

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

CLASIFICACIÓN Y CONFIDENCIALIDAD

Este documento es clasificado como “**Uso Interno**”.

El presente documento es propiedad del grupo Keralty y está restringido a los colaboradores de la organización que cuenten con la autorización expresa para su consulta.

No se permite la reproducción total o parcial de este documento, así como su transmisión a terceros sin la autorización del responsable designado por el grupo Keralty.

LISTA DE DISTRIBUCIÓN

Este documento es de uso interno del grupo Keralty y su copia debe ser controlada y registrada de acuerdo con los procedimientos establecidos por la organización. Su distribución se debe realizar de acuerdo con la lista definida en la tabla de distribución maestra SGSI.

Todo cambio realizado a este documento debe ser controlado, documentado de acuerdo con el procedimiento de control documental y registrados en la tabla de control de cambios del presente documento.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

TABLA DE CONTENIDO

1. OBJETIVO.	3
2. ALCANCE.	3
3. DEFINICIONES	3
4. CONTENIDO.	8
4.1. Arquitectura de Ansible	8
4.2. Balanceo de la Aplicación	10
4.3. Integración de GIT para la creación de proyectos	11
4.4. Integración con Azure AD	12
4.5. Proceso Automatizado de Reinicios	12
4.5.1. Caso de Uso de Reinicios	12
4.6. Playbook de Reinicio Unificado	13
4.6.1. Inventario	13
4.6.2. Encabezado	13
4.6.3. Reinicio de Instancias	14
5. CONTROL DE CAMBIOS.	21
6. FLUJO DE APROBACIÓN.	22

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

1. OBJETIVO.

Describir el proceso automatizado que se tiene en Ansible para el reinicio periódico que se hace diariamente en las instancias de aplicación de las siguientes aplicaciones: Avicena Reportes, Conector OSI y Portal de EPS.

2. ALCANCE.

El proceso automatizado podrá reiniciar de forma automática o a demanda las instancias de las siguientes aplicaciones: Avicena Reportes, Conector OSI y Portal de EPS

3. DEFINICIONES

Ansible Automation Platform: Es una plataforma de automatización de código abierto diseñada para la gestión de configuraciones, implementación de aplicaciones, automatización de tareas y orquestación de tecnologías de la información. Simplifica tareas y flujos de trabajo complejos al permitir que los usuarios definan la automatización en un lenguaje declarativo, utilizando YAML.

Alta Disponibilidad: Se refiere a un sistema o componente que está diseñado y configurado de tal manera que garantiza un funcionamiento continuo durante un período de tiempo largo, minimizando el tiempo de inactividad y asegurando que los servicios o aplicaciones estén disponibles de manera constante, incluso en caso de fallos o interrupciones.

Automatización: Es el proceso de realizar tareas, operaciones o procesos de manera automática, sin intervención humana directa.

API: (Application Programming Interface), es un conjunto de reglas y herramientas que permite que diferentes software se comuniquen entre sí. Proporciona un conjunto de funciones y métodos que permiten a desarrolladores de software acceder a las características o datos de una aplicación, servicio o plataforma sin tener que entender su implementación interna.

Arquitectura: Es el marco organizativo que define cómo los diferentes componentes de un sistema interactúan entre sí y cómo se organizan para lograr los objetivos previstos.

Automation Controller: Es un dispositivo o software que gestiona y coordina la automatización de varias tareas o procesos dentro de un sistema. Sirve como un punto centralizado de control para orquestar y ejecutar flujos de trabajo de automatización, asegurando que diferentes componentes trabajen juntos de manera fluida. Los controladores de automatización desempeñan un papel crucial

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

en mejorar la eficiencia, reducir la intervención manual y mantener la consistencia en los procesos automatizados.

Azure: Es una plataforma de servicios en la nube ofrecida por Microsoft. Microsoft Azure proporciona una amplia gama de servicios de computación en la nube, almacenamiento, bases de datos, redes, inteligencia artificial, análisis y más. Estos servicios permiten a las organizaciones construir, implementar y administrar aplicaciones y servicios a través de la infraestructura global de centros de datos de Microsoft.

AD: (Active Directory), es un servicio de directorio desarrollado por Microsoft que se utiliza para almacenar información sobre objetos en una red y facilitar la búsqueda y administración de estos objetos.

Background: Se refiere a un proceso en segundo plano es una tarea o programa que se ejecuta sin interferir directamente con la interacción del usuario. Puede ser una tarea automática, un servicio o una operación que ocurre mientras el usuario realiza otras acciones.

Balanceo: Es el proceso de distribuir de manera equitativa la carga o la demanda entre varios recursos para optimizar el rendimiento, la eficiencia y la disponibilidad. Esta práctica es comúnmente utilizada en diversos contextos tecnológicos para asegurar que los recursos se utilicen de manera equilibrada y que ningún componente esté sobrecargado mientras otros están subutilizados.

Base de Datos: Es un conjunto organizado de datos que se almacenan de manera estructurada y que están accesibles de forma electrónica. En una base de datos, la información se organiza en tablas, que contienen filas y columnas. Cada fila en una tabla representa una entidad individual, mientras que cada columna representa un atributo específico de esas entidades. Las bases de datos permiten almacenar, gestionar, recuperar y actualizar datos de manera eficiente.

Comando: Es una instrucción específica que se da a un sistema informático para llevar a cabo una tarea o realizar una operación.

