



MANUAL DE DESPLIEGUE CON KIT ANSIBLE

Release Notes	4
Compatibilidad de versiones	5
Recomendaciones	6
Control de versiones	6
Best practices	6
Procedimiento de actualización	7
Requerimientos	7
Software requerido	7
Conectividad requerida	8
Particularidades SolR	8
SolR en entornos NO productivos	8
Particularidades RedHat	9
Certificados autofirmados httpd en RedHat8 y Amazon Linux 2	9
Apache2 y SELinux	11
Intérprete de Python	11
Arranque y parada	11
Particularidades Amazon Linux 2	12
Caracteres especiales	12
Arranque y parada	13
Información general	14
Introducción Ansible	14
¿Qué es un rol?	14
¿Qué es un inventario?	14
¿Qué es un playbook?	14
Playbooks disponibles	15

Estructura de los roles	15
Ajuste de IPs, credenciales y variables requeridos	15
Instalación, configuración y mantenimiento del kit	21
Instalación	21
Instalación de runtime ansible	21
Descarga y despliegue	22
Actualización kit Ansible	23
Respaldo de configuración	23
Ajuste de versión + Descarga	23
Actualización kit Ansible	24
Ajustar ips y credenciales de inventarios y variables de roles	26
Logs	27
Despliegue y mantenimiento de instancia de Anjana Data	28
Despliegue de nueva instancia	28
Generación de playbook	28
Generación de inventario de hosts	28
Ajuste de variables	28
Ejecución del playbook	29
Actualización de instancia	29
Actualización de Anjana	29
Actualización de descriptores de servicios	30
Actualización de vhosts de webserver	30
Instalar/Actualizar instancia desde dentro de la propia máquina	30
Actualización de configuración de Anjana	30
Mantenimiento de instancia	33
Arranque, parada y reinicio	33
Backup de los datos	35
Exportación e importación de datos	35
Exportar log de microservicios	37
Desinstalación	38
Despliegue de plataforma de monitorización de métricas	40
Prometheus	41
Solr exporter y Grafana Agent	42
Grafana	43
Tempo	44
Despliegue de plataforma de agregación de logs	46
Loki	46
Grafana Agent	47
Despliegue de plataforma de monitorización completa	48
Detalle de composición del kit	49
Roles disponibles	49

Tags disponibles	51
Funciones adicionales del kit	56
Cambiar el formato del Log de Anjana	56
Descargar artefactos de Anjana	57
Sampledata	57
Data delete	57
Data reset	58
Exportar backups a S3	58
Balanceador/HA para MinIO	59
Carpeta data MinIO en HA	59
Scripts funcionales	59
Compatibilidad con uso de sudo	61
Verificar que anjana esté levantado correctamente.	61
Repositorios proxy	62
Repositorios proxy Apt	62
Repositorio proxy pip	62
Ejemplos de uso	63
Despliegue single node + RDS	63
Despliegue single node	68
Despliegue en HA + resiliencia Solr	72
Generación de nueva instancia de un plugin ya existente	83
Cambio de nombres de microservicios	86
Despliegue single node + máquina plugin sin edusa	90
Arrancar instancia en HA degradado (Falta de algún nodo)	92
Despliegue de Anjana en Oracle o SqlServer	94
Errores conocidos	96
Timeout	96
Fallo remove-anjana	96
Fallo remove-anjana - cluster MinIO	96
Carpeta data MinIO en HA	97
Borrado de colecciones	97
Error connection check a persistencias	98

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	30/11/2023	Dpto DS	Responsable DS	Creación del documento

Release Notes

- Se han actualizado todos los templates para que la configuración necesaria para un despliegue usando AWS S3 como buckets para el almacenamiento sea rellena dinámicamente al marcar el campo correspondiente en el all.yml.
- Nueva funcionalidad/tag “export-log-s3” que sube el log exportado de los microservicios de Anjana al bucket de logs indicado en el all.yml
- Se han actualizado todos los templates de los microservicios para que las cadenas de conexión sean rellenas automáticamente durante el despliegue con la configuración provista en el archivo all.yml. Ya no será necesario ajustar las configuraciones para entornos distribuidos
- Ahora los buckets a desplegar y sus respectivos nombres pueden ser personalizados desde el all.yml
- Se ha unificado el método de instalación del cliente de PostgreSQL para todos los sistemas soportados. Deja de ser necesario habilitar repositorios externos para el uso de la Utilidad de Exportación e Importación, y el despliegue de datos de ejemplo.
- Se ha añadido la posibilidad de desplegar múltiples instancias de un microservicio y/o plugin en un solo nodo o de forma distribuida.
- Se ha añadido la posibilidad de seleccionar el tipo de log generado por los microservicios entre default y monitoring, permitiendo este último generar un log optimizado para su utilización junto con la plataforma de monitorización.
- Se ha extendido la plataforma de monitorización con nuevos dashboards, y la inclusión de tempo y open telemetry collector. También se ha reducido el número de exporters necesarios para la obtención de métricas con la implementación del grafana agent. La configuración de puertos y credenciales relacionados, entre otros, ya se encuentra en el all.yml junto al resto.
- Se ha migrado el despliegue de configuración del rol edusa al anjana-utility. El despliegue de configuración ya puede realizarse aunque edusa no esté habilitado/desplegado. Ahora está soportado el despliegue de configuración para múltiples instancias en plugins y microservicios.
- Se han añadido los plugins aqtiva, ranger y tableau al producto.
- Actualización de la nueva url del repositorio de artefactos de Anjana, configurable en el all.yml.
- Añadido identificador/marca de agua en los templates gestionados por el kit de ansible.
- Añadida rotación de logs para Apache Web Server y el módulo modsecurity.
- Nueva funcionalidad/tag “check_health” para comprobar el estado del entorno.
- Nuevos tags “insert-ldap/insert-bbdd/insert-s3” para la carga de datos de ejemplo.
- Se ha migrado la configuración de la licencia al fichero all.yml para su accesibilidad.
- Nuevos tags “import-bbdd/import-s3/import-solr” para la migración/importación de datos.

Compatibilidad de versiones

Versiones soportadas de Anjana Data:

- Desde “4.4.0” hasta “Release Notes 23.1”

Sistemas operativos soportados:

- Ubuntu 18.04|20.04|22.04
- Redhat 8
- Amazon Linux 2

Imágenes cloud testadas:

- AWS
 - [Amazon Linux 2 AMI \(HVM\), SSD Volume Type \(64-bit x86\)](#)
 - [Red Hat Enterprise Linux 8.8 \(HVM\) \(ami-04a5b6bd429eaf261\)](#)
 - [Ubuntu 18.04 LTS - Bionic \(ami-0f027af257d061f7c\)](#)
 - [Ubuntu 20.04 LTS - Focal](#)
 - [Ubuntu 22.04 LTS - Jammy](#)
- Azure
 - [Ubuntu Server 18.04 LTS](#)
 - [Ubuntu Server 20.04 LTS](#)
 - [Ubuntu Server 22.04 LTS](#)
 - [Red Hat Enterprise Linux 8 \(LVM\)](#)

NOTA: En la presente versión del kit se incluyen preconfiguradas la última versión de bugfix de cada elemento tratado en el momento de la publicación, pudiendo ser publicadas versiones independientes de dichos elementos en fechas posteriores a la publicación del presente kit.

Recuerde revisar/ajustar las versiones de los elementos a desplegar a la última versión de bugfix disponible. Además, es importante aplicar previamente las Release Notes intermedias antes del despliegue para no provocar datos corruptos al tener datos con una versión inferior a la del software.

Recomendaciones

Control de versiones

Para garantizar un control efectivo de las versiones y cambios en nuestros inventarios y templates en el contexto de nuestras operaciones de automatización con Ansible, es crucial que utilicemos un repositorio Git como parte integral de nuestro flujo de trabajo. El uso de un repositorio Git nos permitirá rastrear y gestionar de manera transparente todas las modificaciones, asegurando la trazabilidad de cada cambio a lo largo del tiempo. Esto no solo facilita la colaboración entre miembros del equipo, sino que también proporciona una capa adicional de seguridad al garantizar la reversión a versiones anteriores si es necesario. Por lo tanto, instamos a todos los miembros del equipo a utilizar un repositorio Git para almacenar y gestionar nuestros archivos de inventario y templates de Ansible, manteniendo así un control riguroso de nuestras configuraciones y facilitando una gestión eficiente de versiones y cambios.

Best practices

A parte de esta recomendación puedes encontrar varias “Buenas prácticas” en el uso de Ansible

https://docs.ansible.com/ansible/latest/tips_tricks/ansible_tips_tricks.html#use-version-control

Procedimiento de actualización

No existe ninguna actuación adicional reseñable, solo es necesario seguir lo detallado en apartado del presente documento [Actualización del kit](#)

Requerimientos

Software requerido

Para el uso del kit es necesario tener disponible para instalar desde repositorio de sistema operativo los siguientes paquetes:

- Nodo director (donde se ejecuta el kit)
 - ansible 2.11.12
 - POSTGRESQL-CLIENT 13
 - python-pip
 - python3
 - boto3
 - python-psycopg2
 - python3-psycopg2
 - libselinux-python3
 - lsof
 - acl
 - sudo
 - jq
 - unzip
 - iproute2
 - ldap-utils|openldap-clients
 - OpenJDK 8
 - OpenJDK 11
- Resto de nodos
 - python-pip
 - python3
 - python-psycopg2
 - python3-psycopg2
 - libselinux-python3
 - lsof
 - acl
 - sudo
 - jq
 - iproute2
 - ldap-utils|openldap-clients
 - OpenJDK 8
 - OpenJDK 11
 - Adicional en nodos frontales
 - apache2|httpd
 - Adicional en nodos de persistencia RDBMS (si no se usa servicio externo)
 - POSTGRESQL 13

- libpq-dev or (postgresql13-libs postgresql13-devel)
- libgeos-c1
- postgresql-13-postgis-3.1
- postgresql-13-postgis-scripts
- psycopg2-binary (Repo pip en Amazon Linux 2)
- Adicional en nodos de grafana
 - grafana
 - libnsl.x86_64
- Adicional en nodos de ansible, exportimport y persistenceutility
 - boto3
 - ansible-core==2.11.12

Conectividad requerida

El presente kit, en modos director o remote, descargará todos los recursos necesarios desde el repositorio en servicios centrales Anjana Data por protocolo HTTPS y puerto 80/443 contra el servidor releases.anjanadata.org, por tanto es necesario tener conectividad desde los servidores a provisionar con dicho servicio.

En modo local es necesario descargar el paquete zip de despliegue en modo local, que ha de ser depositado y descomprimido en el nodo director de Ansible desde el cual Ansible hará la distribución e instalación de software al resto de nodos. Solicitar acceso y credencial en cs@anjanadata.com.

- La descarga del paquete está explicado en el apartado [Descargar artefactos de Anjana](#)
- Los archivos descargados son depositados en la carpeta installation.tmpdir del nodo de Ansible, se ha de proceder a indicar el modo "local" y en próximos lanzamientos usará el software/set de datos depositados en dicha carpeta.

```
installation:
mode: local
tmpdir: /tmp/anjana
```

Particularidades SolR

SolR en entornos NO productivos

Para entornos con poca carga de trabajo o de datos no es necesario cumplir la recomendación de 4 Gb de RAM reservada para su uso, sobretodo, teniendo en cuenta que a partir de la versión 23.1 de Anjana los datos de SolR están persistidos en BBDD y se cargan al reindexar.

Para ajustar la RAM de SolR hay que editar la siguiente variable de entorno en el siguiente template:

```
roles > solr > templates > solr.service.j2
12 Environment=ZK_HOST=
13     {%- for server in zookeeper_hosts -%}
14         {{server.ip}}:2181,
15     {%- endfor -%}
16     /solrcloud
17 {% else %}
18     Environment=ZK_HOST=localhost:2181/solrcloud
19 {% endif %}
20 Environment=SOLR_HEAP=4G
21
```

sustituir por :

```
roles > solr > templates > solr.service.j2
12 Environment=ZK_HOST=
13     {%- for server in zookeeper_hosts -%}
14         {{server.ip}}:2181,
15     {%- endfor -%}
16     /solrcloud
17 {% else %}
18     Environment=ZK_HOST=localhost:2181/solrcloud
19 {% endif %}
20 Environment=SOLR_JAVA_MEM="-Xmx1G"
21
```

Particularidades RedHat

Certificados autofirmados httpd en RedHat8 y Amazon Linux 2

Para generar certificados autofirmados debemos de realizar los siguientes pasos.

Instalamos el paquete mod_ssl

```
sudo dnf install mod_ssl
```

Creamos una carpeta en el directorio de httpd, normalmente en /etc/httpd llamada ssl

```
sudo mkdir /etc/httpd/ssl
cd /etc/httpd/ssl
```

Generamos una clave privada, y un CSR , para crear un certificado autofirmado.

```
sudo openssl genrsa -out anjana_private.key 2048
sudo openssl req -new -key anjana_private.key -out anjana_server.csr
```

Para crear el certificado autofirmado ejecutamos el siguiente comando:

```
sudo openssl x509 -req -days 365 -in anjana_server.csr -signkey anjana_private.key -out anjana.crt
```

Una vez tengamos el certificado llamado anjana.crt, debemos de ir a la ruta donde se encuentren los vhosts, en nuestro caso /etc/httpd/conf.d, copiar el archivo anjana.conf y renombrarlo por anjana_ssl.conf

```
cp /etc/httpd/conf.d/anjana.conf /etc/httpd/conf.d/anjana_ssl.conf
```

Editamos el archivo, cambiando el puerto de la primera línea, por el puerto de ssl que queramos, añadimos las líneas que están marcadas en amarillo, y el siguiente bloque de líneas.

```
SSLEngine on
SSLCertificateFile /etc/httpd/ssl/anjana.crt
SSLCertificateKeyFile /etc/httpd/ssl/anjana_private.key
```

```
<VirtualHost *:443>
  ServerName localhost
  DocumentRoot "/opt/htdocs"
  Alias anjana "/opt/htdocs/anjana"
  CustomLog "|/bin/logger -t apache2 -p user.info" "%h %l %u %t \"%r\" %>s %b"

  SSLEngine on
  SSLCertificateFile /etc/httpd/ssl/anjana.crt
  SSLCertificateKeyFile /etc/httpd/ssl/anjana_private_key
```

```
</Directory>

ProxyPreserveHost On
RequestHeader set X-Forwarded-Proto https
RequestHeader set X-Forwarded-Server *
RequestHeader set X-Forwarded-Port 443
```

Una vez cambiado esto, guardamos cambios y reiniciamos httpd. Una vez reiniciado, ya debería de funcionar correctamente por https.

```
systemctl restart httpd
```

También nos podemos asegurar de que funciona, si en el log, nos sale la siguiente línea

```
[root@ip-10-152-20-158 ~]# sudo systemctl restart httpd ; journalctl -fu httpd
-- Logs begin at Mon 2023-06-05 07:02:24 UTC. --
Jun 05 07:02:29 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Jun 05 07:02:29 ip-10-152-20-158.eu-central-1.compute.internal httpd[689]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using ip-10-152-20-158.eu-central-1.compute.internal. Set the 'ServerName' directive globally to suppress this message
Jun 05 07:02:29 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Started The Apache HTTP Server.
Jun 05 07:02:29 ip-10-152-20-158.eu-central-1.compute.internal httpd[689]: Server configured, listening on: port 443, port 80
Jun 05 07:07:17 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Stopping The Apache HTTP Server...
Jun 05 07:07:18 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: httpd.service: Succeeded.
Jun 05 07:07:18 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Stopped The Apache HTTP Server.
Jun 05 07:07:18 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Jun 05 07:07:18 ip-10-152-20-158.eu-central-1.compute.internal systemd[1]: Started The Apache HTTP Server.
Jun 05 07:07:18 ip-10-152-20-158.eu-central-1.compute.internal httpd[4988]: Server configured, listening on: port 443, port 80
```

Apache2 y SELinux

Apache2 necesita SELinux deshabilitado al usar un directorio fuera de la ubicación predeterminada (/var/www/html).

Editamos /etc/selinux/config y ponemos a permissive|disabled: “SELINUX=permissive” o “SELINUX=disabled”

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled  - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum  - Modification of targeted policy. Only selected processes are protected.
#   mls      - Multi Level Security protection.
SELINUXTYPE=targeted
```

También existe la posibilidad de crear y aplicar una política a la carpeta personalizada con los contextos disponibles de apache lanzando los siguientes comandos.

```
sudo yum install policycoreutils-python-utils selinux-policy-devel setroubleshoot-server
sudo setsebool -P httpd_can_network_connect 1
```

Esto hace que el httpd se pueda conectar a internet, aun teniendo el selinux activado

Intérprete de Python

Es necesario que **todos los servidores** a los que conecte Ansible **tengan Python3**.

NOTA: Las imágenes RedHat disponibles en AWS son sin Python3 por defecto por lo que habría que instalarlo previamente en **todos los nodos**.

Arranque y parada

Debido a que el nombre de los microservicios apache2 y postgresql varía para RedHat y pasan a llamarse httpd y postgresql-13 respectivamente, la utilidad de Anjana no funcionará correctamente hasta que se ajusten los nombres.

Estos ajustes se harán en el fichero hosts.yml del inventario que se esté usando, en la variable “service_name”:

```

hosts.yml x
ansible-inventories > localhost > hosts.yml
417 portalhosts:
418   hosts:
419     portall:
420       service_name: apache2 # apache2 for Debian/Ubuntu OS | httpd for RedHat OS/Amazon Linux 2
421       service_port: 80
422       idx: 150
423       ansible_host: localhost
424       ansible_port: 22
425       service_type: microservices
426 solrhosts:

```

```

hosts.yml x
ansible-inventories > localhost > hosts.yml
455 postgresqlhosts:
456   hosts:
457     postgresql1:
458       service_name: postgresql # postgresql for Debian/Ubuntu OS | postgresql-13 for RedHat OS
459       service_port: 5432
460       idx: 20
461       ansible_host: localhost
462       ansible_port: 22
463       service_type: persistences
464       ansible_python_interpreter: python3 # when problems try /usr/bin/python
465 fakesmtserverhosts:

```

Particularidades Amazon Linux 2

Caracteres especiales

En Amazon Linux 2 puede darse lugar a que los caracteres especiales no se muestren correctamente y los objetos se indexan de forma incorrecta tal como se puede ver en la siguiente imagen:



Para solucionar esto antes del despliegue de Anjana sería necesario la creación de un archivo de variables de entorno en caso de no existir o la adición al existente del siguiente contenido con el siguiente comando:

```
sudo vim /etc/environment
```

```
LC_ALL=en_US.UTF-8  
LANG=en_US.UTF-8
```

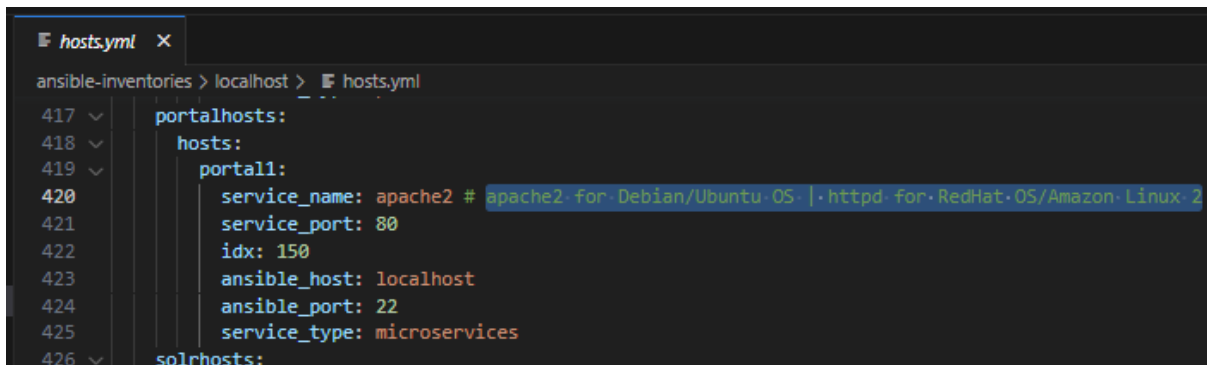
Tras ello sería necesario reiniciar la máquina:

```
sudo reboot
```

Arranque y parada

Debido a que el nombre del microservicios apache2 varía para Amazon Linux 2 y pasa a llamarse httpd, la utilidad de Anjana no funcionará correctamente hasta que se ajusten los nombres.

Estos ajustes se harán en el fichero hosts.yml del inventario que se esté usando, en la variable “service_name”:



```
hosts.yml  
ansible-inventories > localhost > hosts.yml  
417 portalhosts:  
418   hosts:  
419     portal1:  
420       service_name: apache2 # apache2 for Debian/Ubuntu OS | httpd for RedHat OS/Amazon Linux 2  
421       service_port: 80  
422       idx: 150  
423       ansible_host: localhost  
424       ansible_port: 22  
425       service_type: microservices  
426 solrhosts:
```

Información general

Introducción Ansible

¿Qué es un rol?

Los roles son carpetas de tareas que son llamadas desde el playbook en el inventario seleccionado en el comando de ejecución de Ansible. Cada carpeta es una herramienta/software o trabajo distinto.

Las tareas utilizan variables que se inicializan según el inventario escogido para poder diferenciar entre entornos con un solo rol.

¿Qué es un inventario?

Cada carpeta de inventario es un entorno distinto donde se encuentran las variables, hostname y credenciales para desplegar los roles que marque el playbook.

¿Qué es un playbook?

Es una receta entregada a Ansible donde se declaran los roles (tareas y procesos a realizar) y los inventarios (grupos de máquinas) donde se desplegará.

```
├── README.md
├── anjana.yml
├── ansible-inventories
│   ├── k8s
│   ├── localhost
│   └── sample
├── k8s.yml
└── roles
    ├── anjana-utility
    ├── ansible
    ├── apache-exporter-prometheus
    ├── apacheds
    ├── clean
    ├── common
    ├── common-microservices
    ├── drittesta
    ├── edusa
    ├── export
    ├── fakesmtpserver
    ├── grafana
    ├── grafana-agent
    ├── hecate
    ├── hermes
    ├── horus
    ├── import
```

Playbooks disponibles

En el kit se incluye un playbook de ejemplo a usar como plantilla para un despliegue basado en distribución de binarios sobre máquina física o virtual.

anjana.yml → En este yml tenemos lo necesario para que funcione anjana, sin ponerle ninguna característica adicional, sólo está lo imprescindible.

Estructura de los roles

- En esta parte, describiremos la estructura de los roles y las peculiaridades de cada uno de ellos. **Tareas genéricas** → Son tareas que se encuentran en varios roles.
 - Crear directorios de instalación, configuración y backup
 - Establecer los descriptores de servicio sea init o systemd
 - Descargar los artefactos desde el repositorio de Anjana
 - Borrar los artefactos si son antiguos comparando el hash
 - Se establecen los permisos necesarios
 - Para e inicia el servicio

Ajuste de IPs, credenciales y variables requeridos

- **Fichero de nodos** → Variables para la conexión a los distintos nodos
 - <ruta_inventario>/hosts.yml
 - Ajustar credenciales para la conexión a los nodos

```
ansible-inventories > sample > hosts.yml
1  all:
2    vars:
3      ansible_connection: smart
4      ansible_user: <ssh user>
5      ansible_ssh_user: <ssh user>
6      ansible_ssh_private_key_file: /root/.ssh/key.pem
7      force_service_mgr: systemd # systemd or initd
8    children:
9      directorhosts:
10     hosts:
```

- Ajustar IPs o hostname de los nodos de los microservicios


```

ansible-inventories > sample > hosts.yml
55     edusahosts:
56         hosts:
57             edusa1:
58                 service_name: edusa
59                 service_port: 8888
60                 config_profile: native
61                 config_file: configserver
62                 idx: 50
63                 ansible_host: <server_ip>
64                 ansible_port: 22
65                 service_type: microservices
66     hecatehosts:
67         hosts:
68             hecate1:
69                 service_name: hecate
70                 service_port: 50761
71                 config_profile: default
72                 idx: 60
73                 ansible_host: <server_ip>
74                 ansible_port: 22
75                 service_type: microservices
76     viatorhosts:
77         hosts:
78             viator1:
79                 service_name: viator
80                 service_port: 8085
81                 config_profile: default
82                 idx: 110
83                 ansible_host: <server_ip>
84                 ansible_port: 22
85                 service_type: microservices
86     zeushosts:

```

- **Ficheros de variables** → Variables para cada microservicio necesarias para su funcionamiento, las cuales pueden ser encontradas en <ruta_inventario>/group_vars/
 - All.yml:
 - **versión.<microservicio>** → Aquí pondremos la versión que quiere ser desplegada en el servidor de los microservicios.

```

ansible-binary > ansible-inventories > sample > group_vars > all.yml > {} ve
1  #-----#
2  #   Ansible Kit 4.a6   #
3  #-----#
4  version:
5  # MICROSERVICES -----#
6  edusa: 4.5.0
7  hecate: 4.5.0
8  hermes: 4.5.0
9  horus: 4.5.0
10 kerno: 4.5.0
11 minerva: 4.5.0
12 portuno: 4.5.0
13 tot: 4.5.0
14 viator: 4.5.0
15 zeus: 4.5.0
16 portal: 4.5.0
17 admin: 4.5.0
18 drittesta: 4.5.0

```

- **anjana** Permite especificar la carpeta relativa a la raíz del sistema de ficheros donde se despliega el software de Anjana. Con ConfigURL indicamos los servidores de configuración que luego van en el comando de ejecución de los microservicios y en license establecemos la licencia que nos han asignado para el entorno. En hecateReplicas añadimos todos los Hecates que existan en la instalación.

```

ansible-binary > ansible-inventories > sample > group_vars > all.yml > {} installation > {} owner > shell
168 anjana:
169 folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
170 configURL: optional:configserver:http://edusaserver:8888 # HA -> optional:configserver:http://edusaservernode1:8888,
optional:configserver:http://edusaservernode2:8888
171 hecateReplicas: http://hecatserver:50761/eureka # HA -> http://hecatserver:50761/eureka,http://hecatserver2:50761/eureka
172 license:
173 installationCode: AAAAAAAA-InstallationCode-AAAAAAA
174 privateKey: AAAAAAAA-PrivateKey-AAAAAAA
175 anjanaPublicKey: AAAAAAAA-AnjanaPublicKey-AAAAAAA

```

- **etc.hosts** → Si se tratara de un entorno distribuido podemos ajustar aquí los alias e ips que luego irán al archivo /etc/hosts de cada máquina

```

all.yml M X
ansible-binary > ansible-inventories > sample > group_vars > all.yml > ...
85 etc:
86 hosts:
87 - "127.0.0.1 indexservice zkserver s3service s3servicenode1 ndbservice ldapserver"
88 - "127.0.0.1 hecatserver edusaserver zeusservice hermestserver viatorserver viatorservernode1 minervaserver kernoserver totserver
portunoserver"
89 - "127.0.0.1 totpluginactivaserver totpluginawsglueserver totpluginawsiamserver totpluginaws3server totpluginazureadserver
totpluginazurefileserver totpluginazureserver totpluginazureserver totpluginazureserver totpluginazureserver"
90 - "127.0.0.1 totpluginazureserver totpluginazureserver totpluginazureserver totpluginazureserver totpluginazureserver"
totpluginazureserver totpluginazureserver totpluginazureserver totpluginazureserver totpluginazureserver"
91 - "127.0.0.1 temposervice grafanaservice prometheusservice lokiservice"
92 - "127.0.0.1 otelcolservice grafanaagentservice solrextenservice"

```

- **artifactory.user y artifactory.password [NECESARIO]** → Credenciales proporcionadas por el equipo de CS de Anjana para el uso del repositorio de artefactos de Anjana

```

artifactory:
  user: <artifactory_user>
  password: <artifactory_password>

```

- **persistences.s3 | .bbdd | .solr [NECESARIO]** → URL y credenciales para las persistencias. Se usará tanto si se usa el kit para el despliegue como para export, import, backup, scripts de update, ...
 - NOTA: Para el despliegue de Anjana con SqlServer u Oracle hay que ajustar la variable persistences.bbdd.type con el valor correcto ya que esto luego habilitará en los templates de la configuración de los microservicios la parte correcta mediante lógica.
 - NOTA: Mediante comentarios se indican qué variables se usan según para S3 o MinIO y cuales para Postgre, SqlServer u Oracle. Por ejemplo los usuarios y passwords para estos dos últimos son uno por schema, en cambio, para Postgre se usa un solo usuario para todo.

