



MANUAL DE DESPLIEGUE CON KIT ANSIBLE

| | |
|---|-----------|
| Release Notes | 4 |
| Compatibilidad de versiones | 6 |
| Procedimiento de actualización | 7 |
| Requerimientos | 7 |
| Software requerido | 7 |
| Conectividad requerida | 8 |
| Particularidades RedHat | 8 |
| Apache2 y SELinux | 8 |
| Intérprete de Python | 9 |
| Arranque y parada | 9 |
| Información general | 10 |
| Introducción Ansible | 10 |
| ¿Qué es un rol? | 10 |
| ¿Qué es un inventario? | 10 |
| ¿Qué es un playbook? | 10 |
| Playbooks disponibles | 11 |
| Estructura de los roles | 11 |
| Instalación, configuración y mantenimiento del kit | 16 |
| Instalación | 16 |
| Instalación de runtime ansible | 16 |
| Descarga y despliegue | 16 |
| Actualización kit Ansible | 18 |
| Respaldo de configuración | 18 |
| Ajuste de versión + Descarga | 18 |
| Actualización kit Ansible | 19 |

| | |
|---|-----------|
| Ajustar ips y credenciales de inventarios y variables de roles | 21 |
| Logs | 23 |
| Despliegue y mantenimiento de instancia de Anjana Data | 24 |
| Despliegue de nueva instancia | 24 |
| Generación de playbook | 24 |
| Generación de inventario de hosts | 24 |
| Ajuste de variables | 24 |
| Ejecución del playbook | 25 |
| Actualización de instancia | 25 |
| Actualización de Anjana | 25 |
| Actualización de descriptores de servicios | 26 |
| Actualización de vhosts de webserver | 26 |
| Instalar/Actualizar instancia desde dentro de la propia máquina | 26 |
| Actualización de configuración de Anjana | 26 |
| Mantenimiento de instancia | 29 |
| Arranque, parada y reinicio | 29 |
| Backup de los datos | 31 |
| Exportación e importación de datos | 31 |
| Exportar log de microservicios | 33 |
| Desinstalación | 33 |
| Despliegue de plataforma de monitorización de métricas | 35 |
| Despliegue de plataforma de agregación de logs | 38 |
| Detalle de composición del kit | 41 |
| Roles disponibles | 41 |
| Tags disponibles | 43 |
| Funciones adicionales del kit | 48 |
| Descargar artefactos de Anjana | 48 |
| Sampledata | 48 |
| Data delete | 48 |
| Data reset | 49 |
| Exportar backups a S3 | 49 |
| Balanceador/HA para MinIO | 50 |
| Scripts funcionales | 50 |
| Compatibilidad con uso de sudo | 52 |
| Ejemplos de uso | 53 |
| Despliegue single node + RDS | 53 |
| Despliegue single node | 57 |
| Despliegue en HA + resiliencia Solr | 60 |
| Generación de nueva instancia de un plugin ya existente | 68 |
| Errores conocidos | 70 |
| Timeout | 70 |

| | |
|-------------------------------------|----|
| Fallo remove-anjana | 70 |
| Fallo remove-anjana - cluster MinIO | 71 |

| Versión | Fecha de publicación | Responsable | Aprobador | Resumen de cambios |
|---------|----------------------|-------------|----------------|---|
| 1.0 | 18/04/2023 | Dpto DS | Responsable DS | Creación del documento |
| 2.0 | 7/06/2023 | Dpto DS | Responsable DS | Soporte a nuevos plugins Tableau y Ranger Soporte cambio config en plugin LDAP 4.4.2 |
| 3.0 | 12/07/2023 | Dpto DS | Responsable DS | Cambio url de repositorios de artefactos |

Release Notes

- Se mueven los templates a las carpetas de inventario para facilitar su respaldo y mejorar en permisos y seguridad en su gestión.
 - Se expone la posibilidad de editar y ampliar los virtual host del role de portal a través de las variables del inventario
 - Los templates de configuración de los microservicios se reubican al inventario
 - Los descriptores de servicios de los microservicios se reubican al inventario
- Se crea el tag “adjust-ansible-folder” para ajustar los permisos del kit de Ansible
- Se han unificado las credenciales a las persistencias en el archivo principal de variables all.yml para todos los roles que puedan necesitarlo (import, export, insert, delete, queries de utilidades, ...)
- Se han movido al archivo principal de variables all.yml el control de qué roles se ejecutan a través de unas variables booleanas. Ejemplo de uso: si se va a hacer uso de AWS S3 hay que pasar a “false” la variable “export-role.minio”
- Nueva funcionalidad/tag “log” que exporta el log de los microservicios de Anjana
- Nueva funcionalidad/tag “update-anjana-config” que actualiza los descriptores de los microservicios
- Nueva funcionalidad/tag “update-apache-vhost” que actualiza los virtual host del servidor web
- Nueva funcionalidad/tag “update-anjana-services” que actualiza los descriptores de microservicios
- Nueva funcionalidad/tag de reseteo de secuencias en bbdd
- Nueva funcionalidad/tag de borrado de datos en persistencia
- Nueva funcionalidad/tag de borrado de datos + configuración funcional
- Nueva funcionalidad/tag instalación WAF para Apache2 (Modsec)
- Nueva funcionalidad de despliegue de entorno de monitorización de métricas (Grafana + Prometheus + exporters)
- Nueva funcionalidad de despliegue de entorno de monitorización de logs (Grafana + Grafana-agent + Loki + Promtail)
- Se han unificado los roles export e import en export-import-utility
- El rol export-import-utility añade compatibilidad con RDS y AWS S3
- Se añaden varias comprobaciones previas para el role export-import-utility
- El tag s3-dump ahora es compatible con AWS S3
- Añadido Proxypass textarea en el virtual host del role Portal
- Soporte del sistema operativo Amazon Linux 2
- Añadido la posibilidad de despliegue de Horus
- Se han añadido comprobaciones de integridad de las variables de cada role
- Se han unificado tareas para bajar el tiempo de ejecución
- Se ajustan los permisos y propietarios del software desplegado con el kit para mejorar en seguridad y facilidad de gestión técnica

- Se añade en documentación un ejemplo la configuración del archivo sudoers para un técnico de Anjana
- Corregido un error que impedía la ejecución del tag backup para respaldar Solr en una instalación con un kit de Ansible anterior a 4.a4
- Se ha incorporado una utilidad para facilitar el arranque y parada de los microservicios de Anjana de forma selectiva y ordenada. Más información en [Utilidad de Anjana](#) y [Listado de tags](#)
- Nuevos tags: [Listado de tags](#)

Compatibilidad de versiones

Versiones soportadas de Anjana Data:

- Desde “4.4.0” hasta “Release Notes 05/04/2023”

Sistemas operativos soportados:

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

Imagenes cloud testadas:

- AWS
 - [Amazon Linux 2 AMI \(HVM\), SSD Volume Type \(64-bit x86\)](#)
 - [Red Hat Enterprise Linux 8.6 \(HVM\)](#) (ami-06640050dc3f556bb)
 - [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.

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
 - python-psycopg2
 - python3-psycopg2
 - lsof
 - acl
 - sudo
 - jq
 - iproute2
 - ldap-utils|openldap-clients
 - OpenJDK 8
 - OpenJDK 11
- Resto de nodos
 - python-pip
 - python3
 - python-psycopg2
 - python3-psycopg2
 - lsof
 - acl
 - sudo
 - jq
 - iproute2
 - ldap-utils
 - OpenJDK 8
 - OpenJDK 11
 - Adicional en nodos frontales
 - apache2|httpd
 - Adicional en nodos de persistencia RDBMS (si no se usa servicio externo)
 - POSTGRESQL 13

Conectividad requerida

El presente kit, en modos director o remote, descargara 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 RedHat