Caso de Uso: Es una descripción detallada de cómo un usuario interactúa con un sistema para lograr un objetivo específico. Los casos de uso son una herramienta fundamental en el análisis y diseño de sistemas, particularmente en el desarrollo de software, ya que ayudan a capturar los requisitos funcionales del sistema desde la perspectiva del usuario final.

Directorio: Es una estructura de almacenamiento que organiza y gestiona archivos y otros directorios en un sistema de archivos. Los directorios permiten agrupar archivos de manera lógica, facilitando la organización, administración y acceso a los

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

datos almacenados en un dispositivo de almacenamiento como un disco duro, una unidad de estado sólido, una unidad de red, etc.

Execution Environment: Es el conjunto de condiciones y recursos en los que se lleva a cabo la ejecución de un programa o aplicación informática. El entorno de ejecución proporciona el contexto necesario para que un software se ejecute correctamente y alcance sus objetivos.

Flujo: Es el movimiento o la secuencia de datos, información o control a través de un sistema o programa.

GIT: Es un sistema de control de versiones distribuido ampliamente utilizado para el seguimiento de cambios en el código fuente durante el desarrollo de software. Fue creado por Linus Torvalds en 2005 y es conocido por su velocidad, flexibilidad y capacidad para gestionar proyectos de cualquier tamaño. Git es particularmente popular en la comunidad de desarrollo de software de código abierto.

Instancia: Este término se refiere a una ocurrencia específica de un objeto, dato, proceso o servicio, dependiendo del contexto en el que se esté utilizando.

Interfaz Web: Es la parte visual y funcional de un sitio web o aplicación web con la que los usuarios interactúan. Es el medio a través del cual los usuarios pueden acceder y utilizar las funciones y contenidos de una página web.

Inventario: En Ansible, es el archivo que contiene información sobre los nodos que Ansible gestionará. Este archivo describe los hosts (máquinas) y grupos de hosts que forman parte de la infraestructura que se desea administrar con Ansible.

IP: Es un conjunto de reglas que rigen el formato de los datos enviados a través de internet o una red local. La IP permite que los datos se transmitan entre dispositivos (computadoras, servidores, impresoras, etc.) en una red. Cada dispositivo conectado a una red que utiliza el Protocolo de Internet se identifica de manera única mediante una dirección IP.

Interfaz de Red: Es un componente de hardware o software que conecta una computadora u otro dispositivo a una red de computadoras. La interfaz de red permite la comunicación y el intercambio de datos entre dispositivos en la red, actuando como un intermediario entre el dispositivo y los medios de transmisión, como cables de red o señales inalámbricas.

JBoss: Es una marca que abarca una serie de productos y proyectos de software, principalmente relacionados con el desarrollo y la implementación de aplicaciones empresariales en Java. El más conocido de estos productos es el JBoss Enterprise

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

Application Platform (EAP), un servidor de aplicaciones de código abierto que implementa la especificación Java EE (Java Enterprise Edition).

Killrear: Se refiere a la acción de terminar o detener un proceso o tarea en ejecución en un sistema operativo. Este término se usa comúnmente en el contexto de la administración de sistemas y programación, especialmente en entornos Unix y Linux, donde se utilizan comandos específicos para finalizar procesos.

On-premise: Se refiere a la ubicación física de sistemas informáticos, infraestructura y software en las instalaciones físicas de una organización o empresa, en lugar de utilizar recursos de la nube o servicios alojados externamente. En un entorno on-premise, las organizaciones poseen, mantienen y gestionan su propia infraestructura de tecnología de la información (TI) dentro de sus propias instalaciones, ya sea en centros de datos internos o en servidores locales. Esto implica que tienen el control total sobre la configuración, el mantenimiento, la seguridad y la gestión de todos los componentes tecnológicos utilizados para respaldar sus operaciones comerciales.

Oracle Database: Es conocido por su escalabilidad, fiabilidad y capacidades avanzadas de gestión de datos. Ofrece soporte para una variedad de modelos de datos, incluyendo datos estructurados y no estructurados, así como también incluye características de seguridad, alta disponibilidad y rendimiento optimizado. Oracle Database se utiliza en una amplia variedad de aplicaciones empresariales críticas en industrias como finanzas, telecomunicaciones, comercio electrónico, atención médica, entre otras.

Playbook: Es un archivo que contiene una serie de instrucciones (tareas) que describen los pasos que Ansible debe seguir para configurar, administrar o automatizar un conjunto de hosts. Los playbooks son escritos en YAML (YAML Ain't Markup Language) y proporcionan una forma de expresar la configuración deseada y las tareas a realizar en un formato claro y legible.

PostgreSQL: Es un sistema de gestión de bases de datos relacional de código abierto (RDBMS, por sus siglas en inglés) que enfatiza la extensibilidad y la conformidad con los estándares. Es uno de los sistemas de bases de datos más avanzados y potentes disponibles en la actualidad. PostgreSQL es conocido por su capacidad para manejar grandes cantidades de datos, su soporte para consultas complejas y su compromiso con los estándares del lenguaje SQL.

Private Automation Hub: Es un repositorio local que permite a las empresas con entornos desconectados gestionar, compartir y organizar el contenido generado de forma interna y controlar el acceso al contenido creado

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

PID: Se refiere a un número único asignado por el sistema operativo a cada proceso en ejecución. Este identificador se utiliza para gestionar y controlar los procesos dentro del sistema operativo.