```

ansible-inventories > sample > group_vars > all.yml
90 persistences:
91   s3:
92     type: minio          # minio|aws_s3
93     access_key: anjana
94     secret_key: anjanadata
95     host: http://s3service # Ex. http://s3service or http://10.10.10.10 # MinIO Only
96     port: 9000          # MinIO Only
97     region: <region>    # AWS S3 Only
98     buckets:            # Replace the name attribute value with the custom bucket name
99     - { bucket: cdn, name: cdn }
100    - { bucket: dsa, name: dsa }
101    - { bucket: imports, name: imports }
102    - { bucket: textarea, name: textarea }
103    - { bucket: workflows, name: workflows }
104    - { bucket: datadump, name: datadump }
105    - { bucket: anjanalogs, name: anjanalogs }
106   bdd:
107     type: postgresql    # sqlserver, postgresql or oracle
108     host: rdbservice    # Ex. rdbservice or 10.10.10.10
109     port: 5432          # postgresql --> 5432 | oracle --> 1521 | sqlserver --> 1433
110     database: anjana
111
112     user: anjana        # PostgreSQL Only
113     pass: T0pSecret     # PostgreSQL Only
114
115     user_anjana: anjanakerno # SQLServer and Oracle Only
116     pass_anjana: T0pSecret  # SQLServer and Oracle Only
117     user_hermes: anjanahermes # SQLServer and Oracle Only
118     pass_hermes: T0pSecret  # SQLServer and Oracle Only
119     user_minerva: anjanaminerva # SQLServer and Oracle Only
120     pass_minerva: T0pSecret  # SQLServer and Oracle Only
121     user_portuno: anjanaportuno # SQLServer and Oracle Only
122     pass_portuno: T0pSecret  # SQLServer and Oracle Only
123     user_tot: anjanatot    # SQLServer and Oracle Only
124     pass_tot: T0pSecret    # SQLServer and Oracle Only
125     user_zeus: anjanazeus  # SQLServer and Oracle Only
126     pass_zeus: T0pSecret   # SQLServer and Oracle Only
127
128   solr:
129     host: http://indexservice # Ex. http://indexservice or http://10.10.10.10
130     port: 8983
131     collections:
132     - snapshots
133     - kerno
134     - audit_logs
135     - recommendations

```

- **import_role.<role>** → Con este control booleano podremos desactivar roles en el kit.

Ejemplo: Si usamos AWS S3 y RDS de un cloud tendremos que poner minio y postgresql a false.

- **installation.mode** → Actualmente están contemplados 3 modos de despliegue dentro del kit
 - **Director:** La descarga de artefactos se produce en la máquina que está ejecutando Ansible y los archivos son transferidos a la máquina destino.
 - **Remote:** La descarga la efectúa la máquina destino directamente.
 - **Local:** Los archivos han sido descargados previamente en la máquina que ejecutará ansible, y están localizables en la ruta indicada por la variable **tmpdir**. Solo se efectúa la transferencia de archivos a la máquina destino y el resto de tareas, sin descarga.

IMPORTANTE: El directorio tmpdir es temporal. Todos los datos ubicados en él serán borrados al final de la ejecución.

- **installation.startup** → Es systemd para la mayoría de los casos. Se puede cambiar a none para instalaciones donde no esté disponible el systemd
- **installation.type** → Para que descargue el binario launcher y el jar hay que poner external.
- **installation.javaPATH** → path de la máquina virtual java
- **installation.owner** → cuenta de servicio para la instancia, si no existe la crea automáticamente

```

installation:
  mode: director
  tmpdir: /tmp/anjana # IMPORTANTE: El contenido de esta carpeta es
                    # volátil y será borrado al finalizar las tareas
  eurekaPreferIpAddress: true
  failFast: true # HA -> false
  pathReport: /{{anjana.folder}}/version_report
  javaPATH: /usr/lib/jvm
  # startup: none|systemd
  startup: systemd
  # type: internal|external
  type: external
  owner:
    user: anjana
    uid: 3000
    group: anjana
    gid: 3000
    home: /{{anjana.folder}}
    shell: /bin/bash

```

- **configtemplates.core y configtemplates.plugins** → Las líneas que descomentamos, se añadirán la configuración cuando lancemos edusa o el tag update-anjana-config

```

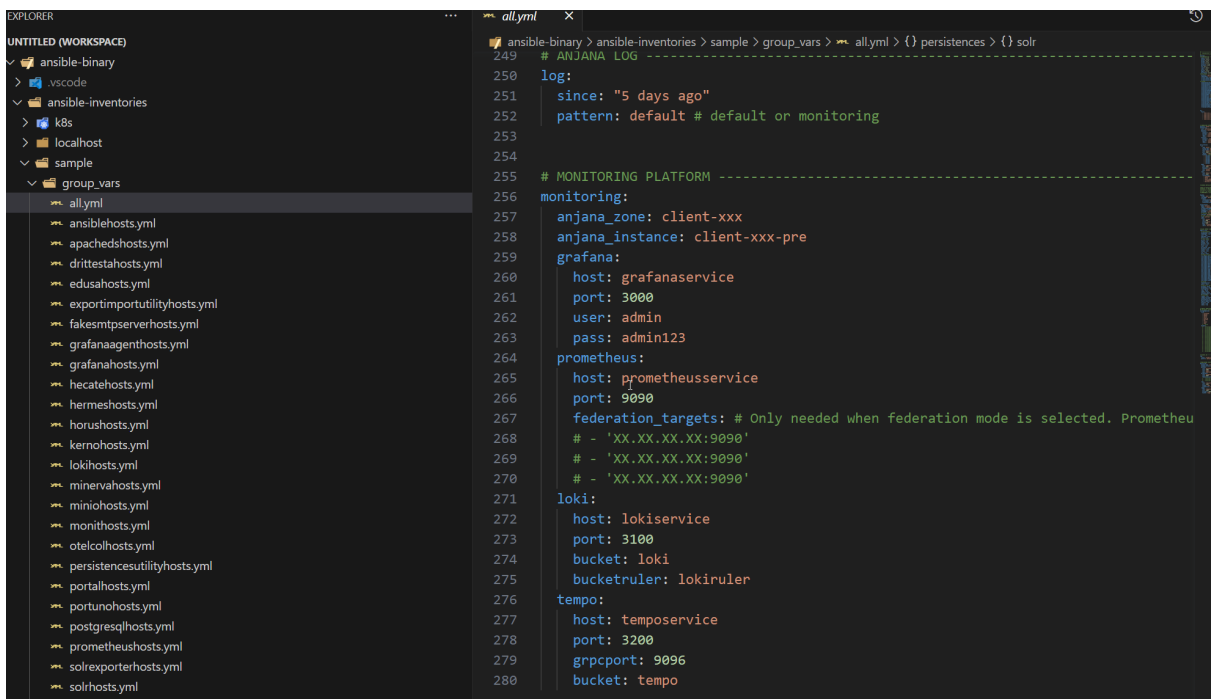
# CONFIG TEMPLATES DEPLOYMENT -----
## Uncomment config as needed
configtemplates:
  core:
    - drittesta
    - hecate
    - hermes
    - kerno
    - minerva
    - portuno
    - tot
    - viator
    - zeus
    # - horus
  plugins:
    # - tot-plugin-aws-glue
    # - tot-plugin-aws-iam
    # - tot-plugin-aws-s3
    # - tot-plugin-azure-ad
    # - tot-plugin-azure-files
    # - tot-plugin-azure-storage
    # - tot-plugin-gcp-bigquery
    # - tot-plugin-gcp-iam
    # - tot-plugin-gcp-storage
    # - tot-plugin-hive
    # - tot-plugin-hdfs
    # - tot-plugin-jdbc
    # - tot-plugin-jdbc-denodo
    # - tot-plugin-jdbc-oracle
    # - tot-plugin-jdbc-redshift
    # - tot-plugin-jdbc-qlserver
    # - tot-plugin-ldap
    # - tot-plugin-aqtiva
    # - tot-plugin-powerbi
    # - tot-plugin-ranger
    # - tot-plugin-tableau

```

- **Log:** A cambiar si queremos que guarde los días que se guarda el log de anjana, y para que el patrón de ese log esté por defecto o adaptado para Grafana y Prometheus.

```
# ANJANA LOG -----
log:
  since: "5 days ago"
  pattern: default # default or monitoring
```

- **Monitoring:** Rellenar todo lo necesario para que al lanzar un tag relacionado con monitorización funcione correctamente el lanzamiento.



```
249 # ANJANA LOG -----
250 log:
251   since: "5 days ago"
252   pattern: default # default or monitoring
253
254
255 # MONITORING PLATFORM -----
256 monitoring:
257   anjana_zone: client-xxx
258   anjana_instance: client-xxx-pre
259   grafana:
260     host: grafanaservice
261     port: 3000
262     user: admin
263     pass: admin123
264   prometheus:
265     host: prometheusservice
266     port: 9090
267     federation_targets: # Only needed when federation mode is selected. Prometheus
268     # - 'XX.XX.XX.XX:9090'
269     # - 'XX.XX.XX.XX:9090'
270     # - 'XX.XX.XX.XX:9090'
271   loki:
272     host: lokiservice
273     port: 3100
274     bucket: loki
275     bucketruler: lokiruler
276   tempo:
277     host: temposervice
278     port: 3200
279     grpcport: 9096
280     bucket: tempo
```

○ **Edusahosts.yml:**

- **config.local.path** → Si está comentado traerá la configuración de git. De tenerlo descomentado usará los archivos configserver-localfiles.yaml
- **config.symboliclink** → Crea un link simbólico de /opt/data/configrepo a /opt/data/minio/configrepo para modificar la configuración desde minio.

```
15 config:
16   local:
17     # UNCOMMENT IF EDUSA USES configserver-localfiles.yaml
18     path: /opt/data/configrepo
19     # If true creates configrepo symbolic link in symboliclink_path to access
20     symboliclink: true
21     symboliclink_path: /opt/data/minio/configrepo
```

Instalación, configuración y mantenimiento del kit

El kit entregado está configurado para utilizar los repositorios centrales de Anjana Data para los cuales es necesario solicitar acceso y credencial en cs@anjanadata.com.

En caso de que se desee usar otro repositorio es necesario realizar los cambios oportunos.

Instalación

Instalación de runtime ansible

Para poder instalar ansible, tenemos que lanzar el siguiente comando en

Ubuntu 18.04|20.04|22.04:

```
sudo apt update -y
sudo apt install python3-pip -y
sudo pip3 install ansible-core==2.11.12

sudo wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/community-general-7.4.0.tar.gz

sudo wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/community-postgresql-3.2.0.tar.gz

sudo ansible-galaxy collection install ./community-general-7.4.0.tar.gz -p
/usr/share/ansible/collections
sudo ansible-galaxy collection install ./community-postgresql-3.2.0.tar.gz -p
/usr/share/ansible/collections
```

RedHat 8 | Amazon Linux 2:

```
# Install Ansible
sudo yum update -y
sudo yum install python3-pip -y
sudo pip3 install ansible-core==2.11.12

# Add binary to PATH
sudo ln -s /usr/local/bin/ansible /usr/bin/ansible
sudo ln -s /usr/local/bin/ansible-galaxy /usr/bin/ansible-galaxy
sudo ln -s /usr/local/bin/ansible-playbook /usr/bin/ansible-playbook

sudo yum install wget -y

sudo wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/community-general-7.4.0.tar.gz

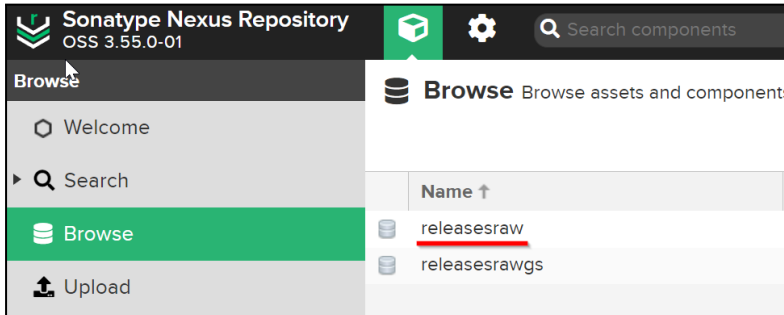
sudo wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/community-postgresql-3.2.0.tar.gz

sudo ansible-galaxy collection install ./community-general-7.4.0.tar.gz -p
/usr/share/ansible/collections
sudo ansible-galaxy collection install ./community-postgresql-3.2.0.tar.gz -p
/usr/share/ansible/collections
```

También instalará Python3 como dependencia de Ansible.

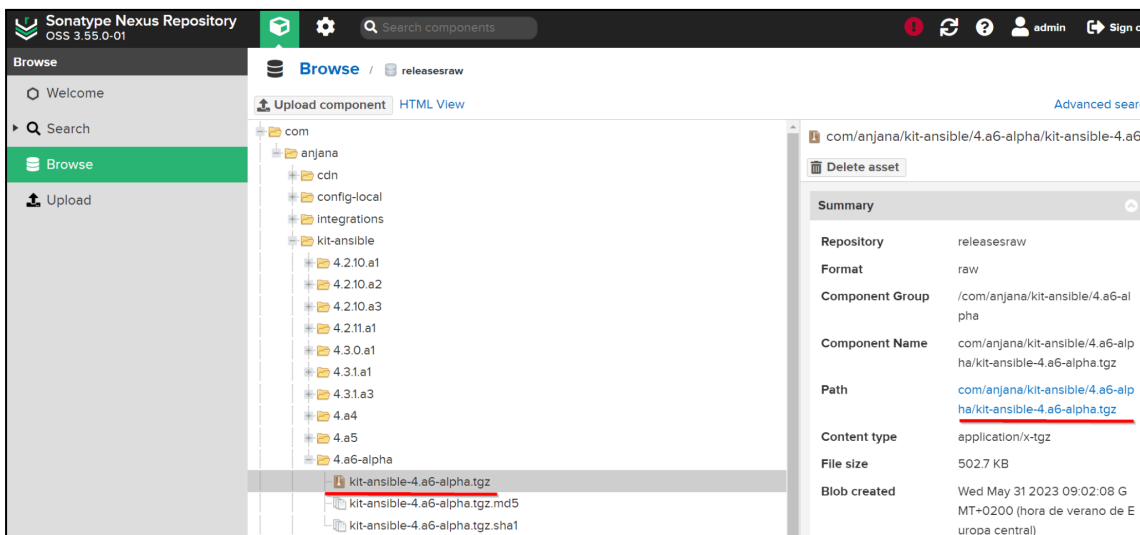
Descarga y despliegue

Accedemos al repositorio de artefactos en la url: <https://releases.anjanadata.org/>, iniciamos sesión y entramos al repositorio RAW “releasesraw”.



Desplegamos hasta llegar al kit de Ansible en la versión deseada (4.a6 a la fecha de la publicación de esta documentación) :

- `com/anjana/kit-ansible/4.a6` pulsamos y en el panel derecho nos aparece el link de descarga



Una vez descargado lo subimos al nodo de Ansible (servidor desde el cual se va a efectuar el despliegue, si es un single-node puede ser el mismo en localhost) y descomprimos:

```
sudo mkdir <ruta_anjana>/ansible # normalmente /opt/ansible

sudo wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/4.a6/kit-ansible-4.a6.
tgz

sudo tar -zxpvf kit-ansible-4.a6.tgz -C <ruta_anjana>/ansible

# ajustamos variables necesarias para un primer despliegue en all.yml y ejecutamos el tag
# "adjust-ansible-folder" para cerrar con permisos y owners correctos la carpeta de Ansible
# folders 770
# files 660
# owners anjana:anjana

# Para ejecutar en localhost el tag siguiente hay que poner el usuario y password en el inventario
localhost pero si luego se va a usar otro inventario hay que añadir usuario y password de nuevo en
el inventario correspondiente
```

```
sudo vim /opt/ansible/ansible-inventories/localhost/group_vars/all.yml
```

```
60
61 # CONNECTION STRINGS -----
62 nexusurl: https://releases.anjanadata.org
63 artifactory:
64   user: <artifactory_user>
65   password: <artifactory_password>
66   internal:
```

y ya podremos lanzar el tag de ajuste de permisos, owner, ...

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags adjust-ansible-folder
```

Añadir el usuario del técnico al grupo Anjana para editar el kit

Ej. sudo usermod -a -G anjana ec2-user

```
sudo usermod -a -G anjana <user-tecnico>
```

```
-rw-rw---- 1 anjana anjana 2768 Mar  9 12:39 README.md
-rw-rw---- 1 anjana anjana 14814 Mar  9 12:39 anjana.yml
drwxrwx--- 5 anjana anjana 4096 Mar  9 12:39 ansible-inventories/
-rw-rw---- 1 anjana anjana 172 Mar  9 12:39 k8s.yml
drwxrwx--- 57 anjana anjana 4096 Mar  9 12:39 roles/
root@ip-10-152-10-240: /opt#
```

Una vez tenemos el kit en el nodo pasamos a ajustar todo lo necesario para poder desplegar Anjana. Esto podemos hacerlo siguiendo el siguiente apartado: [Desplegar Anjana](#)

Actualización kit Ansible

Respaldo de configuración

Debido al gran cambio en la estructura del kit para mejorar la seguridad, el uso, la personalización y futuras actualizaciones, este upgrade no puede ser incremental o parcial. Por este motivo lo primero que debemos hacer es mover la carpeta actual de Ansible para ejecutar el tag ansible y que no se sobrescriba.

```
sudo mv /opt/ansible/ /opt/ansible-old
```

Ajuste de versión + Descarga

Para proceder a la actualización primero debemos seleccionar la versión más reciente disponible en el archivo de versiones all.yml ubicado en la carpeta de inventario que estuviéramos usando.

Ejemplo /opt/ansible-old/ansible-inventories/localhost/group_vars/all.yml.

Para ello editamos el fichero y sustituimos la versión existente por la versión a la cual se quiere actualizar.


```
ansible-binary > ansible-inventories > sample > group_vars > all.yml
1  #-----#
2  #   Ansible Kit 4.a6   #
3  #-----#
4  version:
5  # MICROSERVICES -----
6  |   edusa: 4.4.0
7  |   hecate: 4.4.0
8  |   horus: 4.4.0
9  |   hermes: 4.4.1
10 |   kerno: 4.4.3
11 |   minerva: 4.4.0
12 |   portuno: 4.4.2
13 |   tot: 4.4.1
14 |   viator: 4.4.0
15 |   zeus: 4.4.2
16 |   portal: 4.4.3
17 |   admin: 4.4.0
18 |   drittesta: 4.4.0
19 # UTILITIES -----
20 |   ansible: 4.a6
21 |   compiledat: 4.4 sample 05-04-2022
```

A continuación lanzamos el comando para actualizar el kit de Ansible (es necesario lanzarlo en el nodo que contenga Ansible en caso de ser una instalación multinodo).

```
sudo ansible-playbook -i /opt/ansible-old/ansible-inventories/anjana_xxx/hosts.yml
/opt/ansible-old/anjana_xxx.yml --tags ansible
```

En cuanto se haya completado ya dispondremos del kit en la versión seleccionada en la carpeta original (normalmente /opt/ansible).

Actualización kit Ansible

Una vez tenemos un backup del kit anterior y el kit nuevo en la carpeta adecuada debemos proceder a transferir las personalizaciones del kit antiguo al actualizado.

NOTA: Para abrir el archivo del inventario/kit nuevo y del backup/antiguo podemos ayudarnos del comando:

```
# Abrir vim con 2 paneles en vertical que permite ir editando y viendo a la vez en terminal
# para ir cambiando de panel hay que presionar "CTRL+w"
sudo vim -O /opt/ansible/.../all.yml /opt/ansible-old/.../all.yml
```

```

#-----#
# Ansible Kit 4.a5 #
#-----#
version:
# MICROSERVICES -----#
edusa: 4.4.0
hecate: 4.4.0
horus: 4.4.0
hermes: 4.4.1
kerno: 4.4.3
minerva: 4.4.0
portuno: 4.4.2
tot: 4.4.1
viator: 4.4.0
zeus: 4.4.2
portal: 4.4.3
admin: 4.4.0
drittesta: 4.4.0
# UTILITIES -----#
ansible: 4.a5
sampledata: 4.4-sample-05_04_2023
script reset sequences: 4.4
<ansible-inventories/localhost/group vars/all.yml 5,1 Top
all.yml" [dos] 295L, 9599C

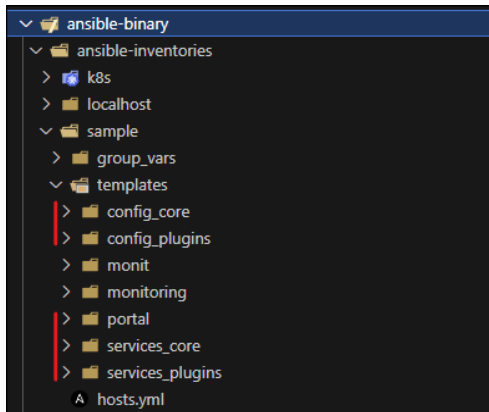
#-----#
# Ansible Kit 4.a6 #
#-----#
version:
# MICROSERVICES -----#
edusa: 4.5.0
hecate: 4.5.0
horus: 4.5.0
hermes: 4.5.0
kerno: 4.5.0
minerva: 4.5.0
portuno: 4.5.0
tot: 4.5.0
viator: 4.5.0
zeus: 4.5.0
portal: 4.5.0
admin: 4.5.0
drittesta: 4.5.0
# UTILITIES -----#
ansible: 4.a6
sampledata: 4.5-sample
script reset sequences: 4.4
all.yml 6,3 Top

```

Con "-d" se pueden ver las diferencias entre los archivos pero es más difícil de interpretar
vim -d /opt/ansible/.../all.yml /opt/ansible-old/.../all.yml

Puntos a revisar:

- all.yml
 - Ajustar versiones de producto
 - Alias personalizados en "etc.hosts:"
 - Credenciales repositorio artefactos
 - En el nuevo apartado "persistences.s3|bbdd|solr:" debemos ajustar hosts, puertos, credenciales, buckets, ... de las persistencias del producto. Esta info es usada para updates, inserts, deletes, reset, ... y es **imprescindible**
 - El resto de propiedades ya existían en la versión anterior y hay que ajustarlas si tuvieron alguna personalización
- templates configuración
 - Se han centralizado los templates de las configuraciones de los microservicios de Anjana en la carpeta del inventario/templates /config_XXX. Habrá que ajustar las personalizaciones que se hicieran en el kit anterior.
- templates systemd services
 - Se han centralizado los templates de los descriptores de servicio de Anjana en la carpeta del inventario/templates /services_XXX/systemd . Habrá que ajustar las personalizaciones que se hicieran en el kit anterior.
- templates virtual hosts de portal
 - Se han centralizado los templates de los virtualhosts del front de Anjana en la carpeta del inventario/templates /portal. Habrá que ajustar las personalizaciones que se hicieran en el kit anterior.

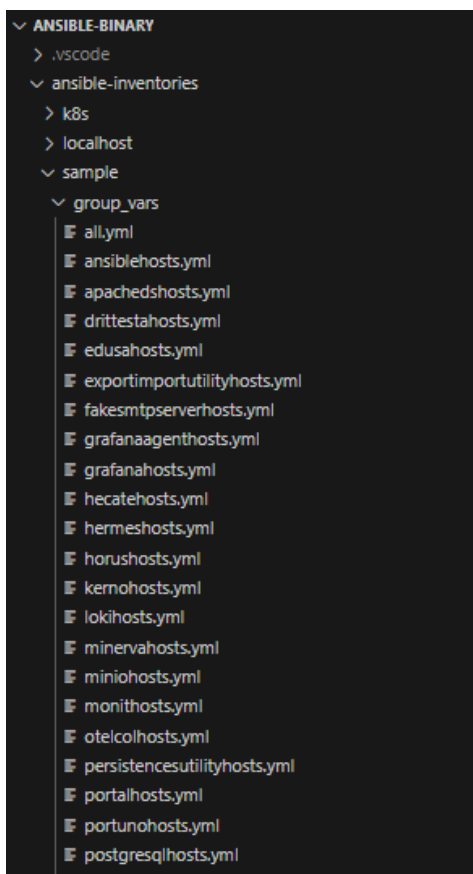


Ajustar ips y credenciales de inventarios y variables de roles

Es necesario volver a rellenar los archivos hosts, con las ips y credenciales de los nodos donde vamos a desplegar y las variables que fueran sustituidas en los archivos de variables de los roles en group_vars.

Se añaden un par de grupos que hay que tener en cuenta

- tempohosts y otelcolhosts-> nodos de software necesario para monitorización del log



El no revisar y portar los cambios de los inventarios personalizados a los archivos actualizados puede causar que el kit no funcione correctamente.

Logs

Para el registro de la actividad del kit existe un fichero ubicado en `/opt/version_report` por defecto. La ubicación puede ser alterada mediante una variable que podemos encontrar en el archivo `all.yml` dentro del inventario que estemos utilizando.

```
ansible-binary > ansible-inventories > sample > group_vars > all.yml
177 installation:
178   mode: director
179   tmpdir: /tmp/anjana # IMPORTANTE: El contenido de esta carpeta
    es volátil y será borrado al finalizar las tareas
180   eurekaPreferIpAddress: true
181   failFast: true # HA -> false
182   pathReport: {{anjana.folder}}/version_report
183   javaPATH: /usr/lib/jvm
```

En el archivo se registran las acciones de los tags que se ejecutan, imagen a continuación:

```
root@ANSIBLE-ControlNode:/opt/ansible# cat /opt/version_report
-----
STARTING DEPLOYMENT
-----
START DATE '2022-10-20T11:10:14Z'
-----
HOST '10.150.100.94'
-----
OK - User & Group
OK - Alias in /etc/hosts
OK - Packages // Dependencies
OK - Artifactory connection check
OK - Tmpdir available
-----
OK - OpenJDK 8 available
-----
STARTING EXPORT TASKS
-----
2022-10-20T11:10:23Z --EXPORTED-- MinIO Buckets to /opt/export-import/minio/anjana_minio.tgz
2022-10-20T11:10:23Z --EXPORTED-- BBDD to /opt/export-import/bbdd/anjana_postgresql.sql
2022-10-20T11:10:23Z --EXPORTED-- Solr Collections to /opt/export-import/solr
-----
2022-10-20T11:10:23Z --UPLOADED-- exports to Bucket datadump
-----
STARTING IMPORT TASKS
-----
2022-10-20T11:10:24Z --IMPORTED-- MinIO Buckets from /opt/export-import/minio/anjana_minio.tgz
2022-10-20T11:10:24Z --IMPORTED-- BBDD from /opt/export-import/bbdd/anjana_postgresql.sql
2022-10-20T11:10:24Z --IMPORTED-- Solr Collections from /opt/export-import/solr
-----
```

Despliegue y mantenimiento de instancia de Anjana Data

Despliegue de nueva instancia

Generación de playbook

Se surte un playbook de ejemplo el cual aplica los roles a los diferentes hosts definidos en el inventario (a un host se pueden aplicar varios roles o incluso la totalidad).

El playbook de ejemplo /opt/ansible/anjana.yml será sobrescrito por el kit al actualizar el propio kit, por tanto ha de usarse como base para generar su propio playbook.

Generación de inventario de hosts

Es necesario generar un fichero de inventario acorde a la infraestructura en la que se ha de desplegar el aplicativo, para ello se puede usar como base una copia del inventario sample que viene incluido en el kit y renombrarlo para evitar pérdidas de configuración.

Para un single-node se puede usar el inventario “localhost” o para un entorno distribuido se puede usar “sample” y cambiarle el nombre para identificarlo mejor, por ejemplo “pro”

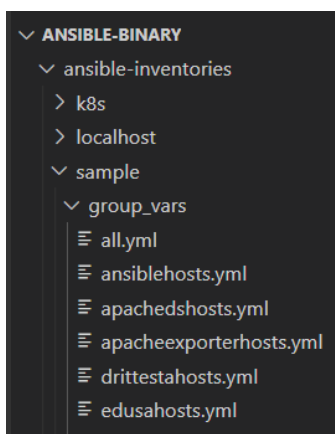
```
#En un single-node
cp -r /opt/ansible/ansible-inventories/localhost /opt/ansible/ansible-inventories/pro

#En un nodo de Ansible donde el despliegue sea en otra u otras máquinas
cp -r /opt/ansible/ansible-inventories/sample /opt/ansible/ansible-inventories/pro
```

Los inventarios de ejemplo sample y localhost serán sobrescritos por el kit al actualizar el propio kit, por tanto han de usarse como base para generar su propios inventarios.

Ajuste de variables

Es imprescindible ajustar las variables contenidas en el fichero global de variables del inventario (ansible-inventories/xxxx/group_vars/all.yml), adicionalmente se pueden ajustar resto de ficheros de variables, los cuales están segmentados por rol (ansible-inventories/xxxx/group_vars/<role>hosts.yml).



