



PUCP

TEL141 - INGENIERÍA DE REDES CLOUD
PRESENTACIÓN FINAL - PLAN DE PRUEBAS EJECUTADO

Código - Nombres y Apellidos:
1 20200334 - Gil Zanabria
2 20195922 - Carlos Pisco
3 20202073 - Rodrigo Barrios

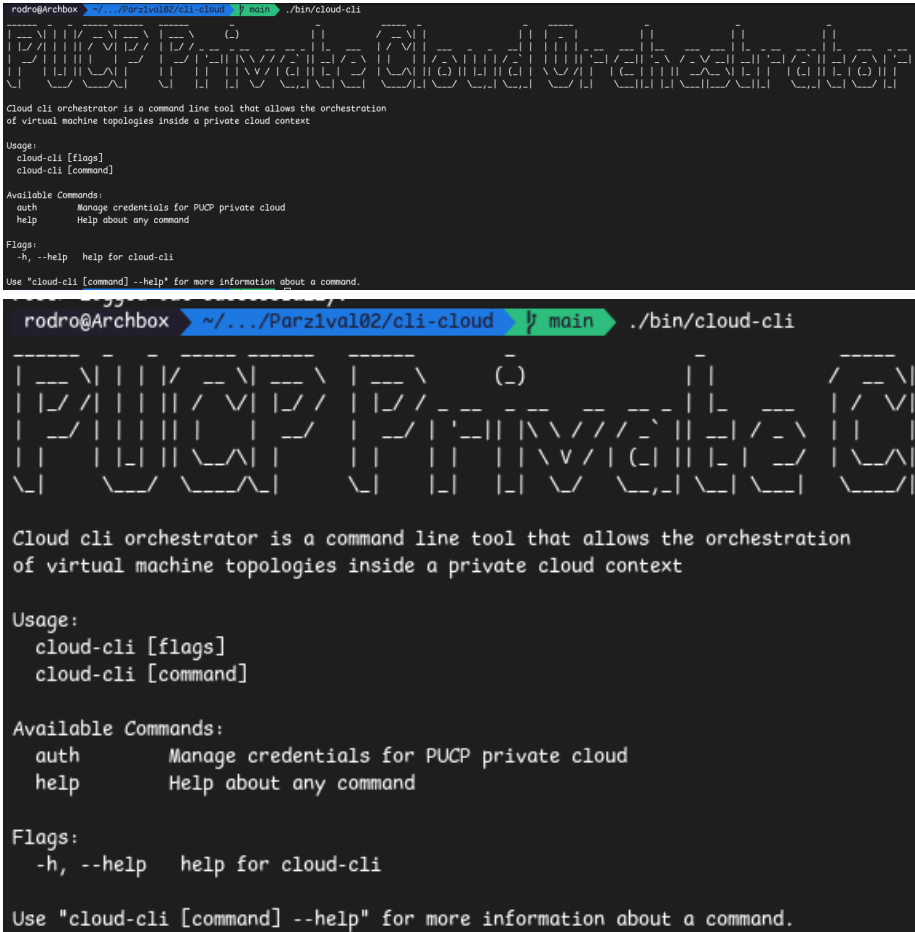
OUTLINE DE PRUEBAS

1. Módulo de interfaz de usuario (CLI)
 - 1.1. Ejecución del CLI
 - 1.2. Denegación de autenticación con credenciales inválidas
 - 1.3. Autenticación y Autorización de usuario administrador con credenciales válidas
 - 1.4. Autenticación y Autorización de usuario cliente con credenciales válidas
 - 1.5. Cierre de sesión de usuario
2. Módulo api gateway / AAA
 - 2.1. Validación de credenciales
 - 2.2. Generación y asignación de tokens de sesión
 - 2.3. Omisión de token de sesión en solicitud
 - 2.4. Token de sesión inválido en solicitud
 - 2.5. Reenvío de solicitudes http
3. Módulo gestor de plantillas
 - 3.1. Listado de plantillas
 - 3.2. Detalle de configuración de plantilla
 - 3.3. Creación de una nueva plantilla
 - 3.4. Edición de una plantilla
 - 3.5. Borrado de una plantilla
 - 3.6. Exportar plantilla
 - 3.7. Importar plantilla
 - 3.8. Graficar plantilla
4. Módulo de persistencia
 - 4.1. Verificación de acciones CRUD
5. Módulo de sistema de colas
 - 5.1. Ejecución correcta del sistema de colas
6. Módulo de scheduler (VM placement)
 - 6.1. Asignación de VMs según capacidad de los servidores.

- 7. Módulo de recursos disponibles
 - 7.1. Recursos extraídos de cada worker
- 8. Módulo Linux Cluster driver
 - 8.1. Despliegue de slices utilizando infraestructura de Linux Cluster
 - 8.2. Comunicación de nodos en un mismo slice utilizando infraestructura de Linux Cluster
 - 8.3. Borrado de slices utilizando infraestructura de Linux Cluster
 - 8.4. Generación de logs
- 9. Módulo de agente worker
 - 9.1. Registro de recursos computacionales en la base de datos
- 10. Módulo Openstack driver
 - 10.1. Despliegue de slices utilizando infraestructura de OpenStack
 - 10.2. Obtención de URL para acceso VNC a instancias en infraestructura de OpenStack
 - 10.3. Borrado de slices utilizando infraestructura de OpenStack

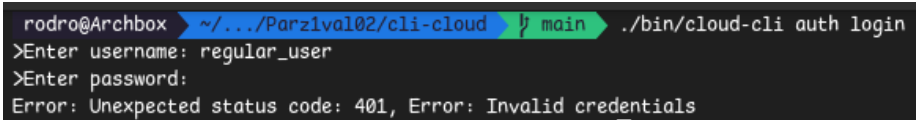
Módulo de interfaz de usuario

1.1 Ejecución del CLI

Objetivo del test	Prueba de ejecución del CLI
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Estar conectado a la red PUCP.2. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none">1. Ejecutar el binario `cloud-cli`
Resultados esperados	El CLI debe mostrar la pantalla principal con los subcomandos disponibles a ejecutar, auth y help.
Resultado del test	PASS
Evidencias	 <pre>rodre@Archbox ~/.../Parzival02/cli-cloud\$./bin/cloud-cli PARZIVAL Cloud cli orchestrator is a command line tool that allows the orchestration of virtual machine topologies inside a private cloud context Usage: cloud-cli [flags] cloud-cli [command] Available Commands: auth Manage credentials for PUCP private cloud help Help about any command Flags: -h, --help help for cloud-cli Use "cloud-cli [command] --help" for more information about a command. rodre@Archbox ~/.../Parzival02/cli-cloud\$ main ./bin/cloud-cli PARZIVAL Cloud cli orchestrator is a command line tool that allows the orchestration of virtual machine topologies inside a private cloud context Usage: cloud-cli [flags] cloud-cli [command] Available Commands: auth Manage credentials for PUCP private cloud help Help about any command Flags: -h, --help help for cloud-cli Use "cloud-cli [command] --help" for more information about a command.</pre>
Comentarios y/o consideraciones	<ol style="list-style-type: none">1. Al ser la primera vez que se ejecuta el binario, solo se tendrán disponibles dos comandos a ejecutar, auth y help.

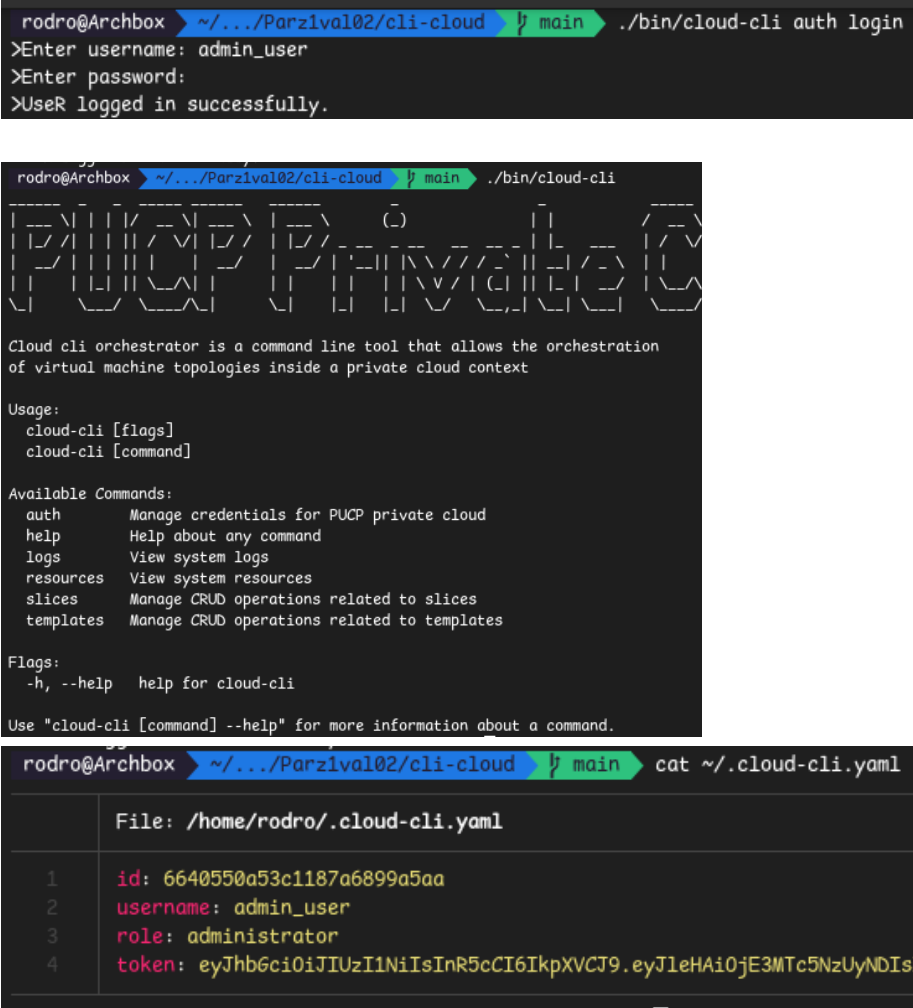
	<ol style="list-style-type: none"> El comando <code>auth</code> se encarga de los sub comandos relacionados con autenticación. El comando <code>help</code> permite visualizar mayor información de otros comandos.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.2 Denegación de autenticación con credenciales inválidas

Objetivo del test	Denegación de autenticación con credenciales inválidas
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> Estar conectado a la red PUCP. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none"> Ejecutar el binario <code>`cloud-cli`</code> con el comando <code>auth</code> y el subcomando <code>login</code> Suministrar credenciales incorrectas
Resultados esperados	El CLI avisa al usuario que las credenciales suministradas son incorrectas.
Resultado del test	PASS
Evidencias	 <pre> rodro@Archbox ~/.../Parzival02/cli-cloud \$ main ./bin/cloud-cli auth login >Enter username: regular_user >Enter password: Error: Unexpected status code: 401, Error: Invalid credentials </pre>
Comentarios y/o consideraciones	N/A

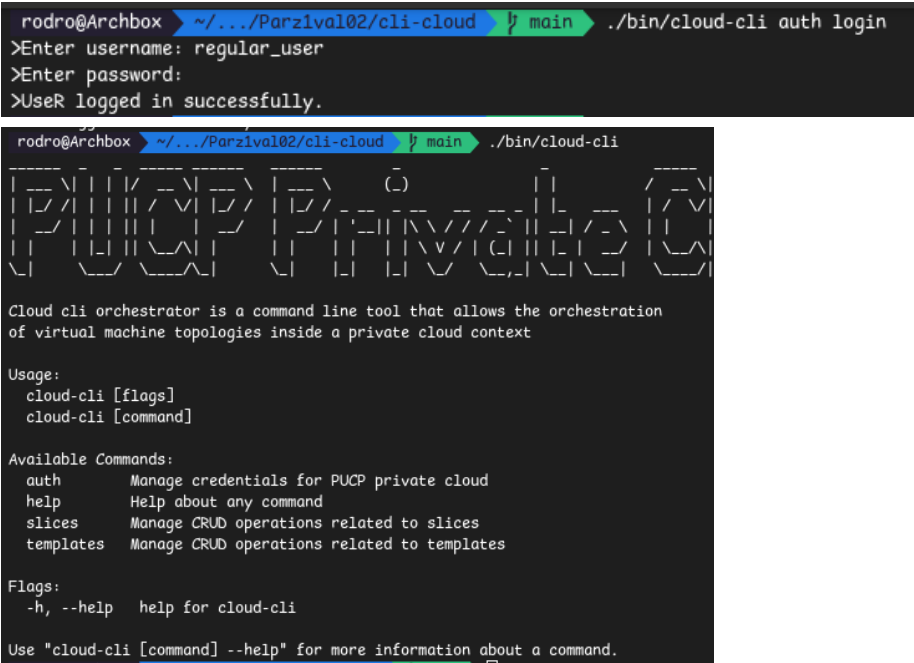
1.3 Autenticación y Autorización de administrador con credenciales válidas

Objetivo del test	Autenticación y autorización de administrador con credenciales válidas
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> Estar conectado a la red PUCP. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none"> Ejecutar el binario <code>`cloud-cli`</code> con el comando <code>auth</code> y el subcomando <code>login</code>

	<p>2. Suministrar credenciales correctas para un usuario con el rol de administrador</p>
Resultados esperados	<ol style="list-style-type: none"> 1. El usuario con rol de administrador se autentica de manera correcta en el sistema a través del uso del CLI. 2. Al autenticarse de manera correcta un usuario con rol de administrador, se desbloquean nuevos comandos que lo autorizan a realizar acciones adicionales en el sistema. <ol style="list-style-type: none"> a. templates: Este comando permite realizar operaciones CRUD sobre plantillas b. slices: Este comando permite realizar operaciones CRUD sobre slices c. logs: Este comando permite visualizar logs del sistema d. resources: Este comando permite visualizar resources del sistema e. zones: Este comando permite realizar operaciones CRUD sobre zonas de disponibilidad
Resultado del test	PASS
Evidencias	 <pre> rodro@Archbox ~/.../Parzival02/cli-cloud 1 main ./bin/cloud-cli auth login >Enter username: admin_user >Enter password: >User logged in successfully. rodro@Archbox ~/.../Parzival02/cli-cloud 1 main ./bin/cloud-cli Cloud cli orchestrator is a command line tool that allows the orchestration of virtual machine topologies inside a private cloud context Usage: cloud-cli [flags] cloud-cli [command] Available Commands: auth Manage credentials for PUCP private cloud help Help about any command logs View system logs resources View system resources slices Manage CRUD operations related to slices templates Manage CRUD operations related to templates Flags: -h, --help help for cloud-cli Use "cloud-cli [command] --help" for more information about a command. rodro@Archbox ~/.../Parzival02/cli-cloud 1 main cat ~/.cloud-cli.yaml File: /home/rodro/.cloud-cli.yaml 1 id: 6640550a53c1187a6899a5aa 2 username: admin_user 3 role: administrator 4 token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTc5NzUyNDIs </pre>

Comentarios y/o consideraciones	1. Para mantener al usuario en sesión, se almacena un archivo yaml en la carpeta home del usuario en el cual se guarda información como el id, username, rol y token.
---------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.4 Autorización de usuario con credenciales válidas

Objetivo del test	Autenticación y autorización de usuario con credenciales válidas
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Estar conectado a la red PUCP. 2. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none"> 1. Ejecutar el binario `cloud-cli` con el comando <i>auth</i> y el subcomando <i>login</i> 2. Suministrar credenciales correctas para un usuario
Resultados esperados	El usuario se autentica de manera correcta en el sistema a través del uso del CLI.
Resultado del test	PASS
Evidencias	 <pre> rodre@Archbox ~/.../Parzival02/cli-cloud \$./bin/cloud-cli auth login >Enter username: regular_user >Enter password: >User logged in successfully. rodre@Archbox ~/.../Parzival02/cli-cloud \$./bin/cloud-cli Cloud cli orchestrator is a command line tool that allows the orchestration of virtual machine topologies inside a private cloud context Usage: cloud-cli [flags] cloud-cli [command] Available Commands: auth Manage credentials for PUCP private cloud help Help about any command slices Manage CRUD operations related to slices templates Manage CRUD operations related to templates Flags: -h, --help help for cloud-cli Use "cloud-cli [command] --help" for more information about a command. </pre>