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 `all.yml` del inventario que se esté usando, en la sección `anjana_utility`:

```
ansible-binary
ansible-inventories
kbs
localhost
group_vars
  allyaml
    ansiblehosts.yml
    apachedhosts.yml
    apacheexporterhosts.yml
    dritttestahosts.yml
    edusahosts.yml
    exportimportilityhosts.yml
    fakesmtpserverhosts.yml
    grafanaagenthosts.yml
    grafanahosts.yml
    hecatehosts.yml
    hemeshosts.yml
    horushosts.yml
    kernohosts.yml
    lokihosts.yml
    minervahosts.yml
    miniohosts.yml
    monithosts.yml
    nodeexporterhosts.yml
    persistencelityhosts.yml
    portalhosts.yml
    portunohosts.yml
    postgresexporterhosts.yml
    postgresfhosts.yml
    prometheushosts.yml
    prontalitohosts.yml
    solhosts.yml
    tothosts.yml
    totpluginaqthosts.yml
    totpluginaqthosts.yml

  222      # - tot-plugin-ldap
  223      # - tot-plugin-aqativa
  224      # - tot-plugin-powerbi
  225
  226
  227      anjana_utility:
  228      - { host: zookeeper, service: zookeeper, port: 2181, dir: zookeeper, idx: 1 }
  229      # - { host: postgresql, service: postgresql-13, port: 5432, dir: postgresql, idx: 2 }
  230      - { host: minio, service: minio, port: 9000, dir: minio, idx: 3 }
  231      - { host: solr, service: solr, port: 8983, dir: solr, idx: 4 }
  232      - { host: edusa, service: edusa, port: 8888, dir: edusa, idx: 5 }
  233      - { host: hecate, service: hecate, port: 50761, dir: hecate, idx: 6 }
  234      - { host: portuno, service: portuno, port: 8998, dir: portuno, idx: 7 }
  235      - { host: zeus, service: zeus, port: 8088, dir: zeus, idx: 8 }
  236      - { host: kerno, service: kerno, port: 8881, dir: kerno, idx: 9 }
  237      - { host: minerva, service: minerva, port: 8984, dir: minerva, idx: 10 }
  238      - { host: viator, service: viator, port: 8885, dir: viator, idx: 11 }
  239      - { host: hermes, service: hermes, port: 8887, dir: hermes, idx: 12 }
  240      - { host: drittesta, service: drittesta, port: 8995, dir: drittesta, idx: 13 }
  241      - { host: tot, service: tot, port: 15000, dir: tot, idx: 14 }
  242      # - { host: portal, service: httpd, port: 88, dir: htddocs, idx: 15 }
  243      - { host: horus, service: horus, port: 9999, dir: horus, idx: 16 }
  244      - { host: totpluginawsglue, service: tot-plugin-aws-glue, port: 15019, dir: tot-plugin-aws-glue, idx: 17 }
  245      - { host: totpluginawsiam, service: tot-plugin-aws-iam, port: 15008, dir: tot-plugin-aws-iam, idx: 18 }
  246      - { host: totpluginaws3, service: tot-plugin-aws-s3, port: 15007, dir: tot-plugin-aws-s3, idx: 19 }
  247      - { host: totpluginazuread, service: tot-plugin-azure-ad, port: 15009, dir: tot-plugin-azure-ad, idx: 20 }
  248      - { host: totpluginazurerefiles, service: tot-plugin-azure-files, port: 15011, dir: tot-plugin-azure-files, idx: 21 }
  249      - { host: totpluginazurerestorage, service: tot-plugin-azure-storage, port: 15012, dir: tot-plugin-azure-storage, idx: 22 }
  250      - { host: totplugginglepbigquery, service: tot-plugin-gcp-bigquery, port: 15002, dir: tot-plugin-gcp-bigquery, idx: 23 }
  251      - { host: totplugginglepam, service: tot-plugin-gcp-iam, port: 15010, dir: tot-plugin-gcp-iam, idx: 24 }
  252      - { host: totplugginglepsstorage, service: tot-plugin-gcp-storage, port: 15003, dir: tot-plugin-gcp-storage, idx: 25 }
  253      - { host: totplugginglehive, service: tot-plugin-hive, port: 15006, dir: tot-plugin-hive, idx: 26 }
  254      - { host: totplugginglehdfs, service: tot-plugin-hdfs, port: 15004, dir: tot-plugin-hdfs, idx: 27 }
  255      - { host: totplugginjdbc, service: tot-plugin-jdbc, port: 15001, dir: tot-plugin-jdbc, idx: 28 }
  256      - { host: totplugginjdbcdemodo, service: tot-plugin-jdbc-demodo, port: 15018, dir: tot-plugin-jdbc-demodo, idx: 29 }
  257      - { host: totplugginjdbcoracle, service: tot-plugin-jdbc-oracle, port: 15013, dir: tot-plugin-jdbc-oracle, idx: 30 }
  258      - { host: totplugginjdbredshift, service: tot-plugin-jdbc-redshift, port: 15017, dir: tot-plugin-jdbc-redshift, idx: 31 }
  259      - { host: totplugginjdcsqlserver, service: tot-plugin-jdbc-sqlserver, port: 15005, dir: tot-plugin-jdbc-sqlserver, idx: 32 }
  260      - { host: totplugginldap, service: tot-plugin-ldap, port: 15015, dir: tot-plugin-ldap, idx: 33 }
  261      - { host: totplugginpowerbi, service: tot-plugin-powerbi, port: 15016, dir: tot-plugin-powerbi, idx: 34 }
  262      - { host: totplugginaqativa, service: tot-plugin-aqativa, port: 15020, dir: tot-plugin-aqativa, idx: 35 }
  263
```

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

- **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 Kit 4.a5  #
#-----#
version:
  edusa: 4.4.0
  hecate: 4.4.0
  horus: 4.4.0
  hermes: 4.4.0
  kerno: 4.4.0
  minerva: 4.4.0
  portuno: 4.4.0
  tot: 4.4.0
  viator: 4.4.0
  ansible: 4.a4
  sampledata: 4.4-sample
  zeus: 4.4.0
  portal: 4.4.0

```

- **anjana.folder** → Permite especificar la carpeta relativa a la raíz del sistema de ficheros donde se despliega el software de Anjana.

```
anjana:  
  folder: opt # WITHOUT SLASH '/' Ex. "opt" or "software/data"
```

- **anjana.configURL** → String con los servidores de configuración que luego van en el comando de ejecución de los microservicios

```
anjana:  
  configURL: http://edusaserver:8888
```

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

```
etc:  
  hosts:  
    - "127.0.0.1 indexservice zkse  
    - "127.0.0.1 hecateserver heca  
    - "127.0.0.1 totpluginawsglues  
    - "127.0.0.1 totpluggingcpstor
```

- **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, ...

```

persistences:
  s3:
    type: minio # minio|aws_s3
    access_key: anjana
    secret_key: anjanadata
    # MinIO Only -----
    host: <buckets-service-url> # Ex. s3service or 10.10.10.10
    port: 9000
    dump_bucket: datadump
    # AWS S3 Only -----
    region: <region>
    buckets:
      - { bucket: cdn, name: <cdn-bucket> }
      - { bucket: dsa, name: <dsa-bucket> }
      - { bucket: imports, name: <imports-bucket> }
      - { bucket: textarea, name: <textarea-bucket> }
      - { bucket: workflows, name: <workflows-bucket> }
      - { bucket: datadump, name: <datadump-bucket> }
  bbdd:
    host: <bbdd/rds-url> # Ex. rdbbservice or 10.10.10.10
    port: 5432
    database: anjana
    user: anjana
    pass: anjana
  solr:
    host: <solr-url> # Ex. http://indexservice or http://10.10.10.10
    port: 8983
    collections:
      - snapshots
      - kerno
      - audit_logs
      - recommendations
      - business_tags
  |

```

- **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.

```

import_role:
  # Persistence
  postgresql: true
  zookeeper: true
  solr: true
  apacheds: true
  minio: true
  # Core Anjana
  portal: true
  edusa: true
  hecate: true
  portuno: true
  kerno: true
  zeus: true
  drittesta: true
  minerva: true

```

- **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 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
- **anjana_utility:** → En el nuevo apartado “anjana_utility:” debemos ajustar puertos y nombres de servicios si hubiera alguna personalización para la funcionalidad de anjana-start, anjana-stop, remove-anjana, ...

```
anjana_utility:
  - { host: zookeeper, service: zookeeper, port: 2181, dir: zookeeper, idx: 1 }
  - { host: postgresql, service: postgresql, port: 5432, dir: postgresql, idx: 2 }
  - { host: minio, service: minio, port: 9000, dir: minio, idx: 3 }
  - { host: solr, service: solr, port: 8983, dir: solr, idx: 4 }
  - { host: edusa, service: edusa, port: 8888, dir: edusa, idx: 5 }
  - { host: hecate, service: hecate, port: 50761, dir: hecate, idx: 6 }
  - { host: portuno, service: portuno, port: 8998, dir: portuno, idx: 7 }
  - { host: zeus, service: zeus, port: 8088, dir: zeus, idx: 8 }
  - { host: kerno, service: kerno, port: 8081, dir: kerno, idx: 9 }
```

- **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
  puede ser borrado al finalizado de las tareas
  eurekaPreferIpAddress: true
  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
```

- 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
22 |       -> /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 ansible-galaxy collection install community.general
sudo ansible-galaxy collection install community.postgresql
sudo ansible-galaxy collection install community.aws
```

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

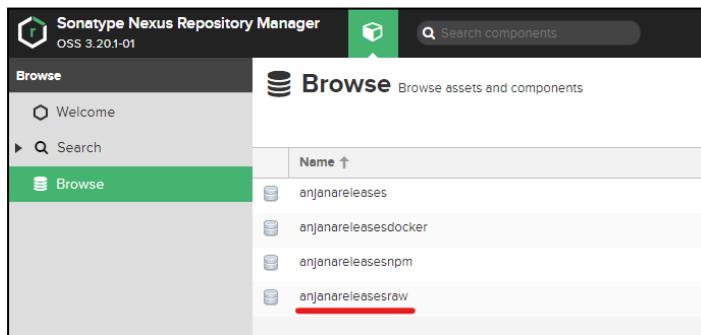
# Add binary to PATH
sudo ln -s /usr/local/bin/ansible /usr/bin/ansible
sudo ln -s /usr/local/bin/ansible-galaxy /usr/bin/ansible-galaxy
sudo ln -s /usr/local/bin/ansible-playbook /usr/bin/ansible-playbook

# Install dependencies
sudo ansible-galaxy collection install community.general
sudo ansible-galaxy collection install community.postgresql
sudo ansible-galaxy collection install community.aws
```

También instalará la Python3 como dependencia de Ansible.