El detalle de las variables, incluso las que son necesarias de ajustar antes de ejecutar, se encuentra en el apartado: [Ajuste de IPs, credenciales y variables requeridos](#)

Ejecución del playbook

Para ello, necesitaremos una máquina con Ansible instalado en la versión 2.11.12 (se explica más arriba como instalarlo). En la máquina lanzaremos los siguientes comandos.

NOTA: ES NECESARIO SER LANZADO CON SUDO O CON USUARIO ROOT

NOTA: ES NECESARIO QUE TODAS LAS MÁQUINAS TENGAN PYTHON3 (RedHat en AWS es desplegado sin Python3 por lo que habría que instalarlo previamente en todos los nodos)

```
sudo ansible-playbook -i <inventario_lanzamiento> <playbook_lanzamiento>

# EJEMPLO DESPLIEGUE ANJANA SIN DATOS DE EJEMPLO
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml
-t update-anjana-config
# Con este despliegue se levanta Anjana sin datos pero si que faltaría desplegar la configuración
de los microservicios. Para hacerlo cogiendo los templates sería con:

# EJEMPLO DESPLIEGUE ANJANA CON DATOS DE EJEMPLO EN PERSISTENCIA
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml
-t anjana-sample
```

Los roles disponen de tags para restringir la ejecución a un determinado grupo, caso típico la ejecución de tareas de un solo microservicio para su despliegue:.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml
--tags edusa
```

NOTA: LA CARPETA DE INVENTARIOS SE DEBE LLAMAR "ansible-inventories"

Actualización de instancia

Actualización de Anjana

Permite ejecutar sólo las tareas de update de los launcher, los jar de los microservicios y el front de Anjana.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml
/opt/ansible/anjana.yml --tags update
```

También es posible actualizar los artefactos de forma individual usando el tag específico para cada microservicio

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml
/opt/ansible/anjana.yml --tags update-edusa
```

Actualización de descriptores de servicios

Si se realiza un cambio en los templates de los descriptores de servicio y queremos desplegarlo para hacerlo efectivo, podemos lanzar el siguiente comando

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml  
/opt/ansible/anjana.yml --tags update-anjana-services
```

Actualización de vhosts de webserver

Si se realiza un cambio en los templates de los virtual hosts de Apache2 y queremos desplegarlo para hacerlo efectivo, lanzaremos el siguiente comando

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml  
/opt/ansible/anjana.yml --tags update-apache-vhosts
```

Instalar/Actualizar instancia desde dentro de la propia máquina

El rol Ansible deja una copia local del kit de Ansible y además crea un script `/opt/ansible/actualizacion.sh` que solo lanza las tareas de actualización de artefactos (solo microservicios Anjana)

Actualización de configuración de Anjana

Ejecutando el kit con el tag “update-anjana-config” podremos actualizar/desplegar la configuración de los microservicios.

Los templates de configuración los encontramos dentro del inventario, en una carpeta llamada `templates/config_core` si es del core de anjana, o `templates/config_plugins` si es la configuración de los plugins.

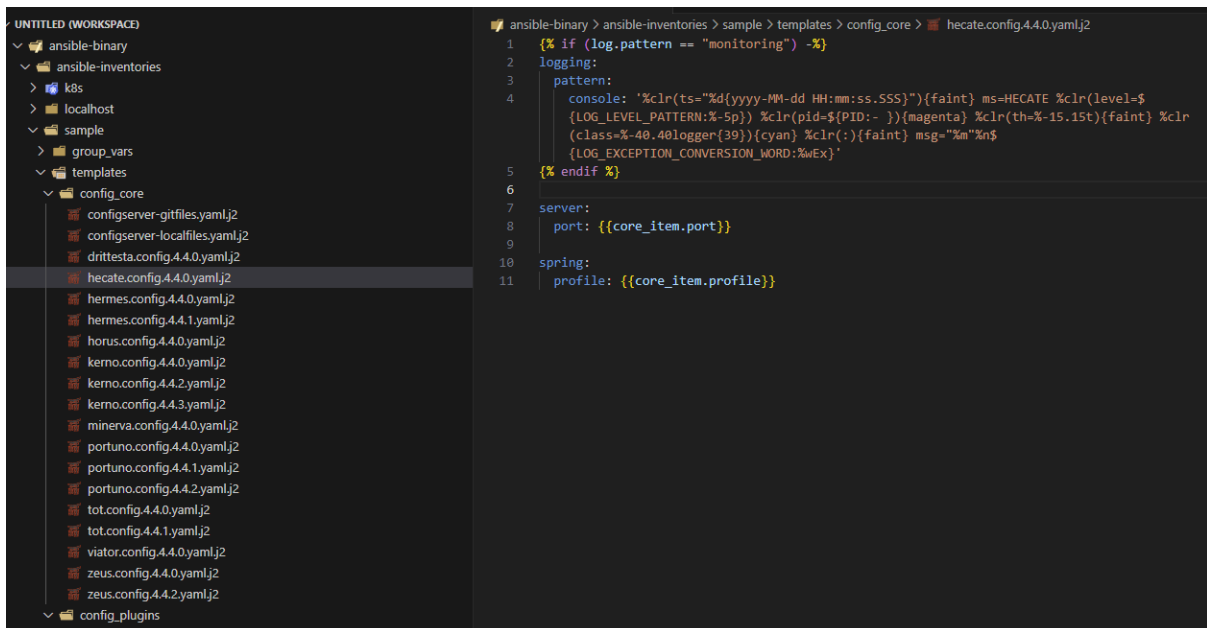
IMPORTANTE: Lanzando este tag, se actualizarán TODOS los archivos de configuración de los microservicios.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml  
/opt/ansible/anjana.yml --tags update-anjana-config
```

Para editar la configuración, iremos a la ruta `<ruta_inventario>/templates/config_core` o `<ruta_inventario>/templates/config_plugins`, en la cual encontraremos varios archivos con la nomenclatura `<microservicio>.config.yaml.j2`, los cuales son los archivos que tenemos que editar para cambiar los archivos de configuración.



Si entramos en alguno, veremos que es el archivo de configuración tal cual.



Una vez reemplazada la configuración que queremos, vamos a ir al archivo all.yml ubicado en nuestro inventario, y descomentamos en este caso, el plugin de jdbc para que lo despliegue.


```

UNTITLED (WORKSPACE)
└─ ansible-binary
   └─ ansible-inventories
      └─ k8s
         └─ localhost
            └─ sample
               └─ group_vars
                  └─ all.yml
                     A ansiblehosts.yml
                     A apachedshosts.yml
                     A drittestahosts.yml
                     A edusahosts.yml
                     A exportimportutilityhosts.yml
                     A fakesmtpserverhosts.yml
                     A grafanaagenthosts.yml
                     A grafanahosts.yml
                     A hecatehosts.yml
                     A hermeshosts.yml
                     A horushosts.yml
                     A kernohosts.yml
                     A lokihosts.yml
                     A minervahosts.yml
                     A miniohosts.yml
                     A monithosts.yml
                     A otelcolhosts.yml
                     A persistencesutilityhosts.yml
                     A portalhosts.yml
                     A portunohosts.yml
                     A postgresqlhosts.yml
                     A prometheushosts.yml
                     A solrexporterhosts.yml
                     A solrhosts.yml
                     A tempohosts.yml
                     A tothosts.yml

ansible-binary > ansible-inventories > sample > group_vars > A all.yml
203
204 # CONFIG TEMPLATES DEPLOYMENT -----
205 ## Uncomment config as needed
206 configtemplates:
207   core:
208     - drittesta
209     - hecate
210     - hermes
211     - kerno
212     - minerva
213     - portuno
214     - tot
215     - viator
216     - zeus
217     # - horus
218   plugins:
219     # - tot-plugin-aws-glue
220     # - tot-plugin-aws-iam
221     # - tot-plugin-aws-s3
222     # - tot-plugin-azure-ad
223     # - tot-plugin-azure-files
224     # - tot-plugin-azure-storage
225     # - tot-plugin-gcp-bigquery
226     # - tot-plugin-gcp-iam
227     # - tot-plugin-gcp-storage
228     # - tot-plugin-hive
229     # - tot-plugin-hdfs
230     # - tot-plugin-jdbc
231     # - tot-plugin-jdbc-denodo
232     # - tot-plugin-jdbc-oracle
233     # - tot-plugin-jdbc-redshift
234     # - tot-plugin-jdbc-sqlserver
235     # - tot-plugin-ldap
236     # - tot-plugin-aqtiva
237     # - tot-plugin-powerbi
238     # - tot-plugin-ranger
239     # - tot-plugin-tableau
240

```

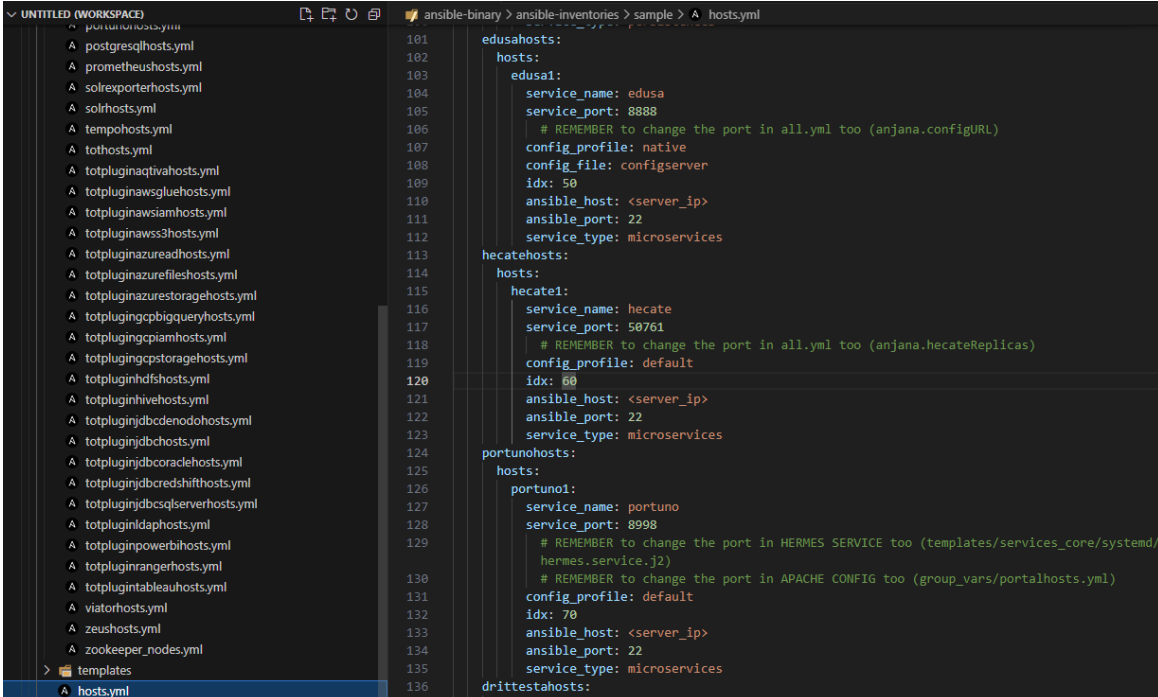
En el caso de arriba, lanzando el tag `update-anjana-config`, actualizará la configuración de todas las líneas descomentadas.

Mantenimiento de instancia

Arranque, parada y reinicio

Esta utilidad permite el arranque y parada de los microservicios de forma selectiva y ordenada para una instalación existente, de esta manera se evitan los errores derivados de la detención de un microservicio sin haber tenido en cuenta sus dependencias.

Esta utilidad tiene en cuenta las variables adicionales descritas en el inventario que hacen referencia al nombre del servicio, puerto, perfil de configuración, número en el índice de arranque, etc... Un ejemplo de uso sería la parada del backend para hacer un upgrade de Anjana.



```

edusahosts:
  hosts:
    edusa1:
      service_name: edusa
      service_port: 8888
      # REMEMBER to change the port in all.yml too (anjana.configURL)
      config_profile: native
      config_file: configserver
      idx: 50
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: microservices
  hecatehosts:
    hosts:
      hecate1:
        service_name: hecate
        service_port: 50761
        # REMEMBER to change the port in all.yml too (anjana.hecateReplicas)
        config_profile: default
        idx: 60
        ansible_host: <server_ip>
        ansible_port: 22
        service_type: microservices
  portunohosts:
    hosts:
      portuno1:
        service_name: portuno
        service_port: 8998
        # REMEMBER to change the port in HERMES SERVICE too (templates/services_core/systemd/
        # hermes.service.j2)
        # REMEMBER to change the port in APACHE CONFIG too (group_vars/portalhosts.yml)
        config_profile: default
        idx: 70
        ansible_host: <server_ip>
        ansible_port: 22
        service_type: microservices
  drittestahosts:
    hosts:

```

NOTA: La utilidad es compatible con múltiples instancias de un mismo microservicio (EJ: edusa1, edusa2, kerno1, kerno2, etc) pero no puede detener las instancias duplicadas de forma individual. Seleccionar la parada de kerno detendrá kerno1 y kerno2 respectivamente.

NOTA: Para RedHat hay que revisar y ajustar el nombre de los servicios, como apache2 (httpd) y postgresql (postgresql-13) para que la utilidad funcione correctamente.

La funcionalidad provista por esta utilidad es la siguiente:

- **La detención y arranque ordenado de todo anjana.** Se produce en el orden adecuado para evitar excepciones no intencionadas.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags stop-anjana
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags start-anjana
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags restart-anjana
```

```

root@anjana22: /mnt/d/Repo
skipping: [ansible] => (item=[{"host": "portal", "service": "apache2", "port": 80, "idx": 14, {"hostname": "export1", "ip": "10.150.100.147"}}])
skipping: [ansible] => (item=[{"host": "portal", "service": "apache2", "port": 80, "idx": 14, {"hostname": "import1", "ip": "10.150.100.147"}}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "zookeeper", "service": "zooke
"ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "postgresql", "service": "postg
1", "ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "minio", "service": "minio",
50.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "solr", "service": "solr", "p
100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "edusa", "service": "edusa",
50.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "hecate", "service": "hecate",
10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "portuno", "service": "portun
"ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "zeus", "service": "zeus", "p
100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "kerno", "service": "kerno",
50.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "minerva", "service": "minerv
"ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "viator", "service": "viator",
10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "hermes", "service": "hermes",
10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "tot", "service": "tot", "port
00.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "portal", "service": "apache2
0.150.100.147"}])

TASK [anjana-utility : Starting [zookeeper] service in [zookeeper1] host if not running] *****
ok: [ansible]

TASK [anjana-utility : Checking if [zookeeper] service is started in [zookeeper1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [postgresql] service in [postgresql] host if not running] *****
ok: [ansible]

TASK [anjana-utility : Checking if [postgresql] service is started in [postgresql] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [minio] service in [minio1] host if not running] *****
ok: [ansible]

TASK [anjana-utility : Checking if [minio] service is started in [minio1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [solr] service in [solr1] host if not running] *****
ok: [ansible]

```

- **Detención y arranque selectivo de un microservicio concreto.** Se arrancarán todos los microservicios necesarios para que el seleccionado funcione. En caso de la parada se detendrán todos los microservicios que dependen del seleccionado de forma previa a la detención.

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags stop-kerno

```

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags start-kerno

```

```

root@anjana22: /mnt/d/Repo
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "zookeeper", "
"ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "postgresql", "
1", "ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "minio", "serv
50.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "solr", "servi
100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "edusa", "serv
50.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "hecate", "ser
10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "portuno", "se
"ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "zeus", "sevi
100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible => (item=[{"host": "kerno", "serv
50.100.147"}])

TASK [anjana-utility : Starting [zookeeper] service in [zookeeper1] host if not running] *****
changed: [ansible]

TASK [anjana-utility : Checking if [zookeeper] service is started in [zookeeper1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [postgresql] service in [postgresql] host if not running] *****
changed: [ansible]

TASK [anjana-utility : Checking if [postgresql] service is started in [postgresql] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [minio] service in [minio1] host if not running] *****
changed: [ansible]

TASK [anjana-utility : Checking if [minio] service is started in [minio1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [solr] service in [solr1] host if not running] *****
changed: [ansible]

TASK [anjana-utility : Checking if [solr] service is started in [solr1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [edusa] service in [edusa1] host if not running] *****
changed: [ansible]

TASK [anjana-utility : Checking if [edusa] service is started in [edusa1] host] *****
ok: [ansible]

TASK [anjana-utility : Starting [hecate] service in [hecate1] host if not running] *****
changed: [ansible]

```

Backup de los datos

Permite hacer backup en la ruta indicada en la variable backup de cada role (bbdd, jar, launcher, cdn, etc):

NOTA: Ajustar las colecciones en el archivo all.yml

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml --tags backup
```

Exportación e importación de datos

Es posible exportar e importar las persistencias para una transferencia entre máquinas/entornos con el role export-import-utility.

NOTA: En el archivo group_vars/all.yml del inventario usado hay que poner "installation.mode: director".

Por defecto, las rutas seleccionadas para la exportación e importación de datos son las que se muestran a continuación, pudiendo ser alteradas en el archivo:

ansible-inventories/sample/group_vars/exportimportutilityhosts.yml

```
ansible-binary > ansible-inventories > sample > group_vars > exportimportutilityhosts.yml
1  ---
2  export_import:
3    path: /{{anjana.folder}}/export-import
4    utility: '{{installation.tmpdir}}/export-import-utility'
5  ei_minio:
6    datapath: '{{minio.datapath}}'
7    path: /{{anjana.folder}}/minio
8    exp_tmpdir: '{{installation.tmpdir}}/minio_export'
9    export_import_path: '{{export_import.path}}/minio'
10   artifact_url: /minio/mc
11   imp_tmpdir: '{{installation.tmpdir}}/minio_import'
12  ei_aws_s3:
13   datapath: /{{anjana.folder}}/data/aws_s3
14   path: /{{anjana.folder}}/aws_s3
15   exp_tmpdir: '{{installation.tmpdir}}/aws_s3_export'
16   imp_tmpdir: '{{installation.tmpdir}}/aws_s3_export'
17   export_import_path: '{{export_import.path}}/aws_s3'
18  ei_bbdd:
19   export_import_path: '{{export_import.path}}/bbdd'
20   artifact_url: /postgresql/
21  ei_solr:
22   path: /{{anjana.folder}}/solr
23   export_import_path: '{{export_import.path}}/solr'
24   artifact_url: /solr/solr-utility/
25   artifact: solr-utility.tgz
```

Las cadenas de conexión como puertos, host, contraseñas, tipo de cliente S3, entre otros, para bases de datos, MinIO y Solr pueden ser localizados y editados en el archivo all.yml del inventario:

```

ansible-binary > ansible-inventories > sample > group_vars > A all.yml
112 persistences:
113   s3:
114     type: minio           # minio|aws_s3
115     access_key: anjana
116     secret_key: anjanadata
117     host: http://s3service # Ex. http://s3service or http://10.10.10.10 # MinIO Only
118     port: 9000           # MinIO Only
119     region: <region>    # AWS S3 Only
120     buckets:            # Replace the name attribute value with the custom bucket name
121     - { bucket: cdn, name: cdn }
122     - { bucket: dsa, name: dsa }
123     - { bucket: imports, name: imports }
124     - { bucket: textarea, name: textarea }
125     - { bucket: workflows, name: workflows }
126     - { bucket: datadump, name: datadump }
127     - { bucket: anjanalogs, name: anjanalogs }
128   bbdd:
129     type: postgresql # sqlserver, postgresql or oracle
130     host: rdbservice # Ex. rdbservice or 10.10.10.10
131     port: 5432       # postgresql --> 5432 | oracle --> 1521 | sqlserver --> 1433
132     database: anjana
133
134     user: anjana     # PostgreSQL Only
135     pass: anjana     # PostgreSQL Only
136
137     user_anjana: anjanakerno # SQLServer and Oracle Only
138     pass_anjana: T0pSecret   # SQLServer and Oracle Only
139     user_hermes: anjanahermes # SQLServer and Oracle Only
140     pass_hermes: T0pSecret   # SQLServer and Oracle Only
141     user_minerva: anjanaminerva # SQLServer and Oracle Only
142     pass_minerva: T0pSecret   # SQLServer and Oracle Only
143     user_portuno: anjanaportuno # SQLServer and Oracle Only
144     pass_portuno: T0pSecret   # SQLServer and Oracle Only
145     user_tot: anjanatot     # SQLServer and Oracle Only
146     pass_tot: T0pSecret     # SQLServer and Oracle Only
147     user_zeus: anjanazeus   # SQLServer and Oracle Only
148     pass_zeus: T0pSecret    # SQLServer and Oracle Only
149
150   solr:
151     host: http://indexservice # Ex. http://indexservice or http://10.10.10.10
152     port: 8983
153     collections:
154     - snapshots
155     - kerno
156     - audit_logs
157     - recommendations

```

NOTA: Hay que tener en cuenta que los archivos generados serán depositados en el nodo director, donde se estará ejecutando ansible.

Para el uso de esta utilidad solo hay que ejecutar el siguiente comando y todas las persistencias quedarán respaldadas en el directorio elegido:

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml -v --tags export

```

De igual manera para su importación ejecutamos la misma línea alterando el tag :

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml -v --tags import
```

A su vez, se puede lanzar cada import por separado con los siguientes tags:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml -v --tags import-s3

sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml -v --tags import-solr

sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml -v --tags import-bbdd
```

NOTA: En el caso de S3 los buckets tienen que estar creados previamente antes de lanzar la importación. De igual manera para las colecciones de SolR, tienen que estar ya creadas vacías previamente, la utilidad no las crea ni las restablece. (Minerva crea las colecciones vacías durante el arranque). A su vez, también deben de estar creadas todas las carpetas de importación con los permisos correctos antes de lanzar el tag de import. En caso contrario, fallará porque no encuentra las carpetas.

Exportar log de microservicios

Tenemos la funcionalidad de exportar los logs de los microservicios de Anjana a la carpeta de cada nodo del backend <anjana.folder>/anjana_log/<microservicio>.log. **Muy útil para cuando hay que proporcionar logs al equipo de soporte.**

```
ansible-inventories > sample > group_vars > all.yml
70 anjana:
71   folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
```

```
root@ :/opt/ansible# ll -h /opt/anjana_log/
total 1.7M
drwxrwx--- 2 anjana anjana 4.0K Jun 13 15:49 ./
drwxr-xr-x 22 root root 4.0K Jun 13 15:49 ../
-rwxrwx--- 1 anjana anjana 21K Jun 13 15:49 drittesta2_localhost_2023-06-13T15:49:20Z.log*
-rwxrwx--- 1 anjana anjana 21K Jun 13 15:49 drittesta_localhost_2023-06-13T15:49:20Z.log*
-rwxrwx--- 1 anjana anjana 5.1K Jun 13 15:49 edusa2_localhost_2023-06-13T15:49:10Z.log*
-rwxrwx--- 1 anjana anjana 12K Jun 13 15:49 edusa_localhost_2023-06-13T15:49:10Z.log*
-rwxrwx--- 1 anjana anjana 15K Jun 13 15:49 hecate2_localhost_2023-06-13T15:49:12Z.log*
-rwxrwx--- 1 anjana anjana 23K Jun 13 15:49 hecate_localhost_2023-06-13T15:49:13Z.log*
-rwxrwx--- 1 anjana anjana 231K Jun 13 15:49 hermes2_localhost_2023-06-13T15:49:26Z.log*
-rwxrwx--- 1 anjana anjana 231K Jun 13 15:49 hermes_localhost_2023-06-13T15:49:27Z.log*
-rwxrwx--- 1 anjana anjana 226K Jun 13 15:49 kerno2_localhost_2023-06-13T15:49:16Z.log*
-rwxrwx--- 1 anjana anjana 223K Jun 13 15:49 kerno_localhost_2023-06-13T15:49:16Z.log*
-rwxrwx--- 1 anjana anjana 235K Jun 13 15:49 minerva2_localhost_2023-06-13T15:49:22Z.log*
-rwxrwx--- 1 anjana anjana 68K Jun 13 15:49 minerva_localhost_2023-06-13T15:49:22Z.log*
-rwxrwx--- 1 anjana anjana 28K Jun 13 15:49 portuno2_localhost_2023-06-13T15:49:14Z.log*
-rwxrwx--- 1 anjana anjana 28K Jun 13 15:49 portuno_localhost_2023-06-13T15:49:14Z.log*
-rwxrwx--- 1 anjana anjana 14K Jun 13 15:49 tot2_localhost_2023-06-13T15:49:28Z.log*
-rwxrwx--- 1 anjana anjana 14K Jun 13 15:49 tot_localhost_2023-06-13T15:49:28Z.log*
-rwxrwx--- 1 anjana anjana 14K Jun 13 15:49 viator2_localhost_2023-06-13T15:49:25Z.log*
-rwxrwx--- 1 anjana anjana 14K Jun 13 15:49 viator_localhost_2023-06-13T15:49:25Z.log*
-rwxrwx--- 1 anjana anjana 223K Jun 13 15:49 zeus_localhost_2023-06-13T15:49:19Z.log*
```

Para la exportación del log debemos ejecutar con el tag “log” como muestra el siguiente ejemplo:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml --tags log
```

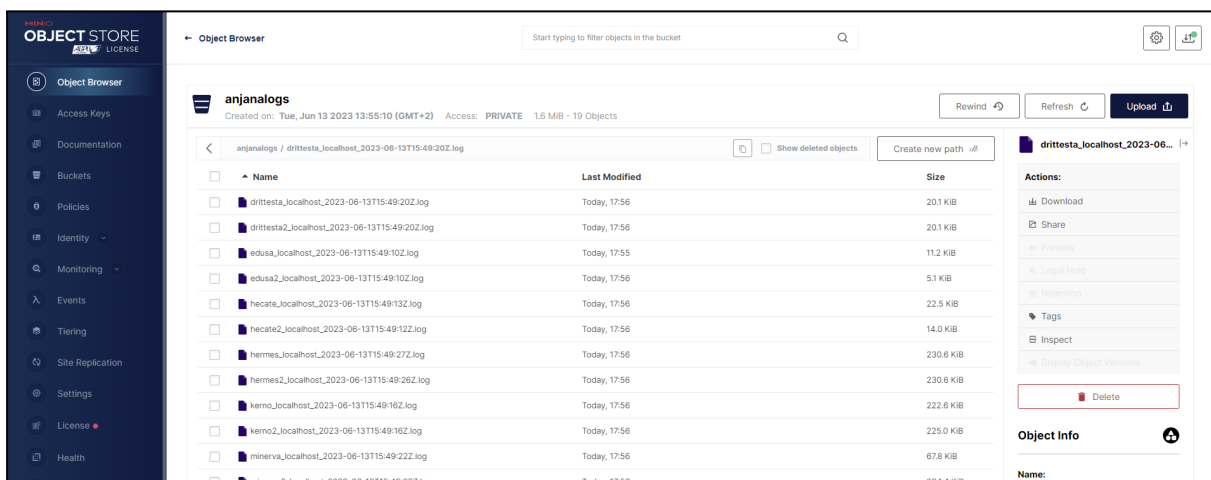
Por defecto exporta los últimos 5 días pero es configurable en tiempo de ejecución:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags log --extra-vars '{"log": {"since": "2 day ago"}}'
```

A su vez, podemos también subir los logs a un servidor s3, que debe de estar configurado (nombre, url y credenciales) en el archivo all.yml. Además, hay que crear el bucket o elegir uno existente. Una vez tengamos los logs, podemos lanzar el siguiente comando para exportarlos

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml /opt/ansible/anjana.yml --tags export-log-s3
```

NOTA: En el archivo group_vars/all.yml del inventario usado hay que poner “installation.mode:” en “director” o “local” para que la utilidad funcione.



Desinstalación

Procede a la desinstalación completa del producto para el inventario seleccionado.

La desinstalación respeta el directorio de **backups**, **export-import**, **ansible** y los archivos de registro de instalación **version_report**, ambos situados en la carpeta principal de Anjana (normalmente /opt).

Para realizar la desinstalación de Anjana se lanza el siguiente comando:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags remove-anjana
```

NOTA: Para que la desinstalación funcione correctamente, el inventario tiene que estar actualizado de acuerdo a la instalación actual, eso incluye: número de máquinas activas, sus correspondientes IPs y nombres de host, usuario de conexión ssh, etc.