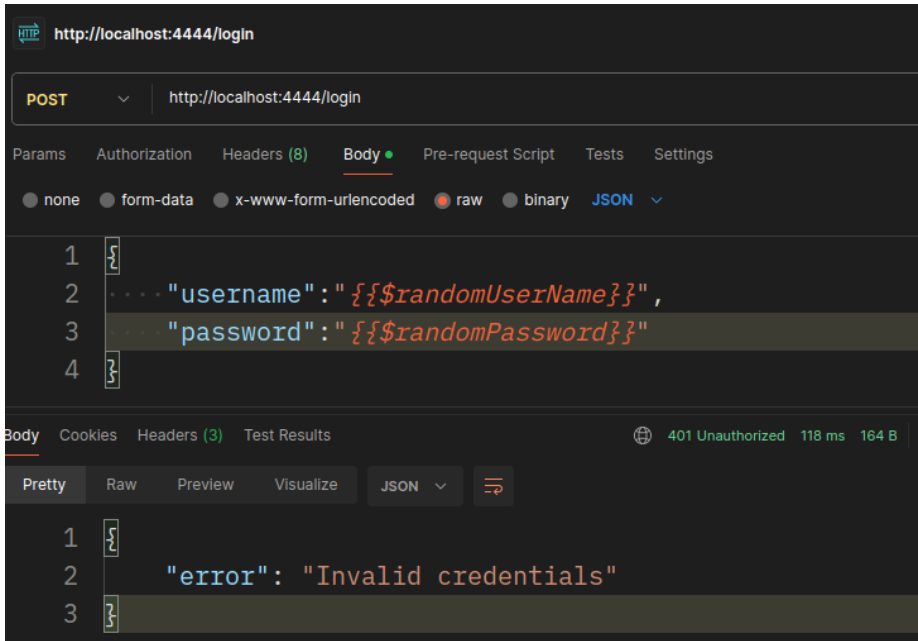
	<pre>rodro@Archbox ~/.../Parzival02/cli-cloud main cat ~/.cloud-cli.yaml File: /home/rodro/.cloud-cli.yaml 1 id: 6640550a53c1187a6899a5ab 2 username: regular_user 3 role: user 4 token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJlE3MTc5NzQ3NDks</pre>
Comentarios y/o consideraciones	<ol style="list-style-type: none"> Al autenticarse de manera correcta, se desbloquean nuevos comandos que lo autorizan a realizar acciones adicionales en el sistema. <ol style="list-style-type: none"> templates: Este comando permite realizar operaciones CRUD sobre plantillas slices: Este comando permite realizar operaciones CRUD sobre slices Para mantener al usuario en sesión, se almacena un archivo yaml en la carpeta home del usuario con información como el id, username, rol y token.

1.5 Cierre de sesión de usuario

Objetivo del test	Cierre de sesión de usuario
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> Estar conectado a la red PUCP. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar. Sesión iniciada de usuario
Procedimiento	<ol style="list-style-type: none"> Ejecutar el binario <code>cloud-cli</code> con el comando <code>auth</code> y el subcomando <code>logout</code>. Ejecutar el comando <code>cat ~/.cloud-cli.yaml</code> para verificar el contenido del archivo.
Resultados esperados	El usuario cierra su sesión en el sistema a través del uso del CLI. Además, no debe existir el archivo <i>cloud-cli.yaml</i> , <i>por lo que, el comando cat muestra error de archivo no encontrado.</i>
Resultado del test	PASS
Evidencias	<pre>rodro@Archbox ~/.../Parzival02/cli-cloud main ./bin/cloud-cli auth logout >User logged out successfully. rodro@Archbox ~/.../Parzival02/cli-cloud main cat ~/.cloud-cli.yaml [bat error]: '/home/rodro/.cloud-cli.yaml': No such file or directory (os error 2)</pre>
Comentarios y/o consideraciones	<ol style="list-style-type: none"> Para cerrar la sesión del usuario, se borra el archivo <code>.cloud-cli.yaml</code> en la carpeta home del usuario.

2. Módulo api gateway / AAA

2.1 Validación de credenciales

Objetivo del test	Prueba de validación de credenciales suministradas por usuarios (verificación de usuarios inválidos)
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Estar conectado a la red PUCP.2. Módulo de api gateway implementado y en funcionamiento en el HeadNode3. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.
Procedimiento	<ol style="list-style-type: none">1. Ejecutar un POST request al endpoint de login que expone el API GATEWAY y suministrar credenciales (username/password) para un usuario invalido (credenciales inválidas).
Resultados esperados	El endpoint del API GATEWAY devuelve la respuesta a la solicitud de inicio de sesión y muestra un mensaje de error (credenciales inválidas).
Resultado del test	PASS
Evidencias	 <p>The screenshot shows a REST client interface. The top section displays the URL 'http://localhost:4444/login' and the method 'POST'. The 'Body' tab is selected, showing a JSON payload: <pre>{ "username": "\${randomUserName}", "password": "\${randomPassword}"}</pre>. The bottom section shows the response status '401 Unauthorized' and the response body: <pre>{ "error": "Invalid credentials"}</pre></p>
Comentarios y/o consideraciones	<ol style="list-style-type: none">1. El api gateway recibe la solicitud de inicio de sesión, la cual contiene un nombre de usuario y contraseña en el cuerpo de la solicitud.2. Valida la existencia del nombre de usuario en MongoDB, así

	<p>como compara la contraseña suministrada con el hash guardado en base de datos.</p> <ol style="list-style-type: none"> 3. En caso las credenciales sean incorrectas, se envía un mensaje de error. 4. En caso las credenciales sean correctas, se procede a crear un json web token para la sesión del usuario
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

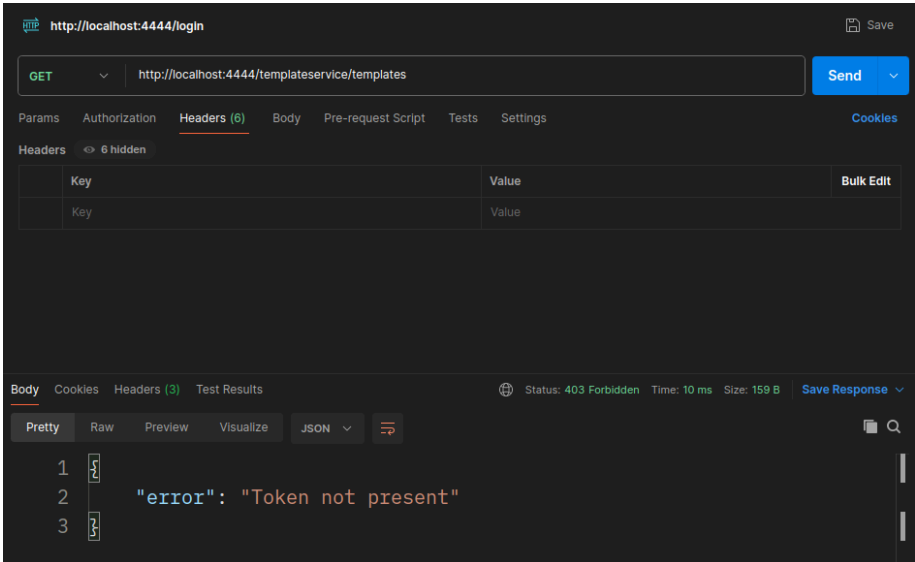
2.2 Generación y asignación de tokens de sesión

Objetivo del test	Prueba de generación y asignación de token de sesión a un usuario logueado.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Estar conectado a la red PUCP. 2. Módulo de api gateway implementado y en funcionamiento en el HeadNode 3. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.
Procedimiento	<ol style="list-style-type: none"> 1. Ejecutar un POST request al endpoint de login que expone el API GATEWAY y suministrar credenciales (username/password) para un usuario válido.
Resultados esperados	El endpoint del API GATEWAY devuelve la respuesta a la solicitud de inicio de sesión y muestra un mensaje de confirmación (credenciales correctas). Se reciben como respuesta campos de datos del usuario logueado (id, username, role, jwt) .Además, se guarda el jwt, así como información adicional del usuario, en un archivo yaml en la carpeta home del usuario para mantener activa la sesión.
Resultado del test	PASS

Evidencias	 <pre> 1 { 2 ... "username": "admin_user", 3 "password": "josemycoach" 4 } Body Cookies Headers (3) Test Results 200 OK 246 ms 435 B Save Response Pretty Raw Preview Visualize JSON 1 { 2 "user": { 3 "id": "6640550a53c1187a6899a5aa", 4 "username": "admin_user", 5 "role": "administrator", 6 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTg1MTExNzAsImkIjojY0MDU1MGE1M2MxMTg3YTY4OTlhNWZhIiwicm9sZSI6ImFkbWluaXN0cmF0b3IiLCJ1c2VybmFtZSI6ImFkbWluX3VzZXIifQ._vdVBL12JXgkz-RPUCBs1oA7Y22F0pkpZtwSV9K6FXo" 7 } 8 }</pre>
Comentarios y/o consideraciones	<ol style="list-style-type: none"> 1. El api gateway recibe credenciales correctas. 2. Se procede a crear y asignar un json web token a la sesión del usuario. 3. Se obtiene información adicional del usuario de la base de datos como id y rol 4. Se setean campos de información de usuario en el jwt como el id, nombre de usuario y rol 5. Se setea un tiempo de expiración al jwt 6. El api gateway responde al CLI con información del usuario, así como con el jwt para mantener activa la sesión del usuario y autorizar futuras peticiones

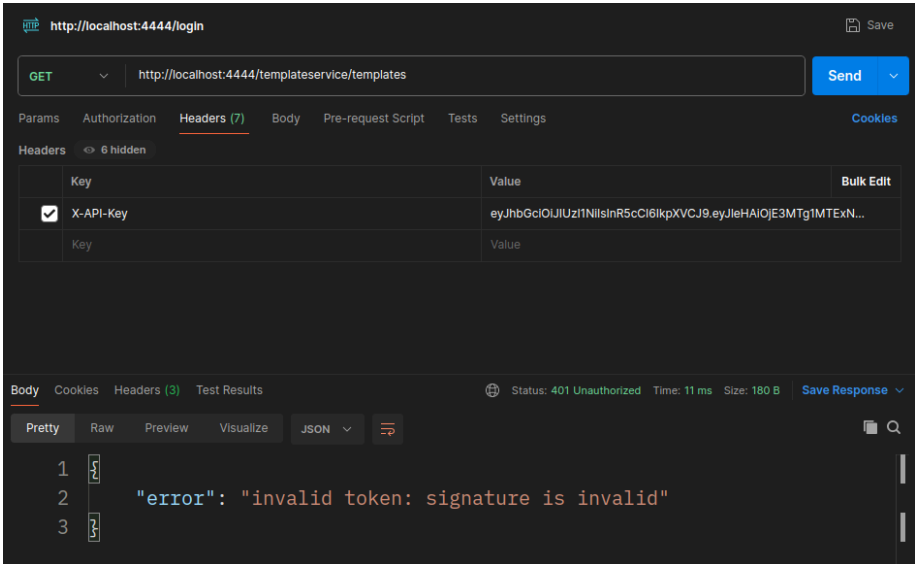
2.3 Omisión de token de sesión en solicitud

Objetivo del test	Prueba de omisión de envío del token de sesión en la cabecera de solicitudes al api gateway
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Estar conectado a la red PUCP. 2. Módulo de api gateway implementado y en funcionamiento en el HeadNode

	3. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.
Procedimiento	<ol style="list-style-type: none"> 1. La cabecera X-API-Key se coloca vacía en el request a realizar al endpoint objetivo (ej: <i>templates</i>). 2. Ejecutar un GET request al endpoint de <i>templates</i> en el API GATEWAY para intentar listar plantillas (respuesta correcta del API GATEWAY en caso se use jwt válido).
Resultados esperados	El API GATEWAY responde a la solicitud realizada con un mensaje de error de autorización. Este mensaje indica que es necesario el jwt en la realización de la solicitud.
Resultado del test	PASS
Evidencias	 <p>The screenshot shows a REST client interface. The URL bar displays 'http://localhost:4444/login'. The request method is 'GET' and the URL is 'http://localhost:4444/templateservice/templates'. The 'Headers' tab is selected, showing a table with 'Key' and 'Value' columns. The 'Body' tab is also visible, showing a JSON response: {\"error\": \"Token not present\"}. The status bar at the bottom indicates 'Status: 403 Forbidden', 'Time: 10 ms', and 'Size: 159 B'.</p>
Comentarios y/o consideraciones	<ol style="list-style-type: none"> 1. El api gateway valida que la cabecera contiene el token de sesión. 2. De no ser el caso, es rechazado el reenvío de la solicitud hacia los módulos internos. 3. Se responde con un mensaje de error y http status code 403 (no permitido)

2.4 Token de sesión inválido en solicitud

Objetivo del test	Prueba de envío de token de sesión inválido en la cabecera de solicitudes al api gateway
Diagrama de red del Test	N/A

Prerrequisitos	<ol style="list-style-type: none"> 4. Estar conectado a la red PUCP. 5. Módulo de api gateway implementado y en funcionamiento en el HeadNode 6. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.
Procedimiento	<ol style="list-style-type: none"> 1. Cambiar la cabecera X-API-Key por un string al azar de caracteres o esperar un tiempo prolongado a que expire el token de sesión 2. Ejecutar un GET request al endpoint de <i>templates</i> en el API GATEWAY para intentar listar plantillas (respuesta correcta del API GATEWAY en caso se use jwt válido).
Resultados esperados	El API GATEWAY responde a la solicitud realizada con un mensaje de error de autorización. Este mensaje indica que es necesario un jwt válido (no falso ni expirado) en la realización de la solicitud.
Resultado del test	PASS
Evidencias	 <p>The screenshot shows a REST client interface. The URL bar displays 'http://localhost:4444/login'. Below it, the request method is 'GET' and the URL is 'http://localhost:4444/templateservice/templates'. The 'Headers' tab is selected, showing a table with one header: 'X-API-Key' with a long alphanumeric value. The 'Body' tab is also visible, showing a JSON response: {\"error\": \"invalid token: signature is invalid\"}. The status bar at the bottom indicates 'Status: 401 Unauthorized', 'Time: 11 ms', and 'Size: 180 B'.</p>
Comentarios y/o consideraciones	<ol style="list-style-type: none"> 1. Se envía la cabecera X-API-Key con el valor del token recogido del archivo yaml 2. El api gateway válida que la cabecera no está vacía 3. El api gateway válida que el token de sesión existe y le pertenece a un usuario del sistema, así como su estado de expiración 4. De fallar una de estas validaciones, se responde con un mensaje de error y http status code 403 (no permitido)

2.5 Reenvío de solicitudes http

Objetivo del test	Prueba de reenvío de solicitudes http a los módulos internos de la solución.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Estar conectado a la red PUCP.2. Módulo de api gateway implementado y en funcionamiento en el HeadNode3. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none">1. Tras iniciar sesión de manera correcta, se asigna un json web token a cada usuario.2. El módulo api gateway recibe todas las solicitudes http que se realizan al ejecutar los diversos flujos desde el CLI.3. El json web token se utiliza en todas las posteriores solicitudes http dirigidas hacia el gateway. Se envía en la cabecera http X-API-Key4. El api gateway recibe la ruta completa, la recorta y reenvía hacia las rutas de los módulos internos.
Resultados esperados	El módulo api gateway recibe la cabecera en cada solicitud. A partir de aquí, valida que el jwt exista y no haya expirado. De ser satisfactorio, redirecciona la solicitud http hacia los módulos internos.
Resultado del test	PASS
Evidencias	
Comentarios y/o consideraciones	<p>Rutas definidas en api gateway:</p> <p>Api-gateway:</p> <ul style="list-style-type: none">• /templateservice/templates• /sliceservice/slices <p>Ruta definidas en módulos internos:</p> <p>Template manager:</p> <ul style="list-style-type: none">• /templates <p>Slice manager:</p> <ul style="list-style-type: none">• /slices