Rama: Se refiere a una línea independiente de desarrollo dentro de un sistema de control de versiones. Las ramas permiten a los desarrolladores trabajar en paralelo en diferentes características, correcciones de errores o experimentos sin afectar la base de código principal. Este concepto es fundamental en sistemas de control de versiones como Git, Mercurial y Subversion.

Repositorio: Es el lugar o sistema donde se almacenan y gestionan archivos y recursos relacionados con un proyecto.

Script: Es un conjunto de instrucciones o comandos escritos en un lenguaje de programación que se utiliza para realizar una tarea específica o automatizar una serie de acciones en un sistema informático. Los scripts son secuencias de comandos que ejecutan tareas sin la necesidad de una intervención manual constante.

Servidor: Es un sistema o software diseñado para proporcionar servicios, recursos o funciones a otros dispositivos o programas, conocidos como "clientes". Los servidores desempeñan un papel fundamental en la comunicación y el intercambio de información en redes de computadoras.

Temporales: Son archivos creados por programas y sistemas operativos para almacenar datos de manera transitoria. Estos archivos se utilizan para una variedad de propósitos, incluyendo el almacenamiento intermedio de datos durante el procesamiento, la gestión de memoria, y la facilitación de la comunicación entre procesos. Los archivos temporales suelen ser eliminados automáticamente una vez que ya no son necesarios, aunque en algunos casos pueden necesitar ser gestionados manualmente.

Terminal: Es una interfaz de usuario que permite a los usuarios interactuar con el sistema operativo mediante la introducción de comandos de texto. Las terminales son esenciales para la administración de sistemas, el desarrollo de software y diversas tareas de programación y automatización. Existen dos tipos principales de terminales: terminales físicas y terminales virtuales.

Token: Es una clave de acceso o un identificador único que se utiliza para autenticar y autorizar el acceso a un recurso, como una red, un sistema o un servicio.

Usuario: Se refiere a una persona, cuenta o entidad que interactúa con un sistema informático, una aplicación o un servicio en línea. El término puede tener diferentes contextos y connotaciones en función de la aplicación específica.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

VLAN: Es una tecnología de redes que permite la segmentación lógica de una red física en múltiples redes virtuales. A través de la implementación de VLANs, se pueden crear segmentos de red independientes, incluso si los dispositivos están conectados al mismo switch físico. Cada VLAN actúa como una red de área local (LAN) separada.

Workflow: Es una secuencia de tareas, pasos o procesos que deben ser realizados para completar un objetivo específico. Los workflows son utilizados para estructurar y automatizar procesos repetitivos y complejos, asegurando que las tareas se realicen de manera eficiente, coherente y en el orden correcto.

YAML: Es un lenguaje de serialización de datos que las personas pueden comprender y suele utilizarse en el diseño de archivos de configuración. Para algunas personas, YAML significa otro lenguaje de marcado más; para otras, es un acrónimo recursivo que quiere decir "YAML no es un lenguaje de marcado", lo que enfatiza la idea de que se utiliza para los datos, no para los documentos. Es un lenguaje de programación popular porque está diseñado para que sea fácil de leer y entender. También se puede utilizar junto con otros lenguajes de programación.

4. CONTENIDO.

4.1. Arquitectura de Ansible

La arquitectura implementada para Ansible Automation Platform se compone de seis (6) servidores on-premise montados en las VLAN productivas las cuales son la VLAN 1 (10.160.1.*) y la VLAN 2 (10.160.2.*) distribuidos de la siguiente manera:

- Tres (3) servidores que contienen los componentes del Automation Controller los cuales incluyen la interfaz web, la API y el motor de configuración de Ansible, los servidores en mención son los siguientes:

10.160.1.235 --- srvvkansiblet1.colsanitas.com

10.160.1.236 --- srvvkansiblet2.colsanitas.com

10.160.1.237 --- srvvkansiblet3.colsanitas.com

- Un (1) servidor para el Private Automation Hub, que proporciona los medios para administrar el contenido de ansible y un registry para almacenar los Execution Environments personalizados, el servidor en mención es el siguiente:

10.160.1.239 --- srvvkansiblehub.colsanitas.com

- Un (1) servidor para almacenar la instancia de la base de datos PostgreSQL v13, el cual mantiene 2 bases de datos: una para los controller y otra para el private automation hub, el servidor en mención es el siguiente:

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

10.160.1.238 --- srvvkansiblebd1.colsanitas.com

- Un (1) servidor que contiene las herramientas necesarias para desarrollar y probar playbooks y execution environments para AAP, el servidor en mención es el siguiente:

10.160.2.37 --- srvvkansibledevsrv.colsanitas.com

A continuación, se muestra el diagrama de la arquitectura de Ansible:

<div> <div>Versión: 3.0</div> <div>Fecha: 28-11-2024</div> <div>Código: SIG-TI-CKE-PR027</div> </div>	<div> <div>PROCEDIMIENTO REINICIOS</div> <div>AUTOMATIZADOS EN ANSIBLE</div> </div>	<div>  </div>
---	---	--