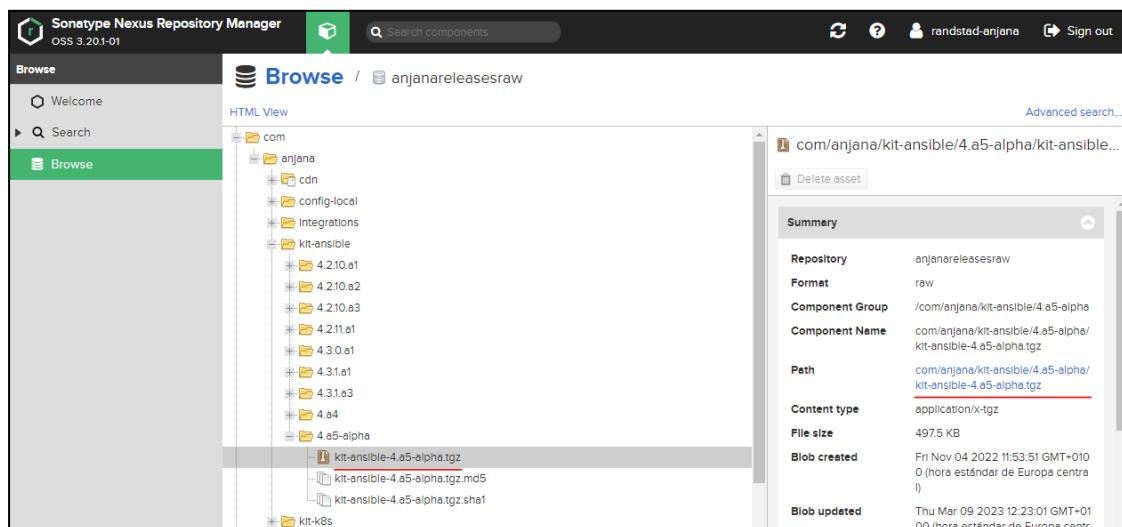
Descarga y despliegue

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



Desplegamos hasta llegar al kit de Ansible en la versión deseada (4.a5 a la fecha de la publicación de esta documentación) :

- com/anjana/kit-ansible/4.a5 pulsamos y en el panel derecho nos aparece el link de descarga



| Summary |
|---|
| Repository anjanareleasesraw Format raw Component Group /com/anjana/kit-ansible/4.a5-alpha Component Name com/anjana/kit-ansible/4.a5-alpha/kit-ansible-4.a5-alpha.tgz Path com/anjana/kit-ansible/4.a5-alpha/kit-ansible-4.a5-alpha.tgz Content type application/x-tgz File size 497.5 KB Blob created Fri Nov 04 2022 11:53:51 GMT+0100 (hora estándar de Europa central) Blob updated Thu Mar 09 2023 12:23:01 GMT+0100 (hora estándar de Europa central) |

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 descomprimimos:

```

mkdir <ruta_base>/ansible # normalmente /opt/ansible
wget --user <user> --password <password>
https://releases.anjanadata.org/repository/releasesraw/com/anjana/kit-ansible/4.a5/kit-ansible-4.a5.tgz
sudo tar -zxpvf kit-ansible-4.a5.tgz -C <ruta_base>/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

sudo ansible-playbook -i <ruta_base>/ansible/ansible-inventories/localhost/hosts.yml
<ruta_anjana>/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#
```

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 sobreescriba.

```
mv -r /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 estuvieramos usando.

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

```
drwxr-xr-x 2 anjana anjana 4096 Oct 10 12:53 .
drwxr-xr-x 3 anjana anjana 4096 Oct 10 13:32 ../
-rw-r--r-- 1 anjana anjana 4364 Oct 10 12:53 all.yml
-rw-r--r-- 1 anjana anjana 299 Oct 10 12:52 ansiblehosts.yml
-rw-r--r-- 1 anjana anjana 1004 Oct 10 12:52 apachedshosts.yml
-rw-r--r-- 1 anjana anjana 772 Oct 10 12:53 dritttestahosts.yml
-rw-r--r-- 1 anjana anjana 1325 Oct 10 12:53 edusahosts.yml
-rw-r--r-- 1 anjana anjana 877 Oct 10 12:52 exporthosts.yml
```

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

```
version:
  edusa: 4.4.0
  config_local: 4.4.0
  hecate: 4.4.0
  horus: 4.4.0
  hermes: 4.4.0
  kerno: 4.4.0
  minerva: 4.4.0
  portuno: 4.4.0
  tot: 4.4.0
  viator: 4.4.0
  ansible: 4.a5
```

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"  
vim -O /opt/ansible/.../all.yml /opt/ansible-old/.../all.yml
```

```

version: #-----#
edusa: 4.4.0 #   Ansible Kit 4.a5  #
config_local: 4.4.0 #-----#
hecate: 4.4.0
horus: 4.4.0
hermes: 4.4.0
kerno: 4.4.0
minerva: 4.4.0
portuno: 4.4.0
tot: 4.4.0
viator: 4.4.0
ansible: 4.a4
sampledata: 4.4-sample
zeus: 4.4.0
portal: 4.4.0
admin: 4.4.0
drittesta: 4.4.0
totpluginaqativa: 4.4.0
totpluginawsiam: 4.4.0
totpluginawsqle: 4.4.0
totpluginaws3: 4.4.0
totpluginazuread: 4.4.0
totpluginazurefiles: 4.4.0
totpluginazurerestorage: 4.4.0
totpluggingcpbiquery: 4.4.0
totpluggingjdbcdenodo: 4.4.0
totpluggingcpiam: 4.4.0
totpluggingpcstorage: 4.4.0
totplugginghdfs: 4.4.0
totplughinhive: 4.4.0
totplughinjdb: 4.4.0
totplughinjdbredshift: 4.4.0
totplughinjdboracle: 4.4.0
totplughinjdbsqlserver: 4.4.0
totplugindldap: 4.4.0
totpluginpowersbi: 4.4.0
#-----#
# UTILITIES -----
ansible: 4.a5-alpha
sampledata: 4.4.0-sample
script_reset_sequences: 4.4
script_delete_data: 4.4
script_delete_data_and_functional_config: 4.4
#-----#
# PLUGINS -----
totpluginawsiam: 4.4.0
totpluginawsqle: 4.4.0
totpluginaws3: 4.4.0
totpluginazuread: 4.4.0
totpluginazurefiles: 4.4.0
#totplughinjazurerestorage: 4.4.0
totpluggingcpbiquery: 4.4.0
totplughinjdbdenodo: 4.4.0
totpluggingcpiam: 4.4.0
totpluggingpcstorage: 4.4.0
totplughinjdfs: 4.4.0
#-----#

```

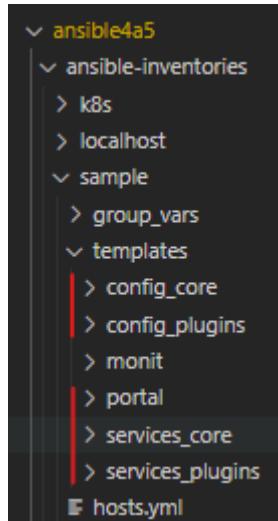
```
# Con “-d” se pueden ver las diferencias entre los archivos pero es más difícil de interpretar  
# vim -d /opt/ansible/.../all.yml /opt/ansible-old/.../all.yml
```

Puntos a revisar:

- `anjana.yml`
 - En la versión anterior controlabamos que role se ejecutaba comentando dentro de este fichero. Ahora no tocamos `anjana.yml` salvo querer editar el intérprete de `python` pero debemos actualizar el apartado “`import_role:`” del `all.yml` a `false` los que estuvieran comentados o no queramos que se ejecuten.

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

- 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
 - En el nuevo apartado “anjana_utility:” debemos ajustar puertos y nombres de servicios si hubiera alguna personalizacion para la funcionalidad de anjana-start, anjana-stop, remove-anjana, ...
 - 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.



```
ansible4a5
├── ansible-inventories
│   ├── k8s
│   ├── localhost
│   └── sample
│       ├── group_vars
│       ├── templates
│       │   ├── config_core
│       │   ├── config_plugins
│       │   ├── monit
│       │   ├── portal
│       │   ├── services_core
│       │   └── services_plugins
└── hosts.yml
```

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

- directorhosts -> nodo en el cual se ejecuta ansible
- persistencesutilityhosts -> nodo en el cual se almacenan los backups de las persistencias y se ejecutan los clientes de bbdd, s3, solr, ...
- exportimportutilityhosts -> nodo en el cual se instalan y ejecutan los clientes de exportación e importación (minio-mc, postgresql-client, solr-utility, etc)
- NOTA: En el inventario sample no dejar ningún hosts sin sustituir "<server_ip>" por una ip, aunque luego no se vaya a utilizar, puede dar problemas en el tag remove-anjana

```
ansible4a5 > 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
10      children:
11        directorhosts:
12          hosts:
13            director1:
14              # Nodo en el cual se ejecuta ansible (nodo director)
15              ansible_host: localhost
16              ansible_port: 22
17              service_hostname: localhost
18              ansible_python_interpreter: python3 # when problems try /usr/bin/python
19              ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
20              # ansible_connection: smart # Only if ansible_host is an IP or DNS
21
22        commonhosts:
23          hosts:
24            common1:
25              # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
26              # Cuentan las máquinas de Anjana, persistencias, ansible, etc
27              ansible_host: localhost
28              ansible_port: 22
29              service_hostname: localhost
30              ansible_python_interpreter: python3 # when problems try /usr/bin/python
31              ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
32            common2:
33              # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
34              # Cuentan las máquinas de Anjana, persistencias, ansible, etc
35              ansible_host: <server_ip>
36              ansible_port: 22
37              service_hostname: localhost
38              ansible_python_interpreter: python3 # when problems try /usr/bin/python
39
40        commonmicroserviceshosts:
41          hosts:
42            commonmicroservices1:
43              # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas
44              # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, etc
```

```
total 200K
-rw-r--r-- 1 anjana anjana 4.2K Oct 10 13:09 all.yml
-rw-r--r-- 1 anjana anjana 290 Oct 10 13:09 ansiblehosts.yml
-rw-r--r-- 1 anjana anjana 979 Oct 10 13:09 apachedhosts.yml
-rw-r--r-- 1 anjana anjana 756 Oct 10 13:09 drittesthosts.yml
-rw-r--r-- 1 anjana anjana 1.3K Oct 10 13:09 edusahosts.yml
-rw-r--r-- 1 anjana anjana 841 Oct 10 13:09 exporthosts.yml
-rw-r--r-- 1 anjana anjana 437 Oct 10 13:09 fakesmtserverhosts.yml
-rw-r--r-- 1 anjana anjana 689 Oct 10 13:09 hecatehosts.yml
-rw-r--r-- 1 anjana anjana 681 Oct 10 13:09 hermeshosts.yml
-rw-r--r-- 1 anjana anjana 817 Oct 10 13:09 importhosts.yml
-rw-r--r-- 1 anjana anjana 683 Oct 10 13:09 kennohosts.yml
-rw-r--r-- 1 anjana anjana 694 Oct 10 13:09 minerahosts.yml
-rw-r--r-- 1 anjana anjana 787 Oct 10 13:09 miniohosts.yml
-rw-r--r-- 1 anjana anjana 174 Oct 10 13:09 monithosts.yml
-rw-r--r-- 1 anjana anjana 1.2K Oct 10 13:09 portalhosts.yml
-rw-r--r-- 1 anjana anjana 712 Oct 10 13:09 portunohosts.yml
-rw-r--r-- 1 anjana anjana 35K Oct 10 13:09 postgresqlhosts.yml
-rw-r--r-- 1 anjana anjana 1.8K Oct 10 13:09 solrhosts.yml
-rw-r--r-- 1 anjana anjana 642 Oct 10 13:09 tohosts.yml
-rw-r--r-- 1 anjana anjana 885 Oct 10 13:09 totpluginawsgluehosts.yml
-rw-r--r-- 1 anjana anjana 871 Oct 10 13:09 totpluginawsiamhosts.yml
-rw-r--r-- 1 anjana anjana 857 Oct 10 13:09 totpluginawss3hosts.yml
-rw-r--r-- 1 anjana anjana 885 Oct 10 13:09 totpluginazureadhosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluginazurerefilehosts.yml
-rw-r--r-- 1 anjana anjana 955 Oct 10 13:09 totpluginazurerestoragehosts.yml
-rw-r--r-- 1 anjana anjana 941 Oct 10 13:09 totpluggincpbigqueryhosts.yml
-rw-r--r-- 1 anjana anjana 871 Oct 10 13:09 totpluggincpiamhosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluggincpstoragehosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluginhdfshosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totplugginhivehosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totplugginjdbdenodohosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totplugginjdbchosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totplugginjdbcoraclehosts.yml
-rw-r--r-- 1 anjana anjana 955 Oct 10 13:09 totplugginjdbredshifthosts.yml
-rw-r--r-- 1 anjana anjana 969 Oct 10 13:09 totplugginjdbcsqldserverhosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluggindaphosts.yml
-rw-r--r-- 1 anjana anjana 876 Oct 10 13:09 totplugginpowerbihosts.yml
-rw-r--r-- 1 anjana anjana 540 Oct 10 13:09 viatorhosts.yml
-rw-r--r-- 1 anjana anjana 655 Oct 10 13:09 zeushosts.yml
-rw-r--r-- 1 anjana anjana 4.7K Oct 10 13:09 zookeeper_nodes.yml
```