Api-gateway

```
[GIN] 2024/07/01 - 06:59:37 | 200 | 12.08531ms | 10.8.0.133 | GET | "/templateservice/templates"
2024/07/01 07:00:03 Request received for /templateservice/templates/668235472411d21efc2a53a8
[GIN] 2024/07/01 - 07:00:03 | 200 | 8.37313ms | 10.8.0.133 | GET | "/templateservice/templates/668235472411d21efc2a53a8"
```

Template manager

```
127.0.0.1 - - [01/Jul/2024 01:59:37] "GET /templates HTTP/1.1" 200 -
127.0.0.1 - - [01/Jul/2024 02:00:03] "GET /templates/668235472411d21efc2a53a8 HTTP/1.1" 200 -
```

Api-gateway

```
[GIN] 2024/07/01 - 07:03:56 | 200 | 21.325333ms | 10.8.0.133 | GET "/sliceservice/slices"
2024/07/01 07:04:05 Request received for /sliceservice/slices/6682224fa17c42d433848971
[GIN] 2024/07/01 - 07:04:05 | 200 | 8.429353ms | 10.8.0.133 | GET "/sliceservice/slices/6682224fa17c42d433848971"
```

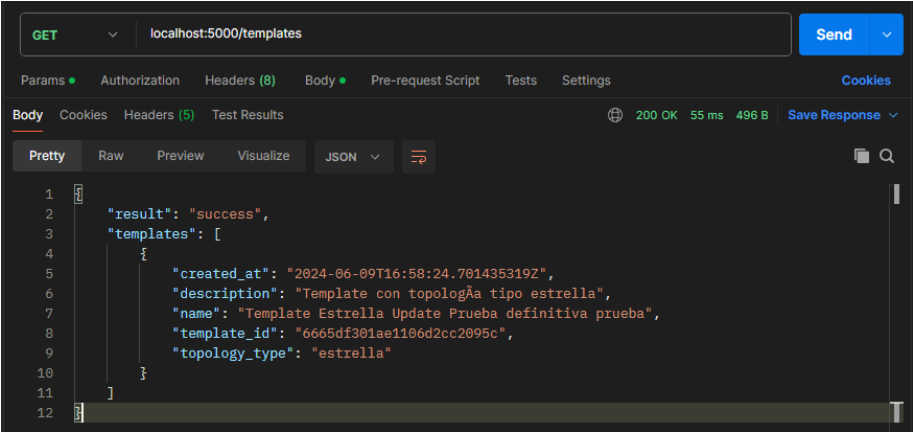
Slice manager

```
127.0.0.1 - - [01/Jul/2024 02:03:56] "GET /slices HTTP/1.1" 200 -
127.0.0.1 - - [01/Jul/2024 02:04:05] "GET /slices/6682224fa17c42d433848971 HTTP/1.1" 200 -
```

3. Módulo gestor de plantillas

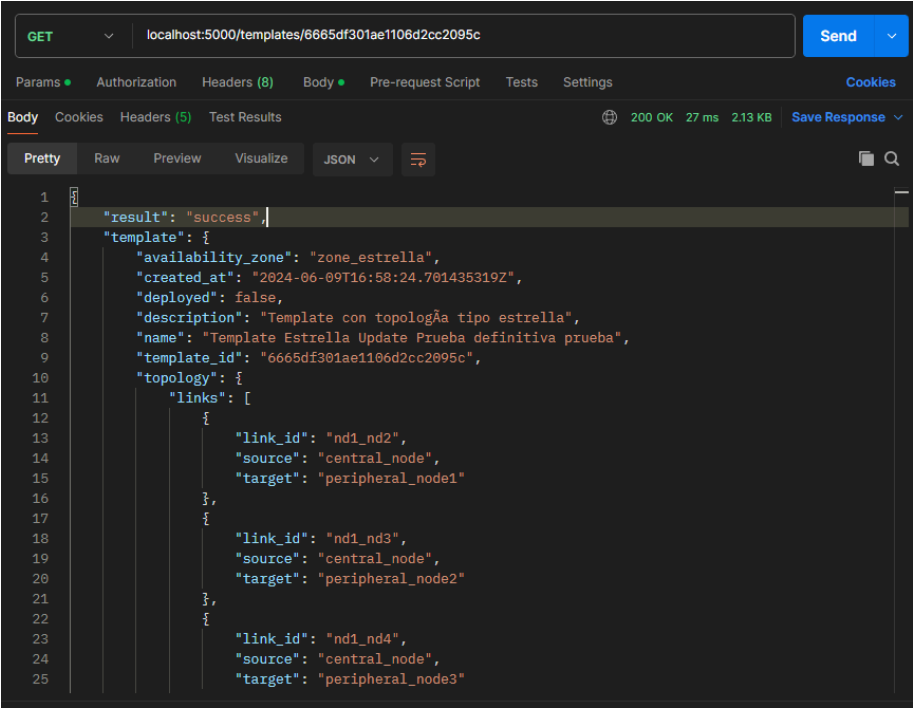
3.1 Listado de plantillas

Objetivo del test	Prueba de listado de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas.
Diagrama de red del Test	N/A
Prerrequisitos	<div>1. Módulo gestor de plantillas implementado y en funcionamiento en Headnode.</div> <div>2. Módulo de api gateway implementado y en funcionamiento en el HeadNode</div> <div>3. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.</div> <div>4. Estar conectado a la red PUCP.</div> <div>5. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente.</div> <div>6. El usuario debe tener plantillas creadas.</div>
Procedimiento	<div>1. Ejecutar un request GET hacia el endpoint de listado que expone el módulo al ejecutarse localmente: localhost:5000/templates.</div> <div>2. En el request, colocar las cabeceras: ["X-User-Role"] y ["X-User-ID"] del archivo .yaml del usuario en sesión. Esto le permitirá pasar las validaciones del módulo de plantillas.</div>
Resultados esperados	Se debe recibir un json con el resultado exitoso de la solicitud y un arreglo de plantillas con 5 campos cada una: created_at, description, name, template_id, topology_type.
Resultado del test	PASS

Evidencias	
Comentarios y/o consideraciones	<p>RAZÓN DE CABECERAS EN PROCEDIMIENTO:</p> <ol style="list-style-type: none">1. Cuando se tiene la integración del sistema, el API GATEWAY (en nuestro unit testing fue manual) envía el id y el rol del usuario que realiza la solicitud en la cabecera de la misma. Esto con el objetivo de validar solo el listado de plantillas que le pertenezcan al usuario logueado.

3.2 Detalle de configuración de plantilla

Objetivo del test	Prueba de listado de detalle de configuración de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Módulo gestor de plantillas implementado y en funcionamiento en Headnode.2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.3. Estar conectado a la red PUCP.4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente.5. El usuario debe tener plantillas creadas.
Procedimiento	<ol style="list-style-type: none">1. Ejecutar un request GET hacia el endpoint de listado por ID que expone el módulo al ejecutarse localmente: localhost:5000/templates/{template_id}.2. Colocar en el URL un template_id válido.
Resultados esperados	Se debe recibir un json con el resultado exitoso de la solicitud y la plantilla (correspondiente al template_id enviado) con 7 campos cada una: created_at, description, name, template_id, topology (links, nodes), topology_type, user_id.

Resultado del test	PASS
Evidencias	
Comentarios y/o consideraciones	N/A

3.3 Creación de una nueva plantilla

Objetivo del test	Prueba de creación de plantilla por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Módulo gestor de plantillas implementado y en funcionamiento en Headnode.2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.3. Estar conectado a la red PUCP.4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente.
Procedimiento	<p>Ejecutar un request POST hacia el endpoint de creación que expone el módulo al ejecutarse localmente: localhost:5000/templates. En esta solicitud, enviar la plantilla completa en formato json en el body. El json mostrado es solo un ejemplo general:</p> <pre>{</pre>

```

"created_at": "datetime_UTC_format",
"description": "template_description",
"name": "template_name",
"topology": {
  "links": [
    {
      "link_id": "nd1_nd2",
      "source": "nd1",
      "target": "nd2",
      "source_port": "nd1_port0",
      "target_port": "nd2_port0"
    },
    {
      "link_id": "nd2_nd3",
      "source": "nd2",
      "target": "nd3",
      "source_port": "nd2_port1",
      "target_port": "nd3_port0"
    },
    {
      "link_id": "nd3_nd4",
      "source": "nd3",
      "target": "nd4",
      "source_port": "nd3_port1",
      "target_port": "nd4_port0"
    },
    {
      "link_id": "nd4_nd1",
      "source": "nd4",
      "target": "nd1",
      "source_port": "nd4_port1",
      "target_port": "nd1_port1"
    }
  ],
  "nodes": [
    {
      "node_id": "nd1",
      "name": "node_1",
      "image": "image_id",
      "flavor": {
        "id": "flavor_id",
        "name": "flavor_name",
        "cpu": number_vCPU,
        "memory": number_RAM_MB,
        "storage": number_disk_GB
      },
      "security_rules": [
        22
      ],
      "ports": [
        {
          "node_id": "nd1_port0"
        },
      ],

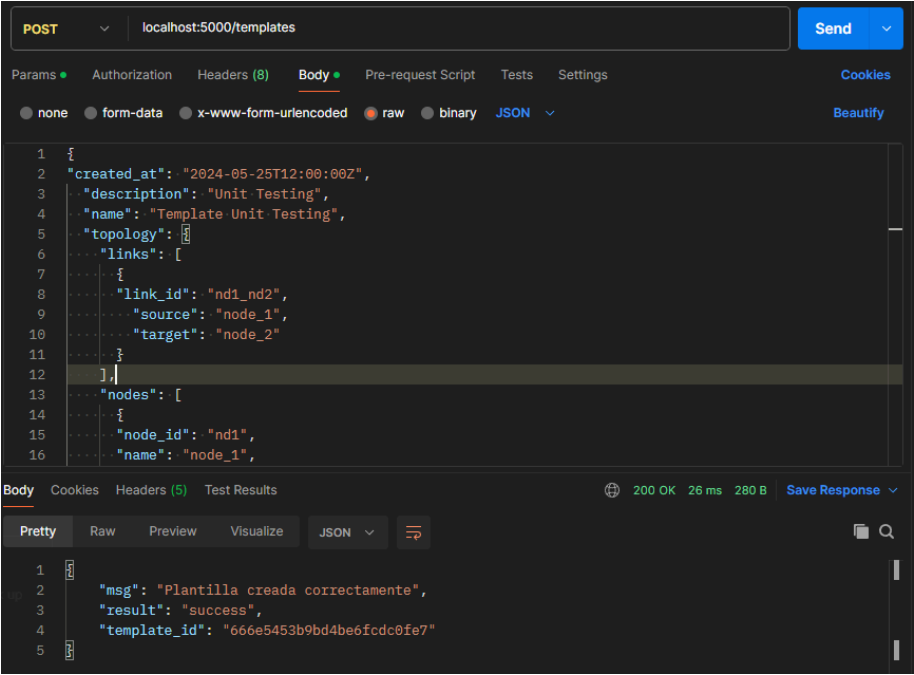
```

```

        {
            "node_id": "nd1_port1"
        }
    ],
    },
    {
        "node_id": "nd2",
        "name": "node_2",
        "image": "image_id",
        "flavor": {
            "id": "flavor_id",
            "name": "flavor_name",
            "cpu": number_vCPU,
            "memory": number_RAM_MB,
            "storage": number_disk_GB
        },
        "security_rules": [
            22
        ],
        "ports": [
            {
                "node_id": "nd2_port0"
            },
            {
                "node_id": "nd2_port1"
            }
        ]
    },
    {
        "node_id": "nd3",
        "name": "node_3",
        "image": "image_id",
        "flavor": {
            "id": "flavor_id",
            "name": "flavor_name",
            "cpu": number_vCPU,
            "memory": number_RAM_MB,
            "storage": number_disk_GB
        },
        "security_rules": [
            22
        ],
        "ports": [
            {
                "node_id": "nd3_port0"
            },
            {
                "node_id": "nd3_port1"
            }
        ]
    },
    {
        "node_id": "nd4",

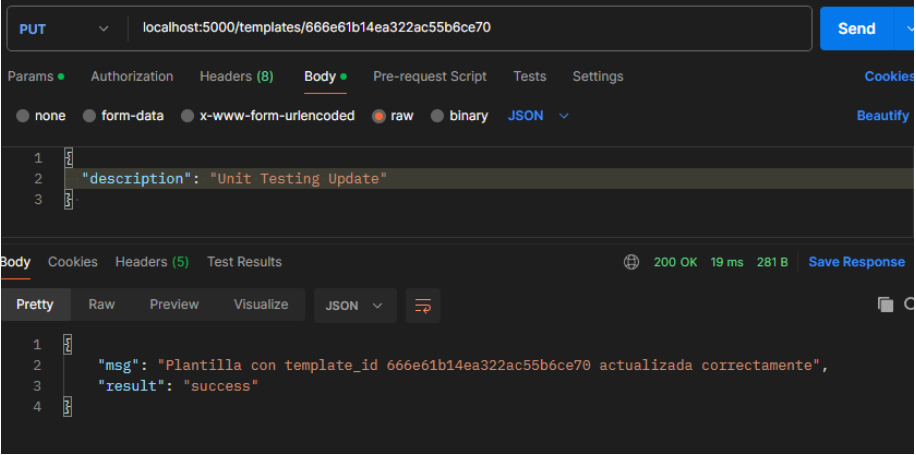
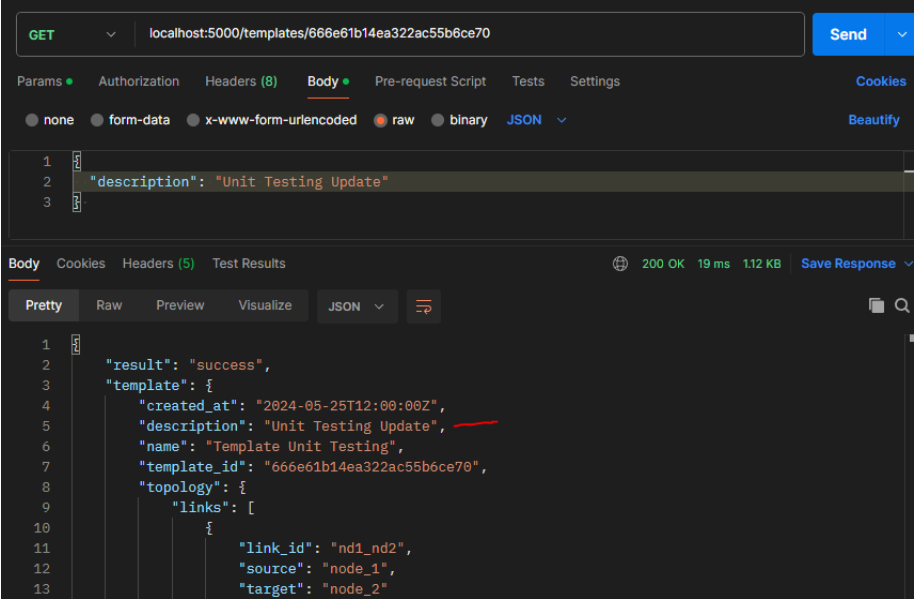
```

	<pre> "name": "node_4", "image": "image_id", "flavor": { "id": "flavor_id", "name": "flavor_name", "cpu": number_vCPU, "memory": number_RAM_MB, "storage": number_disk_GB }, "security_rules": [22], "ports": [{ "node_id": "nd4_port0" }, { "node_id": "nd4_port1" }] }], }, "user_id": "user_id", "topology_type": "topology_type" } </pre>
Resultados esperados	Se recibe un json confirmando la creación de la plantilla (resultado exitoso) junto con el nuevo template_id creado.
Resultado del test	PASS