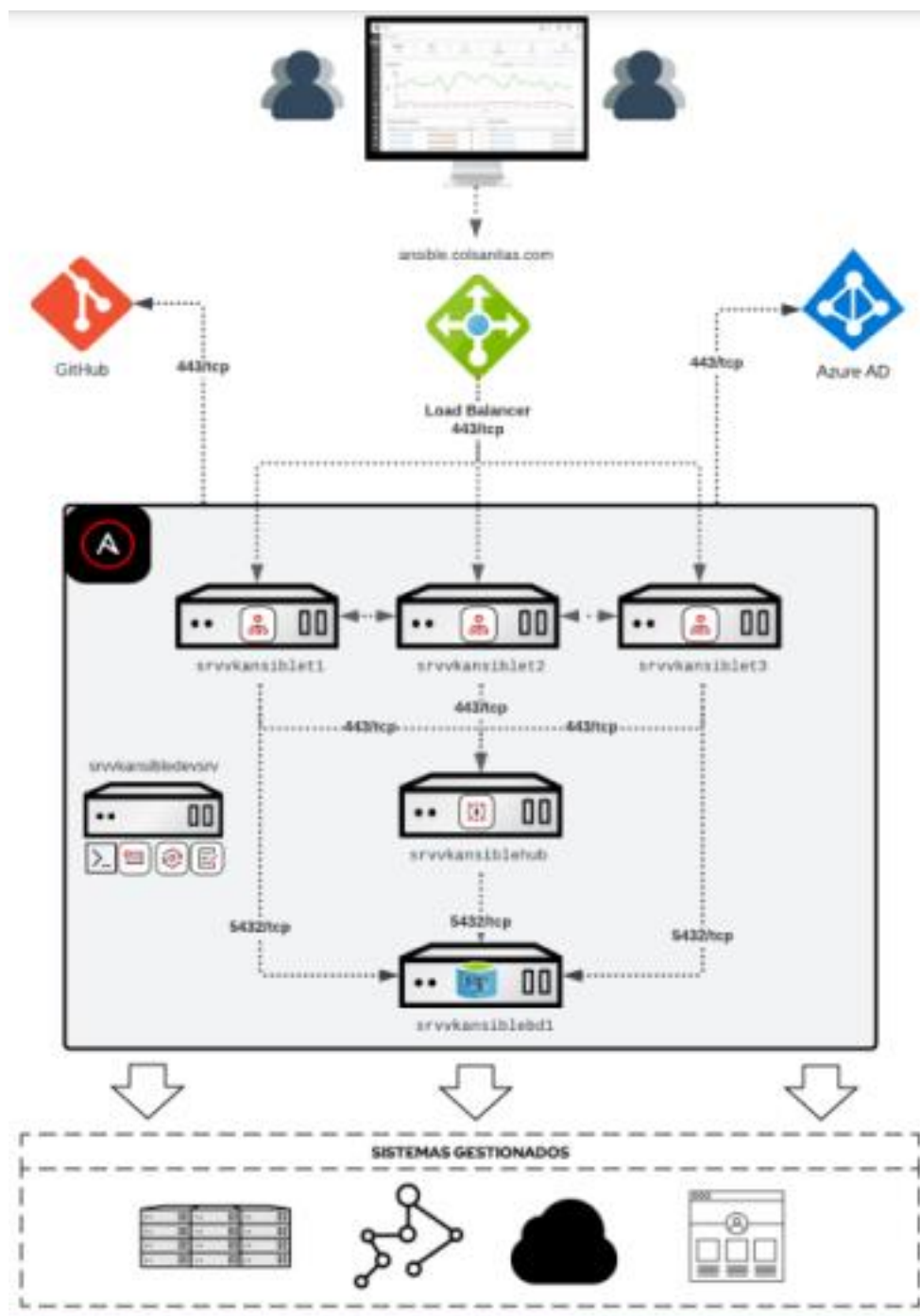


Figura 1. Arquitectura de Ansible

4.2. Balanceo de la Aplicación

El acceso a la consola de Ansible Automation Platform (AAP) se encuentra balanceada en la siguiente URL: <https://ansible.colsanitas.com/#/login> sobre los 3 servidores que contienen el Automation Controller.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

También se desplegaron tres (3) servidores del automation controller para mantener una alta disponibilidad y aumentar la capacidad de la plataforma durante la ejecución de las plantillas de trabajo, se puede acceder al AAP directamente desde los nodos a través de las siguientes URL:

<https://10.160.1.235/#/login>

<https://10.160.1.236/#/login>

<https://10.160.1.237/#/login>

A continuación, se relaciona la información detallada del balanceo de la plataforma:

Nombre	Dirección IP	Observación
ansible.colsanitas.com	172.22.150.81	Balanceo de Consola Web y API

Tabla 1. Información de Balanceo Ansible

4.3. Integración de GIT para la creación de proyectos

Como pudimos ver en la arquitectura ansible almacena sus playbooks y directorios de playbooks desde un repositorio en GIT, este repositorio pertenece a la rama máster y lo podemos encontrar en la siguiente URL: https://github.com/keraltyansible/operaciones_ansible

Este es el repositorio de git asignado para el proyecto de ansible, a este repositorio se accede a través de una autenticación gestionado por Azure AD y se encuentra integrado al ansible a través de un token generado por GIT y configurado en el Automation Controller de la siguiente manera:


