché de hechos `os_linux`

Si este trabajo se completa correctamente, sabrá que tiene acceso a sus hosts y que está listo para ejecutar otros trabajos.

Para configurar una funcionalidad adicional, vaya a las tareas opcionales a continuación o simplemente diviértase.

Próximo



**instruct** Interfaz de usuario del controlador Código VS Nodo 1 Web web win1 setup 1/3 11m También se encontró: configurar install startup TIG DE WINDOWS / DISA para estar

Vistas Panel de control Trabajos Programaciones Flujo de actividad Aprobaciones del flujo de trabajo Recursos Plantillas Credenciales Proyectos Inventarios Servidores Acceso Organizaciones Usuarios Equipos Administración Tipos de credencial Notificación

Trabajos > 8 - SETUP Salida

Jugadas 1 Tareas 20 Servidores 1 Tiempo transcurrido 00:01:04

SETUP Correctamente

Stdout Procesamiento de eventos completo. Descargar salida

```
c: [[question_name : 'Server name or Pattern', type : 'text', variable : '_host', required : false, {'question_name': 'Service state to query?', 'type': 'multiplechoice', 'variable': 'service_state', 'require': false, 'default': 'Running', 'choices': ['Running', 'Stopped']}]], ansible_loop_var: '_controller_template_item'}
```

77 FAILED - RETRYING: [localhost] Configure Controller Job Templates | Wait for finish the job templates creation (30 retries left).

78 FAILED - RETRYING: [localhost] Configure Controller Job Templates | Wait for finish the job templates creation (29 retries left).

79 changed: [localhost] => (item='failed': 0, 'started': 1, 'finished': 0, 'ansible\_job\_id': 'j455448259118356', 'results\_file': '/tmp/ansible\_async/j455448259118356', 'changed': False, '\_controller\_template\_item': {'name': 'WINDOWS / Configuring Password Requirements', 'job\_type': 'run', 'inventory': 'Workshop Inventory', 'project': 'Ansible official demo project', 'playbook': 'windows/powershell\_desc.yml', 'no\_notification': true, 'templates\_started': 0, 'notification\_templates\_success': 0, 'notification\_templates\_error': 0, 'telemetry': 'Telemetry', 'credentials': 'Workshop Credential', 'survey\_enabled': True, 'survey': '(n/a)', 'description': ''}, 'spec': [{"question\_name": "Server Name or Pattern", "type": "text", "variable": "\_host", "required": false}], 'ansible\_loop\_var': '\_controller\_template\_item'})

80 changed: [localhost] => (item='failed': 0, 'started': 1, 'finished': 0, 'ansible\_job\_id': 'j358928083310373', 'results\_file': '/tmp/ansible\_async/j358928083310373', 'changed': False, '\_controller\_template\_item': {'name': 'WINDOWS / AD / Create Domain', 'job\_type': 'run', 'inventory': 'Workshop Inventor'}}

Tarea 2: Ejecutar la plantilla de trabajo STIG para Windows

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

Localice el trabajo **WINDOWS / DISA STIG** y haga clic en el botón . Ingrese al campo de límite y Lanzar. oe\_windows

En esta plantilla de trabajo, Ansible se conecta a un servidor Windows, se ejecuta en la nube, usa WinRM y luego ejecuta una tarea para cada elemento STIG (la mayoría requiere una sola tarea).

WinRM se puede configurar para que sea totalmente compatible con STIG mediante la aplicación de HTTPS y el uso de un método de autenticación cifrado.

Tarea 3 - (opcional) Ejecutar trabajos adicionales

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

Busque **WINDOWS / Ejecutar Powershell** en la sección Plantillas y haga clic en el botón . Entrar en la encuesta y Lanza. oe\_windows

Este trabajo ejecutará un comando de PowerShell determinado en el grupo de hosts. oe\_windows

Si este trabajo se completa correctamente, sabrá que tiene acceso a sus hosts y que está listo para ejecutar otros trabajos.

**Bien hecho!**

Cargando tu próximo reto...

NETWORK / DISA STIG 4/4 También se encontró: linux / disa stig

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

Localice el trabajo **NETWORK / DISA STIG** y haga clic en el botón . Se selecciona Intro en el menú desplegable Tipo de trabajo y, a continuación, en Iniciar clásico.

Esta plantilla de trabajo se conecta a un router Cisco que ejecuta IOS XE y verifica el estado actual de su cumplimiento STIG.