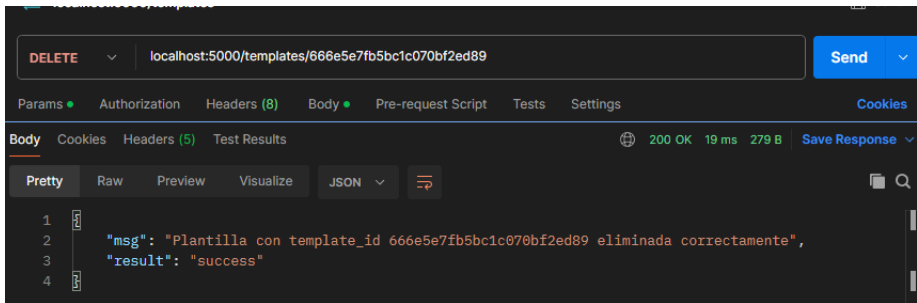
Evidencias	 <p>The screenshot shows a REST client interface with a POST request to <code>localhost:5000/templates</code>. The request body is a JSON object with the following structure:</p> <pre> 1 { 2 "created_at": "2024-05-25T12:00:00Z", 3 "description": "Unit Testing", 4 "name": "Template Unit Testing", 5 "topology": { 6 "links": [7 { 8 "link_id": "nd1_nd2", 9 "source": "node_1", 10 "target": "node_2" 11 } 12], 13 "nodes": [14 { 15 "node_id": "nd1", 16 "name": "node_1", </pre> <p>The response body is a JSON object with the following structure:</p> <pre> 1 { 2 "msg": "Plantilla creada correctamente", 3 "result": "success", 4 "template_id": "666e5453b9bd4be6fcdc0fe7" 5 } </pre>
Comentarios y/o consideraciones	N/A

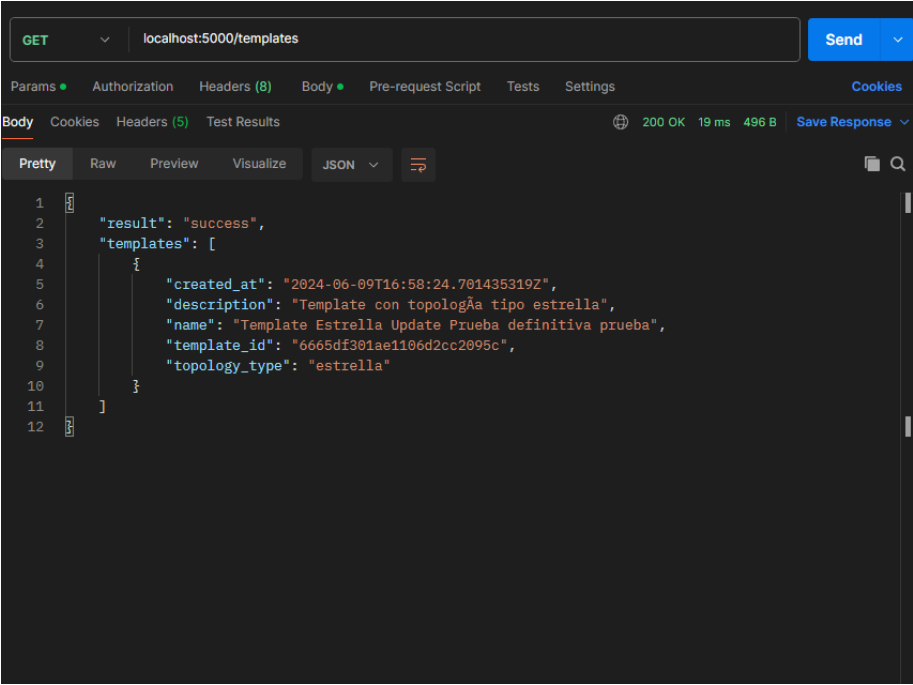
3.4 Edición de una plantilla

Objetivo del test	Prueba de edición de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Módulo gestor de plantillas implementado en Headnode. 2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode. 3. Estar conectado a la red PUCP 4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente. 5. El usuario debe tener plantillas creadas.
Procedimiento	<ol style="list-style-type: none"> 1. Enviar mediante el método PUT una solicitud al endpoint de edición de plantillas que expone el módulo al ejecutarse localmente: <code>localhost:5000/templates/{template_id}</code>. En esta solicitud, enviar solo los campos (rellenados) que se deseen editar en formato json en el body. 2. Verificar el update haciendo una solicitud GET al endpoint de listado de plantilla por id: <code>localhost:5000/templates/{template_id}</code>.

Resultados esperados	<ol style="list-style-type: none">1. Se recibe la confirmación exitosa de edición de la plantilla con determinado template_id.2. De ocurrir un error, se recibe un json con campo result "error".
Resultado del test	PASS
Evidencias	<p>Ejecución de solicitud PUT:</p>  <p>Listado de configuración de plantilla anteriormente editada para comprobar el cambio.</p> 
Comentarios y/o consideraciones	N/A


3.5 Borrado de una plantilla

Objetivo del test	Prueba de borrado de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Módulo gestor de plantillas implementado en Headnode.2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.3. Estar conectado a la red PUCP4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente.5. El usuario debe tener plantillas creadas.
Procedimiento	<ol style="list-style-type: none">1. Enviar mediante el método DELETE una solicitud al endpoint de borrado de plantillas que expone el módulo al ejecutarse localmente: localhost:5000/templates/{template_id}.2. Verificar el borrado en la lista de plantillas al endpoint GET localhost:5000/templates.
Resultados esperados	<ol style="list-style-type: none">1. Se debe recibir un json con el resultado exitoso de la solicitud y la confirmación del borrado de la plantilla con determinado template_id.2. En el listado de plantillas del endpoint GET no debe aparecer la plantilla borrada.
Resultado del test	PASS
Evidencias	

	 <pre> 1 { 2 "result": "success", 3 "templates": [4 { 5 "created_at": "2024-06-09T16:58:24.701435319Z", 6 "description": "Template con topologAa tipo estrella", 7 "name": "Template Estrella Update Prueba definitiva prueba", 8 "template_id": "6665df301ae1106d2cc2095c", 9 "topology_type": "estrella" 10 } 11] 12 } </pre>
Comentarios y/o consideraciones	N/A

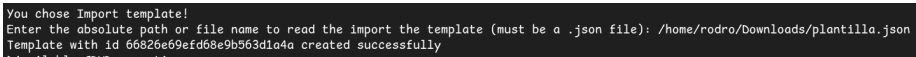
3.6 Exportar plantilla

Objetivo del test	Prueba de exportación de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas (endpoint de listado por id) y el CLI.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Módulo gestor de plantillas implementado en Headnode. 2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode. 3. Estar conectado a la red PUCP 4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente en el CLI del orquestador. 5. El usuario debe tener plantillas creadas.
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar al comando <i>templates</i> del CLI. 2. Escoger la opción <i>List Templates</i> del menú de opciones. 3. Seleccionar mediante la tecla "enter" una plantilla de la tabla mostrada. 4. Escoger la opción <i>Export Template</i> del segundo menú de

	<p>opciones.</p> <ol style="list-style-type: none"> 5. Ingresar un path absoluto (ubicación de guardado del archivo a crear) o el nombre del archivo .json. 6. Ingresar al path indicado para verificar la creación del archivo .json
Resultados esperados	El CLI confirma la exportación de la plantilla seleccionada e imprime la ruta de guardado del archivo json. El archivo con el nombre ingresado se debe encontrar en dicha ruta.
Resultado del test	PASS
Evidencias	 <pre> You chose Export template! Enter the absolute path or file name to export the template (must be a .json file): /home/rodra/Downloads/plantilla.json Template exported successfully to /home/rodra/Downloads/plantilla.json </pre>

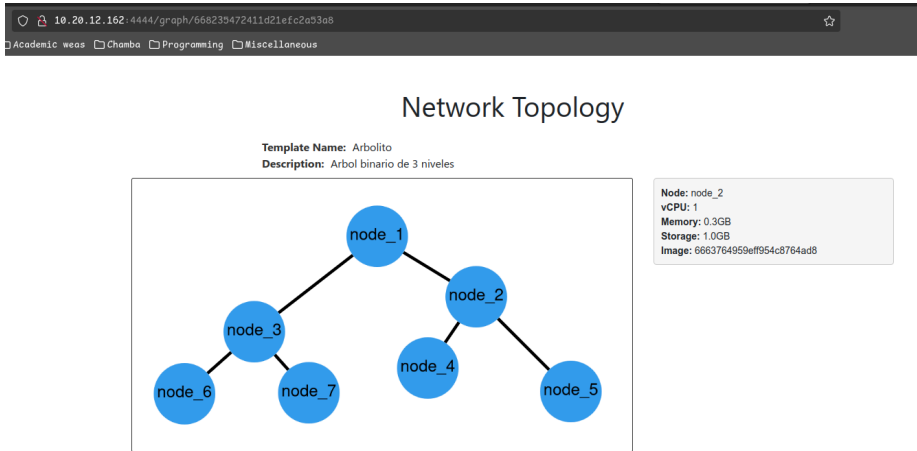
	<div data-bbox="486 215 1401 1520"> <pre> File: /home/rodro/Downloads/plantilla.json 1 { 2 "created_at": "2024-07-01T07:46:47.089822208Z", 3 "description": "Topologia lineal", 4 "name": "Lineal", 5 "topology": { 6 "links": [7 { 8 "link_id": "nd1_nd2", 9 "source": "nd1", 10 "target": "nd2" 11 }, 12 { 13 "link_id": "nd2_nd3", 14 "source": "nd2", 15 "target": "nd3" 16 } 17], 18 "nodes": [19 { 20 "node_id": "nd1", 21 "name": "node_1", 22 "image": "6663764959eff954c8764adc", 23 "security_rules": [24 22 25], 26 "flavor": { 27 "id": "665275b98c45f0c2b8a2e230", 28 "name": "256MBRAM_1VCPU_16BDisk", 29 "cpu": 1, 30 "memory": 0.3, 31 "storage": 1 32 } 33 }, 34 { 35 "node_id": "nd2", 36 "name": "node_2", </pre> </div>
Comentarios y/o consideraciones	<p>Al ingresar el path absoluto o el nombre del archivo, debe especificarse/digitarse explícitamente la extensión del tipo de archivo (.json)</p> <p>Esta funcionalidad utiliza el mismo endpoint de listado de plantilla por id (labor del módulo gestor de plantilla) para obtener la configuración de la plantilla. El CLI se encarga de la creación del archivo utilizando esta respuesta. Por lo tanto, en este caso en particular, se considera unit testing de “funcionalidad” puesto que el funcionamiento completo involucra dos módulos.</p>

3.7 Importar plantilla

Objetivo del test	Prueba de importación de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas y el CLI.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Módulo gestor de plantillas implementado en Headnode.2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode.3. Estar conectado a la red PUCP4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente en el CLI del orquestador.
Procedimiento	<ol style="list-style-type: none">1. Ingresar al comando <i>templates</i> del CLI.2. Escoger la opción <i>Import Template</i> del menú de opciones.3. Ingresar un path absoluto (ubicación de guardado del archivo a importar) o el nombre del archivo .json.4. Listar las plantillas del usuario para verificar la importación.
Resultados esperados	El CLI confirma la importación de la plantilla del archivo .json seleccionado e imprime el ID de la plantilla creada. En la lista de plantillas del usuario, aparece la plantilla importada.
Resultado del test	PASS
Evidencias	 <pre>You chose Import template! Enter the absolute path or file name to read the import the template (must be a .json file): /home/rodra/Downloads/plantilla.json Template with id 66826e69efd68e9b563d1a4a created successfully</pre>
Comentarios y/o consideraciones	<p>Al ingresar el path absoluto o el nombre del archivo, debe especificarse/digitarse explícitamente la extensión del tipo de archivo (.json)</p> <p>Esta funcionalidad utiliza el mismo endpoint de creación de plantilla (labor del módulo gestor de plantilla) para crear la plantilla contenida en el archivo. El CLI se encarga de leer y validar el contenido del archivo. Por lo tanto, en este caso en particular, se considera unit testing de “funcionalidad” puesto que el funcionamiento completo involucra dos módulos.</p>

3.8 Graficar plantilla

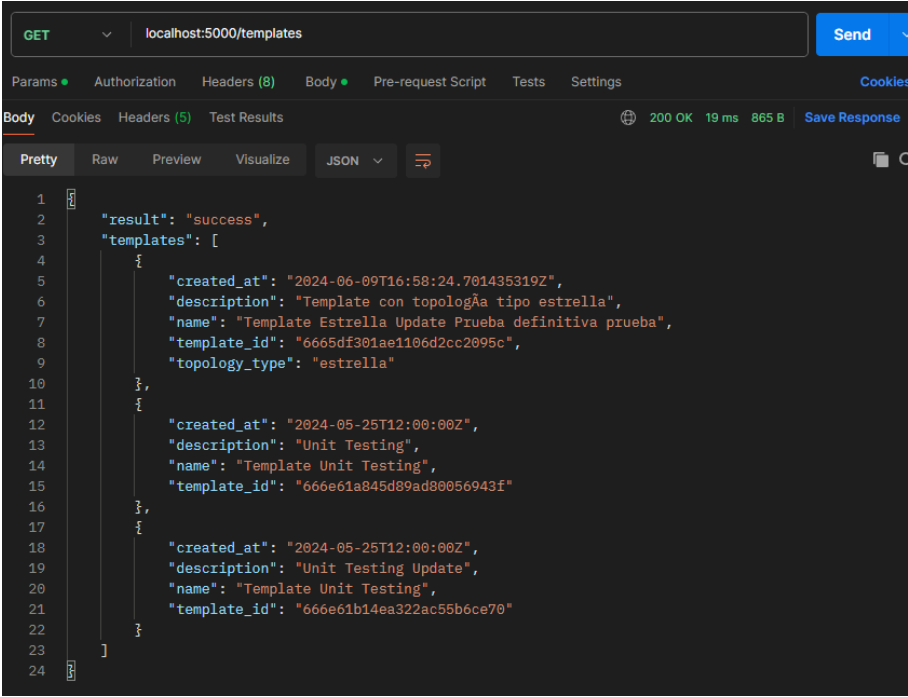
Objetivo del test	Prueba de graficado de plantillas por parte de un usuario administrador o cliente utilizando el módulo de gestor de plantillas y el CLI.
-------------------	------------------------------------------------------------------------------------------------------------------------------------------

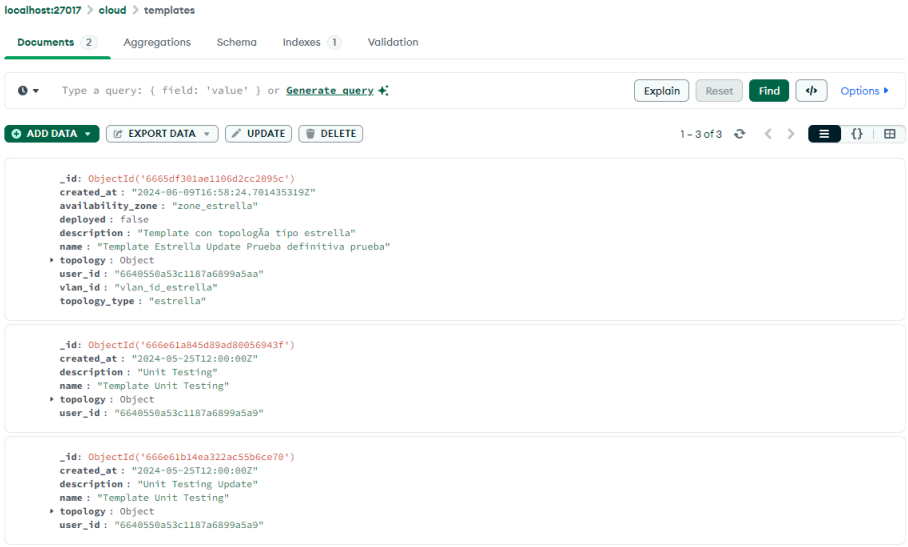
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Módulo gestor de plantillas implementado en Headnode. 2. Módulo de persistencia (MongoDB) implementado y en funcionamiento en Headnode. 3. Estar conectado a la red PUCP 4. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente en el CLI del orquestador.
Procedimiento	<ol style="list-style-type: none"> 1. En el browser, visitar el siguiente link http://10.20.12.162:4444/graph/668235472411d21efc2a53a8.
Resultados esperados	Se despliega el gráfico de la plantilla deseada en un browser.
Resultado del test	PASS
Evidencias	
Comentarios y/o consideraciones	El id de la plantilla se envía como path parameter en la url para graficar.