```

skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginawss31])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginazuread1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginazurefiles1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginazurestorage1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginingcpbigquery1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginingcpiam1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginingcpstorage1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginhdfs1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginhive1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcl])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcdendodo1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcredshift1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcoracle1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcsqlserver1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginldap1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginpowerbi1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [minio1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [portal1])
changed: [ansible1 -> 10.150.100.208] => (item=Removed [/opt/solr-8.6.0] dir from host [solr1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [zookeeper1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [apacheds1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [postgres1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [fakesmtpserver1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [monit1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [ansible1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [export1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [import1])

TASK [anjana-utility : Remove service user] *****
changed: [ansible1 -> 10.150.100.232] => (item=Removed [anjana] user from all hosts)
changed: [ansible1 -> 10.150.100.47] => (item=Removed [anjana] user from all hosts)
changed: [ansible1 -> 10.150.100.208] => (item=Removed [anjana] user from all hosts)

TASK [anjana-utility : Remove service group] *****
ok: [ansible1 -> 10.150.100.232] => (item=Removed [anjana] group from all hosts)
ok: [ansible1 -> 10.150.100.47] => (item=Removed [anjana] group from all hosts)
ok: [ansible1 -> 10.150.100.208] => (item=Removed [anjana] group from all hosts)

TASK [anjana-utility : Writing Version Report] *****
changed: [ansible1]

TASK [anjana-utility : Anjana Removal - End of Procedure] *****
ok: [ansible1] => {
  "msg": [
    "UNINSTALLATION COMPLETED",
    "BACKUPS AVAILABLE AT /opt/backup",
    "RESTART YOUR MACHINE TO FINISH REMOVING TEMPORARY FILES"
  ]
}

PLAY [Import role export] *****

```


Despliegue de plataforma de monitorización de métricas

Hemos añadido al kit de Ansible la plataforma de monitorización de Anjana, que se compone de:

- Grafana, donde podemos crear cuadros de mando y configuramos los Data sources como fuente de datos
 - URL: `http://<url_nodo>:3000`
 - user: admin
 - password: admin123
- Prometheus el cual provee las métricas a los cuadros de mandos
 - URL: `http://<url_nodo>:9090/targets`
- Exporters de Solr y grafana agent, los cuales aportan métricas extras a Prometheus
 - Según el exporter presenta las métricas en un puerto o ruta distinto, se puede consultar en la url de targets de Prometheus o en las configuraciones del propio exporter
- Loki, tempo y otelcol que aportan logs del producto a grafana

Habilitar importación de roles en all.yml

```
ansible-binary > ansible-inventories > localhost > group_vars > A all.yml
6 # Metrics & log monitoring
7 grafana: true
8 prometheus: true
9 solrexporter: true
0 loki: false
1 tempo: false
2 otelcol: false
3 grafanaagent: true
```

Los parámetros relacionados a las cadenas de conexión pueden ser encontrados también en `<ruta_inventario>/group_vars/all.yml`

```
# MONITORING PLATFORM -----
monitoring:
  labels:
    anjana_zone: Cliente-XXX
    anjana_instance: Cliente-XXX-PRE
  grafana:
    host: grafanaservice
    port: 3000
    user: admin
    pass: admin123
  grafanaagent:
    grpcport: 12346
  prometheus:
    host: prometheusservice
    port: 9090
    federation_targets: # Only needed when federation mode is selected. Prometheus servers Public IP/domain:port to scrape from
    # - 'XX.XX.XX.XX:9090'
    # - 'XX.XX.XX.XX:9090'
    # - 'XX.XX.XX.XX:9090'
  loki:
    host: lokiservice
    port: 3100
    bucket: loki
    bucketruler: lokiruler
  tempo:
    host: temposervice
    port: 3200
    grpcport: 9096
    bucket: tempo
```

Para lanzar la plataforma de monitorización es el siguiente comando:

```
# Recuerda crear los buckets loki, lokiruler y tempo para que no fallen los microservicios de Loki y Tempo

# Instalar la dependencia de Grafana en Ansible
sudo ansible-galaxy collection install community.grafana

# Lanzar todo la plataforma de monitorización
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<_inventario>/hosts.yml /opt/ansible/anjana.yml --tags monitoring

# Solo lanzar Grafana o Prometheus
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml /opt/ansible/anjana.yml --tags grafana
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml /opt/ansible/anjana.yml --tags prometheus

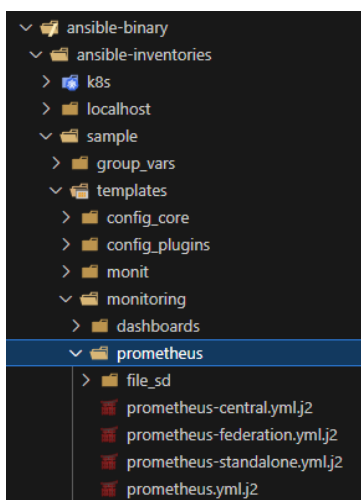
# Solo lanzar un exporter
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml /opt/ansible/anjana.yml --tags solrexporter
```

Prometheus

Hemos creado un rol donde se instala y configura prometheus para la recolección de datos sobre las máquinas de Anjana.

```
ansible-binary > ansible-inventories > sample > hosts.yml > {} all >
507
508     prometheushosts:
509         hosts:
510             prometheus1:
511                 service_name: prometheus
512                 service_port: 9090
513                 prometheus_mode: standalone
514                 ansible_host: <server_ip>
515                 ansible_port: 22
516                 service_type: monitoring
```

La configuración está disponible en el inventario en la carpeta templates/monitoring, estando en esa misma carpeta de monitoring también los archivos file_sd, los cuales no se deberían tocar ya que se rellenan solos.



```
ansible-binary
├── ansible-inventories
│   ├── k8s
│   ├── localhost
│   └── sample
│       ├── group_vars
│       └── templates
│           ├── config_core
│           ├── config_plugins
│           ├── monit
│           └── monitoring
│               ├── dashboards
│               └── prometheus
│                   └── file_sd
│                       ├── prometheus-central.yml.j2
│                       ├── prometheus-federation.yml.j2
│                       ├── prometheus-standalone.yml.j2
│                       └── prometheus.yml.j2
```

La configuración para ambos modos es rellenada automáticamente mediante la información presente en el inventario `<ruta_inventario>/hosts.yml` (nombre de servicio, puerto, IP/dominio, etc...), por lo que tiene que estar actualizada y ser correcta.

Todas las máquinas y/o servicios presentes en el inventario, ya se trate de una instalación de Anjana singlenode, distribuida o en HA serán recogidos en esta configuración.

Aún así puede ser editada en `<ruta_inventario>/templates/monitoring/prometheus/... .`

Solr exporter y Grafana Agent

El exporter de Solr y el Grafana Agent recogen las métricas de los servicios que no tienen endpoint propio de métricas y publican uno para que Prometheus pueda trabajar con las métricas.

Se han sintetizado todos los exporters posibles dentro de Grafana Agent, por lo que pueden habilitarse y deshabilitarse según la necesidad mediante el archivo de variables `<ruta_inventario>/group_vars/grafanaagenthosts.yml`

```
ansible-binary > ansible-inventories > localhost > group_vars > A grafanaagenthosts.yml
1 ---
2 grafanaagent:
3   loglevel: debug
4   path: /etc
5   # If the monitoring node is on another node, indicate the DNS/IP
6   lokihost: '{{monitoring.loki.host}}'
7   lokiport: '{{monitoring.loki.port}}'
8   port: '{{service_port}}'
9   job_type_systemd: loki_systemd
10  job_type_logs: loki_logs
11  hostname: '{{ansible_host}}'
12  zone: '{{monitoring.anjana_zone}}'
13  job: loki
14  integrations:
15    agent: true
16    node_exporter: true
17    apache_http: true
18    postgres_exporter:
19      enabled: true
20      data_source_names: postgresql://{{persistences.bbdd.user}}:{{persistences.bbdd.
pass}}@{{persistences.bbdd.host}}:{{persistences.bbdd.port}}/{{persistences.bbdd.
database}}?sslmode=disable
```

Tanto el Grafana Agent como el Solr exporter tienen sus cadenas de conexión apuntando al apartado de Monitorización situado al final del fichero `<ruta_inventario>/group_vars/all.yml`

En el caso de Grafana Agent hay que editar el inventario y añadir un host de grafanaagent por cada máquina que haya en la instalación de Anjana.

En caso de ser una instalación singlenode sólo será necesario un host de grafanaagent.

```
hosts.yml X
ansible-binary > ansible-inventories > sample > hosts.yml >
562
563
564   grafanaagenthosts:
565     hosts:
566       grafanaagent1:
567         service_name: grafanaagent
568         service_port: 12345
569         ansible_host: <server_ip>
570         ansible_port: 22
571         service_type: monitoring
```

En caso de tratarse de una instalación distribuida o HA de Anjana, habrá que añadir un host de grafanaagent por cada máquina/nodo en la instalación

Grafana

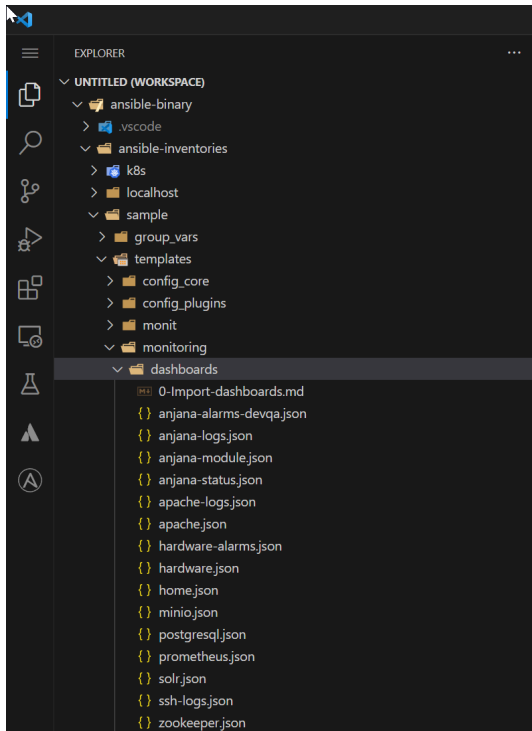
En Grafana se permite configurar las fuentes de datos (datasources) mediante el archivo ubicado en el inventario <ruta_inventario>/group_vars/grafanahosts.yml en caso de querer mayor personalización.

```
ansible-binary > ansible-inventories > localhost > group_vars > A grafanahosts.yml
24 # DATASOURCES -----
25 grafana_datasources:
26   prometheus:
27     imported: true
28     host: '{{monitoring.prometheus.host}}'
29     port: '{{monitoring.prometheus.port}}'
30   postgres:
31     imported: true
32     host: '{{persistences.bbdd.host}}'
33     port: '{{persistences.bbdd.port}}'
34     database: '{{persistences.bbdd.database}}'
35     user: '{{persistences.bbdd.user}}'
36     password: '{{persistences.bbdd.pass}}'
37   solr:
38     imported: true
39     host: '{{persistences.solr.host}}'
40     port: '{{persistences.solr.port}}'
41   loki:
42     imported: true
43     host: '{{monitoring.loki.host}}'
44     port: '{{monitoring.loki.port}}'
45   tempo:
46     imported: true
47     host: '{{monitoring.tempo.host}}'
48     port: '{{monitoring.tempo.port}}'
49
```

Así como los plugins y los dashboards a desplegar.

```
ansible-binary > ansible-inventories > sample > group_vars > A grafanahosts.yml
0 # PLUGINS -----
1 grafana_plugins:
2   - grafana-clock-panel
3   - grafana-simple-json-datasource
4   - grafana-worldmap-panel
5   - grafana-piechart-panel
6   - digiapulssi-breadcrumb-panel
7   - grafana-image-renderer
8   - pue-solr-datasource
9   - jdbramham-diagram-panel
10
11 # DASHBOARDS -----
12 grafana_dashboards:
13   - anjana-apache-logs
14   - anjana-apache
15   - anjana-logs
16   - anjana-minio
17   - anjana-spring-boot
18   - anjana-status
19   - anjana-zookeeper
20   - hardware-extended
21   - node-exporter
22   - postgresql-database
23   - solr-dashboard
24   - syslog-and-applications
25   - systemd-services
26   - home
```

Los dashboards se encuentran en la ruta
/opt/ansible/ansible-inventories/<inventario>/templates/monitoring/dashboards



NOTA: Es recomendable mantener las opciones por defecto si no se conocen bien los productos que componen la plataforma de monitorización o solo se busca una visualización sencilla de métricas.

Una vez lanzado y editado las configuraciones pertinentes, ya podremos entrar a Grafana y ver los dashboards mediante la url, puerto y credenciales indicados en el archivo `<ruta_inventario>/group_vars/all.yml` que se ha mencionado previamente.

Tempo

Hemos añadido a Ansible el rol de Tempo para la recogida de trazas, el cual hay que instalar en la máquina donde se encuentre Grafana. Podemos encontrar sus variables en la ruta `<ruta_inventario>/group_vars/tempohosts.yml`.

Las más importantes son las del puerto, y toda la parte de storage, ya que tempo necesita un bucket de s3 para guardar las métricas.

```

ansible-binary > ansible-inventories > localhost > group_vars > A tempohosts.yml
1  ---
2  tempo:
3    path: '/usr/bin'
4    install:
5      artifact: /tempo/tempo_1.5.0_linux_amd64
6    config:
7      path: '/etc'
8    port: '{{monitoring.tempohosts.port}}'
9    storage:
10     backend: s3
11     endpoint: '{{persistences.s3.host}}:{{persistences.s3.port}}'
12     bucket_name: '{{monitoring.tempohosts.bucket}}'
13     access_key: '{{persistences.s3.access_key}}'
14     secret_key: '{{persistences.s3.secret_key}}'

```

Actualmente estos datos de conexión son manejados desde el all.yml en la sección de S3 y tempo:

```
ansible-binary > ansible-inventories > localhost > group_vars > A all.yml
/4
75 persistences:
76   s3:
77     type: minio # minio|aws_s3
78     access_key: anjana
79     secret_key: anjanadata
80     host: http://s3service # Ex. http://s3service or http://10.10.10.10 # MinIO Only
81     port: 9000 # MinIO Only
```

```
ansible-binary > ansible-inventories > sample > group_vars > A all.yml
331 tempo:
332   host: temposervice
333   port: 3200
334   grpcport: 9096
335   bucket: tempo
```

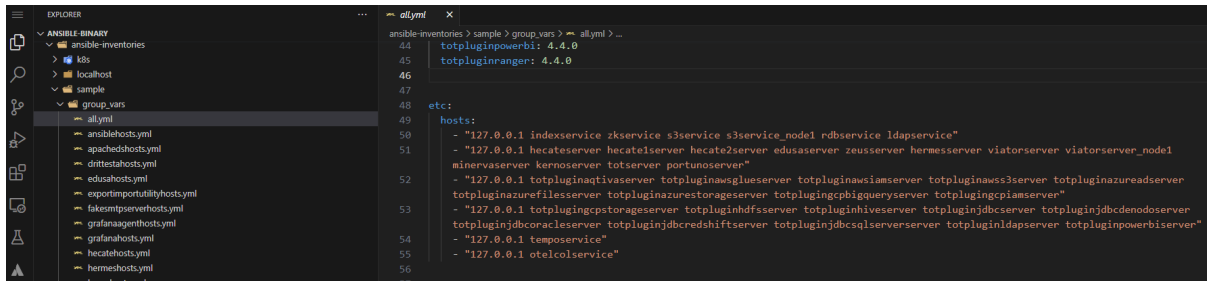
Una vez configurada esta parte, lanzamos el siguiente comando

```
sudo ansible-playbook -i <ruta_inventario>/sample/hosts.yml /opt/ansible/anjana.yml -t tempo
```

Debido a que los logs de Anjana no son de formato Opentelemetry, que son los que acepta tempo, necesitamos usar el Opentelemetry Collector contrib para poder pasarlos a un formato reconocible con tempo. Para ello, hemos creado un rol en Ansible y que haga la adaptación. Para lanzarlo, debemos de tocar el archivo de variables, poniendo lo siguiente

```
otelcolhosts.yml X
ansible-inventories > sample > group_vars > otelcolhosts.yml > {} otelcol > {} metrics > address
Ansible Vars File - Ansible variables File (vars.json)
1 ---
2 otelcol:
3   path: '/usr/bin'
4   install:
5     artifact: /otelcol/otelcol-contrib_0.75.0_linux_amd64.deb
6   config:
7     path: '/etc/otelcol-contrib'
8   exporter:
9     endpoint: temposervice:4317
10  extension:
11    health_check: otelcolservice:13133
12  metrics:
13    address: otelcolservice:19000
```

En la config ubicada en la ruta señalada en amarillo, tenemos que modificar el endpoint del exporter, el cual será la ip y puerto GRPC de tempo, una extension que en el puerto que le indiquemos, saque si todo funciona correctamente, y un puerto para métricas específicas de otelcol. Es importante, que aunque estén los puertos en la máquina localhost, debemos de poner la ip privada, ya que otelcol no acepta localhost o parámetros similares. En el archivo all.yaml de esa misma carpeta, podemos cambiar en la variable etc.hosts la línea donde se indique temposervice localhost por la ip privada de la máquina de tempo, en caso de que no esté en local, e igual con la variable otelcolservice.



```

ansible-inventories > sample > group_vars > all.yaml > ...
44 totpluginpowerbi: 4.4.0
45 totpluginranger: 4.4.0
46
47
48 etc:
49
50 - "127.0.0.1 indexservice zksservice s3service s3service_node1 rdbservice ldapsservice"
51 - "127.0.0.1 hecateserver hecate1server hecate2server edusaserver zeusservice hermesservice viatorservice viatorservice_node1
minervaservice kernoservice totserver portunoservice"
52 - "127.0.0.1 totpluginactivaserver totpluginawsglueserver totpluginawsiamserver totpluginaws3server totpluginazureadservice
totpluginazurefilesservice totpluginazuresstorageserver totpluginazuresblobstoragequeryserver totpluginazurescpiamserver"
53 - "127.0.0.1 totpluginazuresstorageserver totpluginhdfsserver totpluginhivequeryserver totpluginjdbcserver totpluginjdbcnodeoserver
totpluginjdbcoracleserver totpluginjdbcshiftdbserver totpluginjdbcsqlserverserver totpluginldapserver totpluginpowerbiserver"
54 - "127.0.0.1 temposervice"
55 - "127.0.0.1 otelcolservice"
56
57

```

Una vez lo tengamos, lanzamos el siguiente comando

Una vez configurada esta parte, lanzamos el siguiente comando

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/sample/hosts.yaml
/opt/ansible/anjana.yaml -t otelcol

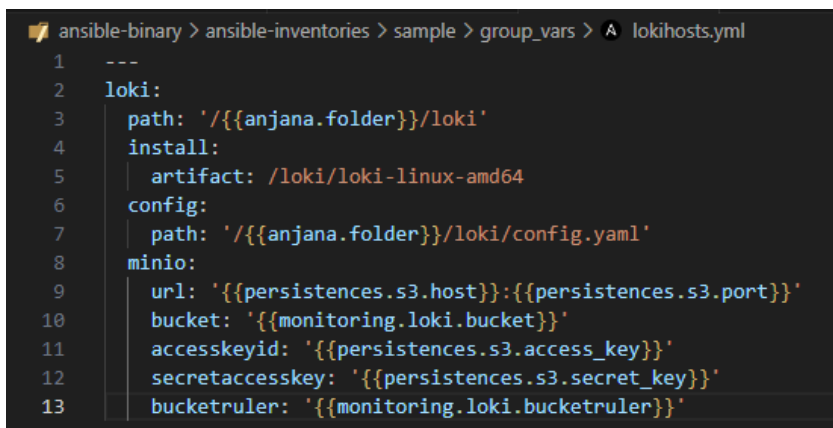
```

Despliegue de plataforma de agregación de logs

Hemos añadido una funcionalidad que utiliza loki y grafana-agent para poder tener los logs de Anjana como datos en Grafana. Loki lo debemos de instalar en la máquina donde se encuentre Grafana, mientras que grafana-agent lo debemos de instalar donde este Anjana.

Loki

Para lanzar Loki, debemos de ajustar las variables, localizadas en el all.yaml para las secciones S3 y loki:



```

ansible-binary > ansible-inventories > sample > group_vars > lokihosts.yaml
1 ---
2 loki:
3   path: '{{anjana.folder}}/loki'
4   install:
5     artifact: /loki/loki-linux-amd64
6   config:
7     path: '{{anjana.folder}}/loki/config.yaml'
8   minio:
9     url: '{{persistences.s3.host}}:{{persistences.s3.port}}'
10    bucket: '{{monitoring.loki.bucket}}'
11    accesskeyid: '{{persistences.s3.access_key}}'
12    secretaccesskey: '{{persistences.s3.secret_key}}'
13    bucketruler: '{{monitoring.loki.bucketruler}}'

```

En el apartado de MinIO, debemos de añadir la URL de un servicio de S3 donde se subirán los logs como MinIO, creando dos buckets con los nombres ahí puestos, y el usuario y contraseña. Las rutas son las de instalación y la de configuración, que por defecto serán /opt/loki

Los buckets deben de estar creados antes de lanzar Loki, si no, fallará.

```
ansible-binary > ansible-inventories > localhost > group_vars > all.yml
/4
75  persistences:
76    s3:
77      type: minio # minio|aws_s3
78      access_key: anjana
79      secret_key: anjanadata
80      host: http://s3service # Ex. http://s3service or http://10.10.10.10 # MinIO Only
81      port: 9000 # MinIO Only
```

```
ansible-binary > ansible-inventories > sample > group_vars > all.yml
264  loki:
265    host: localhost
266    port: 3100
267    bucket: loki
268    bucketruler: lokiruler
```

Para lanzar loki, se lanza de la siguiente manera

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml --tags loki
```

A su vez, si ya tenemos grafana instalada, deberemos de añadir el data source de Loki manualmente. Si lo instalamos desde 0, se instalará con el datasource ya creado.

Grafana Agent

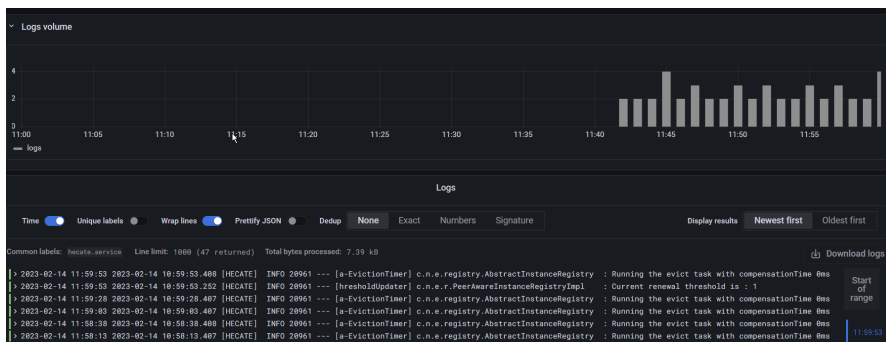
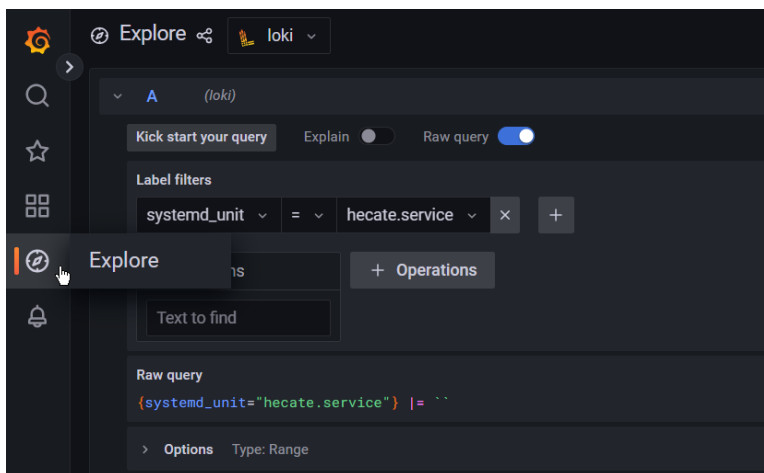
Una vez tengamos Loki, pasaremos a lanzar el grafana-agent en la máquina de Anjana. En ansible, las variables de grafanaagent están ubicadas en <ruta_inventario>/group_vars/grafanaagenthosts.yml.

```
ansible-binary > ansible-inventories > sample > group_vars > grafanaagenthosts.yml
1  ---
2  grafanaagent:
3    loglevel: debug
4    path: /etc
5    # If the monitoring node is on another node, indicate the DNS/IP
6    lokihost: '{{monitoring.loki.host}}'
7    lokiport: '{{monitoring.loki.port}}'
8    port: '{{service_port}}'
9    job_type_systemd: loki_systemd
10   job_type_logs: loki_logs
11   hostname: '{{ansible_host}}'
12   zone: '{{monitoring.anjana_zone}}'
13   job: loki
14   integrations:
15     agent: true
16     node_exporter: true
17     apache_http: true
18     postgres_exporter:
19       enabled: true
20       data_source_names: postgresql://{{persistences.bbdd.user}}:{{persistences.bbdd.
pass}}@{{persistences.bbdd.host}}:{{persistences.bbdd.port}}/{{persistences.bbdd.
database}}?sslmode=disable
```


Aquí, lo importante es la IP y el puerto donde se encuentra loki y el puerto del grafana-agent. Por defecto los datos serán rellenados de forma automática con lo disponible en all.yml. También debemos ajustar según sea necesario los campos de zone, job, loglevel, job_type_systemd y job_type_logs. Para lanzarlo, pondremos el siguiente comando:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
<ruta_ansible_binary>/anjana.yml --tags grafana-agent
```

Para verificar que funciona, vamos a grafana a la sección de explore, y si seleccionamos el datasource de Loki y en label filters seleccionamos una de las opciones que nos salen como la foto siguiente, nos deberían salir logs