◀ Volver a Proyectos					
Detalles Acceso Plantillas de trabajo Notificación Programaciones					
Último estado de la tarea	✔ Correctamente	Nombre	GitHub_operaciones_ ansible	Organización	Default
Tipo de fuente de control	Git	Revisión del control de fuentes	b3fd068 	URL de fuente de control ⓘ	https://github.com/keraltyansible/operaciones_ansible
Credencial de fuente de control	Scm: Github-jlramos_...	Tiempo de espera de la caché	240 Segundos	Ruta base del proyecto ⓘ	/var/lib/awx/projects
Directorio de playbook ⓘ	_8_github_operaciones_ansible	Creado	31/8/2023, 2:13:48 p. m. por admin	Último modificado	23/10/2023, 4:06:38 p. m. por admin

Figura 2. Configuración de Repositorio GIT de Producción en AAP

Nota: La rama mencionada anteriormente es la rama productiva, también llamada rama master y es donde se almacenan los playbooks que se están usando actualmente, sin embargo también tenemos una segunda rama llamada develop ,

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

allí es donde alojamos los playbooks de prueba o aquellos desarrollados de manera transitoria para un único uso y lo podemos encontrar en la siguiente URL: https://github.com/keraltyansible/operaciones_ansible/tree/develop


Volver a Proyectos Detalles Acceso Plantillas de trabajo Notificación Programaciones					
Último estado de la tarea	Correctamente	Nombre	GitHub_OpsAAP_br_develop	Descripción	GitHub en rama develop
Organización	Default	Tipo de fuente de control	Git	Revisión del control de fuentes	c2c9118 
URL de fuente de control ⓘ	https://github.com/keraltyansible/operaciones_ansible	Rama de fuente de control ⓘ	develop	Credencial de fuente de control	Scm: Github-jiramos...
Tiempo de espera de la caché	240 Segundos	Ruta base del proyecto ⓘ	/var/lib/awx/projects	Directorio de playbook ⓘ	_9_github_opsaap_br_develop
Creado	6/9/2023, 4:27:44 p. m. por admin	Último modificado	11/10/2023, 11:33:40 a. m. por admin		

Figura 3. Configuración de Repositorio GIT de Desarrollo en AAP

4.4. Integración con Azure AD

De acuerdo con las políticas de keralty, para la gestión de usuarios y contraseñas el servicio de autenticación del Automation Controller está integrado con Azure AD para permitir la autenticación y gestión de usuarios desde el proveedor de identidad de Azure. Para realizar la integración, el cliente creo la aplicación llamada "ANSIBLE AUTOMATION PLATFORM", cuyo ID es "b4ad74ca-8f7d-4310-b49b-536d6734021b". La URL de callback configurada en Azure AD es:

<https://ansible.colsanitas.com/sso/complete/azuread-oauth2/>

4.5. Proceso Automatizado de Reinicios

4.5.1. Caso de Uso de Reinicios

A continuación, se ilustrará el caso de uso que describe el procedimiento que siguen los operadores del centro y los administradores de sistema para reiniciar los servicios de las aplicaciones mencionadas en el alcance de este documento.

Versión: 3.0 Fecha: 28-11-2024 Código: SIG-TI-CKE-PR027	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
--	---	---

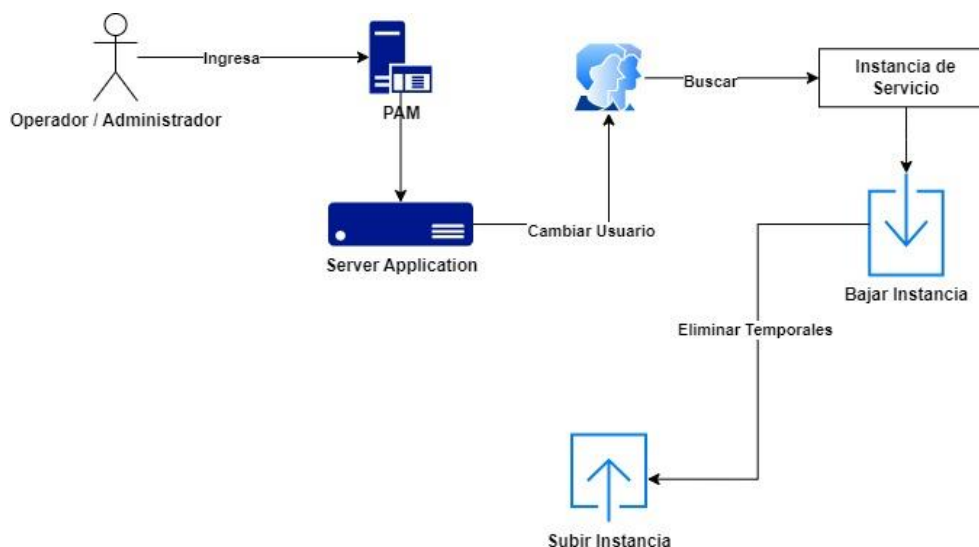


Figura 4. Caso de Uso para Reinicios de Instancias de Servicio Manual

4.6. Playbook de Reinicio Unificado

4.6.1. Inventario

Se configuro un inventario construido con los inventarios ya existentes que incluyen todos los servidores de las aplicaciones Avicena Reportes, Portales EPS, Conector OSI y el servidor del runner el 10.160.2.28 para que l playbook ejecute las tareas correspondientes en cada uno de estos 22 servidores.