4. Módulo de persistencia (MongoDB)

4.1 Verificación de acciones CRUD

Objetivo del test	Verificar la realización de las acciones CRUD sobre la base de datos MongoDB.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Módulo de persistencia (MongoDB) implementado y en

	<p>funcionamiento en Headnode.</p> <ol style="list-style-type: none">2. Estar conectado a la red PUCP.3. Tener sesión activa (autenticada y autorizada) para un usuario administrador/cliente en el CLI del orquestador.
Procedimiento	<ol style="list-style-type: none">1. Ejecutar una solicitud GET hacia el endpoint de listado que expone el módulo al ejecutarse localmente: localhost:5000/templates.2. Verificar que el json de plantillas recibido en postman coincida con los documentos en base de datos MongoDB en Headnode. Se requiere acceder a la base de datos.
Resultados esperados	El response de la solicitud coincide con la información recogida/actualizada en la base de datos MongoDB en HeadNode.
Resultado del test	PASS
Evidencias	 <p>The screenshot shows a Postman interface with a GET request to localhost:5000/templates. The response is a JSON array of three template objects. The status is 200 OK, 19 ms, 865 B. The JSON response is as follows:</p> <pre>1 { 2 "result": "success", 3 "templates": [4 { 5 "created_at": "2024-06-09T16:58:24.701435319Z", 6 "description": "Template con topologÃa tipo estrella", 7 "name": "Template Estrella Update Prueba definitiva prueba", 8 "template_id": "6665df301ae1106d2cc2095c", 9 "topology_type": "estrella" 10 }, 11 { 12 "created_at": "2024-05-25T12:00:00Z", 13 "description": "Unit Testing", 14 "name": "Template Unit Testing", 15 "template_id": "666e61a845d89ad80056943f" 16 }, 17 { 18 "created_at": "2024-05-25T12:00:00Z", 19 "description": "Unit Testing Update", 20 "name": "Template Unit Testing", 21 "template_id": "666e61b14ea322ac55b6ce70" 22 } 23] 24 }</pre>

	
Comentarios y/o consideraciones	N/A

5. Módulo de sistema de colas

5.1 Ejecución correcta del sistema de colas

Objetivo del test	Verificar la funcionalidad del sistema de colas para la gestión de mensajes entre los módulos del sistema orquestador de slices de VMs en Linux Cluster.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Configuración correcta del servidor RabbitMQ.2. Conexión establecida entre los módulos del sistema (slice manager y linux driver) y el servidor RabbitMQ.
Procedimiento	<ol style="list-style-type: none">1. Ejecución del worker celery (linux driver) con el siguiente comando celery -A slice-manager.celery worker --loglevel=info2. Verificar que el servidor rabbitmq es reconocido por el worker
Resultados esperados	<ol style="list-style-type: none">1. Los mensajes enviados a la cola son correctamente procesados por los consumidores.2. No se producen errores de conexión ni pérdida de mensajes.

Resultado del test	PASS
Evidencias	<pre> ----- celery@HeadNode v5.4.0 (opalescent) --- ***** --- -- ***** --- Linux-5.4.0-187-generic-x86_64-with-glibc2.29 2024-07-01 01:57:29 - *** --- * --- - ** ----- [config] - ** ----- .> app: slice-manager:0x7f0b1016d460 - ** ----- .> transport: amqp://guest:**@localhost:5673// - ** ----- .> results: mongodb://localhost:27017/cloud - *** --- * --- .> concurrency: 4 (prefork) -- ***** --- .> task events: OFF (enable -E to monitor tasks in this worker) --- ***** --- ----- [queues] . > celery exchange=celery(direct) key=celery [2024-07-01 01:57:30,076: INFO/MainProcess] Connected to amqp://guest:**@127.0.0.1:5673// </pre>
Comentarios y/o consideraciones	

6. Módulo de scheduler (VM placement)

6.1 Asignación de VMs según capacidad de los servidores

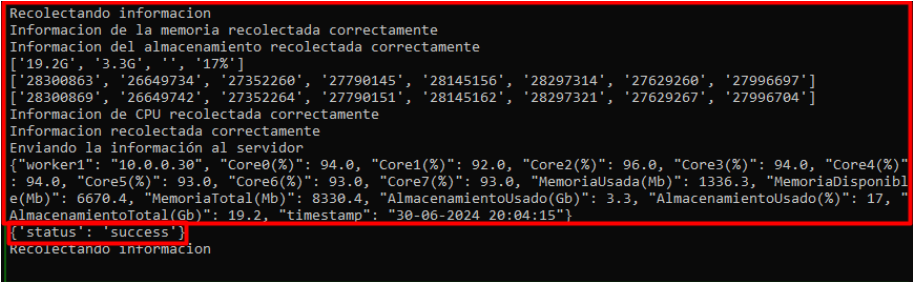
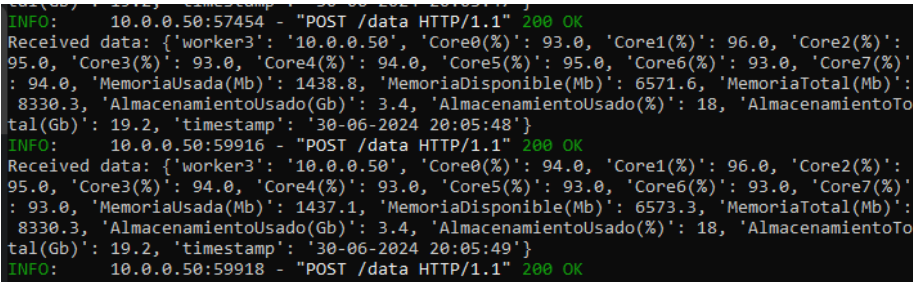
Objetivo del test	Validar la capacidad del módulo de scheduler para realizar scheduling (asignación) eficiente de VMs en los Workers (servidores).
Diagrama de red del Test	N/A
Prerrequisitos	1. Disponibilidad de recursos suficientes en los Workers para desplegar las máquinas virtuales.
Procedimiento	<ol style="list-style-type: none"> 1. Visualizar los recursos actuales (cores CPU, RAM_GB disponible, DISK_GB disponible) de los tres workers mediante los siguientes comandos: 2. Crear un conjunto de solicitudes de despliegue de VMs con diferentes características de vCPU, RAM_GB y DISK_GB. 3. Ejecutar el módulo scheduler. Genera una matriz output de solución de asignaciones (matriz de 1s y 0s. filas: VMs. columnas: workers). 4. Ejecutar los comandos (primer punto) de consumo de recursos en cada worker para verificar que las VMs se asignaron al/los workers con más capacidad de recursos. 5. Comparar el consumo de recursos con el output decisivo del

	scheduler.
Resultados esperados	Las VM son asignadas a los Workers considerando la capacidad y disponibilidad de recursos.
Resultado del test	IN PROGRESS (SÓLO TEÓRICO) IMPLEMENTADO ROUND ROBIN
Evidencias	Registro de las asignaciones de VMs en los Workers Métricas de utilización de recursos de los Workers antes y después de la ejecución del scheduler.
Comentarios y/o consideraciones	N/A

7. Módulo de recursos disponibles

7.1 Recursos extraídos de cada worker

Objetivo del test	Envío de recursos disponibles en los workers hacia el head node para su posterior guardado en base de datos.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Programa en python para la extracción de recursos en los tres trabajadores presentes en la topología. 2. Programa en python para la recepción de recursos en el head node (escuchando en el puerto 9898) y posterior guardado en base de datos de mongo. 3. Servicio de la base de datos de MongoDB corriendo en el puerto 27017 del head node.
Procedimiento	<ol style="list-style-type: none"> 1. Se ejecuta el programa de resources.py en cada worker donde se ejecutan comandos para recolectar información de los recursos y posteriormente enviarla al headnode. 2. Para evaluar el consumo de CPU se revisará el fichero /proc/stat donde se tienen los jiffies correspondientes al tiempo de cada estado en los cores presentes en el worker. 3. Se toman dos capturas tomando un intervalo de un segundo donde se guardan los jiffies correspondientes al tiempo en idle de cada core. 4. Se evalúa el delta de tiempos para hallar el tiempo de uso y se divide sobre el tiempo de medición, de esa manera se obtiene el porcentaje de uso para cada core. 5. Para las medidas de RAM se utiliza el comando free -b y se toman los bytes correspondientes al total, usado y disponible. 6. Para las medidas de Disco se utiliza el comando lsblk -o FSSIZE,FSUSED,FSUSE% donde se toma la primera

	<p>particion sda1 que contiene los archivos raíz de Linux, se recolecta el total, usado y disponible.</p> <ol style="list-style-type: none"> 7. Todos estos recursos son recopilados en un json junto con la etiqueta del worker y timestamp para poder enviar al endpoint expuesto en el puerto 9898 del headnode "http://10.0.10.2:9898/data". 8. El headnode recibe dicha info y responde con un status:success si todo sale de manera correcta. 9. El worker en cuestión recibe dicha respuesta, la imprime y continúa con el loop para seguir scrapeando datos.
Resultados esperados	<p>El correcto envío de los datos desde los workers hacia el head node, para su posterior procesamiento dentro de este.</p> <p>El headnode imprime la data enviada por los workers durante su proceso de escucha.</p> <p>La data es guardada en la base de datos conservando los últimos 1000 registros de acuerdo al timestamp de cada json de recursos recibido.</p>
Resultado del test	PASS
Evidencias	 <pre> Recolectando informacion Informacion de la memoria recolectada correctamente Informacion del almacenamiento recolectada correctamente ['19.2G', '3.3G', '', '17%'] ['28300863', '26649734', '27352260', '27790145', '28145156', '28297314', '27629260', '27996697'] ['28300869', '26649742', '27352264', '27790151', '28145162', '28297321', '27629267', '27996704'] Informacion de CPU recolectada correctamente Informacion recolectada correctamente Enviando la información al servidor {"worker1": "10.0.0.30", "Core0(%)": 94.0, "Core1(%)": 92.0, "Core2(%)": 96.0, "Core3(%)": 94.0, "Core4(%)": 94.0, "Core5(%)": 93.0, "Core6(%)": 93.0, "Core7(%)": 93.0, "MemoriaUsada(Mb)": 1336.3, "MemoriaDisponible(Mb)": 6670.4, "MemoriaTotal(Mb)": 8330.4, "AlmacenamientoUsado(Gb)": 3.3, "AlmacenamientoUsado(%)": 17, "AlmacenamientoTotal(Gb)": 19.2, "timestamp": "30-06-2024 20:04:15"} ('status': 'success') Recolectando informacion </pre>  <pre> INFO: 10.0.0.50:57454 - "POST /data HTTP/1.1" 200 OK Received data: {'worker3': '10.0.0.50', 'Core0(%)': 93.0, 'Core1(%)': 96.0, 'Core2(%)': 95.0, 'Core3(%)': 93.0, 'Core4(%)': 94.0, 'Core5(%)': 95.0, 'Core6(%)': 93.0, 'Core7(%)': 94.0, 'MemoriaUsada(Mb)': 1438.8, 'MemoriaDisponible(Mb)': 6571.6, 'MemoriaTotal(Mb)': 8330.3, 'AlmacenamientoUsado(Gb)': 3.4, 'AlmacenamientoUsado(%)': 18, 'AlmacenamientoTotal(Gb)': 19.2, 'timestamp': '30-06-2024 20:05:48'} INFO: 10.0.0.50:59916 - "POST /data HTTP/1.1" 200 OK Received data: {'worker3': '10.0.0.50', 'Core0(%)': 94.0, 'Core1(%)': 96.0, 'Core2(%)': 95.0, 'Core3(%)': 94.0, 'Core4(%)': 93.0, 'Core5(%)': 93.0, 'Core6(%)': 93.0, 'Core7(%)': 93.0, 'MemoriaUsada(Mb)': 1437.1, 'MemoriaDisponible(Mb)': 6573.3, 'MemoriaTotal(Mb)': 8330.3, 'AlmacenamientoUsado(Gb)': 3.4, 'AlmacenamientoUsado(%)': 18, 'AlmacenamientoTotal(Gb)': 19.2, 'timestamp': '30-06-2024 20:05:49'} INFO: 10.0.0.50:59918 - "POST /data HTTP/1.1" 200 OK </pre>
Comentarios y/o consideraciones	<p>El envío de información desde los workers hacia el head node se da cada 1 segundo.</p> <p>Dichos programas que se encuentran en los wokers y en el headnode se ejecutan en background para poder tener una recolección constante de los recursos a utilizar.</p>

8. Módulo Linux Cluster Driver

8.1 Despliegue de slices utilizando infraestructura de Linux Cluster

Objetivo del test	Verificar la integración y funcionalidad del módulo Linux Cluster Driver en el sistema orquestador de slices de VMs.
Diagrama de red del Test	N/A
Prerrequisitos	1. Conexión establecida entre el slice manager y el Linux driver.
Procedimiento	<ol style="list-style-type: none">1. El usuario decide levantar una plantilla de las que posee.2. Dicha información de la plantilla es enviada al scheduler.3. El algoritmo utilizado por el scheduler (round robin implementado) toma las características y cantidad de nodos para poder decidir en qué worker se desplegará.4. El despliegue se da de manera asíncrona y secuencial en el linux cluster después de haber esperado en la cola.
Resultados esperados	Las VM se despliegan correctamente en el clúster Linux y hay comunicación entre ellas.
Resultado del test	PASS
Evidencias	Colección slices en bd antes de despliegue con deployment_type: "linux"

{deployment_type: "linux"}

+

ADD DATA

EXPORT DATA

UPDATE

DELETE

▶

_id: ObjectId('66822e89a3403456cb45c88d')

created_at: "2024-07-01T04:20:24.992857147Z"

description: "Arbol binario"

name: "Arbol"

▼ topology: Object

▶ links: Array (2)

▼ nodes: Array (3)

▼ 0: Object

node_id: "nd1"

name: "node_1"

image: "6663764959eff954c8764ad8"

flavor: Object

security_rules: Array (1)

ports: Array (2)

process: "291980"

vnc: "6261"

worker: "10.0.0.30"