Al usar el modo, Ansible no realiza ningún cambio y solo informa sobre el cambio que habría realizado. Este resultado se puede utilizar para crear informes de cumplimiento de red automatizados del cliente. Check

Ahora, vuelva a la sección Plantillas y haga clic en para iniciar **NETWORK / DISA STIG**. Sin embargo, esta vez, haga clic en el menú desplegable Tipo de trabajo y seleccione Run.

Al ejecutar la plantilla de trabajo en modo, Ansible realiza los cambios que dice que debe realizar. Una vez completado este trabajo, dejará el router en un estado compatible. Run

Tarea 3 - (Opcional) Ejecutar trabajos adicionales

Vuelva a la sección Plantillas y verá una serie de plantillas de trabajo que se han creado para usted.

Localice el trabajo **NETWORK / Configuration** y haga clic en el botón . Ingrese en el campo de límite, haga clic en siguiente. all

Elija la configuración de red que desea implementar (por ejemplo, acl), haga clic en Siguiente y, a continuación, en Iniciar.

**Bien hecho!**

Has completado el último desafío.

**¡Gracias!**

Sus comentarios se compartirán con el creador de contenido y se utilizarán para mejorar esta pista para otros.

[Añade más comentarios](#)

Powered by  
**instruct**

Full screen

## Getting Started with Windows Automation

Access a Windows virtual desktop, right from your browser

Time limit: 3 hours

Hands-On Lab

Launch →

instruct windows vscode controller Progress 2h, 54m

tabs refresh button. This is due to the RDP session

We have a handy powershell script that can be run to assist with the WinRM configuration of a Windows host. On the Windows host, open Microsoft Edge and navigate to the following URL: <http://gitea:3000>.

**Windows File Explorer**

File Explorer showing the contents of the 'windows\_getting\_started' directory. The 'winrm\_setup' file is selected.

**Progress** (2h, 50m)

You will need to accept the changes being made you can accept all of them with.

Once the script has completed, we can now move our attention to the controller tab, where we have our Automation Controller.

Connecting to our Windows Host

**Automation Controller**

Credenciales > Windows Hosts

Detalles

Nombre	Windows Hosts	Descripción	Machine credential for Windows Hosts
Organización	Default	Tipo de credencial	Machine
Usuario	instrukt	Contraseña	Cifrado
Creado	29/4/2024, 20:51:32 por student	Último modificado	29/4/2024, 20:51:32 por student

**Groups**

No items found

Please add items or update the list

**Progress** (2h, 44m)

Username: instrukt  
Password:

You can leave the rest of the credentials details and save.

Next we need to make sure we have our host in an inventory. We use these inventories when deciding which hosts to act on with Ansible.

Navigate to Resources > Inventories. You will see an inventory has been created already called `servers`. Click on that inventory to get into the details, next select Groups.

**Groups**

No items found

Please add items or update the list

**Progress** (2h, 44m)

Next, click on Add so we can add a group to this inventory. Although we might be working with one or more hosts in this example, it is always beneficial to manage hosts as

**Next**

Trabajos > 1- win\_ping

**Salida**

win\_ping Correctamente

Tiempo transcurrido: 00:00:17

```
Stdout
Procesamiento de eventos completo. Recargar salida
0 SSH password:
1 windows | SUCCESS -> {
2     "changed": false,
3     "ping": "pong"
4 }
```

We need credentials for the host, select **windows Hosts** and proceed to the preview. Lastly we review of the ad-hoc command we just configured, once you are ready you can click on **Launch**. You should have a "successful job completion" to indicate that we can interact with the host using Ansible.

Proyectos > Windows Management

**Detalles**

Último estado de la tarea	Ejecutándose	Nombre	Windows Management
Descripción	My project for Windows automation	Organización	ACME
Tipo de fuente de control	Git	Revisión del control de fuentes	Sincronizar para revisión
URL de fuente de control	<a href="http://gitea:3000/student/windows_getting_started.git">http://gitea:3000/student/windows_getting_started.git</a>	Tiempo de espera de la caché	0 Segundos
Ruta base del proyecto	/var/lib/awx/projects	Directorio de playbook	_8_windows_management
Creado	29/4/2024, 21:05:34 por student	Último modificado	29/4/2024, 21:05:34 por student

Once you select Git as your Source Control Type, then the type details will appear, for our exercise we just need to configure the **Source Control URL**. Please configure this to:

[http://gitea:3000/student/windows\\_getting\\_started](http://gitea:3000/student/windows_getting_started)

Once done, you can save the project. You will see the project page will indicate automation controller is attempting to sync with the repo. Once complete, the Job status will be Successful.