Nombre	Reinicios_Diarios	Último estado de la tarea	✓ Correctamente	Tipo	Inventario construido
Organización	Default	Total groups ②	3	Total hosts ②	22
Total inventory sources ②	0	Update cache timeout ②	0	Inventory sources with failures ②	0
Verbosity ②	0				
Inventarios de entrada	10.160.2.28 avicenareportes epsportals conectorosi				

Figura 5. Inventario Construido para los Reinicios Diarios

4.6.2. Encabezado

```

- name: Playbook para Reiniciar Instacias de PortalEPS, AvicenaReportes y ConectorOSI
  hosts: all
  vars:
    user1: root
    user2: jboss
    user3: devops
  
```

Figura 6. Encabezado del Playbook

<div> Versión: 3.0 </div> <div> Fecha: 28-11-2024 </div> <div> Código: SIG-TI-CKE-PR027 </div>	<div> PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE </div>	<div>  </div>
--	---	--

- ❖ **name:** Aquí definimos el propósito que va a tener el playbook
- ❖ **hosts:** Aquí definimos que este playbook se aplique a todos los hosts definidos en el inventario
- ❖ **vars:** Declaramos en 3 variables los usuarios que usaremos para ejecutar los procesos de los reinicios:
 - **user1:** Usuario **root** para ejecutar el killeo de las instancias de las aplicaciones
 - **user2:** Usuario **jboss** para subir las instancias de Avicena Reportes y Portal EPS
 - **user3:** Usuario **devops** para subir las instancias de Conector OSI desde el servidor del runner (10.160.2.28)

4.6.3. Reinicio de Instancias

4.6.3.1. Killear Instancias de Portal EPS

```
tasks:
#Tarea para Killear Instancias de los Portales 1 al 5
- name: Killeo de Instancias de Portales EPS 1 al 5
  shell: ps -efa | grep -i PortalEPS | grep -v grep | awk '{print $2}' | xargs kill -9
  become: yes
  become_user: "{{ user1 }}"
  when: inventory_hostname in ["10.160.1.120", "10.160.1.129", "10.160.1.122", "10.160.1.123", "10.160.1.124"]
```

Figura 7. Killeo de Instancias de Portal EPS (1 al 5)

- ❖ **Propósito:** Detener todos los procesos de **Portal EPS** en los nodos 1 al 5.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **PortalEPS** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user: "{{ user1 }}"**: Ejecuta el proceso como usuario **root**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.2. Killear Instancias de Conector OSI (1 al 3)

```
#Tarea para Killear Instancias de Conector OSI 1 al 3
- name: Killeo de Instancias de Conector OSI 1 al 3
  shell: ps -efa | grep -i ConectorOSI | grep -v grep | awk '{print $2}' | xargs kill -9
  become: yes
  become_user: "{{ user1 }}"
  when: inventory_hostname in ["10.160.1.147", "10.160.1.148", "10.160.1.149"]
```

Figura 8. Killeo de Instancias de Conector OSI (1 al 3)

<div> Versión: 3.0 </div> <div> Fecha: 28-11-2024 </div> <div> Código: SIG-TI-CKE-PR027 </div>	<div> PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE </div>	<div>  </div>
--	---	--

- ❖ **Propósito:** Detener todos los procesos de **Conector OSI** en los nodos 1 al 3.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **ConectorOSI** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user: "{{ user1 }}"**: Ejecuta el proceso como usuario **root**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.3. Killear Instancias de Avicena Reportes (2 y 3)

```
#Tarea para Killear Instancias de Avicena Reportes 2 y 3
- name: Killeo de Instancias de Reportes Avicena 2 y 3
  shell: ps -efa | grep -i ReportesAvicena | grep -v grep | awk '{print $2}' | xargs kill -9
  become: yes
  become_user: "{{ user1 }}"
  when: inventory_hostname in ["10.160.1.35", "10.160.1.36"]
```

Figura 9. Killeo de Instancias de Avicena Reportes (2 y 3)

- ❖ **Propósito:** Detener todos los procesos de **Avicena Reportes** en los nodos 2 y 3.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **Avicena Reportes** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user: "{{ user1 }}"**: Ejecuta el proceso como usuario **root**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.4. Esperar 2 minutos

```
#Tarea para esperar 2 minutos para subir las instancias de las aplicaciones killeadas
- name: Esperar 2 minutos para subir las instancias
  pause:
    minutes: 2
```

Figura 10. Espera de 2 minutos antes de ejecutar las siguientes tareas

- ❖ **Propósito:** Introduce un retraso de 2 minutos antes de iniciar las instancias nuevamente.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

4.6.3.5. Subir Instancias de Portal EPS (1 al 5)

```
#Tarea para Subir Instancias de los Portales 1 al 5
- name: Subir Instancias Portales EPS 1 al 5
  shell: sh /opt/devops/deploy/shell/Reinicio_PortalePS/subir_servicios_portal.sh
  become: yes
  become_user: "{{ user2 }}"
  when: inventory_hostname in ["10.160.1.120", "10.160.1.129", "10.160.1.122", "10.160.1.123", "10.160.1.124"]
```

Figura 10. Subir Instancias de Portal EPS (1 al 5)