El no revisar y portar los cambios de los archivos actualizados a los inventarios personalizados 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.

```
installation:
  mode: director
  tmpdir: /tmp/anjana # IMPORTANTE: El contenido
  eurekaPreferIpAddress: true
  pathReport: /{{anjana.folder}}/version_report
  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 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.

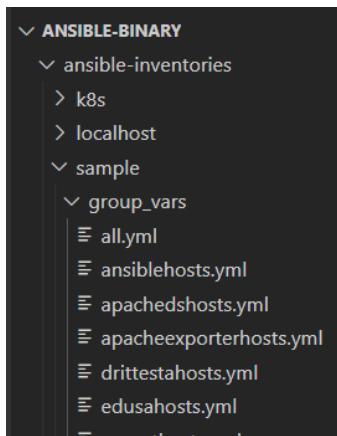
Para un single-node se puede usar el inventario “localhost” o para un entorno distribuido se puede usar “sample” y cambiarle el nombre para identificarlo mejor, por ejemplo “pro”

```
#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: [Estructura de los roles](#)

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>
Ej: sudo ansible-playbook -i ansible-inventories/sample/hosts.yml anjana.yml
```

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 ansible-inventories/sample/hosts.yml 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 ansible-inventories/sample/hosts.yml anjana.yml --tags
update
```

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

```
sudo ansible-playbook -i ansible-inventories/sample/hosts.yml 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 ansible-inventories/sample/hosts.yml 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 ansible-inventories/sample/hosts.yml 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)

Para su uso se requiere actualizar las credenciales del repositorio de Anjana en /opt/ansible/ansible-inventories/localhost/group_vars/all.yml.

```
artifactory:  
  user: <cartifactory_user>  
  password: <cartifactory_user>  
  internal:
```

Actualización de configuración de Anjana

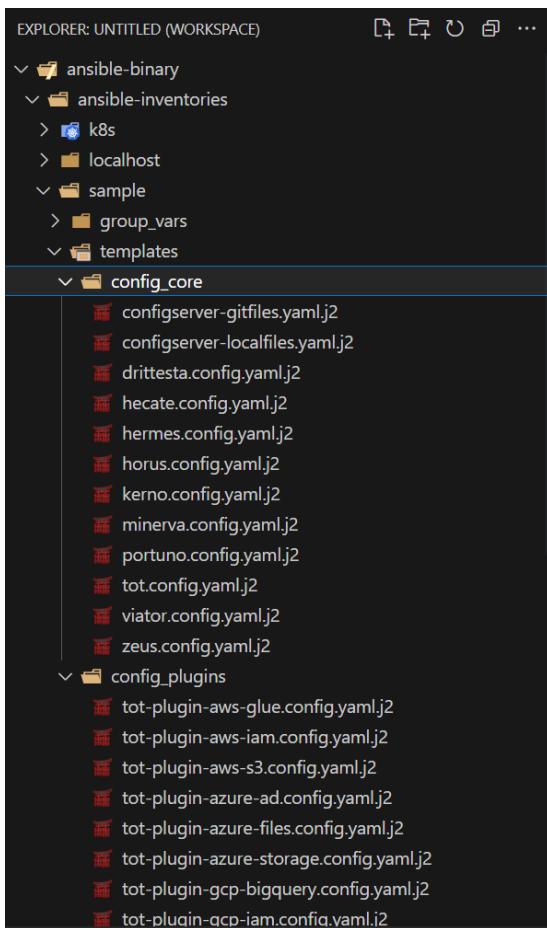
Ejecutando el kit con el tag “update-anjana-config” podremos actualizar/desplegar la configuración de los microservicios.

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.

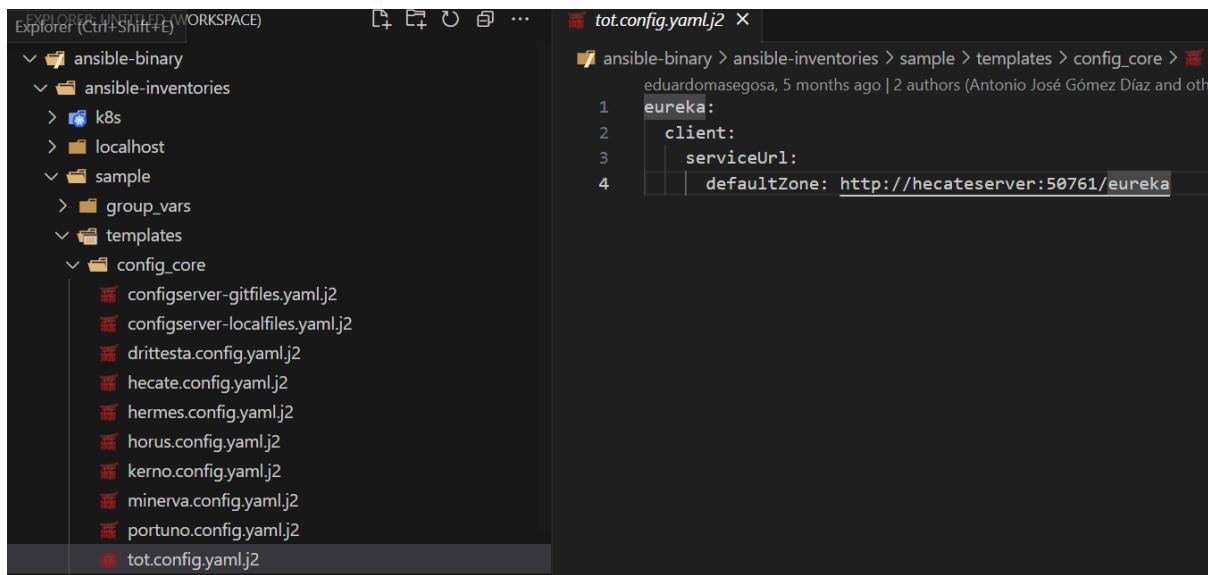
IMPORTANTE: Lanzando este tag, se actualizarán TODOS los archivos de configuración de los microservicios.

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml 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.



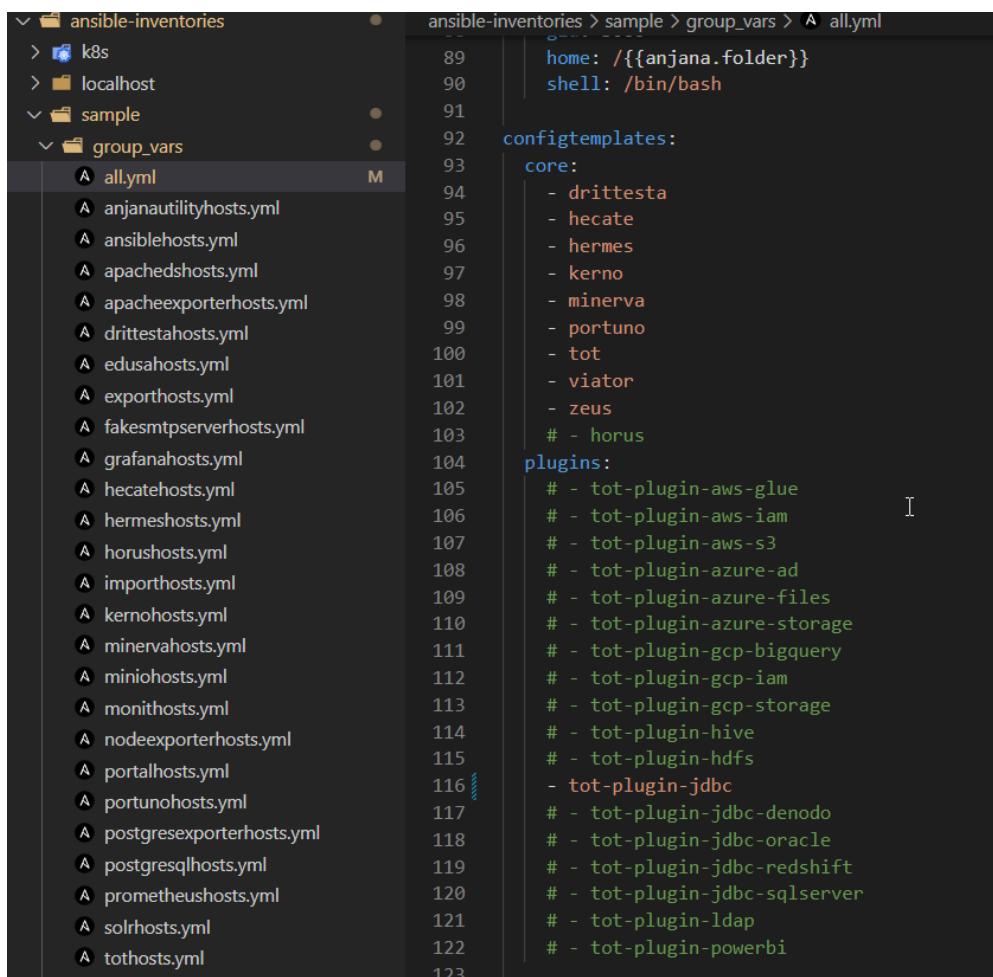
The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file structure under 'WORKSPACE'. It includes 'ansible-binary', 'ansible-inventories' (containing 'k8s', 'localhost', 'sample' (with 'group_vars'), 'templates', and 'config_core' (containing multiple .j2 files like configserver-gitfiles, configserver-localfiles, drittestea, hecate, hermes, horus, kerno, minerva, portuno, and tot)). On the right, the main editor window shows the content of 'tot.config.yaml.j2'. The code defines a 'eureka' client with a service URL pointing to 'http://hecateserver:50761/eureka'.