Despliegue de plataforma de monitorización completa

Hemos añadido un tag con el cual podemos lanzar toda la plataforma de monitorización completa, que comprende Grafana, Prometheus, Exporter de Solr/ZK, Grafana-agent, Loki y Tempo. Tendremos que, como siempre, rellenar el archivo hosts.yml con las distintas máquinas donde va a ir cada uno de los componentes, y tocando la configuración nombrada en los anteriores apartados, lanzamos el siguiente tag e instalará todo.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml --tags monitoring
```

Detalle de composición del kit

Roles disponibles

En el repo de ansible-binary, en la subcarpeta de roles, encontramos los siguientes roles

- Microservicios de Anjana
 - Edusa
 - Hecate
 - Hermes
 - Kerno
 - Minerva
 - Portuno
 - Drittesta
 - Viator
 - Zeus
 - Tot
 - Horus

- Fontales de Anjana (sobre Apache web server 2.4)
 - Portal Anjana
 - Portal Admin

- Otros microservicios necesarios
 - Minio → Servicio S3
 - Postgresql → Servicio de Base de Datos
 - Apacheds → Servicio de LDAP
 - Zookeeper → Servicio de coordinación de aplicaciones distribuido.
 - Solr → Motor de indexación

NOTA: Tanto SOLR como Zookeeper soportan despliegue en modo cluster de 1 a N nodos. Aunque a partir de Anjana 23.1 no hace falta ya que la información está persistida en bbdd relacional.

- Extras:
 - Monit → Interfaz web para control de servicios systemd de Anjana
 - Fakesmtpserver → Servicio de smtp para ver mails enviados desde Anjana
 - Ansible → Crea una copia del kit de Ansible de Anjana Data para poder ejecutar desde dentro de la máquina (/opt/ansible)
 - Anjana-utility → Contiene las funcionalidades contra Anjana tales como apagado, arrancado, ...
 - Export e import → Contiene las funcionalidades de exportado e importado de una instancia de Anjana Data (volcado completo)
 - Persistences-utility → Contiene las funcionalidades de insert, delete, reset de datos en persistencias (S3, SolR y bbdd)

- Grafana, Prometheus y solrexporter → Necesarios si se desea desplegar plataforma de monitorización de métricas
- Loki, Promtail, Grafana y Grafanaagent → Necesarios si se desea desplegar plataforma de monitorización de logs
- Check → Para verificar que anjana esté arrancada correctamente.
- Plugins
 - Tot-plugin-aws-iam
 - Tot-plugin-aws-glue
 - Tot-plugin-aws-s3
 - Tot-plugin-azure-ad
 - Tot-plugin-azure-files
 - Tot-plugin-azure-storage
 - Tot-plugin-gcp-bigquery
 - Tot-plugin-gcp-iam
 - Tot-plugin-gcp-storage
 - Tot-plugin-hdfs
 - Tot-plugin-hive
 - Tot-plugin-jdbc
 - Tot-plugin-jdbc-redshift
 - Tot-plugin-jdbc-oracle
 - Tot-plugin-jdbc-denodo
 - Tot-plugin-jdbc-sqlserver
 - Tot-plugin-ldap
 - Tot-plugin-aqiva
 - Tot-plugin-powerbi
 - Tot-plugin-ranger
 - Tot-plugin-tableau

Tags disponibles

Explicación en detalle de los tags disponibles en el kit de Ansible

Ejecución sin tag	Se despliega Anjana + persistencia sin datos de ejemplo
anjana-sample	El tag más completo: despliega Anjana, hace un insert del set de datos y los yamls de configuración
update	Solo ejecuta las tareas de actualizar los jars, launchers, front y cdn.
export import	Ejecuta los roles de exportación o importación en un entorno Anjana (afecta a la persistencia solr, bbdd y cdn).
import-solr import-bbdd import-s3	Ejecuta el rol de importación en un entorno Anjana de la persistencia de la cual se haya escogido el tag
s3-dump	Se ejecuta normalmente con el tag export y permite subir un comprimido de la exportación a un bucket de dump
edusa hecate portuno kerno zeus dritttesta minerva viator hermes tot horus	Despliega el microservicio seleccionado. Las tareas que hace son respaldar, borrar y descargar los jars, launchers y crear o sobrescribir el existente descriptor del servicio
portal	Despliega Apache2 con el front de Anjana y admin. Crea o sobrescribe los archivos de configuración de Apache para ambos front
tot-plugin-aws-glue tot-plugin-aws-iam tot-plugin-aws-s3 tot-plugin-azure-ad tot-plugin-azure-files tot-plugin-azure-storage tot-plugin-gcp-bigquery tot-plugin-gcp-iam tot-plugin-gcp-storage tot-plugin-hdfs tot-plugin-hive tot-plugin-jdbc tot-plugin-jdbc-denodo tot-plugin-jdbc-oracle tot-plugin-jdbc-redshift tot-plugin-jdbc-sqlserver tot-plugin-ldap tot-plugin-aqtiva tot-plugin-powerbi tot-plugin-ranger tot-plugin-tableau	Despliega el microservicio seleccionado. Las tareas que hace son respaldar, borrar y descargar los jars, launchers y crear o sobrescribir el existente descriptor del servicio
tot-plugins	Lanza todos los plugins de Tot que estén a true en el apartado import_role del all.yml . Las tareas que hace son respaldar, borrar y

	descargar los jars, launchers y crear o sobrescribir el existente el descriptor del servicio
update-edusa update-hecate update-portuno update-keruo update-zeus update-drittesta update-minerva update-viator update-hermes update-tot update-horus	Solo ejecuta las tareas de actualizar los jars y los launchers del microservicio escogido
update-portal	Actualiza el front de Anjana, el de admin y descarga el cdn sin sobrescribir si se encuentra algún conflicto
update-tot-plugin-aws-gluue update-tot-plugin-aws-iam update-tot-plugin-aws-s3 update-tot-plugin-azure-ad update-tot-plugin-azure-files update-tot-plugin-azure-storage update-tot-plugin-gcp-bigquery update-tot-plugin-gcp-iam update-tot-plugin-gcp-storage update-tot-plugin-hdfs update-tot-plugin-hive update-tot-plugin-jdbc update-tot-plugin-jdbc-denodo update-tot-plugin-jdbc-oracle update-tot-plugin-jdbc-redshift update-tot-plugin-jdbc-sqlserver update-tot-plugin-ldap update-tot-plugin-aqtiva update-tot-plugin-powerbi update-tot-plugin-ranger update-tot-plugin-tableau	Solo ejecuta las tareas de actualizar los jars, launchers de los plugins seleccionados
start-anjana	Arranca de forma ordenada todos los microservicios
stop-anjana	Detiene de forma ordenada toda la instalación de Anjana, comenzando por los plugins y microservicios dependientes de otros y terminando en las persistencias
restart-anjana	Realiza una parada y luego un arranque de forma ordenada de todo Anjana
start-zookeeper start-postgresql start-minio start-solr start-edusa start-hecate start-portuno start-zeus start-keruo start-minerva start-horus start-viator start-hermes	Arranca el microservicio seleccionado iniciando primero de forma selectiva y ordenada todos los microservicios y persistencias que necesita para funcionar

<pre> start-tot start-portal start-tot-plugin-aqtiva start-tot-plugin-aws-glue start-tot-plugin-aws-iam start-tot-plugin-aws-s3 start-tot-plugin-azure-ad start-tot-plugin-azure-files start-tot-plugin-azure-storage start-tot-plugin-gcp-bigquery start-tot-plugin-gcp-iam start-tot-plugin-gcp-storage start-tot-plugin-hive start-tot-plugin-hdfs start-tot-plugin-jdbc start-tot-plugin-jdbc-denodo start-tot-plugin-jdbc-oracle start-tot-plugin-jdbc-redshift start-tot-plugin-jdbc-sqlserver start-tot-plugin-ldap start-tot-plugin-aqtiva start-tot-plugin-powerbi start-tot-plugin-ranger start-tot-plugin-tableau </pre>	
<pre> stop-zookeeper stop-postgresql stop-minio stop-solr stop-edusa stop-hecate stop-portuno stop-zeus stop-kerno stop-minerva stop-horus stop-viator stop-hermes stop-tot stop-portal stop-tot-plugin-aqtiva stop-tot-plugin-aws-glue stop-tot-plugin-aws-iam stop-tot-plugin-aws-s3 stop-tot-plugin-azure-ad stop-tot-plugin-azure-files stop-tot-plugin-azure-storage stop-tot-plugin-gcp-bigquery stop-tot-plugin-gcp-iam stop-tot-plugin-gcp-storage stop-tot-plugin-hive stop-tot-plugin-hdfs stop-tot-plugin-jdbc stop-tot-plugin-jdbc-denodo stop-tot-plugin-jdbc-oracle stop-tot-plugin-jdbc-redshift stop-tot-plugin-jdbc-sqlserver stop-tot-plugin-ldap stop-tot-plugin-aqtiva stop-tot-plugin-powerbi stop-tot-plugin-ranger stop-tot-plugin-tableau </pre>	<p>Detiene el microservicio seleccionado parando primero de forma selectiva y ordenada todos los microservicios que dependen de él</p>

remove-anjana	Procede a una desinstalación completa de Anjana, incluido el software de terceros que se haya desplegado con el kit, en el inventario seleccionado. La carpeta de Ansible no se borra porque impediría la finalización de la ejecución. Se mantienen los backups y las exportaciones existentes.
common	Crea todo lo necesario para el correcto funcionamiento de anjana, como crear el usuario y grupo, crear la carpeta temporal, ajustar los alias de /etc/hosts, instala jq/unzip/iproute2 y comprobar la conexión al repositorio
common-microservices	Tareas comunes de provisión
ansible	Lanza el rol de ansible, el cual deja el kit de Anjana dentro de la máquina, e instala Ansible con las colecciones necesarias, dejando también el script de actualización
clean	Borra lo descargado en la carpeta temporal
persistences	Instala todos los software de persistencia (minio, solr, zk, postgresql)
persistences-sample	Hace lo mismo que el tag de persistences, añadiendo el sample data que hayamos seleccionado en all.yml
postgresql zookeeper solr apacheds minio	Instala el software de persistencia
skip-backup	Se usa lanzandolo con skip-tags "--skip-tags skip-backup", para evitar que se haga backup
backup	Realiza el backup de los jar, launcher, configrepo y de los datos de las persistencias. NOTA para la persistencia (PostgreSQL, Solr, MinIO, ...): Si la persistencia no se ha desplegado con Ansible hay que lanzar el role "Export" que permite atacar a persistencias externas para realizar el respaldo
delete	Borra los datos en minio, solr, y postgresql
insert	Se realiza un insert con el set de datos escogido en all.yml en minio y postgresql
reset	Hace lo mismo que el tag delete + el tag insert seguido
download	El modo local requiere tener descargado todo el software previamente. Este tag permite tener esta descarga previa. Se lanza en modo director y descarga jars, launchers, set de datos, software, utilidades que requiere Anjana para ser copiado a otro nodo Ansible sin conectividad a internet y desplegar Anjana en modo "local"
grafana	Despliega Grafana junto a los dashboards y datasources.
prometheus	Despliega Prometheus
monitoring	Despliega la plataforma de monitorización compuesta por Grafana, Prometheus, Exporter de Solr, Grafana-agent, Loki, Tempo y el

	Opentelemetry Collector
solrexporter	Despliega el exporter de Solr, que recoge métricas sobre el/los Solr indicados
grafana-agent	Despliega Grafana-agent, para la captura de métricas y logs
otelcol	Despliega el Opentelemetry Collector, necesario para trabajar con tempo
Loki	Despliega Loki, el cual pasa las métricas que recibe desde Grafana-agent y Promtail a Grafana
update-anjana-config	Actualiza/despliega los archivos de configuración de los microservicios. CUIDADO, reemplaza la configuración de todos los microservicios al ejecutar, con los templates en el inventario
update-apache-vhosts	Actualiza/despliega los archivos de configuración de los virtualhosts de Apache. CUIDADO, reemplaza todos los vhosts
update-anjana-services	Actualiza/despliega los descriptores de servicio de los microservicios de Anjana. CUIDADO, reemplaza todos los descriptores de Anjana
modsecurity	Instala WAF modsecurity en modo "Detection-Only" y reinicia Apache2.
log	Exporta los logs de los microservicios de Anjana a la carpeta <anjana-folder>/anjana_log/<microservicio>.log. Por defecto exporta los últimos 5 días pero es configurable en tiempo de ejecución: ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml /opt/ansible/anjana.yml --tags log --extra-vars 'log_since="2 day ago"'
update-anjana-sequences	Resetea las secuencias de Anjana en la base de datos
delete-anjana-data	Borra los datos de Anjana en la base de datos
delete-anjana-data-and-functional-config	Borra los datos y la configuración funcional de Anjana en la base de datos
adjust-ansible-folder	Ajusta los permisos de la carpeta descargada de ansible
restart_anjana_microservices	Reinicia los servicios de Anjana sin ordenar
export-log-s3	Recoge los logs ya disponibles de la carpeta /opt/anjana_logs y los sube a un s3 configurado en el all.yaml
check_health	Verifica que los microservicios y persistencias de anjana, junto a sus plugins, si existen, están correctamente levantados o no
fakesmtpserver	Instala un servicio SMTP con interfaz web para ver los emails enviados por Anjana
monit	Instala un software con interfaz web para ver el estado de los microservicios de Anjana además de reiniciarlos o pararlos. El usuario tiene que tener acceso al puerto de monit en la máquina de anjana y que el entorno sea singlenode

Funciones adicionales del kit

Cambiar el formato del Log de Anjana

Podemos cambiar el log de Anjana y Apache al formato key:value para que sea más fácil para leer al usuario y para mejorar su captación en la plataforma de monitorización. Para ello, en el archivo all.yml de nuestro inventario, debemos de irnos al final del documento, y cambiar la variable log.pattern de default a monitoring.



```
ansible-inventories > sample > group_vars > all.yml > {} log
262 log:
263   since: "5 days ago"
264   pattern: default # default or monitoring
```

Una vez cambiada esa variable, lanzamos el siguiente comando para actualizar las configuraciones con el nuevo pattern y reiniciar todos los microservicios

```
# Para actualizar los archivos de apache2
ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml -t update-apache-vhosts

# Para actualizar los archivos de configuración
ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml -t update-anjana-config

# Para reiniciar todos los microservicios
ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml
/opt/ansible/anjana.yml -t restart-anjana
```

Los Logs pasan de este formato:

```
root@ip-10-152-10-108:~# journalctl -fu hermes
-- Logs begin at Wed 2023-04-19 14:13:17 UTC. --
May 10 05:52:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 05:52:31.649 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 05:57:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 05:57:31.649 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:02:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:02:31.649 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:07:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:07:31.650 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:12:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:12:31.650 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:17:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:17:31.650 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:22:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:22:31.651 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:27:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:27:31.652 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:32:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:32:31.653 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
May 10 06:37:31 ip-10-152-10-108 hermeslauncher[3069]: 2023-05-10 06:37:31.654 [HERMES] INFO 3069 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
```

A este:

```
root@ip-10-152-10-108:~# journalctl -fu hermes --since 06:46
-- Logs begin at Wed 2023-04-19 14:13:17 UTC. --
May 10 06:46:42 ip-10-152-10-108 hermeslauncher[49001]: ts="2023-05-10 06:46:42.486" ms=HERMES level=INFO pid=49001 th=trap-executor-0 class=c.n.d.s.r.aws.ConfigClusterResolver : msg="Resolving eureka endpoints via configuration"
May 10 06:51:42 ip-10-152-10-108 hermeslauncher[49001]: ts="2023-05-10 06:51:42.486" ms=HERMES level=INFO pid=49001 th=trap-executor-0 class=c.n.d.s.r.aws.ConfigClusterResolver : msg="Resolving eureka endpoints via configuration"
May 10 06:56:42 ip-10-152-10-108 hermeslauncher[49001]: ts="2023-05-10 06:56:42.487" ms=HERMES level=INFO pid=49001 th=trap-executor-0 class=c.n.d.s.r.aws.ConfigClusterResolver : msg="Resolving eureka endpoints via configuration"
```

Descargar artefactos de Anjana

Es posible la descarga del total de artefactos de Anjana para poderlos usar en un despliegue posterior sin conexión al repositorio. Para ello se lanza:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml --tags download
```

NOTA: este tag solo puede usarse en modo de despliegue “director” para que las descargas se efectúen en el nodo de Ansible.

```
✓ installation:
  mode: director
  tmpdir: /tmp/anjana
```

Sampledata

Despliega el set de datos de ejemplo de uso de Anjana seleccionado en el archivo principal de variables all.yml

```
## DEPLOY ANJANA WITH ALL SAMPLEDATA (no hay backup previo)
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml -v -t anjana-sample
```

```
## DEPLOY ALL SAMPLEDATA
```

```
# En bbdd no deben de existir los schemas y el backend de Anjana debe de estar parado para evitar conflictos
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml -v --tags insert
```

Data delete

Hace una limpieza de los datos escogidos en el entorno

NOTA: Recordar detener previamente el backend

```
## DELETE/CLEAN ALL DATA
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml -v --tags stop-anjana
```

```
sudo systemctl start minio postgresql solr
```

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml -v --tags delete
```

Data reset

Hace una limpieza de los datos y posteriormente despliega el set de datos de ejemplo de uso de Anjana seleccionado en el archivo principal de variables all.yml

```
## DELETE/CLEAN AND INSERT ALL DATA
# El backend de Anjana debe de estar parado para evitar conflictos

sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml
-v --tags reset

sudo reboot
```

Exportar backups a S3

El rol Ansible tiene la funcionalidad de comprimir las exportaciones del role “export” y subirlo a un bucket de S3.

La variable con la ruta que comprime y posteriormente sube a S3, se ajusta en “<ansible-folder>/ansible-inventories/<inventory>/group_vars/exportimportutilityhosts.yml” y es:

```
ansible-inventories > localhost > group_vars > exportimportutilityhosts.yml
1 ---
2 export_import:
3   path: /{{anjana.folder}}/export-import
4   utility: '{{installation.tmpdir}}/export-import-utility'
5 ei_minio:
6   datapath: /{{anjana.folder}}/data/minio
7   path: /{{anjana.folder}}/minio
```

La variable con el bucket, url y credenciales se puede ajustar en el archivo “<ansible-folder>/ansible-inventories/<inventory>/group_vars/all.yml”:

```
ansible-binary > ansible-inventories > sample > group_vars > all.yml
0 persistences:
1   s3:
2     type: minio # minio|aws_s3
3     access_key: anjana
4     secret_key: anjanadata
5     host: http://s3service # Ex. http://s3service or http://10.10.10.10 # MinIO Only
6     port: 9000 # MinIO Only
7     region: <region> # AWS S3 Only
8     buckets: # Replace the name attribute value with the custom bucket name
9       - { bucket: cdn, name: cdn-pro }
10      - { bucket: dsa, name: dsa-pro }
11      - { bucket: imports, name: imports-pro }
12      - { bucket: textarea, name: textarea-pro }
13      - { bucket: workflows, name: workflows-pro }
14      - { bucket: datadump, name: datadump-pro }
15      - { bucket: anjanalogs, name: anjanalogs-pro }
```

El comando a lanzar es:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags s3-dump
```

El archivo generado tiene el siguiente formato para no tener problemas de sobrescritura:

```
anjana_exportdump_2023-01-30T12:26:46Z.tgz
```

Balancedor/HA para MinIO

Se facilita en el virtualhost del servidor web un balanceador para entornos con varios nodos de MinIO.

En el archivo portalhost.yml hay que tener en cuenta:

- port.minioProxyPass para indicar el puerto que tendrá el balanceador que luego habrá que tenerlo en cuenta para la configuración por ejemplo de Kerno, Hermes, etc...
- balancer.minio_nodes para indicar los nodos de MinIO que se van a balancear

```

A exportimportutilityhosts.yml
A fakesmtpserverhosts.yml
A grafanaagenthosts.yml
A grafanahosts.yml
A hecatehosts.yml
A hermeshosts.yml
A horushosts.yml
A kernohosts.yml
A lokihosts.yml
A minervahosts.yml
A miniohosts.yml
A monithosts.yml
A nodeexporterhosts.yml
A persistencesutilityhosts.yml
A portalhosts.yml
A portunohosts.yml
A postgresexporterhosts.yml
A postgresqlhosts.yml
A prometheushosts.yml
A promtailhosts.yml
A solrhosts.yml
A tothosts.yml
A totpluginactivahosts.yml
A totpluginawsgluehosts.yml
A totpluginawsiamhosts.yml
A totpluginaws3hosts.yml
A totpluginazureadhosts.yml
A totpluginazurefilehosts.yml
14 backup: /{{anjana.folder}}/backup/portal
15 port:
16   http: 80
17   https: 443
18   minioProxyPass: 9100
19 vhosts:
20   - anjana_ssl.conf
21   - anjana.conf
22   - minio balancer.conf
23 modules:
24   - rewrite
25   - proxy
26   - proxy_http
27   - proxy_balancer
28   - lbmethod_bytraffic
29 modsecurity: false
30 balancer:
31   swagger: |
32     BalancerMember http://viatorserver_node1:8085/swagger
33   swaggerapi: |
34     BalancerMember http://viatorserver_node1:8085/v3/api-docs
35   gateway: |
36     BalancerMember http://viatorserver_node1:8085
37   portuno: |
38     BalancerMember http://portunoserver:8998/portuno
39   minio_nodes:
40     - '{{persistences.s3.host}}' ## Main MinIO node. Do not modify this one
41     # For HA Deployment / MinIO Cluster
42     # - <http(s)://minio_node2>
43     # - <http(s)://minio_node3>
44     # - <http(s)://minio_node4>

```

```

minio:
  url: http://<front_node>:9100
  accessKey: anjana
  secretKey: anjanadata
# Mail connection properties

```

Carpeta data MinIO en HA

Carpeta de datos de un MinIO en HA debe ser un disco montado o saldrá el siguiente error al hacer “journalctl -u minio -n100”

```

Sep 06 14:11:22 minio1 minio[3860]: API: SYSTEM()
Sep 06 14:11:22 minio1 minio[3860]: Time: 14:11:22 UTC 09/06/2023
Sep 06 14:11:22 minio1 minio[3860]: Error: Drive 'http://10.152.20.196:9000/opt/data/minio' is part of root drive, will not be used (*errors.errorString)
Sep 06 14:11:22 minio1 minio[3860]: 8: internal/logger/logger.go:258:logger.LogIf()
Sep 06 14:11:22 minio1 minio[3860]: 7: cmd/erasure-sets.go:1072:cmd.markRootDisksAsDown()
Sep 06 14:11:22 minio1 minio[3860]: 6: cmd/format-erasure.go:785:cmd.initFormatErasure()
Sep 06 14:11:22 minio1 minio[3860]: 5: cmd/prepare-storage.go:214:cmd.connectLoadInitFormats()
Sep 06 14:11:22 minio1 minio[3860]: 4: cmd/prepare-storage.go:305:cmd.waitForFormatErasure()
Sep 06 14:11:22 minio1 minio[3860]: 3: cmd/erasure-server-pool.go:103:cmd.newErasureServerPools()
Sep 06 14:11:22 minio1 minio[3860]: 2: cmd/server-main.go:785:cmd.newObjectLayer()
Sep 06 14:11:22 minio1 minio[3860]: 1: cmd/server-main.go:611:cmd.serverMain()
Sep 06 14:11:22 minio1 minio[3860]: API: SYSTEM()
Sep 06 14:11:22 minio1 minio[3860]: Time: 14:11:22 UTC 09/06/2023

```

Scripts funcionales

Se han facilitado incluidos en el kit unos scripts para la gestión de los datos y la configuración funcional de Anjana, entre los que se incluyen:

- Actualización de secuencias
- Borrado de datos
- Borrado de datos y configuración funcional

La versión del script que se lanza puede ser ajustada en el archivo all.yml del inventario:

```
kerno: 4.4.0
minerva: 4.4.0
portuno: 4.4.0
tot: 4.4.0
viator: 4.4.0
zeus: 4.4.0
portal: 4.4.0
admin: 4.4.0
drittesta: 4.4.0
# UTILITIES -----
ansible: 4.a5
sampledata: 4.4-sample
script_reset_sequences: 4.4
script_delete_data: 4.4
script_delete_data_and_functional_config: 4.4
# PLUGINS -----
```

Y para lanzar los scripts se haría con los siguientes tags respectivamente:

```
ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags update-anjana-sequences

ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags delete-anjana-data

ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml
/opt/ansible/anjana.yml --tags delete-anjana-data-and-functional-config
```

NOTA: Al tener cambios en la persistencia de la bdd es importante **reindexar** después de la ejecución para tener datos consistentes en SolR.

Compatibilidad con uso de sudo

Debido a la necesidad de que la ejecución del kit requiere de sudo, podemos restringir a los usuarios los comandos necesarios para el kit editando el archivo `/etc/sudoers`.

```
sudo visudo
```

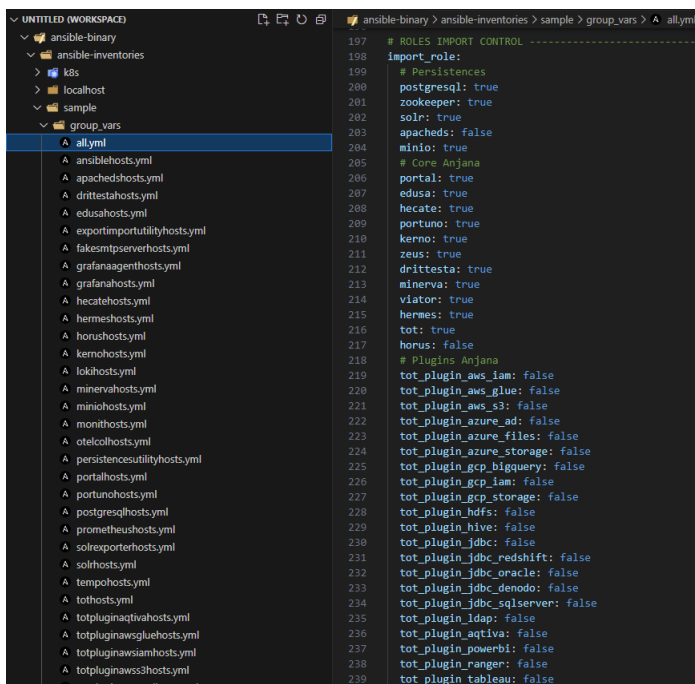
Reglas por defecto para que funcione Ansible, y las tareas que los técnicos como mínimo deben tener. Ejemplo de reglas:

```
#GS
Cmdnd_Alias SERVICES      = /usr/bin/journalctl *, /usr/bin/systemctl *
Cmdnd_Alias READ          = /usr/bin/cat *, /usr/bin/less *, /usr/bin/tail *
#Cmdnd_Alias WRITE        = /usr/bin/nano /opt/*, /usr/bin/vim /opt/*
Cmdnd_Alias NETWORK       = /usr/bin/netstat *, /usr/bin/ss *
Cmdnd_Alias APT           = /usr/bin/apt *
Cmdnd_Alias ANSIBLE       = /usr/bin/nano /opt/ansible/anjana.yml, /usr/bin/nano
/opt/ansible/ansible-inventories/*
Cmdnd_Alias ANSIBLE_PLAYBOOK = /usr/bin/ansible-playbook *
Cmdnd_Alias ANSIBLE_VIM   = /usr/bin/vim /opt/ansible/anjana.yml, /usr/bin/vim
/opt/ansible/ansible-inventories/*

Cmdnd_Alias ANJANA_ALL    = SERVICES, READ, NETWORK, APT, ANSIBLE, ANSIBLE_VIM,
ANSIBLE_PLAYBOOK
%<grupo_técnicos> ALL=ANJANA_ALL
<usuario_técnico> ALL=ANJANA_ALL
```

Verificar que anjana esté levantado correctamente.

Hemos creado un role, llamado check, en el cual verificamos que anjana esté levantado atacando a la api `/actuador/health` de los microservicios y plugins de anjana que estén a true en el apartado de `import_role` del archivo `all.yml` de nuestro inventario.



```
197 # ROLES IMPORT CONTROL
198 import_role:
199 # Persistences
200 postgresql: true
201 zookeeper: true
202 solr: true
203 apacheds: false
204 minio: true
205 # Core Anjana
206 portal: true
207 edusa: true
208 hecate: true
209 portuno: true
210 kerno: true
211 zeus: true
212 drittesta: true
213 minerva: true
214 viator: true
215 hermes: true
216 tot: true
217 horus: false
218 # Plugins Anjana
219 tot_plugin_aws_iam: false
220 tot_plugin_aws_glue: false
221 tot_plugin_aws_s3: false
222 tot_plugin_azure_ad: false
223 tot_plugin_azure_files: false
224 tot_plugin_azure_storage: false
225 tot_plugin_gcp_bigquery: false
226 tot_plugin_gcp_iam: false
227 tot_plugin_gcp_storage: false
228 tot_plugin_hdfs: false
229 tot_plugin_hive: false
230 tot_plugin_jdbc: false
231 tot_plugin_jdbc_redshift: false
232 tot_plugin_jdbc_oracle: false
233 tot_plugin_jdbc_denodo: false
234 tot_plugin_jdbc_sqlserver: false
235 tot_plugin_ldap: false
236 tot_plugin_activa: false
237 tot_plugin_powerbi: false
238 tot_plugin_ranger: false
239 tot_plugin_tableau: false
```

Aunque el role se llame check, tendremos que lanzar el role con el tag check_health.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml  
/opt/ansible/anjana.yml --tags check_health
```

Repositorios proxy

Se han creado 2 repositorios proxy apt, uno para seguridad y otro para archivos, y un repositorio proxy pip disponible para que los clientes mejoren la descarga de los ficheros necesarios para la instalación de Ansible y de anjana.

Los repositorios proxy vienen de los repositorios oficiales tanto de pip, como de ubuntu.

Repositorios proxy Apt

Para configurarlo, vamos a ir a la máquina Ubuntu, y creamos el fichero /etc/apt/sources.list.d/anjana.list con el siguiente contenido

```
deb https://releases.anjanadata.org/repository/aptproxy/ focal focal-update  
focal-security focal-backports main restricted universe multiverse  
  
deb https://releases.anjanadata.org/repository/aptproxyssecurity/ focal focal-update  
focal-security focal-backports main restricted universe multiverse
```

Para la autenticación en el repositorio nexus externo, debemos de crear el archivo /etc/apt/auth.conf.d/anjana.conf con el siguiente contenido

```
machine releases.anjanadata.org login <usuario> password <password> debian
```

Una vez creados ambos archivos, ya podemos ir a la ruta /etc/apt/sources.list y comentar las líneas de los demás repositorios.

Repositorio proxy pip

En la máquina donde tengamos PIP instalado, editamos el archivo /etc/pip.conf, en el cual añadiremos lo siguiente.

```
[global]  
index = https://<user>:<password>@releases.anjanadata.org/repository/Pypi-proxy/pypi  
  
index-url =  
https://<user>:<password>@releases.anjanadata.org/repository/Pypi-proxy/simple
```

Para verificar que funciona, lanzaremos el comando pip config list -v , donde nos mostrará la información del archivo /etc/pip.conf si todo está bien configurado.

Ejemplos de uso

Despliegue single node + RDS

A continuación se detalla el proceso de despliegue de Anjana en un único nodo con base de datos en RDS de cloud.

1. Creamos una máquina virtual
2. En la carpeta ansible-inventories hay que clonar de sample una nueva carpeta con un nombre identificador del nuevo entorno
3. Editar en hosts.yml los parámetros `ansible_host`, `ansible_user` y `ansible_ssh_private_key_file` para que Ansible pueda acceder.