- ❖ **Propósito:** Ejecutar script de inicio de instancia de **PortalEPS**.
- ❖ **shell:** Invoca el script **subir_servicios.sh** ubicado en la ruta: **/opt/devops/deploy/shell/Reinicio_PortalePS**.
- ❖ **become_user:** "{{ user2 }}": Ejecuta el proceso como usuario **jboss**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.6. Subir Instancias de Conector OSI (1 al 3)

```
#Tarea para Subir Instancias de Conector OSI 1 al 3
- name: Subir Instancia Conector OSI 147
  shell: ssh tomcat@10.160.1.147 "sh /opt/devops/deploy/shell/Reinicio_Conector/subir_instancia_conector.sh"
  become: yes
  become_user: "{{ user3 }}"
  when: inventory_hostname == '10.160.2.28'

- name: Subir Instancia Conector OSI 148
  shell: ssh tomcat@10.160.1.148 "sh /opt/devops/deploy/shell/Reinicio_Conector/subir_instancia_conector.sh"
  become: yes
  become_user: "{{ user3 }}"
  when: inventory_hostname == '10.160.2.28'

- name: Subir Instancia Conector OSI 149
  shell: ssh tomcat@10.160.1.149 "sh /opt/devops/deploy/shell/Reinicio_Conector/subir_instancia_conector.sh"
  become: yes
  become_user: "{{ user3 }}"
  when: inventory_hostname == '10.160.2.28'
```

Figura 11. Subir Instancias de Conector OSI (1 al 3)

- ❖ **Propósito:** Ejecutar remotamente desde el servidor 10.160.2.28 el script de inicio de instancia de **Conector OSI**.
- ❖ **shell:** Usa ssh para ejecutar remotamente desde el servidor 10.160.2.28 con el usuario **devops** el script de subida de instancias de **Conector OSI** sobre los servidores 10.160.1.147, 10.160.1.148 y 10.160.1.149
- ❖ **become_user:** "{{ user3 }}": Ejecuta el proceso como usuario **devops**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

<div> Versión: 3.0 </div> <div> Fecha: 28-11-2024 </div> <div> Código: SIG-TI-CKE-PR027 </div>	<div> PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE </div>	
--	---	---

4.6.3.7. Subir Instancias de Reportes Avicena

```
#Tarea para Subir Instancias de Reportes Avicena 1 al 3
- name: Reinicio de Instancias de Reportes
  shell: sh /opt/jboss/EAP-7.2.0/subir/subir_servicios_reportes_avicena.sh
  become: yes
  become_user: "{{ user2 }}"
  when: inventory_hostname in ["10.160.1.35", "10.160.1.36"]
```

Figura 12. Subir Instancias de Avicena Reportes (2 y 3)

- ❖ **Propósito:** Ejecutar script de inicio de instancias de **Avicena Reportes**.
- ❖ **shell:** Invoca el script **subir_servicios_reportes_avicena.sh** ubicado en la ruta: **/opt/jboss/EAP-7.2.0/subir**.
- ❖ **become_user:** "{{ user2 }}": Ejecuta el proceso como usuario **jboss**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.8. Esperar 5 minutos

```
#Tarea para esperar 5 minutos para subir las instancias de las aplicaciones killeadas
- name: Esperar 5 minutos para subir las instancias
  pause:
    minutes: 5
```

Figura 13. Espera de 5 minutos antes de ejecutar las siguientes tareas

- ❖ **Propósito:** Introduce un retraso de 5 minutos antes de iniciar las instancias nuevamente.

4.6.3.9. Killar Instancias de Portal EPS (6 al 9)

```
#Tarea para Killar Instancias de los Portales 6 al 9
- name: Reinicio de Instancias de Portales EPS 6 al 9
  shell: ps -efa | grep -i PortalEPS | grep -v grep | awk '{print $2}' | xargs kill -9
  become: yes
  become_user: "{{ user1 }}"
  when: inventory_hostname in ["10.160.1.125", "10.160.1.126", "10.160.1.246", "10.160.1.14"]
```

Figura 14. Killeo de Instancias de Portal EPS (6 al 9)

- ❖ **Propósito:** Detener todos los procesos de **Portal EPS** en los nodos 6 al 9.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **PortalEPS** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user:** "{{ user1 }}": Ejecuta el proceso como usuario **root**

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.10. Killear Instancias de Conector OSI (1 al 3)

```
#Tarea para Killear Instancias de Conector OSI 4 y 5
- name: Killeo de Instancias de Conector OSI 4 y 5
  shell: ps -efa | grep -i ConectorOSI | grep -v grep | awk '{print $2}' | xargs kill -9
  become: yes
  become_user: "{{ user1 }}"
  when: inventory_hostname in ["10.160.2.31", "10.160.2.32"]
```

Figura 15. Killeo de Instancias de Conector OSI (4 y 5)

- ❖ **Propósito:** Detener todos los procesos de **Conector OSI** en los nodos 4 y 5.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **ConectorOSI** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user: "{{ user1 }}"**: Ejecuta el proceso como usuario **root**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.11. Killear Instancias de Avicena Reportes (4 y 5)

```
#Tarea para Killear Instancias de Avicena Reportes 4 y 5
- name: Killeo de Instancias de Reportes Avicena 1 al 3
  shell: ps -efa | grep -i ReportesAvicena | grep -v grep | awk '{print $2}' | xargs kill -9
  when: inventory_hostname in ["10.160.1.37", "10.160.1.38"]
```

Figura 16. Killeo de Instancias de Avicena Reportes (4 y 5)