```

tot.config.yaml.j2
1 eureka:
2   client:
3     serviceUrl:
4       defaultZone: http://hecateserver:50761/eureka

```

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



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays 'ansible-inventories' (containing 'k8s', 'localhost', 'sample', and 'group_vars'). 'group_vars' is expanded, showing numerous .yml files including 'all.yml', 'anjanauilityhosts.yml', 'ansiblehosts.yml', 'apachedshosts.yml', 'apacheexporterhosts.yml', 'dritttestahosts.yml', 'edusahosts.yml', 'exporthosts.yml', 'fakesmtpserverhosts.yml', 'grafanahosts.yml', 'hecatehosts.yml', 'hermeshosts.yml', 'horushosts.yml', 'importhosts.yml', 'kernohosts.yml', 'minervahosts.yml', 'miniohosts.yml', 'monithosts.yml', 'nodeexporterhosts.yml', 'portalhosts.yml', 'portunohosts.yml', 'postgresexporterhosts.yml', 'postgresqlhosts.yml', 'prometheushosts.yml', 'solrhosts.yml', and 'tohosts.yml'. On the right, the main editor window shows the content of 'all.yml'. The code defines 'home' as '/{{anjana.folder}}' and 'shell' as '/bin/bash'. It also defines 'configtemplates' and 'core' sections containing host names like dritttesta, hecate, hermes, kerno, minerva, portuno, tot, viator, and zeus. The 'plugins' section contains many commented-out entries, including 'tot-plugin-jdbc' which is now uncommented.

```

ansible-inventories > sample > group_vars > all.yml
89 home: /{{anjana.folder}}
90 shell: /bin/bash
91
92 configtemplates:
93   core:
94     - dritttesta
95     - hecate
96     - hermes
97     - kerno
98     - minerva
99     - portuno
100    - tot
101    - viator
102    - zeus
103    # - horus
104 plugins:
105   # - tot-plugin-aws-glue
106   # - tot-plugin-aws-iam
107   # - tot-plugin-aws-s3
108   # - tot-plugin-azure-ad
109   # - tot-plugin-azure-files
110   # - tot-plugin-azure-storage
111   # - tot-plugin-gcp-bigquery
112   # - tot-plugin-gcp-iam
113   # - tot-plugin-gcp-storage
114   # - tot-plugin-hive
115   # - tot-plugin-hdfs
116   - tot-plugin-jdbc
117   # - tot-plugin-jdbc-denodo
118   # - tot-plugin-jdbc-oracle
119   # - tot-plugin-jdbc-redshift
120   # - tot-plugin-jdbc-sqlserver
121   # - tot-plugin-ldap
122   # - tot-plugin-powerbi
123

```

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 una lista previamente confeccionada de microservicios y puertos, editables y disponibles en el archivo `all.yml` ubicado en el inventario. Un ejemplo de uso sería la parada del backend para hacer un upgrade de Anjana.

```
anjana.utility:
  - { host: zookeeper, service: zookeeper, port: 2181, dir: zookeeper, idx: 1 }
  - { host: postgresql, service: postgresql, port: 5432, dir: postgresql, idx: 2 }
  - { host: minio, service: minio, port: 9000, dir: minio, idx: 3 }
  - { host: solr, service: solr, port: 8983, dir: solr, idx: 4 }
  - { host: edusa, service: edusa, port: 8888, dir: edusa, idx: 5 }
  - { host: hecate, service: hecate, port: 50761, dir: hecate, idx: 6 }
  - { host: portuno, service: portuno, port: 8998, dir: portuno, idx: 7 }
  - { host: zeus, service: zeus, port: 8088, dir: zeus, idx: 8 }
  - { host: kerno, service: kerno, port: 8081, dir: kerno, idx: 9 }
  - { host: minerva, service: minerva, port: 8084, dir: minerva, idx: 10 }
  - { host: viator, service: viator, port: 8085, dir: viator, idx: 11 }
  - { host: hermes, service: hermes, port: 8087, dir: hermes, idx: 12 }
  - { host: dritttesta, service: dritttesta, port: 8095, dir: dritttesta, idx: 13 }
  - { host: tot, service: tot, port: 15000, dir: tot, idx: 14 }
  - { host: portal, service: apache2, port: 80, dir: htdocs, idx: 15 }
  - { host: horus, service: horus, port: 9999, dir: horus, idx: 16 }
# - { host: tot-plugin-aws-glue, service: tot-plugin-aws-glue, port: 15019, dir: tot-plugin-aws-glue, idx: 17 }
# - { host: tot-plugin-aws-iam, service: tot-plugin-aws-iam, port: 15008, dir: tot-plugin-aws-iam, idx: 18 }
# - { host: tot-plugin-aws-s3, service: tot-plugin-aws-s3, port: 15007, dir: tot-plugin-aws-s3, idx: 19 }
# - { host: tot-plugin-azure-ad, service: tot-plugin-azure-ad, port: 15009, dir: tot-plugin-azure-ad, idx: 20 }
# - { host: tot-plugin-azure-files, service: tot-plugin-azure-files, port: 15011, dir: tot-plugin-azure-files, idx: 21 }
# - { host: tot-plugin-azure-storage, service: tot-plugin-azure-storage, port: 15012, dir: tot-plugin-azure-storage, idx: 22 }
# - { host: tot-plugin-gcp-bigquery, service: tot-plugin-gcp-bigquery, port: 15002, dir: tot-plugin-gcp-bigquery, idx: 23 }
# - { host: tot-plugin-gcp-iam, service: tot-plugin-gcp-iam, port: 15010, dir: tot-plugin-gcp-iam, idx: 24 }
# - { host: tot-plugin-gcp-storage, service: tot-plugin-gcp-storage, port: 15003, dir: tot-plugin-gcp-storage, idx: 25 }
# - { host: tot-plugin-hive, service: tot-plugin-hive, port: 15006, dir: tot-plugin-hive, idx: 26 }
# - { host: tot-plugin-hdfs, service: tot-plugin-hdfs, port: 15004, dir: tot-plugin-hdfs, idx: 27 }
# - { host: tot-plugin-jdbc, service: tot-plugin-jdbc, port: 15001, dir: tot-plugin-jdbc, idx: 28 }
# - { host: tot-plugin-jdbc-denodo, service: tot-plugin-jdbc-denodo, port: 15018, dir: tot-plugin-jdbc-denodo, idx: 29 }
# - { host: tot-plugin-jdbc-oracle, service: tot-plugin-jdbc-oracle, port: 15013, dir: tot-plugin-jdbc-oracle, idx: 30 }
# - { host: tot-plugin-jdbc-redshift, service: tot-plugin-jdbc-redshift, port: 15017, dir: tot-plugin-jdbc-redshift, idx: 31 }
# - { host: tot-plugin-jdbc-sqlserver, service: tot-plugin-jdbc-sqlserver, port: 15005, dir: tot-plugin-jdbc-sqlserver, idx: 32 }
# - { host: tot-plugin-ldap, service: tot-plugin-ldap, port: 15015, dir: tot-plugin-ldap, idx: 33 }
# - { host: tot-plugin-powerbi, service: tot-plugin-powerbi, port: 15016, dir: tot-plugin-powerbi, idx: 34 }
```

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 ansible-inventories/localhost/hosts.yml anjana.yml --tags
stop-anjana
```