```
ansible-inventories > sample > hosts.yml
1  all:
2    vars:
3      ansible_connection: smart
4      ansible_user: <ssh_user>
5      ansible_ssh_user: <ssh_user>
6      ansible_ssh_private_key_file: /root/.ssh/key.pem
7      ansible_become: true
8      force_service_mgr: systemd # systemd or initd
9  children:
10  directorhosts:
11    hosts:
12      director1:
13        # Nodo en el cual se ejecuta ansible (nodo director)
14        ansible_host: localhost
15        ansible_port: 22
16        service_hostname: localhost
17        ansible_python_interpreter: python3 # when problems try /usr/bin/python
18        ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
19        # ansible_connection: smart # Only if ansible_host is an IP or DNS
20  commonhosts:
21    hosts:
22      common1:
23        # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
24        # Cuentan las máquinas de Anjana, persistencias, ansible, etc
25        ansible_host: localhost
26        ansible_port: 22
27        service_hostname: localhost
28        ansible_python_interpreter: python3 # when problems try /usr/bin/python
29        ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
30      common2:
31        # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
32        # Cuentan las máquinas de Anjana, persistencias, ansible, etc
33        ansible_host: <server_ip>
34        ansible_port: 22
35        service_hostname: localhost
36        ansible_python_interpreter: python3 # when problems try /usr/bin/python
37  commonmicroserviceshosts:
38    hosts:
39      commonmicroservices1:
40        # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) ta
41        # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookee
42        ansible_host: <server_ip>
43        ansible_port: 22
44        service_hostname: localhost
45  persistencesubitubosts:
```

4. Editar `group_vars/all.yml` con la versión deseada para el despliegue de Anjana

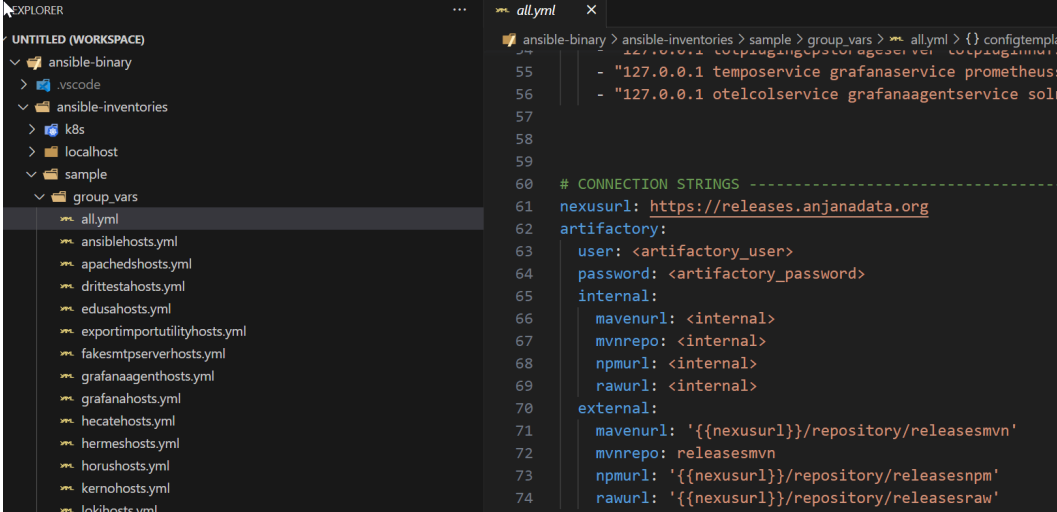

```

ansible-binary > ansible-inventories > sample > group_vars > all.yml
1 #-----#
2 #   Ansible Kit 4.a6   #
3 #-----#
4 version:
5 # MICROSERVICES -----#
6   edusa: 4.5.0
7   hecate: 4.5.0
8   hermes: 4.5.0
9   horus: 4.5.0
10  kerno: 4.5.0
11  minerva: 4.5.0
12  portuno: 4.5.0
13  tot: 4.5.0
14  viator: 4.5.0
15  zeus: 4.5.0
16  portal: 4.5.0
17  admin: 4.5.0
18  drittesta: 4.5.0

```

5. Editar group_vars/all.yml

- a. Usuario y contraseña proporcionado por Anjana para el acceso al repositorio de artefactos



```

55 - "127.0.0.1 tempodbservice grafanaservice prometheus"
56 - "127.0.0.1 otelcolservice grafanaagentservice sol"
57
58
59
60 # CONNECTION STRINGS -----#
61 nexusurl: https://releases.anjanadata.org
62 artifactory:
63   user: <artifactory_user>
64   password: <artifactory_password>
65   internal:
66     mavenurl: <internal>
67     mvnrepo: <internal>
68     npmurl: <internal>
69     rawurl: <internal>
70   external:
71     mavenurl: '{{nexusurl}}/repository/releasesmvn'
72     mvnrepo: releasesmvn
73     npmurl: '{{nexusurl}}/repository/releasesnpm'
74     rawurl: '{{nexusurl}}/repository/releasesraw'

```

- b. Url y credenciales de RDS

```

persistences:
  s3:
    type: minio # minio|aws_s3
    access_key: anjana
    secret_key: anjanadata
    host: # MinIO Only
      - http://s3service # Ex. http://s3service or http://10.10.10.10
    # For HA Deployment / MinIO Cluster
    # - <http(s)://minio_node2>
    # - <http(s)://minio_node3>
    # - <http(s)://minio_node4>
    port: 9000 # MinIO Only
    region: <region> # AWS S3 Only
    buckets: # Replace the name attribute value with the custom bucket name
      - { bucket: cdn, name: cdn-pro }
      - { bucket: dsa, name: dsa-pro }
      - { bucket: imports, name: imports-pro }
      - { bucket: textarea, name: textarea-pro }
      - { bucket: workflows, name: workflows-pro }
      - { bucket: datadump, name: datadump-pro }
      - { bucket: anjanalogs, name: anjanalogs-pro }
  bidd:
    host: rdbservice # Ex. rdbservice or 10.10.10.10
    port: 5432
    database: anjana
    user: anjana
    pass: anjana

```

c. Archivos de configuración a añadir

```

# CONFIG TEMPLATES DEPLOYMENT -----
## Uncomment config as needed
configtemplates:
  core:
    - drittista
    - hecate
    - hermes
    - kerno
    - minerva
    - portuno
    - tot
    - viator
    - zeus
    # - horus
  plugins:
    # - tot-plugin-aws-glue
    # - tot-plugin-aws-iam
    # - tot-plugin-aws-s3
    # - tot-plugin-azure-ad
    # - tot-plugin-azure-files
    # - tot-plugin-azure-storage
    # - tot-plugin-gcp-bigquery
    # - tot-plugin-gcp-iam
    # - tot-plugin-gcp-storage
    # - tot-plugin-hive
    # - tot-plugin-hdfs
    # - tot-plugin-jdbc
    # - tot-plugin-jdbc-denodo
    # - tot-plugin-jdbc-oracle
    # - tot-plugin-jdbc-redshift
    # - tot-plugin-jdbc-sqlserver
    # - tot-plugin-ldap
    # - tot-plugin-aqtiva
    # - tot-plugin-powerbi
    # - tot-plugin-ranger
    # - tot-plugin-tableau

```

d. La licencia de Anjana

```
ansible-inventories > sample > group_vars > all.yml
117
118 anjana:
119   folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data
120   configURL: optional:configserver:http://edusaserver:8888 #
121             edusaservernode1:8888,optional:configserver:http://edusase
122   license:
123     installationCode: AAAAAAAAA-InstallationCode-AAAAAAAAA
124     privateKey: AAAAAAAAA-PrivateKey-AAAAAAAAA
125     anjanaPublicKey: AAAAAAAAA-AnjanaPublicKey-AAAAAAAAA
```

e. Microservicios y plugins a instalar.

```
all.yml x
ansible binary > ansible-inventories > sample > group_vars >
141 ROLES IMPORT CONTROL -----
142 import_role:
143   # Persistences
144   postgresql: true
145   zookeeper: true
146   solr: true
147   apacheds: false
148   minio: true
149   # Core Anjana
150   portal: true
151   edusa: true
152   hecate: true
153   portuno: true
154   kerno: true
155   zeus: true
156   drittesta: true
157   minerva: true
158   viator: true
159   hermes: true
160   tot: true
161   horus: false
162   # Plugins Anjana
163   tot_plugin_aws_iam: false
164   tot_plugin_aws_glue: false
165   tot_plugin_aws_s3: false
166   tot_plugin_azure_ad: false
167   tot_plugin_azure_files: false
168   tot_plugin_azure_storage: false
169   tot_plugin_gcp_bigquery: false
170   tot_plugin_gcp_iam: false
171   tot_plugin_gcp_storage: false
172   tot_plugin_hdfs: false
173   tot_plugin_hive: false
174   tot_plugin_jdbc: false
175   tot_plugin_jdbc_redshift: false
176   tot_plugin_jdbc_oracle: false
177   tot_plugin_jdbc_denodo: false
178   tot_plugin_jdbc_sqlserver: false
179   tot_plugin_ldap: false
180   tot_plugin_aqtiva: false
181   tot_plugin_powerbi: false
182   tot_plugin_ranger: false
```

6. Las credenciales y ruta de conexión a la base de datos serán rellenados automáticamente durante el despliegue utilizando los disponibles en el all.yml del inventario. Pueden alterarse manualmente en los templates de la configuración de los microservicios en “ansible-inventories/<inventario>/templates/config_core/xxxx”

```

D (WORKSPACE)  ansible-binary > ansible-inventories > sample > templates > config_core > kerno.config.4.4.3.ymlj2
1  spring:
2  datasource:
3      #If schema changes, change hibernate.default_schema few lines after
4      url: jdbc:postgresql://{{persistences.bbdd.host}}:{{persistences.bbdd.port}}/{{persistences.bbdd.database}}?currentSchema=anjana
5      username: {{persistences.bbdd.user}}
6      password: {{persistences.bbdd.pass}}
7  jpa:
8      properties:
9          hibernate:
10             default_schema: anjana
11

```

```

anjana:
  datasource:
    driver-class-name: org.postgresql.Driver
    url: jdbc:postgresql://anjanards.cgikxvfi33gk.eu-west-1.rds.amazonaws.com:5432/anjanardsdatabase
    password: V4xq7Fc4edfUPhD
    username: anjanouser
  hikari:
    schema: anjana
    leak-detection-threshold: 20000
    connection-timeout: 3000

```

7. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos ([Conectividad requerida](#)).
8. Una vez los siguientes puntos preparados ya podemos lanzar el despliegue:
 - a. acceso por ssh a la máquina
 - b. establecidas las variables con los datos correctos
 - c. acceso al repositorio de artefactos desde la máquina

En este caso lo vamos a lanzar para que despliegue con datos de ejemplo PERO va a saltar desplegar postgresql para que no lo instale ya que vamos a usar RDS debido al ajuste anterior de import-role.postgre a false en all.yml:

```

import_role:
  # Persistence
  postgresql: false
  zookeeper: true
  solr: true
  apacheds: false
  minio: true
  # Core Anjana
  portal: true

```

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml
/opt/ansible/anjana.yml -tags anjana-sample

```

```

EXPLORER  ...  hosts.yml  X
ANSIBLE-BINARY
├── ansible-inventories
│   ├── sample
│   └── singlenode_&_rds
│       ├── group_vars
│       └── hosts.yml
└── roles
    └── anjana.yml
hosts.yml
1  all:
2      children:
3          edusahosts:
4              hosts:
5                  edusa1:
6                      ansible_host: 13.94.171.246
7                      ansible_port: 22
8                      service_hostname: localhost
9          hecatehosts:
10             hosts:
11                 hecate1:

```

```

root@anjana17: /mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&_rds/hosts.yml anjana.yml --extra-vars={'data_overwrite_config_local=true','data_overwrite_ldap=true','data_overwrite_bbdd=true','data_overwrite_wf=true'} --skip-tags postgresql
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change, but still be user configurable on deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtpserverhosts

PLAY [Import role postgresql] *****

TASK [Gathering Facts] *****
ok: [postgresql]

PLAY [Import role zookeeper] *****

TASK [Gathering Facts] *****

```

9. Cuando finaliza presenta un reporte con las tareas de cada rol. Podemos ver como PostgreSQL lo ha saltado.

```

PLAY RECAP *****
apacheds1      : ok=27  changed=18  unreachable=0  failed=0  skipped=5  rescued=0  ignored=0
dritteta1     : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
edusa1        : ok=34  changed=19  unreachable=0  failed=0  skipped=9  rescued=0  ignored=0
fakesmtpserver1 : ok=10  changed=6   unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
hecate1       : ok=26  changed=11  unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
hermes1       : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
kerno1        : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
minerva1      : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
minio1        : ok=26  changed=17  unreachable=0  failed=0  skipped=5  rescued=0  ignored=1
monit1        : ok=7   changed=5   unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
portal1       : ok=34  changed=23  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
portuno1      : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
postgresql1   : ok=1   changed=0   unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
solr1         : ok=26  changed=11  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
viator1       : ok=24  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
zeus1         : ok=26  changed=10  unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
zookeeper1    : ok=27  changed=11  unreachable=0  failed=0  skipped=6  rescued=0  ignored=0

```

Despliegue single node

1. Creamos una máquina virtual
2. En la carpeta ansible-inventories hay que clonar de sample una nueva carpeta con un nombre identificador del nuevo entorno

3. Editar en hosts.yml los parámetros ansible_host, ansible_user y ansible_ssh_private_key_file para que Ansible pueda acceder.

```

ansible-inventories > sample > hosts.yml
1  all:
2  vars:
3  ansible_connection: smart
4  ansible_user: <ssh_user>
5  ansible_ssh_user: <ssh_user>
6  ansible_ssh_private_key_file: /root/.ssh/key.pem
7  ansible_become: true
8  force_service_mgr: systemd # systemd or initd
9  children:
10 directorhosts:
11   hosts:
12    director1:
13     # Nodo en el cual se ejecuta ansible (nodo director)
14     ansible_host: localhost
15     ansible_port: 22
16     service_hostname: localhost
17     ansible_python_interpreter: python3 # when problems try /usr/bin/python
18     ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
19     # ansible_connection: smart # Only if ansible_host is an IP or DNS
20 commonhosts:
21   hosts:
22    common1:
23     # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
24     # Cuentan las máquinas de Anjana, persistencias, ansible, etc
25     ansible_host: localhost
26     ansible_port: 22
27     service_hostname: localhost
28     ansible_python_interpreter: python3 # when problems try /usr/bin/python
29     ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
30    common2:
31     # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
32     # Cuentan las máquinas de Anjana, persistencias, ansible, etc
33     ansible_host: <server_ip>
34     ansible_port: 22
35     service_hostname: localhost
36     ansible_python_interpreter: python3 # when problems try /usr/bin/python
37 commonmicroserviceshosts:
38   hosts:
39    commonmicroservices1:
40     # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) ta
41     # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookee
42     ansible_host: <server_ip>
43     ansible_port: 22
44     service_hostname: localhost
45

```

4. Editar group_vars/all.yml con la versión deseada para el despliegue de Anjana

```

ansible-binary > ansible-inventories > sample > group_vars > all.yml
1  #-----#
2  #  Ansible Kit 4.a6  #
3  #-----#
4  version:
5  # MICROSERVICES -----#
6  edusa: 4.5.0
7  hecate: 4.5.0
8  hermes: 4.5.0
9  horus: 4.5.0
10 kerno: 4.5.0
11 minerva: 4.5.0
12 portuno: 4.5.0
13 tot: 4.5.0
14 viator: 4.5.0
15 zeus: 4.5.0
16 portal: 4.5.0
17 admin: 4.5.0
18 dritttesta: 4.5.0

```

5. Editar group_vars/all.yml con el usuario y contraseña proporcionado por Anjana para el acceso al repositorio de artefactos y resto de variables necesarias pre despliegue.

```

55  artifactory:
56    user: <artifactory_user>
57    password: <artifactory_password>
58    internal:
59      mavenurl: <internal>
60      mvnrepo: <internal>
61      npmurl: <internal>
62      rawurl: <internal>
63
64    external:
65      mavenurl: https://artifactory.anjanadata.org:8443/repository/anjanareleases
66      mvnrepo: anjanareleases
67      npmurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesnpm
68      rawurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesraw
69

```

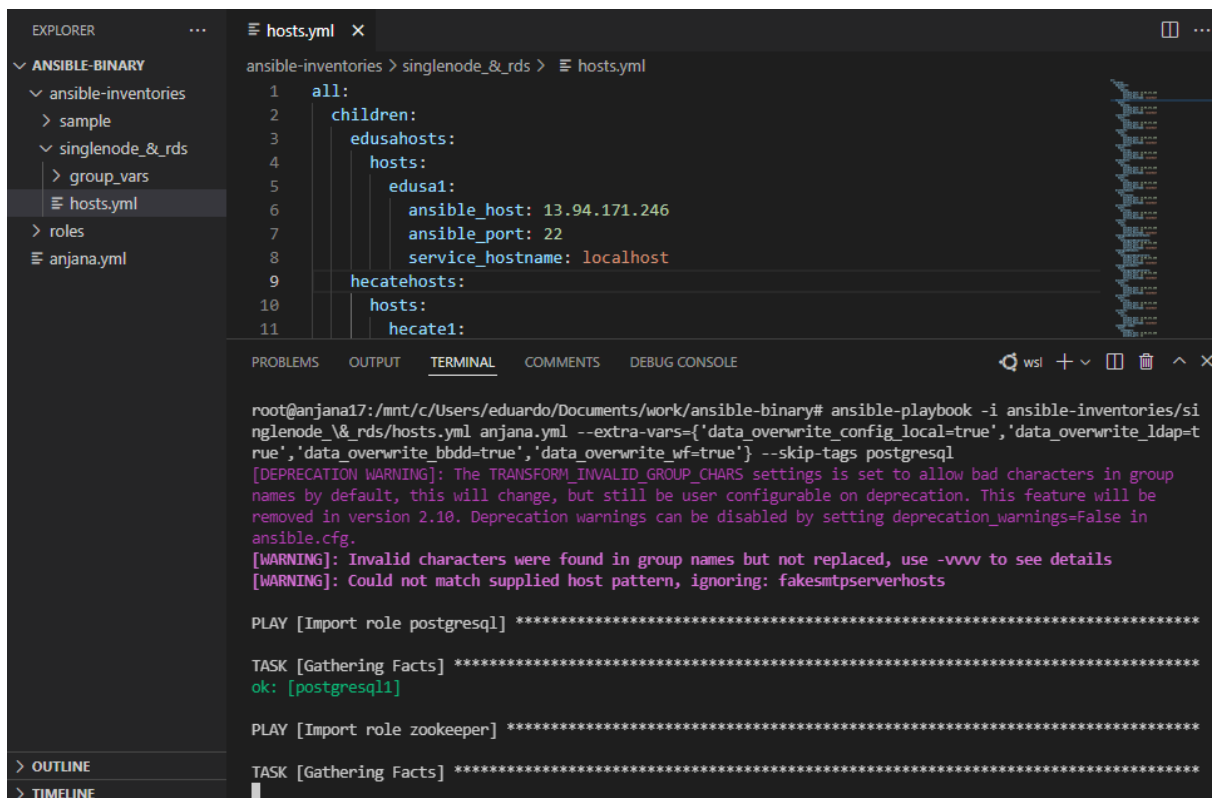
6. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos.
7. Una vez los siguientes puntos preparados ya podemos lanzar el despliegue:
 - a. acceso por ssh a la máquina
 - b. establecidas las variables con los datos correctos
 - c. acceso al repositorio de artefactos desde la máquina

En este caso lo vamos a lanzar para que despliegue con datos de ejemplos

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml
/opt/ansible/anjana.yml -v -tags anjana-sample

```



```

EXPLORER  ...  hosts.yml X
└─ ANSIBLE-BINARY
  └─ ansible-inventories
    └─ sample
      └─ singlenode_&rds
        └─ group_vars
          └─ hosts.yml
        └─ roles
          └─ anjana.yml

ansible-inventories > singlenode_&rds > hosts.yml
1  all:
2  |   children:
3  |     |   edusahosts:
4  |     |     |   hosts:
5  |     |     |     |   edusa1:
6  |     |     |     |     |   ansible_host: 13.94.171.246
7  |     |     |     |     |   ansible_port: 22
8  |     |     |     |     |   service_hostname: localhost
9  |     |   hecatehosts:
10 |     |   hosts:
11 |     |   hecate1:

PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE
root@anjana17:/mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&rds/hosts.yml anjana.yml --extra-vars={'data_overwrite_config_local=true','data_overwrite_ldap=true','data_overwrite_hbdd=true','data_overwrite_wf=true'} --skip-tags postgresql
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change, but still be user configurable on deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtserverhosts

PLAY [Import role postgresql] *****

TASK [Gathering Facts] *****
ok: [postgresq11]

PLAY [Import role zookeeper] *****

TASK [Gathering Facts] *****

```

8. Cuando finaliza presenta un reporte con las tareas de cada rol

```
PLAY RECAP *****
apacheds1      : ok=27  changed=18  unreachable=0  failed=0  skipped=5  rescued=0  ignored=0
dritttestal   : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
edusal        : ok=33  changed=18  unreachable=0  failed=0  skipped=10  rescued=0  ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
hecate1       : ok=26  changed=11  unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
hermes1       : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
kerno1        : ok=30  changed=11  unreachable=0  failed=0  skipped=4  rescued=0  ignored=0
minerval      : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
minio1        : ok=26  changed=17  unreachable=0  failed=0  skipped=5  rescued=0  ignored=1
monit1        : ok=7   changed=5   unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
portali       : ok=34  changed=23  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
portuno1      : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
postgresql1   : ok=40  changed=21  unreachable=0  failed=0  skipped=55  rescued=0  ignored=0
solr1         : ok=26  changed=12  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
viator1       : ok=24  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
zeus1         : ok=30  changed=12  unreachable=0  failed=0  skipped=4  rescued=0  ignored=0
zookeeper1    : ok=27  changed=15  unreachable=0  failed=0  skipped=6  rescued=0  ignored=0
```


Despliegue en HA + resiliencia Solr

Para el modo de despliegue HA + resiliencia creamos las siguientes máquinas virtuales.

- 2 vm de front
- 2 vm de back
- 3 vm de solr+zk
 - NOTA: Solo para 4.4, no hace falta cluster de SolR a partir de Anjana 23.1 porque la información pasa a ser completamente gestionada de forma autónoma por el producto y está persistida en bbdd relacional.
- 4 vm de MinIO (Cluster)
- PostgreSQL en RDS

Estos son los pasos a seguir para el ajuste de Ansible, posterior ejecución del despliegue y ajuste de configuración.

1. En la carpeta ansible-inventories hay que clonar sample a una nueva carpeta con un nombre identificador del nuevo entorno
2. Para desplegar en varios nodos desde Ansible hay que copiar las líneas de cada nodo en "hosts.yml" cambiando el nombre, ip, ...

Ejemplo de common:

```

commonhosts:
  hosts:
    common1:
      # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
      # Cuentan las máquinas de Anjana, persistencias, ansible, etc
      # En mode = remote este nodo no va hacer check de conexión con el repositorio de Anjana
      ansible_host: localhost
      ansible_port: 22
      service_type: utilities
      ansible_python_interpreter: python3 # when problems try /usr/bin/python
      ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
    common2:
      # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
      # Cuentan las máquinas de Anjana, persistencias, ansible, etc
      ansible_host: <server_ip> #FRONT1
      ansible_port: 22
      service_type: utilities
      ansible_python_interpreter: python3 # when problems try /usr/bin/python
    common3:
      # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
      # Cuentan las máquinas de Anjana, persistencias, ansible, etc
      ansible_host: <server_ip> #FRONT2
      ansible_port: 22
      service_type: utilities
      ansible_python_interpreter: python3 # when problems try /usr/bin/python
    common4:
      # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
      # Cuentan las máquinas de Anjana, persistencias, ansible, etc
      ansible_host: <server_ip> #BACK1
      ansible_port: 22
      service_type: utilities
      ansible_python_interpreter: python3 # when problems try /usr/bin/python

```

NOTA: para el rol de common hay que copiar las líneas tantas veces como nodos distintos tenga el despliegue. El rol de common se asegura de que todos los nodos cuenten con los permisos, usuarios y requisitos mínimos para el despliegue de anjana, así que tiene que ejecutarse en todos ellos.

Ejemplo de common-microservices:

```

commonmicroserviceshosts:
  hosts:
    commonmicroservices1:
      # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas usen Java 8
      # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, etc
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: utilities
    commonmicroservices2:
      # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas usen Java 8
      # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, etc
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: utilities
    commonmicroservices3:
      # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas usen Java 8
      # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, etc
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: utilities
    commonmicroservices4:
      # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas usen Java 8
      # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, etc
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: utilities

```

NOTA: de igual manera que para el rol de common, en el rol de common-microservices es necesario copiar tantas veces las líneas como nodos haya pero esta vez solo para los servicios que hagan uso de java. PostgreSQL y MiniIO por ejemplo quedarían excluidos.

Ejemplo de back:

```
edusahosts:
  hosts:
    edusa1:
      service_name: edusa
      service_port: 8888
      # REMEMBER to change the port in all.yml too (anjana.configURL)
      config_profile: native
      config_file: configserver
      idx: 50
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: microservices
    edusa2:
      service_name: edusa
      service_port: 8888
      # REMEMBER to change the port in all.yml too (anjana.configURL)
      config_profile: native
      config_file: configserver
      idx: 50
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: microservices
```

Ejemplo de front

```
portalhosts:
  hosts:
    portal1:
      service_name: apache2 # apache2 for Debian/Ubuntu OS | httpd for RedHat OS/Amazon Linux 2
      service_port: 80
      idx: 150
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: microservices
    portal2:
      service_name: apache2 # apache2 for Debian/Ubuntu OS | httpd for RedHat OS/Amazon Linux 2
      service_port: 80
      idx: 150
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: microservices
```

Ejemplo de Solr

```
solrhosts:
  hosts:
    solr1:
      service_name: solr
      service_port: 8983
      idx: 20
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: persistences
    solr2:
      service_name: solr
      service_port: 8983
      idx: 20
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: persistences
    solr3:
      service_name: solr
      service_port: 8983
      idx: 20
      ansible_host: <server_ip>
      ansible_port: 22
      service_type: persistences
```

3. Editar en hosts.yml los parámetros ansible_host, ansible_user y ansible_ssh_private_key_file para que Ansible pueda acceder.

```
ansible-inventories > sample > hosts.yml
1 all:
2 vars:
3   ansible_connection: smart
4   ansible_user: <ssh_user>
5   ansible_ssh_user: <ssh_user>
6   ansible_ssh_private_key_file: /root/.ssh/key.pem
7   ansible_become: true
8   force_service_mgr: systemd # systemd or initd
9 children:
10  directorhosts:
11    hosts:
12      director1:
13        # Nodo en el cual se ejecuta ansible (nodo director)
14        ansible_host: localhost
15        ansible_port: 22
16        service_hostname: localhost
17        ansible_python_interpreter: python3 # when problems try /usr/bin/python
18        ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
19        # ansible_connection: smart # Only if ansible_host is an IP or DNS
20  commonhosts:
21    hosts:
22      common1:
23        # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
24        # Cuentan las máquinas de Anjana, persistencias, ansible, etc
25        ansible_host: localhost
26        ansible_port: 22
27        service_hostname: localhost
28        ansible_python_interpreter: python3 # when problems try /usr/bin/python
29        ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
30      common2:
31        # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
32        # Cuentan las máquinas de Anjana, persistencias, ansible, etc
33        ansible_host: <server_ip>
34        ansible_port: 22
35        service_hostname: localhost
36        ansible_python_interpreter: python3 # when problems try /usr/bin/python
```

4. Solr y Zookeeper en modo cluster requieren de los siguientes ajustes:
 - a. NOTA: A partir de Anjana 23.1 no es necesario un cluster de Zookeeper ni de SolR porque la información que almacenaba SolR pasa a ser completamente gestionada de forma autónoma por el producto y está persistida en bbdd relacional.
 - b. En “sample/group_vars/solrhosts.yml” ajustar la ip de conexión con cada nodo de zookeeper desde los nodos solr (dejar localhost en singlenode)

a. Versiones de producto

```

ansible-binary > ansible-inventories > sample > group_vars > A all.yml
1  #-----#
2  #  Ansible Kit 4.a6  #
3  #-----#
4  versión:
5  # MICROSERVICES -----#
6  edusa: 4.5.0
7  hecate: 4.5.0
8  hermes: 4.5.0
9  horus: 4.5.0
10 kerno: 4.5.0
11 minerva: 4.5.0
12 portuno: 4.5.0
13 tot: 4.5.0
14 viator: 4.5.0
15 zeus: 4.5.0
16 portal: 4.5.0
17 admin: 4.5.0
18 drittesta: 4.5.0

```

- b. Ajustar sección etc.hosts donde hay que indicar los alias de las máquinas del entorno anjana y su persistencia. En la configuración que vienen con el kit de Ansible se usan muchos esos alias, por lo que es mejor dejar los alias que están pero ajustando las ips y añadir alias nuevos al ser HA añadiendo node1, node2, ... donde deba. Así en todo el kit de Ansible ya se pueden utilizar los alias si se prefiere antes que la ip pero es opcional su uso.