▶ 1: Object

▶ 2: Object

user_id: "6640550a53c1187a6899a5aa"

topology_type: "arbol binario"

availability_zone: "Zone 1"

deployment_type: "linux"

internet: false

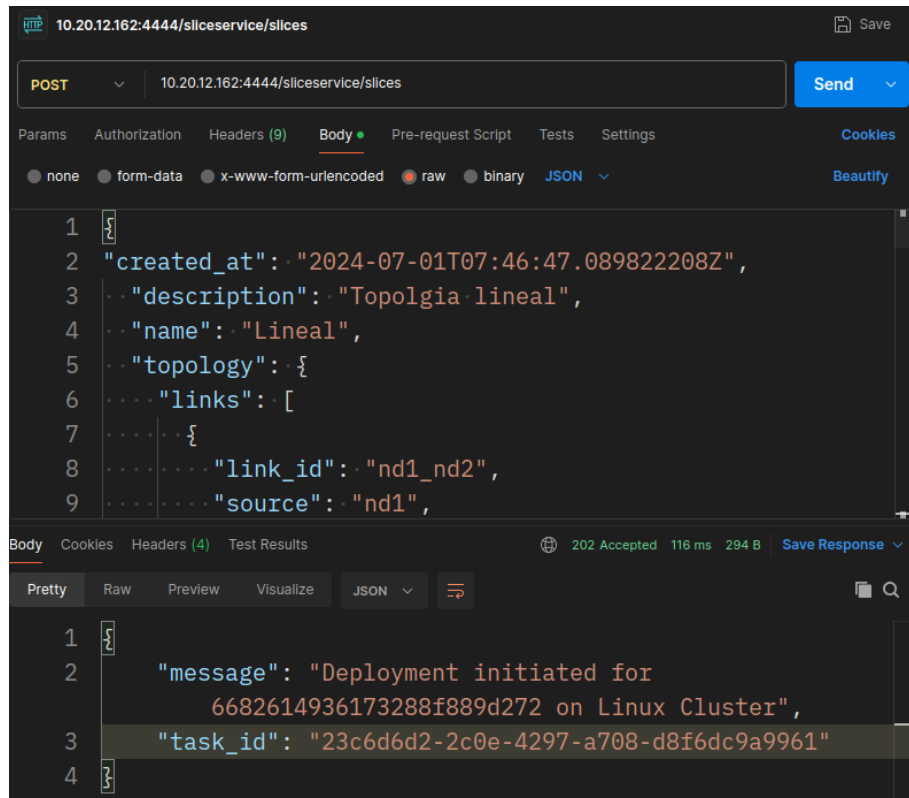
template_id: "66821c2213337f06ea618cda"

vlan_id: "202"

Nodos en el worker 1 antes de desplegar

```
ubuntu@Worker1:~$ ps aux | grep qemu | grep vlan
root      291980  1.0  1.9 659876 156824 ?        S1   Jun30   1:57 qemu-system-x86_64
-enable-kvm -vnc 0.0.0.0:361 -netdev tap,id=nd1vlan202-tap,ifname=nd1vlan202-tap,script=no,downscript=no -device e1000,netdev=nd1vlan202-tap,mac=20:20:20:20:34:89 -daemonize -snapshot cirros-0.6.2-x86_64-disk.img -name nd1
```

Inicializador asíncrono y secuencial del despliegue:



The screenshot displays a REST client interface with a POST request to `10.20.12.162:4444/sliceservice/slices`. The request body is a JSON object. The response, with status `202 Accepted`, is also a JSON object containing a message and a task ID.

```
1 {
2   "created_at": "2024-07-01T07:46:47.089822208Z",
3   "description": "Topolgia lineal",
4   "name": "Lineal",
5   "topology": {
6     "links": [
7       {
8         "link_id": "nd1_nd2",
9         "source": "nd1",
```

```
1 {
2   "message": "Deployment initiated for
               6682614936173288f889d272 on Linux Cluster",
3   "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961"
4 }
```

Revisión del estado del task de despliegue:

The screenshot shows a REST client interface with the following details:

- URL:** 10.20.12.162:4444/sliceservice/tasks/23c6d6d2-2c0e-4297-a708-d8f6dc9a9961
- Method:** GET
- Authorization:** No Auth
- Status:** 200 OK
- Time:** 121 ms
- Size:** 395 B
- Response Body (JSON):**

```
{
  "message": "Task completed successfully.",
  "result": {
    "message": "Slice with slice id
        6682614936173288f889d272 deployed
        successfully on Linux Cluster"
  },
  "status": "SUCCESS",
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961"
}
```

Nodos en el worker 1 después de desplegar

```
ubuntu@Worker1:~$ ps aux | grep qemu | grep vlan
root      291980  0.9  1.9 659876 156824 ?        S1   Jun30   2:06 qemu-system-x86_64 -enable-kvm -vnc 0.0.0.0:361 -netdev tap,id=nd1v1
on202-tap,ifname=nd1v1on202-tap,script=no,downscript=no -device e1000,netdev=nd1v1on202-tap,mac=20:20:20:34:89 -daemonize -snapshot
cirros-0.6.2-x86_64-disk.img -name nd1
root      333281 23.5  1.9 586144 156856 ?        S1   02:57   0:55 qemu-system-x86_64 -enable-kvm -vnc 0.0.0.0:481 -netdev tap,id=nd1v1
on256-tap,ifname=nd1v1on256-tap,script=no,downscript=no -device e1000,netdev=nd1v1on256-tap,mac=20:20:20:79:00 -daemonize -snapshot
cirros-0.6.2-x86_64-disk.img -name nd1
```

Nodo creado en el worker 1 tiene el proceso 333281, id nd1 y puerto vnc 6381. Pertenece al slice creado con la vlan 256.

🕒 {deployment_type:"linux"}

➕ ADD DATA ▾

📄 EXPORT DATA ▾

✎ UPDATE

🗑 DELETE

```
_id: ObjectId('66822e89a3403456cb45c88d')
created_at: "2024-07-01T04:20:24.992857147Z"
description: "Arbol binario"
name: "Arbol"
▶ topology: Object
  user_id: "6640550a53c1187a6899a5aa"
  topology_type: "arbol binario"
  availability_zone: "Zone 1"
  deployment_type: "linux"
  internet: false
  template_id: "66821c2213337f06ea618cda"
  vlan_id: "202"
```

▶

```
_id: ObjectId('6682614936173288f889d272')
created_at: "2024-07-01T07:46:47.089822208Z"
description: "Topologia lineal"
name: "Lineal"
▼ topology: Object
  ▶ links: Array (2)
  ▼ nodes: Array (3)
    ▼ 0: Object
      node_id: "nd1"
      name: "node_1"
      image: "6663764959eff954c8764adc"
      ▶ security_rules: Array (1)
      ▶ flavor: Object
        process: "333281"
        vnc: "6381"
        worker: "10.0.0.30"
      ▶ 1: Object
      ▶ 2: Object
  user_id: "6640550a53c1187a6899a5aa"
  topology_type: "lineal"
  availability_zone: "Zone 1"
  deployment_type: "linux"
  internet: false
  template_id: "66825ee7efd68e9b563d1a49"
  vlan_id: "256"
```

Ejecución del despliegue desde el worker de celery

	<pre> [2024-07-01 02:57:01,077: WARNING/ForkPoolWorker-4] sudo -S bash vm_script.sh nd1 br-vlan 256 6381 [2024-07-01 02:57:01,094: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:01,138: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:01,993: INFO/ForkPoolWorker-4] 10.0.0.40 is assigned nodes: 2 [2024-07-01 02:57:01,993: WARNING/ForkPoolWorker-4] sudo -S bash vm_script.sh nd2 br-vlan 256 6308 [2024-07-01 02:57:02,009: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:02,050: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:02,896: INFO/ForkPoolWorker-4] 10.0.0.50 is assigned nodes: 3 [2024-07-01 02:57:02,896: WARNING/ForkPoolWorker-4] sudo -S bash vm_script.sh nd3 br-vlan 256 6330 [2024-07-01 02:57:02,914: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:02,956: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:03,798: WARNING/ForkPoolWorker-4] sudo -S bash obtain_worker.sh 256 [2024-07-01 02:57:03,814: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:03,854: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:04,595: WARNING/ForkPoolWorker-4] ['ubuntu', '[sudo] password for ubuntu: ', 'nd1 333281 6381 10.0.0.30'] [2024-07-01 02:57:04,595: INFO/ForkPoolWorker-4] Node nd1 is assigned to 10.0.0.30. Process is 333281 and vnc port is 6381 [2024-07-01 02:57:04,595: WARNING/ForkPoolWorker-4] sudo -S bash obtain_worker.sh 256 [2024-07-01 02:57:04,612: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:04,650: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:05,397: WARNING/ForkPoolWorker-4] ['ubuntu', '[sudo] password for ubuntu: ', 'nd2 333883 6308 10.0.0.40'] [2024-07-01 02:57:05,398: INFO/ForkPoolWorker-4] Node nd2 is assigned to 10.0.0.40. Process is 333883 and vnc port is 6308 [2024-07-01 02:57:05,398: WARNING/ForkPoolWorker-4] sudo -S bash obtain_worker.sh 256 [2024-07-01 02:57:05,415: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH_8.2p1) [2024-07-01 02:57:05,456: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 02:57:06,200: WARNING/ForkPoolWorker-4] ['ubuntu', '[sudo] password for ubuntu: ', 'nd3 342111 6330 10.0.0.50'] [2024-07-01 02:57:06,200: INFO/ForkPoolWorker-4] Node nd3 is assigned to 10.0.0.50. Process is 342111 and vnc port is 6330 </pre>
Comentari os y/o considerac iones	<p>En mongodb se guarda información relevante de cada nodo como el worker en el que se desplegó, el puerto vnc y el número del proceso en ejecución. A nivel de slice, en bd se guarda el número de vlan.</p> <p>Se conecta a las VMs utilizando el puerto VNC expuesto.</p> <p>El usuario puede comprobar las comunicaciones entre VMs de un mismo slice.</p>

8.2 Comunicación de nodos en un mismo slice utilizando infraestructura de Linux Cluster

Objetivo del test	Verificar la comunicación de diferentes nodos dentro de un mismo slice en Linux CLuster
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Conexión establecida entre el sistema orquestador (solución) y el clúster Linux. 2. Datos de recursos disponibles en el momento de ejecución del scheduler.
Procedimiento	<ol style="list-style-type: none"> 1. El usuario se conecta a los nodos por vnc 2. Ejecuta ping entre los nodos de un mismo slice para verificar la comunicación
Resultados esperados	Los nodos dentro de un mismo slice se pueden comunicar
Resultado del test	PASS
Evidencias	

```

▼ 0: Object
  node_id: "nd1"
  name: "node_1"
  image: "6663764959eff954c8764adc"
  ▶ security_rules: Array (1)
  ▶ flavor: Object
    process: "333281"
    vnc: "6381"
    worker: "10.0.0.30"
▼ 1: Object
  node_id: "nd2"
  name: "node_2"
  image: "6663764959eff954c8764adc"
  ▶ security_rules: Array (1)
  ▶ flavor: Object
    process: "333883"
    vnc: "6308"
    worker: "10.0.0.40"
▼ 2: Object
  node_id: "nd3"
  name: "node_3"
  image: "6663764959eff954c8764adc"
  ▶ security_rules: Array (1)
  ▶ flavor: Object
    process: "342111"
    vnc: "6330"
    worker: "10.0.0.50"

```

SSH tunneling para poder entrar a los nodos

```

X rodرو@Archbox ~ ssh -NL 5901:10.0.0.30:6381 ubuntu@10.20.12.162 -p 5800
ubuntu@10.20.12.162's password:

```

```

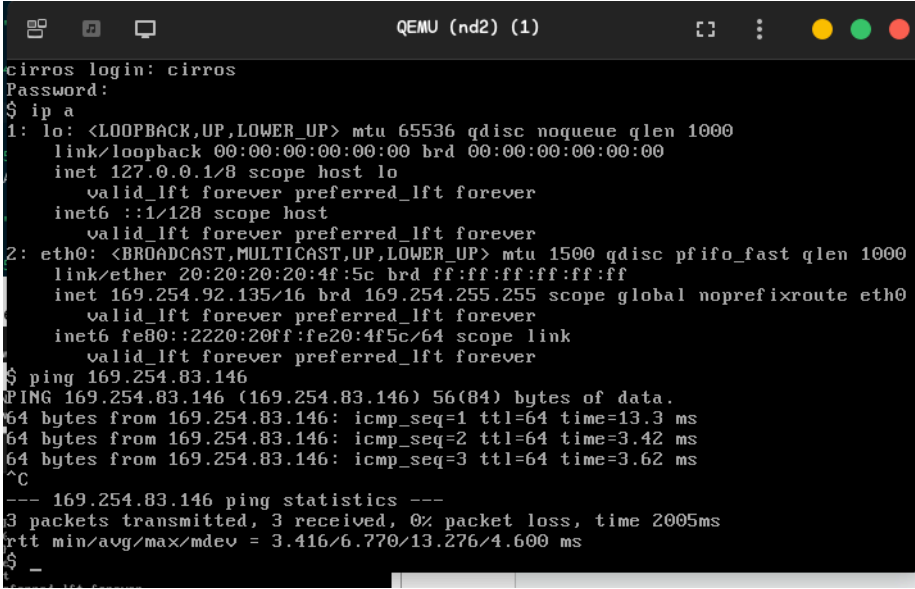
rodرو@Archbox ~ ssh -NL 5902:10.0.0.40:6308 ubuntu@10.20.12.162 -p 5800
ubuntu@10.20.12.162's password:

```

```

QEMU (nd1) (1)
cirros login: cirros
password:
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 20:20:20:20:79:00 brd ff:ff:ff:ff:ff:ff
    inet 169.254.83.146/16 brd 169.254.255.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::2220:20ff:fe20:7900/64 scope link
        valid_lft forever preferred_lft forever
$ ping 169.254.92.135
PING 169.254.92.135 (169.254.92.135) 56(84) bytes of data.
64 bytes from 169.254.92.135: icmp_seq=1 ttl=64 time=6.54 ms
64 bytes from 169.254.92.135: icmp_seq=2 ttl=64 time=3.50 ms
64 bytes from 169.254.92.135: icmp_seq=3 ttl=64 time=3.25 ms
^C
--- 169.254.92.135 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.251/4.428/6.537/1.494 ms
$ s_

```


	 <pre> QEMU (nd2) (1) cirros login: cirros Password: \$ ip a 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000 link/ether 20:20:20:20:4f:5c brd ff:ff:ff:ff:ff:ff inet 169.254.92.135/16 brd 169.254.255.255 scope global noprefixroute eth0 valid_lft forever preferred_lft forever inet6 fe80::2220:20ff:fe20:4f5c/64 scope link valid_lft forever preferred_lft forever \$ ping 169.254.83.146 PING 169.254.83.146 (169.254.83.146) 56(84) bytes of data. 64 bytes from 169.254.83.146: icmp_seq=1 ttl=64 time=13.3 ms 64 bytes from 169.254.83.146: icmp_seq=2 ttl=64 time=3.42 ms 64 bytes from 169.254.83.146: icmp_seq=3 ttl=64 time=3.62 ms ^C --- 169.254.83.146 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2005ms rtt min/avg/max/mdev = 3.416/6.770/13.276/4.600 ms \$ - </pre>
Comentarios y/o consideraciones	N/A

8.3 Borrado de slices utilizando infraestructura de Linux Cluster

Objetivo del test	Verificar la integración y funcionalidad del módulo Linux Cluster Driver en el sistema orquestador de slices de VMs.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> Conexión establecida entre el sistema orquestador (solución) y el clúster Linux. Datos de recursos disponibles en el momento de ejecución del scheduler.
Procedimiento	<ol style="list-style-type: none"> El usuario decide borrar un slice que tiene desplegado Dicha información del slice es enviada al módulo. El borrado se da en el linux cluster después de haber esperado en la cola.
Resultados esperados	Las VM desplegadas se borran correctamente en el clúster Linux. En base de datos se borra el documento del slice.
Resultado del test	PASS
Evidencias	Inicializador asíncrono y secuencial del borrado:

10.20.12.162:4444/sliceservice/slices/6682614936173288f889d272

DELETE

10.20.12.162:4444/sliceservice/slices/6682614936173288f889d272

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookies

Headers6 hidden

	Key	Value
<input checked="" type="checkbox"/>	X-API-Key	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJlE3MTk
	Key	Value