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

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

```

root@anjana22:/mnt/d/Repo x + ~
skipping: [ansible1] => (item=[{"host": "portal", "service": "apache2", "port": 80, "idx": 10}, {"hostname": "export1", "ip": "10.150.100.147"}])
skipping: [ansible1] => (item=[{"host": "portal", "service": "apache2", "port": 80, "idx": 10}, {"hostname": "import1", "ip": "10.150.100.147"}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "zookeeper", "service": "zoo
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "postgres", "service": "pos
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "minio", "service": "minio",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "solr", "service": "solr",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "edusa", "service": "edusa",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "hecate", "service": "hecate",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "portuno", "service": "portun
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "zeus", "service": "zeus",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "kerno", "service": "kerno",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "minerva", "service": "minerv
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "viator", "service": "viator",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "hermes", "service": "hermes",
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "tot", "service": "tot", "por
'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "portal", "service": "apache2
'ip': '10.150.100.147'}])

TASK [anjana-utility : Starting [zookeeper] service in [zookeeper1] host if not running] *****
ok: [ansible1]

TASK [anjana-utility : Checking if [zookeeper] service is started in [zookeeper1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [postgresql] service in [postgresql1] host if not running] *****
ok: [ansible1]

TASK [anjana-utility : Checking if [postgresql] service is started in [postgresql1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [minio] service in [minio1] host if not running] *****
ok: [ansible1]

TASK [anjana-utility : Checking if [minio] service is started in [minio1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [solr] service in [solr1] host if not running] *****
ok: [ansible1]

```

- **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 ansible-inventories/localhost/hosts.yml anjana.yml --tags
stop-kerno

```

```

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

```

```

root@anjana22:/mnt/d/Repo x + ~
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "zookeeper", 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "postgres", 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "minio", 'service': 'minio', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "solr", 'service': 'solr', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "edusa", 'service': 'edusa', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "hecate", 'service': 'hecate', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "portuno", 'service': 'portuno', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "zeus", 'service': 'zeus', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "kerno", 'service': 'kerno', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "minerva", 'service': 'minerva', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "viator", 'service': 'viator', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "hermes", 'service': 'hermes', 'ip': '10.150.100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "tot", 'service': 'tot', 'port': '100.147'}])
included: /mnt/d/Repositorios/ansible-binary/roles/anjana-utility/tasks/anjana-start.yml for ansible1 => (item=[{"host": "portal", "service": "apache2", "ip": '10.150.100.147'}])

TASK [anjana-utility : Starting [zookeeper] service in [zookeeper1] host if not running] *****
changed: [ansible1]

TASK [anjana-utility : Checking if [zookeeper] service is started in [zookeeper1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [postgresql] service in [postgresql1] host if not running] *****
changed: [ansible1]

TASK [anjana-utility : Checking if [postgresql] service is started in [postgresql1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [minio] service in [minio1] host if not running] *****
changed: [ansible1]

TASK [anjana-utility : Checking if [minio] service is started in [minio1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [solr] service in [solr1] host if not running] *****
changed: [ansible1]

TASK [anjana-utility : Checking if [solr] service is started in [solr1] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [edusa] service in [edusal] host if not running] *****
changed: [ansible1]

TASK [anjana-utility : Checking if [edusa] service is started in [edusal] host] *****
ok: [ansible1]

TASK [anjana-utility : Starting [hecate] service in [hecatel] host if not running] *****
changed: [ansible1]

```

Backup de los datos

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

```
sudo ansible-playbook -i ansible-inventories/sample/hosts.yml 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-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: /{{anjana.folder}}/data/minio
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    artifacts:
22      - pgsql-libs-13.10
23      - pgsql-13.10
24  ei_solr:
25    path: /{{anjana.folder}}/solr
26    export_import_path: '{{export_import.path}}/solr'
27    artifact_url: /solr/solr-utility/
28    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:

```

persistences:
  s3:
    ●type: minio # minio|aws_s3
    ●access_key: anjana
    ●secret_key: anjanadata
    # MinIO Only -----
    ●host: http://s3service # Ex. http://s3service or http://10.10.10.10
    ●port: 9000
    ●dump_bucket: datadump
    # AWS S3 Only -----
    ●region: <region>
    ●buckets:
      - { bucket: cdn, name: <cdn-bucket> }
      - { bucket: dsa, name: <dsa-bucket> }
      - { bucket: imports, name: <imports-bucket> }
      - { bucket: textarea, name: <textarea-bucket> }
      - { bucket: workflows, name: <workflows-bucket> }
      - { bucket: datadump, name: <datadump-bucket> }

  bbdd:
    ●host: rdbservice # Ex. rdbservice or 10.10.10.10
    ●port: 5432
    ●database: anjana
    ●user: anjana
    ●pass: anjana

  solr:
    ●host: http://indexservice # Ex. http://indexservice or http://10.10.10.10
    ●port: 8983
    ●collections:
      - snapshots
      - kerno
      - audit_logs
      - recommendations
      - business_tags

```

NOTA: Para que esta herramienta funcione correctamente hay que asignar en el archivo hosts.yml dentro del inventario las IP's/dominios correspondientes a los siguientes nodos tal como se muestra en la siguiente imagen y sus comentarios:

```

ansiblehosts:
  hosts:
    ansible1:
      # Nodo en el cual se instala y ejecuta ansible (nodo director)
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost
  exportimportutilityhosts:
    hosts:
      exportimportutility1:
        # Nodo en el cual se instalan y ejecutan los clientes de exportación e importación (minio-mc, postgresql-client, solr-utility, etc)
        # La exportación y los datos a importar son almacenados en este nodo
        ansible_host: <server_ip>
        ansible_port: 22
        service_hostname: localhost

```

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 ansible-inventories/localhost/hosts.yml anjana.yml -v
--tags export

```

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

```

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

```

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 al arranque)

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"
72    url: https://localhost:8080
```

```
root@ga44:~# ll -h /opt/anjana_log/*
-rwxrwx--- 1 anjana anjana 357K Mar  6 14:20 /opt/anjana_log/drittesta.log*
-rwxrwx--- 1 anjana anjana 102K Mar  6 14:20 /opt/anjana_log/edusa.log*
-rwxrwx--- 1 anjana anjana 1.2M Mar  6 14:20 /opt/anjana_log/hecate.log*
-rwxrwx--- 1 anjana anjana 3.6M Mar  6 14:20 /opt/anjana_log/hermes.log*
-rwxrwx--- 1 anjana anjana 6.0M Mar  6 14:20 /opt/anjana_log/kerno.log*
-rwxrwx--- 1 anjana anjana 5.6M Mar  6 14:20 /opt/anjana_log/minerva.log*
-rwxrwx--- 1 anjana anjana 507K Mar  6 14:20 /opt/anjana_log/portuno.log*
-rwxrwx--- 1 anjana anjana 31M Mar  6 14:21 /opt/anjana_log/tot.log*
-rwxrwx--- 1 anjana anjana 1.3M Mar  6 14:20 /opt/anjana_log/viator.log*
-rwxrwx--- 1 anjana anjana 3.9M Mar  6 14:20 /opt/anjana_log/zeus.log*
root@ga44:~#
```

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

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

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

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

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 ansible-inventories/localhost/hosts.yml 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 [totpluginazurerestorage1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluggingcpbigquery1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluggingcpiam1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluggingcpstorage1])
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 [totpluginjdb1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginjdbcdenodo1])
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 [totplugindap1])
skipping: [ansible1] => (item=Removed [/opt/solr-8.6.0] dir from host [totpluginpowershell1])
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: grafana_user
 - password: contraseñaapoderosa123

```
ansible-binary > ansible-inventories > localhost > group_vars >  grafanahosts.yml
 1  ---
 2  # defaults file for ansible-role-grafana
 3
 4  grafana_configure: true
 5  #grafana_packages: moved to os specific vars
 6
 7  grafana_admin_user: grafana_user
 8  grafana_admin_password: contraseñaapoderosa123
 9  grafana_http_port: 3000
10  grafana_conf_file: /etc/grafana/grafana.ini
```
- Prometheus el cual provee las métricas a los cuadros de mandos
 - URL: http://<url_nodo>:9090/targets
- Exporters de Solr/Zookeeper, Apache, postgresql y node, 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 de los propios exporters

Descomentar roles en all.yml

```
ansible-binary > ansible-inventories > localhost > group_vars >  all.yml
176  # Metrics & log monitoring
177  grafana: true
178  prometheus: true
179  nodeexporter: true
180  postgresexporter: true
181  solrexporter: true
182  apacheexporter: true
183  jmespath: true
```

Para lanzar la plataforma de monitorización es el siguiente comando.

```
# Dependencias necesarias en nodos de ansible y en nodo de monitorización
sudo pip3 install jmespath
sudo ansible-galaxy collection install community.grafana

# Lanzar Grafana, Prometheus y exporters
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags monitoring

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

Para lanzar la todos los exporters, es el siguiente comando.

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags exporters
# Solo lanzar uno de los exporters
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags nodeexporter
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags postgresexporter
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags solrexporter
sudo ansible-playbook -i <ruta_inventario>/hosts.yml anjana.yml --tags apacheexporter
```

Debemos de configurar los siguientes archivos para decirle a Prometheus y los exporters de que máquinas van a coger las métricas. Para ello, vamos a ir a la ruta de la máquina de Prometheus /etc/prometheus/file_sd y añadir a cada archivo, la nueva máquina donde hayamos instalado los exporters.