```

ansible-inventories > ha > group_vars > all.yml
49 etc:
50 hosts:
51 - "10.0.0.1 indexservicenode1 zkservicenode1 indexservice zkservice"
52 - "10.0.0.2 indexservicenode2 zkservicenode2"
53 - "10.0.0.3 indexservicenode2 zkservicenode2"
54 - "10.0.0.4 s3servicenode1 s3service"
55 - "10.0.0.5 s3servicenode2"
56 - "10.0.0.6 s3servicenode3"
57 - "10.0.0.7 s3servicenode4"
58 - "10.0.0.8 rdbservicenode1 rdbservice"
59 - "10.0.0.9 hecateservicenode1 edusaservicenode1 zeusservicenode1 hermesservicenode1 viatorservicenode1 minivaservicenode1 kernoservicenode1 totservicenode1 portunoservicenode1"
60 - "10.0.0.9 hecateserver edusaserver zeusservicenode2 hermesservicenode2 viatorservicenode2 minivaservicenode2 kernoservicenode2 totservicenode2 portunoservicenode2"
61 - "10.0.0.10 hecateservicenode2 edusaservicenode2 zeusservicenode2 hermesservicenode2 viatorservicenode2 minivaservicenode2 kernoservicenode2 totservicenode2 portunoservicenode2"
62

```

- c. usuario y contraseña proporcionado por Anjana para el acceso al repositorio de artefactos

```

# CONNECTION STRINGS -----#
nexusurl: https://releases.anjanadata.org
artifactory:
  user: <artifactory_user>
  password: <artifactory_password>
  internal:
    mavenurl: <internal>
    mvnrepo: <internal>
    npmurl: <internal>
    rawurl: <internal>
  external:
    mavenurl: '{{nexusurl}}/repository/releasesmvn'
    mvnrepo: releasesmvn
    npmurl: '{{nexusurl}}/repository/releasesnpm'
    rawurl: '{{nexusurl}}/repository/releasesraw'

```

- d. Licencia de Anjana

```

ansible-inventories > ha > group_vars > all.yml
-----
125     license:
126         installationCode: AAAAAAAA-InstallationCode-AAAAAAA
127         privateKey: AAAAAAAA-PrivateKey-AAAAAAA
128         anjanaPublicKey: AAAAAAAA-AnjanaPublicKey-AAAAAAA
129

```

- e. La variable configURL ajusta los descriptores de servicio de los microservicios de Anjana para pedir la configuración a los 2 servidores de config en HA

```

ansible-inventories > localhost > group_vars > all.yml
-----
169     anjana:
170         folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
171         configURL: optional:configserver:http://edusaserver:8888 # HA -> optional:configserver:http://edusaservernode1:8888,optional:configserver:http://edusaservernode2:8888
172         hecateReplicas: http://hecatserver:50761/eureka # HA -> http://hecatserver:50761/eureka,http://hecatserver2:50761/eureka
173         license:

```

- f. Para habilitar el HA en Hecate tenemos que ir a la variable hecateReplicas y ajustar con los 2 nodos como indica el comentario.

```

ansible-inventories > localhost > group_vars > all.yml
-----
169     anjana:
170         folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
171         configURL: optional:configserver:http://edusaserver:8888 # HA -> optional:configserver:http://edusaservernode1:8888,optional:configserver:http://edusaservernode2:8888
172         hecateReplicas: http://hecatserver:50761/eureka # HA -> http://hecatserver:50761/eureka,http://hecatserver2:50761/eureka
173         license:

```

- 7. Este punto solo a partir de Anjana 23.1, en el template del descriptor de servicios de Hermes hay que añadir un parámetro en el comando de arranque. Se trata de el 2º valor del array de nodos de Portuno donde pedir la configuración:

```

ansible-inventories > localhost > templates > services_core > systemd > hermes.service.j2
-----
1  {{ ('Generated by:\nAnsible Kit 4.40\nFrom:\n'+Inventory_dir+'/templates/services_core/systemd/'+service_name+'.service.j2') | comment('plain', prefix=('#'*100), postfix=('#'*100)) }}
2
3
4  [Unit]
5  Description=Anjana hermes server
6  Requires=network.target
7  After=network.target edusa.service
8
9  [Service]
10 Environment+HOSTNAME={{ ansible_host }}
11 {% if (ansible_pkg_mgr == "yum" or ansible_pkg_mgr == "dnf") -%}
12 Environment=JAVA_HOME={{(installation.javaPATH)}/jre-1.8.0-openjdk
13 {% else %}
14 Environment=JAVA_HOME={{(installation.javaPATH)}/java-8-openjdk-amd64
15 {% endif %}
16 WorkingDirectory={{ hermes.install.path }}
17 TimeoutStopSec=20
18 ExecStart={{ if (installation.type == "external") }}{{ hermes.install.path }}/hermeslauncher{{ endif }} {{ if (ansible_pkg_mgr == "yum" or ansible_pkg_mgr == "dnf") -%}}{{(installation.javaPATH)}/jre-1.8.0-openjdk/bin/java{{ else %}}{{(installation.javaPATH)}/java-8-openjdk-amd64/bin/java{{ endif }} {{ hermes.install.javaopts }} -Danjana.properties.url[0]=http://portunoserver:8998/internal/v2/appconf?prefix=hermes -javaagent:{{(hermes.install.hibernateagent.path)}/jar-agent-hibernate.jar -jar {{ hermes.install.jar }} --spring.cloud.config.uri={{ hermes.install.edusaurl }} --spring.profiles.active={{ config_profile }} --spring.cloud.config.failFast={{ installation.failFast }} {% if not (installation.eurekaPreferIpAddress) -%}-eureka.instance.preferIpAddress=false{{ endif %}
19
20

```

- 8. Este punto solo a partir de Anjana 23.1, en todos los templates de microservicios que conectan con Portuno añadir el segundo nodo

```

ansible > ansible-inventories > ha > templates > config_core > kerno.config.4.5.0.yaml.j2
-----
59     anjana:
60         file:
61             explorer:
62                 clientType: {{(persistences.s3.type|upper)}} # AWS_S3 ó MINIO
63                 folder:
64                     textArea: {{ (persistences.s3.buckets | selectattr("bucket","=", "textarea")|list)[0]["name"]}}
65                     excelImport: {{ (persistences.s3.buckets | selectattr("bucket","=", "imports")|list)[0]["name"]}}
66                     attach: {{ (persistences.s3.buckets | selectattr("bucket","=", "dsa")|list)[0]["name"]}}
67                 region: {{(persistences.s3.region)}} #solo para AWS_S3
68                 url: {{(persistences.s3.host)}}:{{(persistences.s3.port)}} #solo para MINIO
69                 clientId: {{(persistences.s3.access_key)}}
70                 secretKey: {{(persistences.s3.secret_key)}}
71             properties:
72                 wait: 10
73                 retry: 3
74                 url:
75                     - http://portunoservernode1:{{(hostvars["portuno1"]["service_port"])/internal/v2/appconf?prefix=kerno
76                     - http://portunoservernode2:{{(hostvars["portuno2"]["service_port"])/internal/v2/appconf?prefix=kerno
77                 client_id: a2Vydm8uc2Vydm1jZQ==

```

9. En todos los microservicios con Swagger (Hermes, Kerno, Minerva, Zeus, Portuno) hay que editar la url de este

```
45  swagger:
46    server:
47      url:
48        # change ${HOSTNAME} to anjana url
49        - https://anjana_url.com/gateway
```

10. Este punto solo para la versión anterior a Anjana 23.1 o en 23.1 si se quiere conservar el cluster de Solr por carga de peticiones o de datos, en el archivo de configuración de Minerva hay que añadir los 3 nodos para que minerva pueda balancear de nodo de Solr si alguno no responde:

```
ansible-inventories > ha > templates > config_core > minerva.config.4.4.0.yml.j2
29  solr:
30    urls:
31      - http://indexservicenode1:8983/solr
32      - http://indexservicenode2:8983/solr
33      - http://indexservicenode3:8983/solr
34    client_type: CLOUD
```

11. En group_vars/portalhosts.yml hay que ajustar los endpoints de los proxypass con los nuevos nodos. Los proxies necesarios para los buckets se configurarán automáticamente durante el despliegue indicando los nodos de minio bajo la variable minio_nodes

```
ansible-inventories > sample > group_vars > portalhosts.yml
30  balancer:
31    swagger: |
32      # BalancerMember http://viatorservernode1:8085/swagger
33      # BalancerMember http://viatorservernode1:8085/swagger
34    swaggerapi: |
35      # BalancerMember http://viatorservernode1:8085/v3/api-docs
36      # BalancerMember http://viatorservernode1:8085/v3/api-docs
37    gateway: |
38      BalancerMember http://viatorservernode1:8085
39      BalancerMember http://viatorservernode1:8085
40    portuno: |
41      BalancerMember http://portunoserver:8998/portuno
42      BalancerMember http://portunoserver:8998/portuno
43    minio_nodes:
44      #- '{{persistences.s3.host}}' ## Main MinIO node. Do not modify this one
45      # For HA Deployment / MinIO Cluster
46      - http://minioservicenode1
47      - http://minioservicenode2
48      - http://minioservicenode3
49      - http://minioservicenode4
50
```

12. Solicitar acceso en el firewall del repositorio de artefactos de Anjana desde la ip pública del servidor de Ansible que vaya a desplegar.

13. Una vez los siguientes puntos preparados ya podemos lanzar el despliegue:

- a. acceso por ssh a la máquina
- b. seteadas las variables con los datos correctos
- c. acceso al repositorio de artefactos desde la máquina

RECOMENDACIÓN: Lanzar el comando siguiente de ping de Ansible comprueba que tenemos conectividad con la máquina a desplegar antes de lanzar el playbook

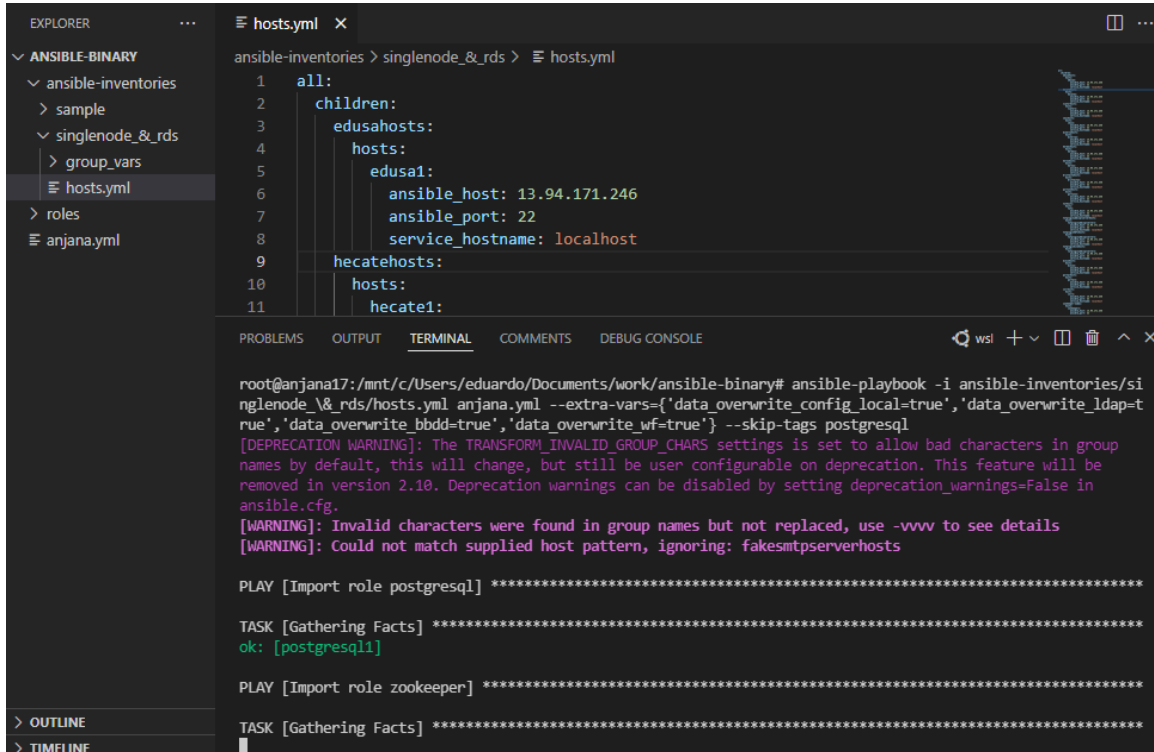
```
sudo ansible -i /opt/ansible/ansible-inventories/sample/hosts.yml all -m ping
```

```
root@anjana17:/mnt/c/Users/eduardo/Documents/work/ansible-binary-Copy# ansible -i ansible-inventories/sample/hosts.yml all -m ping
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change with
deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.c
onfig.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
viator1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hecate2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hecate1 | SUCCESS => {
  "ansible_facts": {
```

14. Lanzamos el despliegue

En este caso lo vamos a lanzar para que despliegue con datos de ejemplos

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml /opt/ansible/anjana.yml -tags anjana-sample
```



The screenshot shows a VS Code editor with a file explorer on the left showing the project structure. The main editor displays the content of `hosts.yml` from the `ansible-inventories > singlenode_&_rds` directory:

```
1 all:
2   children:
3     edusahosts:
4       hosts:
5         edusa1:
6           ansible_host: 13.94.171.246
7           ansible_port: 22
8           service_hostname: localhost
9   hecatehosts:
10    hosts:
11    hecate1:
```

The terminal at the bottom shows the execution of the command:

```
root@anjana17:/mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&_rds/hosts.yml anjana.yml --extra-vars={'data overwrite_config_local=true', 'data overwrite_ldap=true', 'data overwrite_bbdd=true', 'data overwrite_wf=true'} --skip-tags postgresql
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change, but still be user configurable on deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtpserverhosts

PLAY [Import role postgresql] *****

TASK [Gathering Facts] *****
ok: [postgresql]

PLAY [Import role zookeeper] *****

TASK [Gathering Facts] *****
```

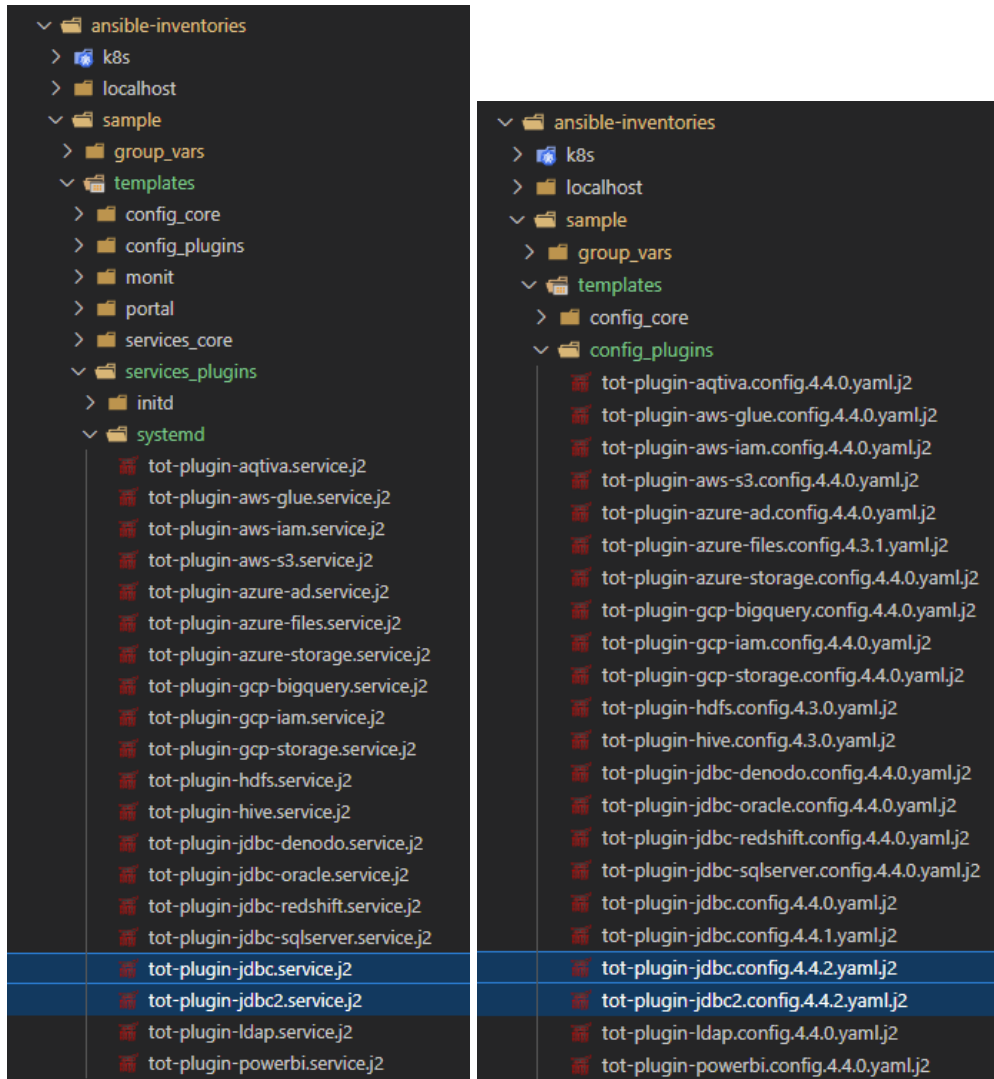
15. Cuando finaliza presenta un reporte con las tareas de cada rol

```
PLAY RECAP *****
*****
apacheds1      : ok=23  changed=10  unreachable=0  failed=0  skipped=10  rescued=0  ignored=0
drittesta1    : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
drittesta2    : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
edusa1        : ok=31  changed=14  unreachable=0  failed=0  skipped=13  rescued=0  ignored=0
edusa2        : ok=31  changed=14  unreachable=0  failed=0  skipped=13  rescued=0  ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0  failed=0  skipped=0   rescued=0  ignored=0
hecate1       : ok=27  changed=12  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
edusa2        : ok=31  changed=14  unreachable=0  failed=0  skipped=13  rescued=0  ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0  failed=0  skipped=0   rescued=0  ignored=0
hecate1       : ok=27  changed=12  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
hecate2       : ok=27  changed=12  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
hermes1       : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
hermes2       : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
kerno1        : ok=31  changed=12  unreachable=0  failed=0  skipped=4   rescued=0  ignored=0
kerno2        : ok=31  changed=12  unreachable=0  failed=0  skipped=4   rescued=0  ignored=0
minerva1      : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
minerva2      : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
minio1        : ok=25  changed=14  unreachable=0  failed=0  skipped=7   rescued=0  ignored=1
monit1        : ok=7   changed=5   unreachable=0  failed=0  skipped=1   rescued=0  ignored=0
porta11       : ok=35  changed=29  unreachable=0  failed=0  skipped=2   rescued=0  ignored=0
porta12       : ok=35  changed=29  unreachable=0  failed=0  skipped=2   rescued=0  ignored=0
portuno1      : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
portuno2      : ok=27  changed=10  unreachable=0  failed=0  skipped=8   rescued=0  ignored=0
```

Generación de nueva instancia de un plugin ya existente

Los puntos a editar/crear son los siguientes:

- Debemos copiar tantos templates como servicios queramos en el apartado de `/opt/ansible/ansible-inventories/<inventario>/templates/service_plugins` y copiar el archivo de ese plugin cambiando el nombre a `tot-plugin-XXX2.service.j2`. Hacemos lo mismo para los templates de configuración añadiendo un segundo template de la versión que corresponda.



- En el archivo de inventario elegido `/opt/ansible/ansible-inventories/<inventario>/hosts.yml` se añaden nuevas entradas tantas como instancias del mismo plugin se quieran añadir, cambiando el puerto, nombre del servicio, nombre de aplicación y perfil de configuración.

```
hosts.yml M X
ansible-binary > ansible-inventories > sample > hosts.yml > {} all
296     ansible_port: 22
297     service_type: plugins
298     totpuginjdbchosts:
299     hosts:
300     totpuginjdbc1:
301     app_name: tot-plugin-jdbc1
302     service_name: tot-plugin-jdbc
303     service_port: 15001
304     tot_service_port: 15000
305     config_profile: default1
306     idx: 280
307     ansible_host: <server_ip>
308     ansible_port: 22
309     service_type: plugins
310     totpuginjdbc2:
311     app_name: tot-plugin-jdbc2
312     service_name: tot-plugin-jdbc
313     service_port: 15031
314     tot_service_port: 15000
315     config_profile: default2
316     idx: 280
317     ansible_host: <server_ip>
318     ansible_port: 22
319     service_type: plugins
```

- En caso de que la configuración de la segunda instancia sea diferente, también será necesario modificar el perfil de configuración para que se aplique correctamente.

```
hosts.yml M X
ansible-binary > ansible-inventories > sample > hosts.yml > {} all
296     ansible_port: 22
297     service_type: plugins
298     totpuginjdbchosts:
299     hosts:
300     totpuginjdbc1:
301     app_name: tot-plugin-jdbc1
302     service_name: tot-plugin-jdbc
303     service_port: 15001
304     tot_service_port: 15000
305     config_profile: default1
306     idx: 280
307     ansible_host: <server_ip>
308     ansible_port: 22
309     service_type: plugins
310     totpuginjdbc2:
311     app_name: tot-plugin-jdbc2
312     service_name: tot-plugin-jdbc
313     service_port: 15031
314     tot_service_port: 15000
315     config_profile: default2
316     idx: 280
317     ansible_host: <server_ip>
318     ansible_port: 22
319     service_type: plugins
```

NOTA: Si no se ajustan los perfiles de configuración de acuerdo a la necesidad, la configuración de todas las instancias del mismo plugin será sobrescrita con el último perfil de configuración indicado. **En multi-instancia, no deberá de existir ningún perfil llamado default**, si no, todas las instancias del plugin cogerán esa configuración.

EJEMPLO: Desplegando multi-instancia para un plugin como tot-plugin-jdbc donde se quiere que cada instancia apunte a una tecnología (base de datos) diferente, habría que modificar en el archivo hosts.yml previamente mencionado el puerto, nombre de servicio, nombre de aplicación y perfil de configuración, y en este caso también, en la configuración que se despliega, el driver que se va a usar para la conexión de base de datos, url, entre otras cosas.

- Lanzamos la actualización de la configuración para incluir el nuevo template con el comando:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml  
/opt/ansible/anjana.yml --tags update-anjana-config
```

- Para desplegar la nueva instancia debemos lanzar el role del plugin con el siguiente comando:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/localhost/hosts.yml  
/opt/ansible/anjana.yml --tags tot-plugin-jdbc
```

Cambio de nombres de microservicios

Para el cambio de nombres de un microservicio de Anjana necesitará seguir los siguientes pasos:

1. En el archivo <ruta_inventario>/hosts.yml hay que indicar el nuevo nombre en la variable `service_name`

```

43     edusahosts:
44         hosts:
45             edusa1:
→ 46+                 service_name: edusa1
47                 service_port: 8888
48                 config_profile: native
49                 config_file: configserver
50                 idx: 50
51                 ansible_host: localhost
52                 ansible_port: 22
53                 service_type: microservices
54     hecatehosts:
55         hosts:
56             hecate1:
→ 57+                 service_name: hecate1
58                 service_port: 50761
59                 config_profile: default
60                 idx: 60
61                 ansible_host: localhost
62                 ansible_port: 22
63                 service_type: microservices
64     viatorhosts:
65         hosts:
66             viator1:
→ 67+                 service_name: viator1
68                 service_port: 8085
69                 config_profile: default

```

- a.
2. Los templates de la configuración de Anjana deben ser sustituidos con el nuevo nombre
 - a. Cambiando '`<ruta_inventario>/templates/config_core/kerno.config.4.5.0.yaml1.j2`' por '`<ruta_inventario>/templates/config_core/kerno1.config.4.5.0.yaml1.j2`'
3. Al igual que el punto anterior, hay que hacer el mismo procedimiento con los servicios
 - a. Cambiando '`<ruta_inventario>/templates/service_core/systemd/kerno.service.j2`' por '`<ruta_inventario>/templates/service_core/systemd/kerno1.service.j2`'
4. En los siguientes templates de configuración de Anjana hay que añadir el nuevo nombre de los distintos microservicios con los que se comunica a través de Hecate.
 - a. Drittesta

```

ansible-inventories > localhost > templates > config_core > drittesta.config.4.5.0.yaml.j2
30
31   client:
32     defaultZone: http://hecatserver:{{hostvars["hecate1"]]["service
33   internal:
34     kerno:
35       name: kerno1
36     zeus:
37       name: zeus1
38

```

i.

b. Hermes

```

ansible-inventories > localhost > templates > config_core > hermes.config.4.5.0.yaml.j2
80  client:
81  |   defaultZone: http://hecatserver:{{hostvars["hecate1"]]["serv
82  |   internal:
83  |     kerno:
84  |       name: kerno1
85  |     zeus:
86  |       name: zeus1
87  |     minerva:
88  |       name: minerva1
89  |     tot:
90  |       name: tot1
91  |       path: tot
92  |     portuno:
93  |       name: portuno1
94  |
95

```

i.

c. Kerno

```

ansible-inventories > localhost > templates > config_core > kerno.config.4.5.0.yaml.j2
53  #database-platform: com.anjana.core.hibernate.dialect.AnjanaOracleDial
54
55  eureka:
56  |   client:
57  |     defaultZone: http://hecatserver:{{hostvars["hecate1"]]["service_port
58  |     internal:
59  |       hermes:
60  |         name: hermes1
61  |       zeus:
62  |         name: zeus1
63  |       minerva:
64  |         name: minerva1
65  |       tot:
66  |         name: tot1
67  |         path: tot
68  |       portuno:
69  |         name: portuno1
70  |       drittesta:
71  |         name: drittesta1
72  |
73

```

i.

d. Minerva

```
ansible-inventories > localhost > templates > config_core > minerva.config.4.5.0.yaml.j2
60 eureka:
61   client:
62     serviceUrl:
63     defaultZone: http://hecatserver:{{hostvars["hecate1"]["service
64   internal:
65     kerno:
66       name: kerno1
67     hermes:
68       name: hermes1
69     zeus:
70       name: zeus1
71     portuno:
72       name: portuno1
73
```

- i.
- e. Portuno

```
ansible-inventories > localhost > templates > config_core > portuno.config.4.5.0.yaml.j2
21 eureka:
22   client:
23     serviceUrl:
24     defaultZone: http://hecatserver:{{hostvars["hecate1"]["service_port
25   internal:
26     kerno:
27       name: kerno1
28     hermes:
29       name: hermes1
30     zeus:
31       name: zeus1
32     minerva:
33       name: minerva1
34     tot:
35       name: tot1
36
```

- i.
- f. Tot

```
drittista.config.4.5.0.yaml.j2 (Working Tree) M tot.config.4.5.0.yaml.j2 M X
ansible-inventories > localhost > templates > config_core > tot.config.4.5.0.yaml.j2
27 eureka:
28   client:
29     serviceUrl:
30     defaultZone: http://hecatserver:{{hostvars["hecate1"]["service_por
31   internal:
32     kerno:
33       name: kerno1
34     portuno:
35       name: portuno1
36     zeus:
37       name: zeus1
38
```

- i.
- g. Viator


```

ansible-inventories > localhost > templates > config_core > viator.config.4.5.0.yaml.j2
21
22 application:
23   services:
24     kerno: kerno1
25     hermes: hermes1
26     zeus: zeus1
27     viator: viator1
28     portuno: portuno1
29     minerva: minerva1

```

- i.
- h. Zeus

```

ansible-inventories > localhost > templates > config_core > zeus.config.4.5.0.yaml.j2
61 eureka:
62   client:
63     serviceUrl:
64       defaultZone: http://hecatserver:{{hostvars["hecate1"]["service_port"]}}/eureka
65   internal:
66     kerno:
67       name: kerno1
68     drittesta:
69       name: drittesta1
70     portuno:
71       name: portuno1
72     hermes:
73       name: hermes1
74     minerva:
75       name: minerva1
76

```

- i.