- ❖ **Propósito:** Detener todos los procesos de **Avicena Reportes** en los nodos 2 y 3.
- ❖ **ps -efa:** Lista todos los procesos del sistema.
- ❖ **grep -i PortalEPS:** Filtra todos los procesos que contienen el nombre **Avicena Reportes** ignorando mayúsculas y minúsculas.
- ❖ **grep -v grep:** Excluye el propio proceso **grep**.
- ❖ **awk '{print \$2}':** Obtiene los PID (IDs de proceso).
- ❖ **xargs kill -9:** Termina forzosamente los procesos.
- ❖ **become_user: "{{ user1 }}"**: Ejecuta el proceso como usuario **root**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

<div> Versión: 3.0 </div> <div> Fecha: 28-11-2024 </div> <div> Código: SIG-TI-CKE-PR027 </div>	<div> PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE </div>	<div>  </div>
--	---	--

4.6.3.12. Subir Instancias de Portal EPS (6 al 9)

```
#Tarea para Subir Instancias de los Portales 6 al 9
- name: Subir Instancias Portales EPS 6 al 9
  shell: sh /opt/devops/deploy/shell/Reinicio_PortalEPS/subir_servicios_portal.sh
  become: yes
  become_user: "{{ user2 }}"
  when: inventory_hostname in ["10.160.1.125", "10.160.1.126", "10.160.1.246", "10.160.1.14"]
```

Figura 17. Subir Instancias de Portal EPS (6 al 9)

- ❖ **Propósito:** Ejecutar script de inicio de instancia de **PortalEPS**.
- ❖ **shell:** Invoca el script **subir_servicios.sh** ubicado en la ruta: **/opt/devops/deploy/shell/Reinicio_PortalEPS**.
- ❖ **become_user:** "{{ user2 }}": Ejecuta el proceso como usuario **jboss**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

4.6.3.13. Subir Instancias de Conector OSI (4 y 5)

```
#Tarea para Subir Instancias de Conector OSI 4 y 5
- name: Subir Instancia Conector OSI 2.31
  shell: ssh tomcat@10.160.2.31 "sh /opt/devops/deploy/shell/Reinicio_Conector/subir_instancia_conector.sh"
  become: yes
  become_user: "{{ user3 }}"
  when: inventory_hostname == '10.160.2.28'

- name: Subir Instancia Conector OSI 2.32
  shell: ssh tomcat@10.160.2.32 "sh /opt/devops/deploy/shell/Reinicio_Conector/subir_instancia_conector.sh"
  become: yes
  become_user: "{{ user3 }}"
  when: inventory_hostname == '10.160.2.28'
```

Figura 18. Subir Instancias de Conector OSI (4 y 5)

- ❖ **Propósito:** Ejecutar remotamente desde el servidor 10.160.2.28 el script de inicio de instancia de **Conector OSI**.
- ❖ **shell:** Usa ssh para ejecutar remotamente desde el servidor 10.160.2.28 con el usuario **devops** el script de subida de instancias de **Conector OSI** sobre los servidores 10.160.2.31 y 10.160.2.32
- ❖ **become_user:** "{{ user3 }}": Ejecuta el proceso como usuario **devops**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

4.6.3.14. Subir Instancias de Reportes Avicena

```
#Tarea para Subir Instancias de Reportes Avicena 4 y 5
- name: Reinicio de Instancias de Reportes
  shell: sh /opt/jboss/EAP-7.2.0/subir/subir_servicios_reportes_avicena.sh
  become: yes
  become_user: "{{ user2 }}"
  when: inventory_hostname in ["10.160.1.37", "10.160.1.38"]
```

Figura 19. Subir Instancias de Avicena Reportes (4 y 5)

- ❖ **Propósito:** Ejecutar script de inicio de instancias de **Avicena Reportes**.
- ❖ **shell:** Invoca el script **subir_servicios_reportes_avicena.sh** ubicado en la ruta: **/opt/jboss/EAP-7.2.0/subir**.
- ❖ **become_user:** **"{{ user2 }}"**: Ejecuta el proceso como usuario **jboss**
- ❖ **when:** Aplica la tarea solo en los servidores listados en la condicional.

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

5. CONTROL DE CAMBIOS.

FECHA	CAMBIO	VERSIÓN
12/12/2023	Creación de Documento.	1.0
04/06/2024	Modificación de Esquemas de Reinicios a través de Workflows.	2.0
28/11/2024	Unificación de reinicios en un solo playbook después de la migración de Avicena.	3.0

Tabla 2. Control de Cambios

Versión: 3.0	PROCEDIMIENTO REINICIOS AUTOMATIZADOS EN ANSIBLE	
Fecha: 28-11-2024		
Código: SIG-TI-CKE-PR027		

6. FLUJO DE APROBACIÓN.

ELABORÓ	REVISÓ	APROBÓ
Nombre: Jose Luis Ramos Bernal (Administrador de sistemas I) Área/Proceso: GESTION CORPORATIVA TECNOLOGIA Y PROCESOS-TRANSVERSAL Fecha: 28/11/2024	Nombre: Jairo Arley Zamudio Tovar (CONSULTOR DE PROYECTOS Y AUTOMATIZACION TI) Área/Proceso: GESTION CORPORATIVA TECNOLOGIA Y PROCESOS-TRANSVERSAL Fecha: 28/11/2024	Nombre: Luisa Gineth Castaño Perea (Director(a) de Operaciones y Comunicaciones TI) Área/Proceso: GESTION CORPORATIVA TECNOLOGIA Y PROCESOS-TRANSVERSAL Fecha: 05/12/2024

Tabla 3. Flujo de Aprobación

Cualquier copia impresa de este documento se considera como **COPIA NO CONTROLADA**.