BodyCookiesHeaders (4)Test Results202 Accepted136 ms299 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Delete workflow initiated for
3     6682614936173288f889d272 on Linux Cluster",
4   "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740"
}
```

Revisión del estado del task de borrado:

10.20.12.162:4444/sliceservice/tasks/51cc7908-329e-4ebb-91c0-51bf4b663740

GET

10.20.12.162:4444/sliceservice/tasks/51cc7908-329e-4ebb-91c0-51bf4b663740

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookies

Headers6 hidden

	Key	Value
<input checked="" type="checkbox"/>	X-API-Key	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJlE3MTk
	Key	Value

BodyCookiesHeaders (4)Test Results200 OK130 ms394 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Task completed successfully.",
3   "result": {
4     "message": "Slice with slice id
5       6682614936173288f889d272 deleted
6       successfully on Linux Cluster"
7   },
8   "status": "SUCCESS",
9   "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740"
}
```

	<p>Nodo 1 en workwer 10.0.0.30 con proceso 333281 borrado</p> <pre>ubuntu@Worker1:~\$ ps aux grep 333281 ubuntu 333948 0.0 0.0 8160 724 pts/4 S+ 03:30 0:00 grep --color=auto 333281 ubuntu@Worker1:~\$</pre> <p>Ejecución del borrado desde el worker de celery</p> <pre>[2024-07-01 03:23:07,904: INFO/ForkPoolWorker-4] Starting to delete VM slice 6682614936173288f889d272 on Linux Cluster [2024-07-01 03:23:07,904: WARNING/ForkPoolWorker-4] bash implement_orchestrator/delete_headnode.sh 256 br-vlan [2024-07-01 03:23:08,040: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH.8.2p1) [2024-07-01 03:23:08,109: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 03:23:08,957: INFO/ForkPoolWorker-4] Node nd1 assigned to 10.0.0.30 with process 333281 has been deleted [2024-07-01 03:23:08,981: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH.8.2p1) [2024-07-01 03:23:09,021: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 03:23:09,854: INFO/ForkPoolWorker-4] Node nd2 assigned to 10.0.0.40 with process 333883 has been deleted [2024-07-01 03:23:09,880: INFO/ForkPoolWorker-4] Connected (version 2.0, client OpenSSH.8.2p1) [2024-07-01 03:23:09,924: INFO/ForkPoolWorker-4] Authentication (password) successful! [2024-07-01 03:23:10,774: INFO/ForkPoolWorker-4] Node nd3 assigned to 10.0.0.50 with process 342111 has been deleted [2024-07-01 03:23:10,778: INFO/ForkPoolWorker-4] Slice with slice id 6682614936173288f889d272 deleted successfully on [2024-07-01 03:23:10,778: WARNING/ForkPoolWorker-4] Slice borrdo exitosamente.</pre>
Comentarios y/o consideraciones	N/A

8.4 Generación de logs

Objetivo del test	Prueba de generación de logs de troubleshooting.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Estar conectado a la red PUCP. 2. Módulo de api gateway implementado y en funcionamiento en el HeadNode 3. Módulo de interfaz de usuario implementado y distribuido como binario para el sistema operativo a utilizar.
Procedimiento	<ol style="list-style-type: none"> 1. Los usuarios ejecutan acciones en el sistema, como el despliegue y borrado de slices utilizando Linux Cluster
Resultados esperados	<p>A medida que se ejecutan las acciones, se generan logs y se guardan en base de datos.</p> <p>Los tasks de celery también se guardan en base de datos</p>
Resultado del test	PASS
Evidencias	Tasks de celery

	<pre>1 _id: "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961" 2 status: "SUCCESS" 3 result: "e id 6682614936173288f889d272 deployed successfully on Linux Cluster" 4 traceback: null 5 children: Array (empty) 6 date_done: 2024-07-01T07:57:06.212+00:00</pre> <pre>1 _id: "51cc7908-329e-4ebb-91c0-51bf4b663740" 2 status: "SUCCESS" 3 result: "ce id 6682614936173288f889d272 deleted successfully on Linux Cluster" 4 traceback: null 5 children: Array (empty) 6 date_done: 2024-07-01T08:23:10.786+00:00</pre>
Comentarios y/o consideraciones	

Logs de despliegue y borrado de slice con id

```

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:56:57,439 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - Starting deployment of VM slice 6682614936173288f889d272 on Linux Cluster",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:01,077 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - 10.0.0.30 is assigned nodes: 1",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:01,993 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - 10.0.0.40 is assigned nodes: 2",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:02,896 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - 10.0.0.50 is assigned nodes: 3",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:04,595 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - Node nd1 is assigned to 10.0.0.30. Process is 333281 and vnc port is 6381",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:05,398 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - Node nd2 is assigned to 10.0.0.40. Process is 333883 and vnc port is 6388",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:06,200 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - Node nd3 is assigned to 10.0.0.50. Process is 342111 and vnc port is 6330",
  "level": "INFO"
}

{
  "_id": {},
  "task_id": "23c6d6d2-2c0e-4297-a708-d8f6dc9a9961",
  "timestamp": {},
  "message": "2024-07-01 02:57:06,205 - task_23c6d6d2-2c0e-4297-a708-d8f6dc9a9961 - INFO - Slice with slice id 6682614936173288f889d272 deployed successfully on Linux Cluster",
  "level": "INFO"
}
```

```

  "id": {},
  "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740",
  "timestamp": {},
  "message": "2024-07-01 03:23:07,904 - task_51cc7908-329e-4ebb-91c0-51bf4b663740 - INFO - Starting to delete VM slice 6682614936173288f889d272 on Linux Cluster",
  "level": "INFO"
}

{
  "id": {},
  "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740",
  "timestamp": {},
  "message": "2024-07-01 03:23:08,957 - task_51cc7908-329e-4ebb-91c0-51bf4b663740 - INFO - Node nd1 assigned to 10.0.0.30 with process 332281 has been deleted",
  "level": "INFO"
}

{
  "id": {},
  "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740",
  "timestamp": {},
  "message": "2024-07-01 03:23:09,854 - task_51cc7908-329e-4ebb-91c0-51bf4b663740 - INFO - Node nd2 assigned to 10.0.0.40 with process 333883 has been deleted",
  "level": "INFO"
}

{
  "id": {},
  "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740",
  "timestamp": {},
  "message": "2024-07-01 03:23:10,774 - task_51cc7908-329e-4ebb-91c0-51bf4b663740 - INFO - Node nd3 assigned to 10.0.0.50 with process 342111 has been deleted",
  "level": "INFO"
}

{
  "id": {},
  "task_id": "51cc7908-329e-4ebb-91c0-51bf4b663740",
  "timestamp": {},
  "message": "2024-07-01 03:23:10,778 - task_51cc7908-329e-4ebb-91c0-51bf4b663740 - INFO - Slice with slice id 6682614936173288f889d272 deleted successfully on Linux Cluster",
  "level": "INFO"
}

```

9. Módulo de agente worker

9.1 Registro de recursos computacionales en la base de datos

Objetivo del test	Recepción de recursos en el head node provenientes de los workers, para su posterior guardado en la base de datos.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Programa en python para la extracción de recursos en los tres trabajadores presentes en la topología.2. Programa en python para la recepción de recursos en el head node (escuchando en el puerto 9898) y posterior guardado en base de datos de mongo.3. Servicio de la base de datos de MongoDB corriendo en el puerto 27017 del head node.
Procedimiento	<ul style="list-style-type: none">- Se ejecuta el programa de extracción de datos en los 3 workers de la topología, de donde se extrae data correspondiente al porcentaje de uso de cada core (consumo en cada segundo), RAM y disco, para poder el json a enviar.- El programa ejecutado en el head node recibe dicha información enviada a un servicio en escucha en su puerto 9898, para poder posteriormente guardarla en la base de datos en la colección de 'resources'.- EL programa mediante queries filtra los recursos enviados por cada worker ej: query1 = {"worker1": "10.0.0.30"} y procede a quedarse con los 1000 más recientes filtrando por timestamp.- Posteriormente dichos recursos son validados mediante el cliente de mongodb manejado, donde se tiene al dirigirse a la colección de 'resources' se verá la data recolectada con el comando { workerX: "10.0.0.Y"} dependiendo del worker en

	cuestión.
Resultados esperados	La data es guardada en la base de datos conservando los ultimos 1000 registros de acuerdo al timestamp de cada json de recursos recibido.
Resultado del test	PASS
Evidencias	 <p>The screenshot displays terminal logs and a database interface. The terminal shows three HTTP POST requests to '/data' with status '200 OK'. Each request contains a JSON object with resource metrics for different workers. The database interface shows a table with columns 'id', 'workers', 'Core0(%)', 'Core1(%)', 'Core2(%)', 'Core3(%)', 'Core4(%)', 'Core5(%)', 'Core6(%)', 'Core7(%)', 'MemoriaUsada(Mb)', 'MemoriaDisponible(Mb)', 'MemoriaTotal(Mb)', 'AlmacenamientoUsado(Gb)', 'AlmacenamientoUsado(%)', 'AlmacenamientoTotal(Gb)', and 'timestamp'. The 'workers' field is highlighted in red in the database view, showing 'worker1' for the first record, 'worker2' for the second, and 'worker3' for the third. The 'timestamp' field is also highlighted in red, showing '30-06-2024 19:47:33'.</p>
Comentarios y/o consideraciones	<p>En los json guardados en la base de datos se guardados en la base de datos cada uno presenta un campo indicando 'worker1', 'worker2' y 'worker3' para poder diferenciarlos.</p> <p>Dichos programas que se encuentran en los wokers y en el headnode se ejecutan en background para poder tener una recolección constante de los recursos a utilizar.</p>

10. Módulo OpenStack Driver

10.1 Despliegue de slices utilizando infraestructura de OpenStack

Objetivo del test	Validar la funcionalidad del módulo de OpenStack Driver para interactuar con la infraestructura de OpenStack y gestionar los recursos necesarios para el despliegue de slices de VMs.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none">1. Instalación y configuración correcta del entorno OpenStack en el entorno de nube privada (HeadNode y Workers).2. Credenciales de acceso válidas para el módulo de OpenStack Driver.
Procedimiento	<ol style="list-style-type: none">1. Generar un despliegue de slices de VMs mediante la invocación de la función <i>openstackDeployment(slice_json, user_name)</i> del módulo de OpenStack Driver. Recibe como parámetros el json del slice a desplegar (template) y el nombre del usuario dueño del futuro slice. <p>Para user_name, usar: regular_user</p> <p>Ejemplo de slice_json</p> <pre>{ "created_at": "2024-06-30T18:30:07.687411Z", "description": "testfinal", "name": "testfinal", "topology": { "links": [{ "link_id": "nd1_nd2", "source": "nd1", "target": "nd2", "source_port": "nd1_port0", "target_port": "nd2_port0" }, { "link_id": "nd2_nd3", "source": "nd2", "target": "nd3", "source_port": "nd2_port1", "target_port": "nd3_port0" }, { "link_id": "nd3_nd4", "source": "nd3", "target": "nd4", "source_port": "nd3_port1",</pre>

```

    "target_port": "nd4_port0"
  },
  {
    "link_id": "nd4_nd1",
    "source": "nd4",
    "target": "nd1",
    "source_port": "nd4_port1",
    "target_port": "nd1_port1"
  }
],
"nodes": [
  {
    "node_id": "nd1",
    "name": "node_1",
    "image": "6663764959eff954c8764ad8",
    "flavor": {
      "id": "665275b98c45f0c2b8a2e230",
      "name": "256MBRAM_1VCPU_1GBDisk",
      "cpu": 1,
      "memory": 0.256,
      "storage": 1
    },
    "security_rules": [
      22
    ],
    "ports": [
      {
        "node_id": "nd1_port0"
      },
      {
        "node_id": "nd1_port1"
      }
    ]
  },
  {
    "node_id": "nd2",
    "name": "node_2",
    "image": "6663764959eff954c8764ad8",
    "flavor": {
      "id": "665275b98c45f0c2b8a2e230",
      "name": "256MBRAM_1VCPU_1GBDisk",
      "cpu": 1,
      "memory": 0.256,
      "storage": 1
    },
    "security_rules": [
      22
    ],
    "ports": [
      {
        "node_id": "nd2_port0"
      },
      {

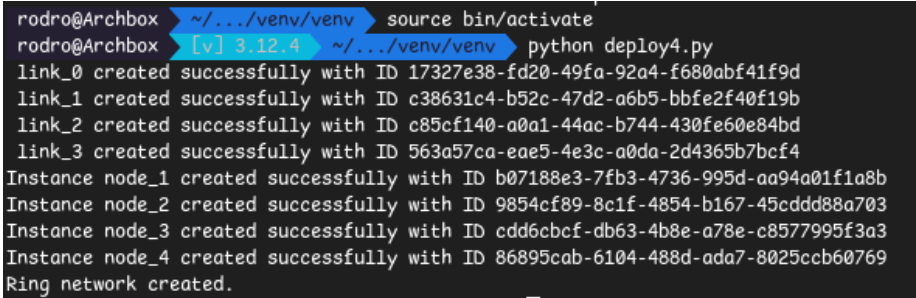
```

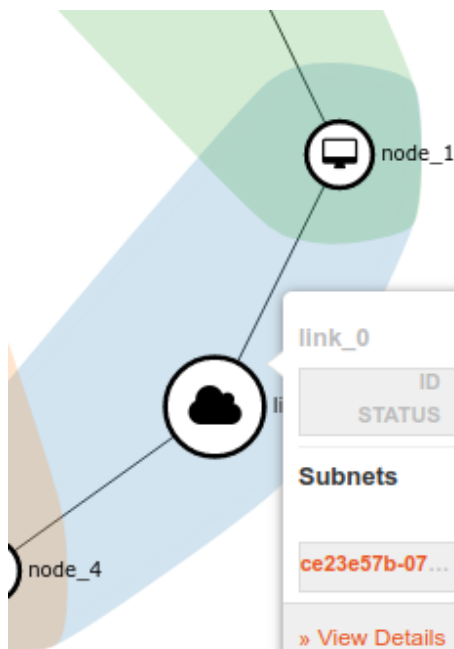
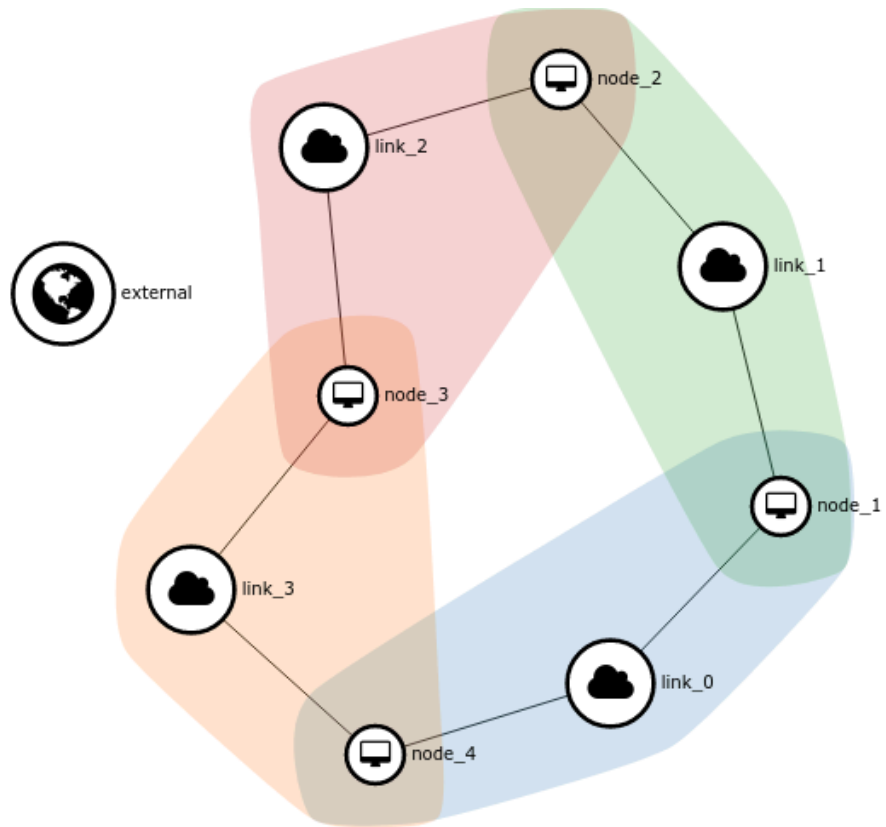


```

        "node_id": "nd2_port1"
    }
]
},
{
    "node_id": "nd3",
    "name": "node_3",
    "image": "6663764959eff954c8764ad8",
    "flavor": {
        "id": "665275b98c45f0c2b8a2e230",
        "name": "256MBRAM_1VCPUs_1GBDisk",
        "cpu": 1,
        "memory": 0.256,
        "storage": 1
    },
    "security_rules": [
        22
    ],
    "ports": [
        {
            "node_id": "nd3_port0"
        },
        {
            "node_id": "nd3_port1"
        }
    ]
},
{
    "node_id": "nd4",
    "name": "node_4",
    "image": "6663764959eff954c8764ad8",
    "flavor": {
        "id": "665275b98c45f0c2b8a2e230",
        "name": "256MBRAM_1VCPUs_1GBDisk",
        "cpu": 1,
        "memory": 0.256,
        "storage": 1
    },
    "security_rules": [
        22
    ],
    "ports": [
        {
            "node_id": "nd4_port0"
        },
        {
            "node_id": "nd4_port1"
        }
    ]
}
}
],
"user_id": "6640550a53c1187a6899a5ab",