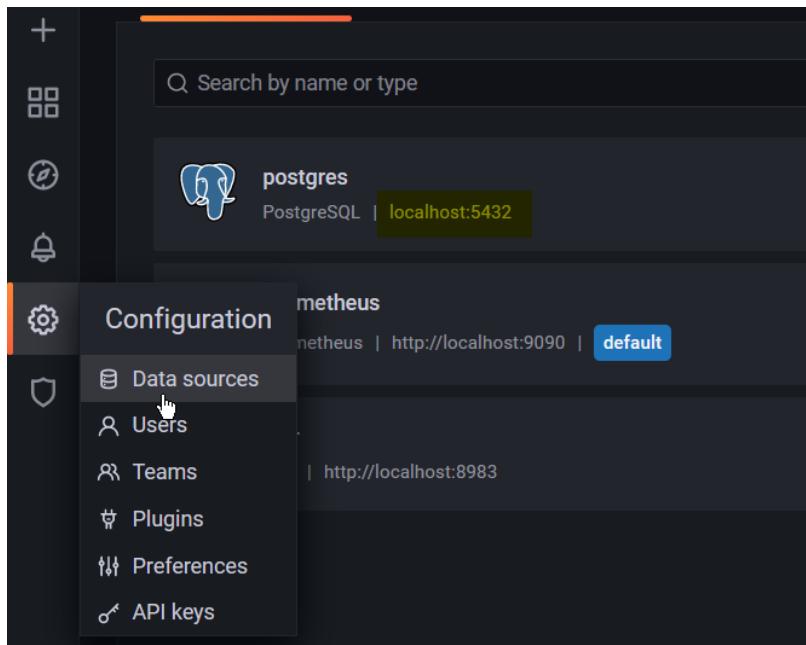
```
root@qa43:~# ll /etc/prometheus/file_sd/
total 32
drwxrwx--- 2 root prometheus 4096 Dec  7 12:15 .
drwxr-xr-x 6 root prometheus 4096 Dec  7 12:20 ..
-rw-r----- 1 root prometheus    94 Dec  7 11:57 apache.yml
-rw-r----- 1 root prometheus   103 Nov 30 11:34 minio.yml
-rw-r----- 1 root prometheus     5 Nov 30 11:34 node.yml
-rw-r----- 1 root prometheus   164 Dec  7 11:58 postgres.yml
-rw-r----- 1 root prometheus   161 Dec  7 11:58 solr.yml
-rw-r----- 1 root prometheus   90 Nov 30 11:34 zookepper.yml
```

Ejemplo con el de Apache: /etc/prometheus/file_sd/apache.yml

```
- labels:
  env: apache
- targets: ['10.150.100.223:9117']
  labels:
    instancename: qa43
- targets: ['10.100.100.1:9117']
  labels:
    instancename: maquina1
```

De esta manera, si tenemos el exporter de apache en cada una de las máquinas, Prometheus cogerá las métricas de ambas máquinas.

Una vez lanzado y editado las configuraciones pertinentes, ya podremos entrar a Grafana y ver los dashboards. En caso de que algún datasource no funcione, puede ser porque no esté bien la URL a donde apunta. Deberemos de cambiarlo en la parte de “Data sources” de la configuración de Grafana.



The screenshot shows the Grafana configuration interface. On the left, there is a sidebar with the following icons and menu items:

- A plus sign icon
- A square icon with four smaller squares
- A magnifying glass icon
- A bell icon
- A gear icon labeled "Configuration"
- A shield icon

The "Configuration" menu is open, showing the following options:

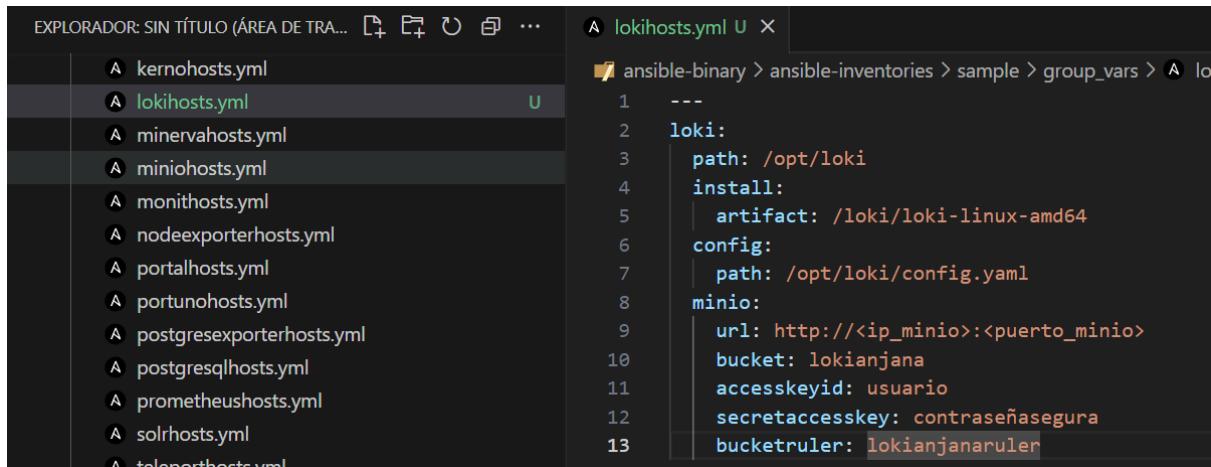
- Data sources (selected, highlighted with a cursor icon)
- Users
- Teams
- Plugins
- Preferences
- API keys

In the main content area, there are two entries listed:

- postgres**: PostgreSQL | localhost:5432
- metheus**: metheus | http://localhost:9090 | default

Despliegue de plataforma de agregación de logs

Hemos añadido una funcionalidad que utiliza loki,promtail 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 promtail y grafana-agent lo debemos de instalar donde este Anjana. Para lanzar Loki, debemos de tocar el archivo de variables, poniendo lo siguiente



```

EXPLORADOR: SIN TÍTULO (ÁREA DE TRABAJO)  lokihosts.yml
  A kernohosts.yml
  A lokihosts.yml
  A minervahosts.yml
  A miniohosts.yml
  A monithosts.yml
  A nodeexporterhosts.yml
  A portalhosts.yml
  A portunohosts.yml
  A postgrexporterhosts.yml
  A postgresqlhosts.yml
  A prometheushosts.yml
  A solrhosts.yml
  A teleorthosts.yml

A lokihosts.yml
  1 ---
  2 loki:
  3   path: /opt/loki
  4   install:
  5     artifact: /loki/loki-linux-amd64
  6   config:
  7     path: /opt/loki/config.yaml
  8   minio:
  9     url: http://<ip_minio>:<puerto_minio>
 10    bucket: lokianjana
 11    accesskeyid: usuario
 12    secretaccesskey: contraseñaasegura
 13    bucketruler: lokianjanaruler

```

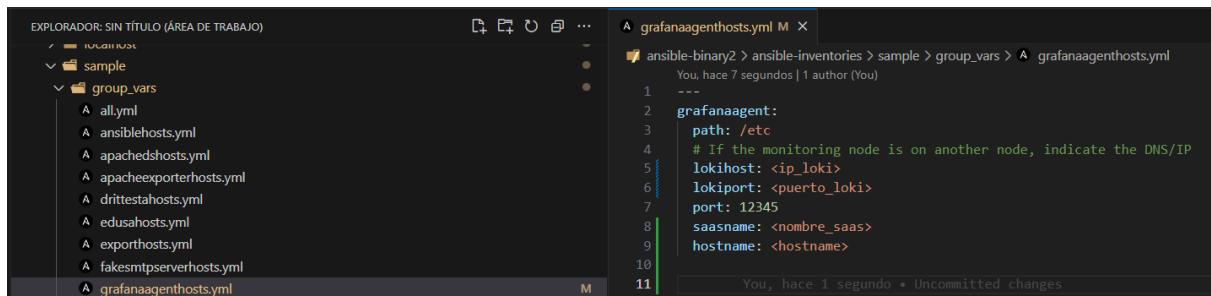
En el apartado de MinIO, debemos de añadir la URL de un servicio de S3 donde se subirán los logs como MinIO, creando dos buckets con los nombres ahí puestos, y el usuario y contraseña. Las rutas son las de instalación y la de configuración, que por defecto serán /opt/loki

Para lanzar loki, se lanza de la siguiente manera

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml <ruta_ansible_binary>/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.

Una vez tengamos Loki, pasaremos a lanzar el grafana-agent en la máquina de Anjana. En ansible, deberemos de modificar el archivo grafanaagenthosts y modificar lo siguiente



```

EXPLORADOR: SIN TÍTULO (ÁREA DE TRABAJO)
  A all.yml
  A ansiblehosts.yml
  A apachedhosts.yml
  A apacheexporterhosts.yml
  A dritttestahosts.yml
  A edusahosts.yml
  A exphosts.yml
  A fakesmtserverhosts.yml
  A grafanaagenthosts.yml

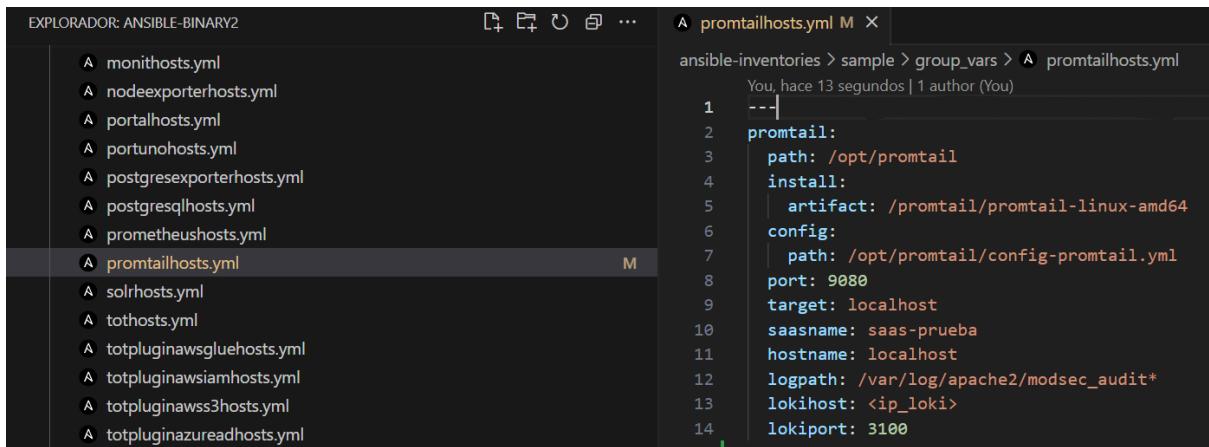
A grafanaagenthosts.yml
  1 ---
  2 grafanaagent:
  3   path: /etc
  4   # If the monitoring node is on another node, indicate the DNS/IP
  5   lokihost: <ip_loki>
  6   lokiport: <puerto_loki>
  7   port: 12345
  8   saasname: <nombre_saas>
  9   hostname: <hostname>
10
11 You, hace 1 segundo * Uncommitted changes