5. Añadir al descriptor de servicio de Hermes el nuevo nombre de Hermes y el de Portuno para su peculiaridad de comunicación antes de la carga de la configuración:
 - a. Hay que poner junto al argumento '-Danjana.properties.url[0]='

```
-Dspring.application.name=hermes1 -Deureka.internal.portuno.name=portuno1
```

```

ansible-inventories > localhost > group_vars > hermeshosts.yml
1 ---
2 hermes:
3   install:
4     jar: /com/anjana/hermes/{{ version.hermes }}/hermes-{{ version.hermes }}.jar
5     launcher: /com/anjana/hermeslauncher/{{ version.hermes }}/hermeslauncher-{{
6     version.hermes }}.bin
7     path: /{{anjana.folder}}/hermes
8     backup: /{{anjana.folder}}/backup/hermes
9     edusaurl: '{{anjana.configURL}}'
10    config: /{{anjana.folder}}/hermes
11    serviceuser: '{{installation.owner.user}}'
12    javaopts: '-XX:+UseG1GC -XX:+UseStringDeduplication -Xmx4G -Danjana.
13    properties.url[0]=http://portunoserver:8998/internal/v2/appconf?prefix=hermes
14    -Dspring.application.name=hermes1 -Deureka.internal.portuno.name=portuno1'
15    groupUrl: 'com.anjana'
16    nameUrl: 'hermes'
17    hibernateagent:

```

- b.

6. Crear un link simbólico de las carpetas de la configuración de Anjana con el nombre nuevo
 - a. El comando sería:

```
cd /opt/data/configrepo/
ln -s kerno kerno1
```


- b. También tocaremos la variable `import_role`, en la cual pondremos todo a `false` menos el plugin de `jdbc`.

```
all.yml M X
ansible-binary > ansible-inventories > sample > group_vars > all.yml >
194 import_role:
195   # Persistences
196   postgresql: false
197   zookeeper: false
198   solr: false
199   apacheds: false
200   minio: false
201   # Core Anjana
202   portal: false
203   edusa: false
204   hecate: false
205   portuno: false
206   kerno: false
207   zeus: false
208   drittesta: false
209   minerva: false
210   viator: false
211   hermes: false
212   tot: false
213   horus: false
214   # Plugins Anjana
215   tot_plugin_aws_iam: false
216   tot_plugin_aws_glue: false
217   tot_plugin_aws_s3: false
218   tot_plugin_azure_ad: false
219   tot_plugin_azure_files: false
220   tot_plugin_azure_storage: false
221   tot_plugin_gcp_bigquery: false
222   tot_plugin_gcp_iam: false
223   tot_plugin_gcp_storage: false
224   tot_plugin_hdfs: false
225   tot_plugin_hive: false
226   tot_plugin_jdbc: true
```

- c. En la variable `configtemplates`, comentaremos todo el core, y en la parte de plugins, descomentaremos el de `jdbc`

```
261 configtemplates:
262   core:
263     # - drittesta
264     # - hecate
265     # - hermes
266     # - kerno
267     # - minerva
268     # - portuno
269     # - tot
270     # - viator
271     # - zeus
272     # - horus
273   plugins:
274     # - tot-plugin-aws-glue
275     # - tot-plugin-aws-iam
276     # - tot-plugin-aws-s3
277     # - tot-plugin-azure-ad
278     # - tot-plugin-azure-files
279     # - tot-plugin-azure-storage
280     # - tot-plugin-gcp-bigquery
281     # - tot-plugin-gcp-iam
282     # - tot-plugin-gcp-storage
283     # - tot-plugin-hive
284     # - tot-plugin-hdfs
285     - tot-plugin-jdbc
286     # - tot-plugin-jdbc-denodo
287     # - tot-plugin-jdbc-oracle
288     # - tot-plugin-jdbc-redshift
289     # - tot-plugin-jdbc-sqlserver
290     # - tot-plugin-ldap
291     # - tot-plugin-aqtiva
292     # - tot-plugin-powerbi
293     # - tot-plugin-ranger
294     # - tot-plugin-tableau
```

4. Modificaremos el hosts.yml, sustituyendo en los hosts de tot-plugin-jdbc y anjana-utility por la IP de la máquina del plugin. También modificaremos las variables de usuario, usuario_ssh y ansible_ssh_private_key_file.
5. Tenemos que modificar el template del servicio del microservicio. Para ello vamos a la ruta sample/templates/services_plugins/systemd y editamos el fichero necesario. Una vez en el fichero, tenemos que editarlo de la siguiente manera.

```
# Línea que quitar
--spring.config.import={{ <nombre_microservicio>.install.edusaurl }}

# Líneas a añadir
--spring.cloud.config.enabled=false
--spring.config.additional-location=<ruta_configuración>/application-default.yaml

# Líneas a cambiar
--spring.cloud.config.failFast={{ installation.failFast }} por
--spring.cloud.config.failFast=false
```

6. Una vez hemos cambiado todo, lanzamos los siguientes comandos

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/sample/hosts.yml
/opt/ansible/anjana.yml -t tot-plugin-jdbc,update-anjana-config
```

Arrancar instancia en HA degradado (Falta de algún nodo)

Si ocurre la situación que 1 nodo de Anjana en HA no está disponible tenemos que realizar una serie de actuaciones manuales para levantar Anjana de nuevo.

1. En el archivo all.yml tenemos que quitar el nodo de Edusa que no está disponible en la variable anjana.configURL:

```
ansible-inventories > ha > group_vars > all.yml
122  anjana:
123     folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
124     # configURL: optional:configserver:http://edusaserver:8888
125     # HA
126     configURL: optional:configserver:http://edusaservernode1:8888optional:configserver:http://edusaservernode2:8888
127     license:
```

2. En el template del descriptor de servicio de Hecate borrar el nodo no disponible

```

ansible-inventories > ha > templates > services_core > systemd > hecate.service.j2
1  {{ ('Generated by:\nAnsible Kit 4.a6\nFrom:\n'+inventory_dir+'/templates/services_core/systemd/'
   +service_name+'.service.j2') | comment('plain', prefix=('#'*100), postfix=('#'*100)) }}
2
3  [Unit]
4  Description=Anjana hecate server
5  Requires=network.target
6  After=network.target edusa.service
7
8  [Service]
9  Environment=HOSTNAME={{ ansible_host }}
10 {% if (ansible_pkg_mgr == "yum" or ansible_pkg_mgr == "dnf") -%}
11     Environment=JAVA_HOME={{ installation.javaPATH }}/jre-1.8.0-openjdk
12 {% else %}
13     Environment=JAVA_HOME={{ installation.javaPATH }}/java-8-openjdk-amd64
14 {% endif %}
15 Environment=HECATE_REPLICAS=http://10.152.20.231:50761/eureka http://10.152.20.239:50761/eureka
16 WorkingDirectory={{ hecate.install.path }}
17 TimeoutStopSec=20
18 ExecStart=% if (installation.type == "external") %>{{ hecate.install.path }}/hecatelauncher{%
   endif %} {% if (ansible_pkg_mgr == "yum" or ansible_pkg_mgr == "dnf") -%}{{ installation.
   javaPATH }}/jre-1.8.0-openjdk/bin/java{% else %}{{ installation.javaPATH }}/java-8-openjdk-amd64/
   bin/java{% endif %} {{ hecate.install.javaopts }} -javaagent:{{ hecate.install.hibernateagent.

```

3. En todos los templates de configuración de microservicios de Anjana donde aparezca la variable para conectar con Hecate hay que borrar el nodo no disponible

```

eureka:
  client:
    serviceUrl:
      defaultZone: http://10.152.20.231:50761/eureka http://hecateservernode2:50761/eureka

```

4. Este punto solo a partir de Anjana 23.1, en el template de Hermes hay que quitar un parámetro en el comando de arranque. Se trata de el valor del array de nodo de Portuno que no está disponible

```

ansible-binary > ansible-inventories > sample > group_vars > hermeshosts.yml
1  hermes:
2
3  install:
4  jar: /com/anjana/hermes/{{ version.hermes }}/hermes-{{ version.hermes }}.jar
5  launchers: /com/anjana/hermes/launcher/{{ version.hermes }}/hermeslauncher-{{ version.hermes }}.bin
6  path: {{(anjana_folder)}}/hermes
7  backup: {{(anjana_folder)}}/backup/hermes
8  edusaurl: '{{(anjana_configURL)}}'
9  config: {{(anjana_folder)}}/hermes
10 serviceusers: '{{installation.owner.user}}'
11 javaopts: '-XX:+UseG1GC -Danjana.properties.url[0]=http://portunoservertime:8998/Internal/v2/appconf?prefix=hermes -Dhermes.nodes={{(hermes_nodes)}} -Dhermes.replicas={{(hermes_replicas)}}'
12 groupUrl: 'com.anjana'
13 name: 'hermes'

```

5. Este punto solo a partir de Anjana 23.1, en todos los templates de configuración de microservicios de Anjana donde aparezca la variable para conectar con Portuno hay que borrar el nodo no disponible

```

ansible > ansible-inventories > ha > templates > config_core > kerno.config.4.5.0.yaml,j2
59  anjana:
60  file:
61  explorer:
62  clientType: {{persistences.s3.type|upper}} # AWS_S3 ó MINIO
63  folder:
64  textarea: {{ (persistences.s3.buckets | selectattr("bucket","=", "textarea")|list)[0]["name"]}
65  excelImport: {{ (persistences.s3.buckets | selectattr("bucket","=", "imports")|list)[0]["name"]}
66  attach: {{ (persistences.s3.buckets | selectattr("bucket","=", "dsa")|list)[0]["name"]}
67  region: {{persistences.s3.region}} #solo para AWS_S3
68  url: {{persistences.s3.host}}:{{persistences.s3.port}} #solo para MINIO
69  clientId: {{persistences.s3.access_key}}
70  secretKey: {{persistences.s3.secret_key}}
71  properties:
72  wait: 10
73  retry: 3
74  url:
75  - http://portunosexplorer:{{hostvars["portuno2"]["service_port"]}}/internal/v2/appconf?prefix=kerno
76  - http://portunosexplorer:{{hostvars["portuno2"]["service_port"]}}/internal/v2/appconf?prefix=kerno
77  client_id: a2Vybm8uc2VydmljZQ==

```

6. En todos los templates de configuración de microservicios de Anjana donde aparezca la variable para conectar con Zeus para obtener el token hay que cambiar el nodo por el disponible

```

ansible > ansible-inventories > ha > templates > config_core > kerno.config.4.5.0.yaml,j2
79  security:
80  oauth2:
81  resource:
82  tokenInfoUri: http://zeusservernode2:{{hostvars["zeus2"]["service_port"]}}/internal/v1/auth/validate-token
83

```

7. Para actualizar el entorno con las variables cambiadas en Ansible tenemos que lanzar:

```

sudo ansible-playbook -i /opt/ansible/ansible-inventories/ha/hosts.yml /opt/ansible/anjana.yml -t
update-anjana-config,update-anjana-services

```

8. El último punto para revisar serían los balanceadores de los archivos de configuración de Apache donde editamos a mano el archivo (normalmente en /etc/apache/sites-enabled/anjianaxxxx.conf). Tenemos que borrar los nodos caídos en los proxypass.

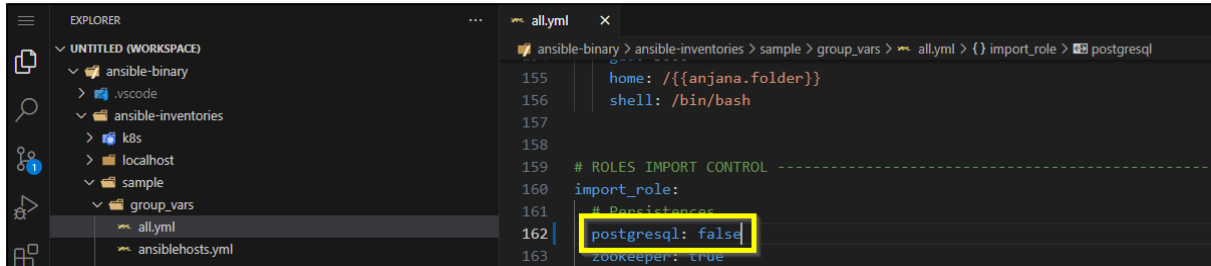
NOTA: No se puede hacer en los templates porque se rellenan con un bucle de los hosts de minio.

9. Para finalizar sería recomendable reiniciar el nodo de Anjana

Despliegue de Anjana en Oracle o SqlServer

En este apartado explicaremos cómo vamos a instalar anjana sobre un oracle o un sqlserver si queremos que estos sean nuestros motores de BBDD.

Para ello, accederemos al inventario que estemos usando, y pondremos a false el role de postgresql para que no se ejecute.

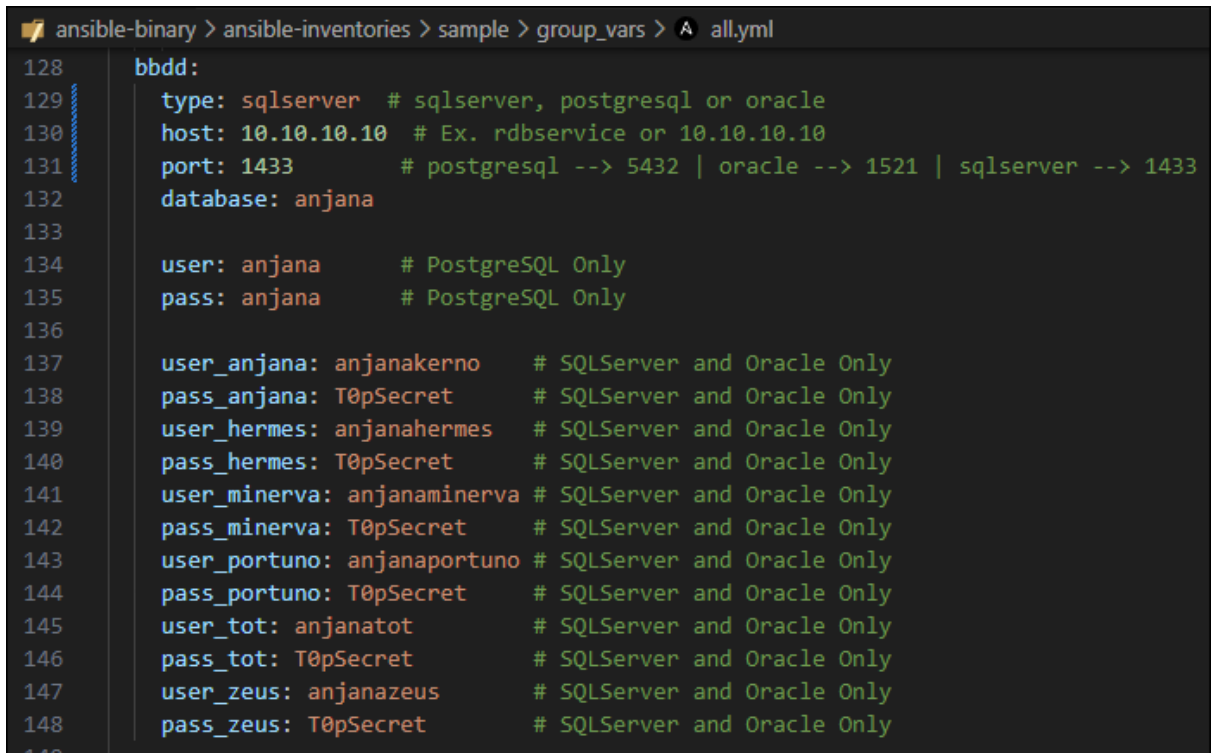


```
ansible-binary > ansible-inventories > sample > group_vars > all.yml > {} import_role > postgresql
155 | home: /{{anjana.folder}}
156 | shell: /bin/bash
157 |
158 |
159 | # ROLES_IMPORT_CONTROL -----
160 | import_role:
161 | # PostgreSQL
162 | postgresql: false
163 | zookeeper: true
```

También tendremos que cambiar toda la configuración del core de Anjana, para que en vez de apuntar a un Postgresql y usar su driver, use el driver de SQLServer o Oracle. En este ejemplo, vamos a usar SQLServer. Todos los usuarios y los schemas deben de estar creados anterior al lanzamiento de ansible, en otro caso, fallará.

En el caso de estos dos motores de BBDD, debemos de tener un usuario por schema, teniendo hasta 6 usuarios, si no, fallará también.

Para actualizar la configuración ajustamos la cadena de conexión de la base de datos y los usuarios necesarios en el all.yml para adecuarse a la base de datos a usar.



```
ansible-binary > ansible-inventories > sample > group_vars > all.yml
128 | bbdd:
129 |   type: sqlserver # sqlserver, postgresql or oracle
130 |   host: 10.10.10.10 # Ex. rdbservice or 10.10.10.10
131 |   port: 1433 # postgresql --> 5432 | oracle --> 1521 | sqlserver --> 1433
132 |   database: anjana
133 |
134 |   user: anjana # PostgreSQL Only
135 |   pass: anjana # PostgreSQL Only
136 |
137 |   user_anjana: anjanakerno # SQLServer and Oracle Only
138 |   pass_anjana: T0pSecret # SQLServer and Oracle Only
139 |   user_hermes: anjanahermes # SQLServer and Oracle Only
140 |   pass_hermes: T0pSecret # SQLServer and Oracle Only
141 |   user_minerva: anjanaminerva # SQLServer and Oracle Only
142 |   pass_minerva: T0pSecret # SQLServer and Oracle Only
143 |   user_portuno: anjanaportuno # SQLServer and Oracle Only
144 |   pass_portuno: T0pSecret # SQLServer and Oracle Only
145 |   user_tot: anjanatot # SQLServer and Oracle Only
146 |   pass_tot: T0pSecret # SQLServer and Oracle Only
147 |   user_zeus: anjanazeus # SQLServer and Oracle Only
148 |   pass_zeus: T0pSecret # SQLServer and Oracle Only
```

Una vez tengamos ajustada la configuración, lanzaremos el comando de ansible-playbook sin tags, ya que así instalará anjana sin sample data, ya que no existe sample data de anjana para estos dos motores de BBDD.

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories/<inventario>/hosts.yml /opt/ansible/anjana.yml
```

Una vez finalizada, deberemos de lanzar el tag update-anjana-config, ya que no tendrán configuraciones los microservicios.

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml /opt/ansible/anjana.yml -t update-anjana-config
```

Errores conocidos

Timeout

Sucede cuando durante la ejecución y a la vez el despliegue la máquina está ocupada levantando

```
TASK [portuno : Download jar, launcher and hibernate] *****
[WARNING]: conditional statements should not include jinja2 templating delimiters such as {{
changed: [portuno1 -> localhost] => (item={'url': 'https://artifactory.anjanadata.org:8443/repo
"local"', 'mode': 'u+rw-x,g+rw-x,o-rwx'})
failed: [portuno1 -> localhost] (item={'url': 'https://artifactory.anjanadata.org:8443/repo
llation.type == "external" and installation.mode != "local"', 'mode': 'u+rw-x,g+rw-x,o-rwx'})
launcher", "mode": "u+rw-x,g+rw-x,o-rwx", "url": "https://artifactory.anjanadata.org:8443/repo
llation.mode != \"local\"", "msg": "Connection failure: The read operation timed out", "url
skipping: [portuno1] => (item={'url': 'https://artifactory.anjanadata.org:8443/repository/an
"download" in ansible_run_tags or (installation.mode != "local" and not hibernate.stat.exi
```

Fallo remove-anjana

A veces quitar los service alguno se queda congelado y falla. Se recomienda para resolver ejecutar:

```
sudo systemctl stop xxxx
sudo systemctl disable --now xxxx
sudo reboot
```

```
TASK [anjana-utility : Disabling now [minerva] service in [minerva] host if running] *****
fatal: [director1 -> 10.150.100.245]: FAILED! => {"changed": false, "msg": "Could not find the requested service minerva: host"}
PLAY RECAP *****
common1 : ok=45 changed=1 unreachable=0 failed=0 skipped=8 rescued=0 ignored=0
```

Fallo remove-anjana - cluster MinIO

El tag remove-anjana intenta borrar las carpetas montadas desde volumen y falla.

```
changed: [director1 -> 10.150.100.95] => (item=Removed [/opt/data] dir)
failed: [director1 -> 10.150.100.43] (item=Removed [/opt/data] dir) => (ansible_loop_var: "item", "changed": false, "item": ["/opt/data", {"ip": "10.150.100.43"}], "msg": "rmtree failed: [Errno 16] Device or resource busy: 'minio'")
failed: [director1 -> 10.150.100.133] (item=Removed [/opt/data] dir) => (ansible_loop_var: "item", "changed": false, "item": ["/opt/data", {"ip": "10.150.100.133"}], "msg": "rmtree failed: [Errno 16] Device or resource busy: 'minio'")
failed: [director1 -> 10.150.100.12] (item=Removed [/opt/data] dir) => (ansible_loop_var: "item", "changed": false, "item": ["/opt/data", {"ip": "10.150.100.12"}], "msg": "rmtree failed: [Errno 16] Device or resource busy: 'minio'")
failed: [director1 -> 10.150.100.66] (item=Removed [/opt/data] dir) => (ansible_loop_var: "item", "changed": false, "item": ["/opt/data", {"ip": "10.150.100.66"}], "msg": "rmtree failed: [Errno 16] Device or resource busy: 'minio'")
ok: [director1 -> 10.150.100.95] => (item=Removed [/opt/data] dir)
```

Para evitarlo hay que comentar la líneas de los nodos de MinIO en hosts.yml


```

service_hostname: localhost
miniohosts:
  hosts:
  #
  # minio1:
  #   ansible_host: 10.150.100.43
  #   ansible_port: 22
  #   service_hostname: localhost
  #
  # minio2:
  #   ansible_host: 10.150.100.138
  #   ansible_port: 22
  #   service_hostname: localhost
  #
  # minio3:
  #   ansible_host: 10.150.100.12
  #   ansible_port: 22
  #   service_hostname: localhost
  #
  # minio4:
  #   ansible_host: 10.150.100.66
  #   ansible_port: 22
  #   service_hostname: localhost
portalhosts:
  hosts:

```

Carpeta data MinIO en HA

Carpeta de datos de un MinIO en HA debe ser un disco montado o saldrá el siguiente error al hacer “journalctl -u minio -n100”

```

Sep 06 14:11:22 minio1 minio[3860]: API: SYSTEM()
Sep 06 14:11:22 minio1 minio[3860]: Time: 14:11:22 UTC 09/06/2023
Sep 06 14:11:22 minio1 minio[3860]: Error: Drive 'http://10.152.20.196:9000/opt/data/minio' is part of root drive, will not be used (*errors.errorString)
Sep 06 14:11:22 minio1 minio[3860]: 8: internal/logger/logger.go:258:logger.LogIf()
Sep 06 14:11:22 minio1 minio[3860]: 7: cmd/erasure-sets.go:1072:cmd.markRootDisksAsDown()
Sep 06 14:11:22 minio1 minio[3860]: 6: cmd/format-erasure.go:785:cmd.initFormatErasure()
Sep 06 14:11:22 minio1 minio[3860]: 5: cmd/prepare-storage.go:214:cmd.connectLoadInitFormats()
Sep 06 14:11:22 minio1 minio[3860]: 4: cmd/prepare-storage.go:305:cmd.waitForFormatErasure()
Sep 06 14:11:22 minio1 minio[3860]: 3: cmd/erasure-server-pool.go:103:cmd.newErasureServerPools()
Sep 06 14:11:22 minio1 minio[3860]: 2: cmd/server-main.go:785:cmd.newObjectLayer()
Sep 06 14:11:22 minio1 minio[3860]: 1: cmd/server-main.go:611:cmd.serverMain()
Sep 06 14:11:22 minio1 minio[3860]: API: SYSTEM()
Sep 06 14:11:22 minio1 minio[3860]: Time: 14:11:22 UTC 09/06/2023

```

Borrado de colecciones

En el archivo all.yml hay que ajustar las colecciones que usamos en Anjana, si no, cuando hagamos un reset, delete, ... nos saldrá un error como el siguiente.

```

TASK [persistences-utility : Checking Solr connection from inventory and/or cloud] *****
failed: [persistences-utility -> localhost] (item=snaphots) => {"ansible_loop_var": "item", "changed": false, "connection": "close", "content_length": "309", "content_security_policy": "default-src 'none'; base-uri 'none'; connect-src 'self'; form-action 'self'; font-src 'self'; frame-ancestors 'none'; img-src 'self'; media-src 'self'; style-src 'self'; unsafe-inline; script-src 'self'; worker-src 'self';", "content_type": "application/json;charset=utf-8", "elapsed": 0, "item": "snaphots", "json": {"error": {"code": 400, "metadata": [{"CLUSTERSTATUS": "NOT FOUND", "error-class": "org.apache.solr.common.SolrException", "root-error-class": "org.apache.solr.common.SolrException"}, "msg": "Collection: snapshots not found"}, "responseHeader": {"QTime": 5, "status": 400}}, "msg": "Status code was 400 and not [200]: HTTP Error 400: Bad Request", "redirected": false, "status": 400, "url": "http://10.150.100.94:8983/api/collections/snaphots", "x_content_type_options": "nosniff", "x_frame_options": "SAMEORIGIN", "x_xss_protection": "1; mode=block"}
ok: [persistences-utility -> localhost] (item=kerno)
failed: [persistences-utility -> localhost] (item=audit_logs) => {"ansible_loop_var": "item", "changed": false, "connection": "close", "content_length": "310", "content_security_policy": "default-src 'none'; base-uri 'none'; connect-src 'self'; form-action 'self'; font-src 'self'; frame-ancestors 'none'; img-src 'self'; media-src 'self'; style-src 'self'; unsafe-inline; script-src 'self'; worker-src 'self';", "content_type": "application/json;charset=utf-8", "elapsed": 0, "item": "audit_logs", "json": {"error": {"code": 400, "metadata": [{"CLUSTERSTATUS": "NOT FOUND", "error-class": "org.apache.solr.common.SolrException", "root-error-class": "org.apache.solr.common.SolrException"}, "msg": "Collection: audit logs not found"}, "responseHeader": {"QTime": 0, "status": 400}}, "msg": "Status code was 400 and not [200]: HTTP Error 400: Bad Request", "redirected": false, "status": 400, "url": "http://10.150.100.94:8983/api/collections/audit_logs", "x_content_type_options": "nosniff", "x_frame_options": "SAMEORIGIN", "x_xss_protection": "1; mode=block"}
ok: [persistences-utility -> localhost] (item=recomandations)
failed: [persistences-utility -> localhost] (item=business_tags) => {"ansible_loop_var": "item", "changed": false, "connection": "close", "content_length": "313", "content_security_policy": "default-src 'none'; base-uri 'none'; connect-src 'self'; form-action 'self'; font-src 'self'; frame-ancestors 'none'; img-src 'self'; media-src 'self'; style-src 'self'; unsafe-inline; script-src 'self'; worker-src 'self';", "content_type": "application/json;charset=utf-8", "elapsed": 0, "item": "business_tags", "json": {"error": {"code": 400, "metadata": [{"CLUSTERSTATUS": "NOT FOUND", "error-class": "org.apache.solr.common.SolrException", "root-error-class": "org.apache.solr.common.SolrException"}, "msg": "Collection: business tags not found"}, "responseHeader": {"QTime": 0, "status": 400}}, "msg": "Status code was 400 and not [200]: HTTP Error 400: Bad Request", "redirected": false, "status": 400, "url": "http://10.150.100.94:8983/api/collections/business_tags", "x_content_type_options": "nosniff", "x_frame_options": "SAMEORIGIN", "x_xss_protection": "1; mode=block"}

PLAY RECAP *****
apacheds1      : ok=0    changed=0    unreachable=0    failed=0    skipped=14   rescued=0     ignored=0
common1       : ok=40   changed=1    unreachable=0    failed=0    skipped=10   rescued=0     ignored=0
common2       : ok=11   changed=1    unreachable=0    failed=0    skipped=6    rescued=0     ignored=0
common3       : ok=11   changed=1    unreachable=0    failed=0    skipped=6    rescued=0     ignored=0
common1(crosservices1) : ok=3   changed=1    unreachable=0    failed=0    skipped=1    rescued=0     ignored=0
common1(crosservices2) : ok=3   changed=1    unreachable=0    failed=0    skipped=1    rescued=0     ignored=0
common1(crosservices3) : ok=3   changed=1    unreachable=0    failed=0    skipped=1    rescued=0     ignored=0
director1     : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0     ignored=0

```

```
solr:  
host: http://10.150.100.94 # Ex. http://  
port: 8983  
collections:  
  #- snapshots  
  - kerno  
  #- audit_logs  
  - recommendations  
  #- business_tags
```

Error connection check a persistencias

Cuando se usa Oracle o SqlServer o no está accesible una de las persistencias según el tag que se lance puede salir error de que no conecta con alguna de las persistencias.

Si no se va a usar la persistencia en el trabajo que lancemos con ansible se puede evitar ese check mediante el argumento en el lanzamiento "--skip-tags connection-check".

Ejemplo:

```
sudo ansible-playbook -i /opt/ansible/ansible-inventories<inventario>/hosts.yml  
/opt/ansible/anjana.yml --tags xxxxx --skip-tags connection-check
```