```

	<pre>"topology_type": "anillo" }</pre> <ol style="list-style-type: none"> Verificar que el módulo provisiona los recursos necesarios en OpenStack (instancias de VM, redes y almacenamiento) utilizando la interfaz gráfica de OpenStack Horizon (apartados de compute, red y topología de red) y accediendo con las credenciales del usuario administrador. Para verificar estado de VMs o enlaces, utilizar los siguientes comandos en HeadNode sobre openstack <p>Comandos para verificar:</p> <ul style="list-style-type: none"> Despliegue completo: <ul style="list-style-type: none"> openstack server list --project slice_json_name openstack network list --project slice_json_name Estado de redes <pre>openstack network show nombre_o_id_de_la_red -f json jq '.status'</pre> Estado de vms <pre>openstack server show nombre_o_id_de_la_instancia -f json jq '.status'</pre>
Resultados esperados	<ol style="list-style-type: none"> La función <i>openstackDeployment(slice_json, user_name)</i> del módulo de OpenStack Driver imprime logs de creación de recursos (VMs, ports, links). La configuración de las instancias de VM y la red es coherente con las especificaciones del slice_json. En Horizon se observan las especificaciones (instancias, ports, links) ya desplegadas. El estado de los recursos es 'ACTIVE'
Resultado del test	PASS
Evidencias	 <pre>rodro@Archbox ~/.../venv/venv\$ source bin/activate rodro@Archbox [v] 3.12.4 ~/.../venv/venv\$ python deploy4.py link_0 created successfully with ID 17327e38-fd20-49fa-92a4-f680abf41f9d link_1 created successfully with ID c38631c4-b52c-47d2-a6b5-bbfe2f40f19b link_2 created successfully with ID c85cf140-a0a1-44ac-b744-430fe60e84bd link_3 created successfully with ID 563a57ca-eae5-4e3c-a0da-2d4365b7bcf4 Instance node_1 created successfully with ID b07188e3-7fb3-4736-995d-aa94a01f1a8b Instance node_2 created successfully with ID 9854cf89-8c1f-4854-b167-45cddd88a703 Instance node_3 created successfully with ID cdd6cbcf-db63-4b8e-a78e-c8577995f3a3 Instance node_4 created successfully with ID 86895cab-6104-488d-ada7-8025ccb60769 Ring network created.</pre>



link_0

ID 17327e38-fd20-49fa-92a4-f680abf41f9d

STATUS Active

Subnets

ce23e57b-07...

10.0.0.0/24


Delete Subnet

+ Create Subnet

» View Details

Delete Network

```
root@HeadNode:~# openstack network show 17327e38-fd20-49fa-92a4-f680abf41f9d -f json | jq '.status'
"ACTIVE"
```



node_1

ID b07188e3-7fb3-4736-995d-aa94a01f1a8b

STATUS Active

IP Addresses

10.0.0.92ce23e57b-07c6-4bee-99ed-4e5f129c2998

10.0.1.242dcd20648-060e-48d6-93fb-00c8506ef248

» View Instance Details

» Open Console

Delete Instance

```
root@HeadNode:~# openstack server show b07188e3-7fb3-4736-995d-aa94a01f1a8b -f json | jq '.status'
"ACTIVE"
```

Despliegue completo

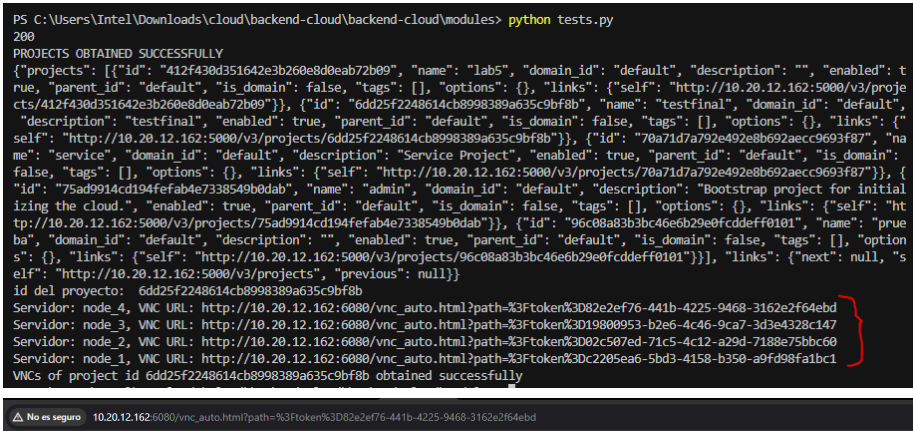
```
root@HeadNode:~# openstack server list --project prueba
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+-----+-----+-----+
| 86895cab-6104-488d-ada7-8025ccb60769 | node_4 | ACTIVE | link_0=10.0.0.104; link_3=10.0.3.124 | cirros | 256MBRAM_1VCPUS_1GBRoot |
| cdd6cbcf-db63-4b8e-a78e-c8577995f3a3 | node_3 | ACTIVE | link_2=10.0.2.93; link_3=10.0.3.70 | cirros | 256MBRAM_1VCPUS_1GBRoot |
| 9854cf89-8c1f-4854-b167-45cdd88a703 | node_2 | ACTIVE | link_1=10.0.1.80; link_2=10.0.2.87 | cirros | 256MBRAM_1VCPUS_1GBRoot |
| b07188e3-7fb3-4736-995d-aa94a01f1a8b | node_1 | ACTIVE | link_0=10.0.0.92; link_1=10.0.1.242 | cirros | 256MBRAM_1VCPUS_1GBRoot |
+-----+-----+-----+-----+-----+-----+

root@HeadNode:~# openstack network list --project prueba
+-----+-----+-----+
| ID | Name | Subnets |
+-----+-----+-----+
| 17327e38-fd20-49fa-92a4-f680abf41f9d | link_0 | ce23e57b-07c6-4bee-99ed-4e5f129c2998 |
| 563a57ca-eae5-4e3c-a0da-2d4369b7bcf4 | link_3 | 7b61869f-206e-4b87-b8f9-20f958e47188 |
| c38631c4-b52c-47d2-a6b5-bbfe2f40f19b | link_1 | dcd20648-060e-48d6-93fb-00c8506ef248 |
| c85cf140-a0a1-44ac-b744-438fe60e84bd | link_2 | 69ade9ab-7fb1-424a-a548-359b62d5d695 |
+-----+-----+-----+
```

Comentarios y/o consideraciones

Módulo de OpenStack Driver debe ser compatible con la versión específica de OpenStack instalada en HeadNode.

10.2 Obtención de URL para acceso VNC a instancias en infraestructura de OpenStack

Objetivo del test	Validar la funcionalidad del módulo de OpenStack Driver para interactuar con la infraestructura de OpenStack y gestionar los recursos necesarios para la obtención de URLs para acceso VNC a VMs.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Instalación y configuración correcta del entorno OpenStack en el entorno de nube privada (HeadNode y Workers). 2. Credenciales de acceso válidas para el módulo de OpenStack Driver.
Procedimiento	<ol style="list-style-type: none"> 1. Generar una obtención de URLs para acceso VNC a instancias de un slice determinado (project_name) mediante la invocación de la función obtainVNCfromProject(project_name) del módulo de OpenStack Driver. Recibe como parámetros el project_name (slice_name). 2. Verificar que el módulo imprime las URLs para acceso VNC de cada instancia del slice (project). 3. Colocar una URL en un navegador (Chrome) para verificar su funcionamiento.
Resultados esperados	<ol style="list-style-type: none"> 1. La función obtainVNCfromProject(project_name) del módulo de OpenStack Driver imprime las URLs para acceso VNC de cada instancia del slice (project).
Resultado del test	PASS
Evidencias	 <pre> PS C:\Users\Intel\Downloads\cloud\backend-cloud\backend-cloud\modules> python tests.py 200 PROJECTS OBTAINED SUCCESSFULLY {"projects": [{"id": "412f43bd351642e3b260e8d0eab72b09", "name": "lab5", "domain_id": "default", "description": "", "enabled": true, "parent_id": "default", "is_domain": false, "tags": [], "options": {}, "links": {"self": "http://10.20.12.162:5000/v3/projects/412f43bd351642e3b260e8d0eab72b09"}}, {"id": "6dd25f2248614cb8998389a635c9bf8b", "name": "testfinal", "domain_id": "default", "description": "testfinal", "enabled": true, "parent_id": "default", "is_domain": false, "tags": [], "options": {}, "links": {"self": "http://10.20.12.162:5000/v3/projects/6dd25f2248614cb8998389a635c9bf8b"}}, {"id": "70a71d7a792e492e8b692a6cc9693f87", "name": "service", "domain_id": "default", "description": "Service Project", "enabled": true, "parent_id": "default", "is_domain": false, "tags": [], "options": {}, "links": {"self": "http://10.20.12.162:5000/v3/projects/70a71d7a792e492e8b692a6cc9693f87"}}, {"id": "75ad9914cd194fefab4e7338549b0dab", "name": "admin", "domain_id": "default", "description": "Bootstrap project for initialing the cloud.", "enabled": true, "parent_id": "default", "is_domain": false, "tags": [], "options": {}, "links": {"self": "http://10.20.12.162:5000/v3/projects/75ad9914cd194fefab4e7338549b0dab"}}, {"id": "96c08a83b3bc46e6b29e0fcddeff0101", "name": "prueba", "domain_id": "default", "description": "", "enabled": true, "parent_id": "default", "is_domain": false, "tags": [], "options": {}, "links": {"self": "http://10.20.12.162:5000/v3/projects/96c08a83b3bc46e6b29e0fcddeff0101"}}, {"id": "http://10.20.12.162:5000/v3/projects", "previous": null}] id del proyecto: 6dd25f2248614cb8998389a635c9bf8b Servidor: node_4, VNC URL: http://10.20.12.162:6080/vnc_auto.html?path=%3Ftoken%3D082e2ef76-441b-4225-9468-3162e2f64ebd Servidor: node_3, VNC URL: http://10.20.12.162:6080/vnc_auto.html?path=%3Ftoken%3D19800953-b2e6-4c46-9ca7-3d3e4328c147 Servidor: node_2, VNC URL: http://10.20.12.162:6080/vnc_auto.html?path=%3Ftoken%3D082c507ed-71c5-4c12-a29d-7188e75bbc60 Servidor: node_1, VNC URL: http://10.20.12.162:6080/vnc_auto.html?path=%3Ftoken%3Dc2205ea6-5bd3-4158-b350-a9fd98fa1bc1 VNCs of project id 6dd25f2248614cb8998389a635c9bf8b obtained successfully </pre>

	 <pre> [3.410498] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found [3.427620] EDD information not available. [3.444700] Freeing unused kernel memory: 1400K (ffffffffff01f42000 - fffffffffff020b4000) [3.468063] Write protecting the kernel read-only data: 14336k [3.485422] Freeing unused kernel memory: 1860K (ffff80000182f000 - fffff80001a00000) [3.509930] Freeing unused kernel memory: 168K (ffff800001d46000 - fffff800001e0000) Further output written to /dev/ttyS0 [4.397366] random: dd urandom read with 23 bits of entropy available [226.241977] random: nonblocking pool is initialized login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root. cirros login: login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root. cirros login: login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root. cirros login: login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root. cirros login: login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root. cirros login: </pre>
Comentarios y/o consideraciones	Módulo de OpenStack Driver debe ser compatible con la versión específica de OpenStack instalada en HeadNode.

10.3 Borrado de slices utilizando infraestructura de OpenStack

Objetivo del test	Validar la funcionalidad del módulo de OpenStack Driver para interactuar con la infraestructura de OpenStack y gestionar los recursos necesarios para el borrado de slices de VMs.
Diagrama de red del Test	N/A
Prerrequisitos	<ol style="list-style-type: none"> 1. Instalación y configuración correcta del entorno OpenStack en el entorno de nube privada (HeadNode y Workers). 2. Credenciales de acceso válidas para el módulo de OpenStack Driver.
Procedimiento	<ol style="list-style-type: none"> 1. Generar un borrado de slices de VMs mediante la invocación de la función <i>openstackDeleteSlice(project_name, slice_id)</i> del módulo de OpenStack Driver. Recibe como parámetros el project_name (slice_name) y slice_id correspondiente al slice guardado en base de datos. 2. Verificar que el módulo libera los recursos utilizados por el slice desplegado en OpenStack (instancias de VM, redes y almacenamiento) mediante la interfaz Horizon accediendo con credenciales de administrador.
Resultados esperados	<ol style="list-style-type: none"> 1. La función <i>openstackDeleteSlice(project_name, slice_id)</i> del módulo de OpenStack Driver imprime logs de borrado de recursos (VMs, ports, links). 2. Al actualizar Horizon, se observa que el slice (project) ha sido borrado (no existente) cuando finaliza por completo la función.
Resultado del test	PASS

Evidencias	<p>Comandos para verificar (antes del borrado del project):</p> <ul style="list-style-type: none">• Listar la cantidad de redes en el proyecto: openstack network list --project project_name• Listar la cantidad de instancias en el proyecto: openstack server list --project project_name <pre>rodrg@Archbox [v] 3.12.4 ~/.../venv/venv python deploy4.py gAAAAABmFpGrEstUFVLVm6xQyXPQ164EdW374-0vcI5P3FohZhEafn-_b1CUNMfbUnanSKM1yJ4eo8Tm8aobLoJuSt09VDDn0AGYVp C6SUE0ATi3gwwA7uU8xRiSWtRtEczhwwUPEXpoHMKNU6xy0-6gY0S35HhR6KH7JM2V7dcuFjUS0P1cMvVken62F_bIuIDhUfmeLCgr Networks list of project 96c08a83b3bc46e6b29e0fcddeff0101 obtained Deleted instance: 86895cab-6104-488d-ada7-8025ccb60769 for link_0 Deleted instance: b07188e3-7fb3-4736-995d-aa94a01f1a8b for link_0 Deleted link with ID: 17327e38-fd20-49fa-92a4-f680abf41f9d Deleted instance: cdd6cbcf-db63-4b8e-a78e-c8577995f3a3 for link_3 Deleted link with ID: 563a57ca-eae5-4e3c-a0da-2d4365b7bcf4 Deleted instance: 9854cf89-8c1f-4854-b167-45cdd88a703 for link_1 Deleted link with ID: c38631c4-b52c-47d2-a6b5-bbfe2f40f19b Deleted link with ID: c85cf140-a0a1-44ac-b744-430fe60e84bd Ring network deleted.</pre> <pre>root@HeadNode:~# openstack network list --project prueba root@HeadNode:~# openstack server list --project prueba root@HeadNode:~# █</pre> 
Comentarios y/o consideraciones	Módulo de OpenStack Driver debe ser compatible con la versión específica de OpenStack instalada en HeadNode.