Great! this means we have synchronised with the repo and

The screenshot shows a web-based interface for VS Code, specifically a "Get Started" guide. On the left, there's a sidebar with tabs for "instruct", "windows", "vscode", and "controller". The main area has tabs for "EXPLORER", "ANSIBLE", "Get Started", and "TERMINAL". The "TERMINAL" tab is active, displaying a terminal session:

```
[rhel@controller ansible]$ git clone http://gitea:3000/student/windows_getting_started.git
Cloning into 'windows_getting_started'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 7.24 KiB | 7.24 MiB/s, done.
[rhel@controller ansible]$ git config --global user.name "student@redhat.com"
[rhel@controller ansible]$ git config --global user.email "student@redhat.com"
[rhel@controller ansible]$ git config --global user.name "student"
[rhel@controller ansible]$
```

To the right, there's a "Progress" bar at 2h 23m. A section titled "Get Started with VS Code in the Web" shows a "Choose the look you want" dialog with two color palette options. Below it, a terminal window shows the command to clone a repository. Further down, instructions for configuring default repo details are shown, along with a terminal window for running "git config" commands.

The screenshot shows the Red Hat Ansible Platform interface. The left sidebar includes sections for "Vistas", "Trabajos", "Programaciones", "Flujo de actividad", "Aprobaciones del flujo de trabajo", "Metricas del host", "Recursos", "Plantillas", "Credenciales", "Proyectos", "Inventarios", "Servidores", "Acceso", "Organizaciones", "Usuarios", "Equipos", and "Administración".

The main area displays a "Trabajos > 3 - Local Account" job. The "Salida" tab is selected, showing a terminal output window. The output shows a successful playbook run for a "Local Account" with 10 tasks completed in 00:03:14:

```
PLAY [windows] *****
  0: PLAY RECAP *****
    1: PLAY [windows] *****
      0: PLAY RECAP *****
```

To the right, there's a "Progress" bar at 2h 25m. A survey titled "Reset Local Account" is displayed, asking for a username and password. The survey questions and answers are:

- Question: Please provide the username for the password reset  
Description: User's Account to be reset  
Answer variable name: username  
Answer type: text
- Question: Please provide the desired password  
Description: New Password  
Answer variable name: user\_password  
Answer type: password

A note at the bottom of the survey says: "Por último, no olvides habilitar la encuesta. Vuelva a Recursos > plantillas e inicie la plantilla. Cuando la Encuesta le pida que rellene los datos, utilice lo siguiente: Reset Local Account".

**Trabajos > 3 - Local Account**

**Salida**

Local Account Correctamente

Jugadas: 1 Tiempo transcurrido: 00:00:14

Stdout:

```
Procesamiento de eventos completo. Recargar salida
0 1 PLAY [windows] *****
05:29
2 3 PLAY RECAP *****
8:05:29
```

**Por último, no olvide habilitar la encuesta. Vuelva a Recursos > plantillas e inicie la plantilla. Cuando la Encuesta le pida que rellene los datos, utilice lo siguiente: Reset Local Account**

**Creación de usuarios y grupos locales**

**Bien hecho!**  
Cargando tu próximo reto...

---

**Registry Editor**

File Edit View Favorites Help

Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Update\AU

Name	Type	Data
AutoReboot	REG_DWORD	0x00000004 (4)
ScheduledInstall...	REG_DWORD	0x00000000 (0)
ScheduledInstall...	REG_DWORD	0x00000003 (3)

**Guarda la plantilla y ahora podemos actualizar el repositorio.**

**git add \***  
**git commit -m "Updated Site 2"**  
**git push**

**¡No olvides sincronizar tu proyecto! Por último, vaya a Vistas > Trabajos y vuelva a ejecutar la última ejecución de plantilla de IIS Webconfig haciendo clic en el cohete.**

**Bien hecho!**  
Esto incluye el laboratorio Introducción a Windows Automation

**Bien hecho!**  
Has completado el último desafío.

**¡Gracias!**

Sus comentarios se compartirán con el creador de contenido y se utilizarán para mejorar esta pista para otros.

[Añade más comentarios](#)