```

Aquí, lo que debemos de modificar, son la IP y el puerto donde se encuentra loki, el puerto del grafana-agent, y debemos de poner el nombre del saas y del host de la máquina, para que en grafana después podamos filtrar por saas e instancia. Para lanzarlo, pondremos el siguiente comando

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml <ruta_ansible_binary>/anjana.yml
--tags grafanaagent
```

También hemos creado un rol llamado promtail que recoge los datos de modsecurity, y lo pasamos a grafana para tener un panel con los logs. Para ello, tenemos que modificar el archivo promtailhosts.yml.



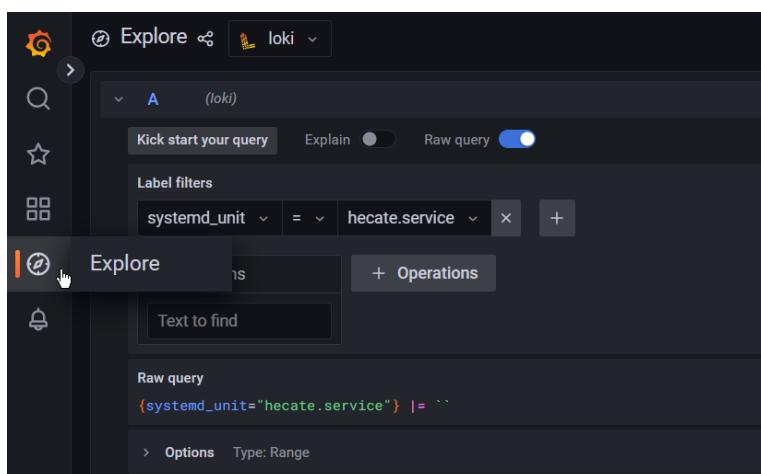
```
ansible-inventories > sample > group_vars > promtailhosts.yml
You, hace 13 segundos | 1 author (You)
1 ---|
2 promtail:
3   path: /opt/promtail
4   install:
5     artifact: /promtail/promtail-linux-amd64
6   config:
7     path: /opt/promtail/config-promtail.yml
8   port: 9080
9   target: localhost
10  saasname: saas-prueba
11  hostname: localhost
12  logpath: /var/log/apache2/modsec_audit*
13  lokihost: <ip_loki>
14  lokiport: 3100
```

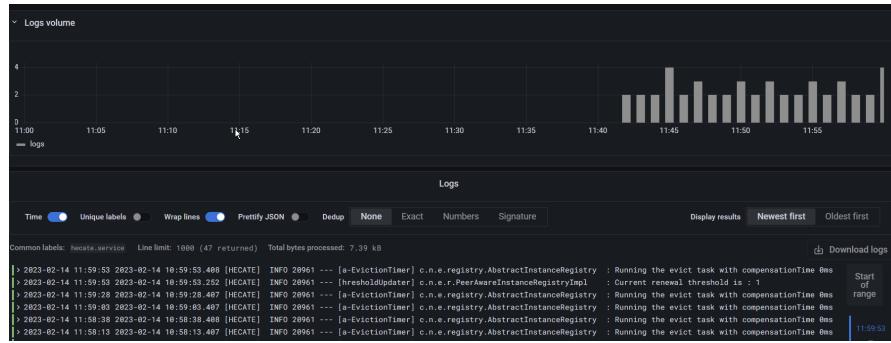
En este archivo podremos cambiar la máquina y puerto donde se encuentra loki, el nombre de la instancia y del Saas, el puerto de promtail, y sus rutas.

Una vez cambiado, lanzamos el siguiente comando

```
sudo ansible-playbook -i <ruta_inventario>/hosts.yml <ruta_ansible_binary>/anjana.yml
--tags promtail
```

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





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.
- 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, nodeexporter, postgresexporter, solrexporter y apacheexporter → 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
- 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-aqativa
 - Tot-plugin-powerbi
 - Tot-plugin-tableau
 - Tot-plugin-ranger

Tags disponibles

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

| | |
|--|---|
| anjana-sample | El tag más completo: despliega Anjana, hace un insert del set de datos y los yaml's 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). |
| 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 drittesta minerva viator hermes tot horus | Despliega el microservicio seleccionado. Las tareas que hace son borrar y descargar los jars, launchers y crear o sobreescribir el existente descriptor del servicio |
| portal | Despliega Apache2 con el front de Anjana y admin. Crea o sobreescribe los archivos de configuración de Apache para ambos front |
| tot-plugin-aqativa 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-aqativa tot-plugin-powerbi tot-plugin-tableau tot-plugin-ranger | Despliega el microservicio seleccionado. Las tareas que hace son borrar y descargar los jars, launchers y crear o sobreescribir el existente descriptor del servicio |
| tot-plugins | Lanza todos los plugins de Tot que estén descomentados en el playbook. Las tareas que hace son borrar y descargar los jars, launchers y crear o sobreescribir el existente descriptor del servicio |
| update-edusa update-hecate update-portuno | Solo ejecuta las tareas de actualizar los jars y los launchers del microservicio escogido |

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

| | |
|--|---|
| <code>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-aqativa start-tot-plugin-powerbi start-tot-plugin-tableau start-tot-plugin-ranger</code> | |
| <code>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-aqativa 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-aqativa stop-tot-plugin-powerbi stop-tot-plugin-tableau stop-tot-plugin-ranger</code> | Detiene el microservicio seleccionado parando primero de forma selectiva y ordenada todos los microservicios que dependen de él |
| <code>remove-anjana</code> | 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. |

| | |
|---|---|
| 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/iprofile2 y comprobar la conexión al artifactory |
| 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. |
| 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 role 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”. Nota: para el cliente de postgresql se necesita conexión a internet (se usa en export-import/utility y persistence/utility). |
| grafana | Despliega Grafana en la máquina que se le indique y los dashboards y datasources. |
| prometheus | Despliega Prometheus en la máquina que se le indique. |
| monitoring | Despliega Grafana y Prometheus con todos los dashboards y datasources necesarios. |
| nodeexporter | Despliega exporter de node, que trae métricas sobre CPU, RAM de la máquina, etc |

| | |
|---|--|
| postgresexporter | Despliega exporter de postgres, que trae métricas sobre la/las BBDD de postgres que se le indique. |
| solrexporter | Despliega exporter de postgres, que trae métricas sobre el/los Solr que se le indique. |
| apacheexporter | Despliega exporter de apache, que trae métricas sobre el/los apache que se le indique. |
| exporters | Despliega todos los exporters. |
| grafanaagent | Despliega Grafana-agent, en los nodos que queramos capturar su log y los manda a Loki |
| promtail | Despliega Promtail, el cual coge los logs de modsecurity y lo manda a Loki |
| 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 del rol de edusa. |
| 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 ansible-inventories/localhost/hosts.yml 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-functionality-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 |

Funciones adicionales del kit

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 ansible-inventories/sample/hosts.yml 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. Para el cliente de postgresql se necesita conexión a internet (se usa en export-import-utility y persistence-utility).

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

Sampleddata

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 ansible-inventories/sample/hosts.yml anjana.yml -v  
--tags 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 ansible-inventories/sample/hosts.yml 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 ansible-inventories/sample/hosts.yml 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 ansible-inventories/sample/hosts.yml anjana.yml -v
--tags reset
```

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-inventories > localhost > group_vars > all.yml
09
70  persistences:
71    s3:
72      type: minio # minio|aws_s3
73      access_key: anjana
74      secret_key: anjanadata
75      # MinIO Only -----
76      host: http://s3service # Ex. http://s3service
77      port: 9000
78      dump_bucket: datadump
79      # AWS S3 Only -----
80      region: <region>
```

El comando a lanzar es:

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

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

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

Balcedor/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 y el registro la url de MinIO de portuno.app_portuno.
- balancer.minioProxyPass para indicar los nodos de minio en los cuales balancear la carga o aplicar la alta disponibilidad

```

  ↻ exphosts.yml
  ↻ fakesmtserverhosts.yml
  ↻ hecatehosts.yml
  ↻ hermeshosts.yml
  ↻ importhosts.yml
  ↻ kernohosts.yml
  ↻ minervahosts.yml
  ↻ miniohosts.yml
  ↻ monithosts.yml
  ↻ portalhosts.yml
  ↻ portunohosts.yml
  ↻ postgresqlhosts.yml
  ↻ solrhosts.yml
  ↻ tothosts.yml
  ↻ totpluginawsgluehosts.yml
  ↻ totpluginawsiamhosts.yml
  ↻ totpluginawss3hosts.yml
  ↻ totpluginazureadhosts.yml
  ↻ totpluginazurefileshosts.yml
  ↻ totpluginazurestoragehosts.yml
  ↻ totpluginawsigquenhosts.yml

  ↻ databaseSchema.yml
  ↻ minio:
    url: http://<front_node>:9100
    accessKey: anjana
    secretKey: anjanadata
  ↻ Mail_connection_properties

```

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 ansible-inventories/localhost/hosts.yml anjana.yml --tags
update-anjana-sequences

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

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

NOTA: Al tener cambios en la persistencia de la bbdd 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
Cmnd_Alias SERVICES      = /usr/bin/journalctl *, /usr/bin/systemctl *
Cmnd_Alias READ          = /usr/bin/cat *, /usr/bin/less *, /usr/bin/tail *
#Cmnd_Alias WRITE         = /usr/bin/nano /opt/*, /usr/bin/vim /opt/*
Cmnd_Alias NETWORK        = /usr/bin/netstat *, /usr/bin/ss *
Cmnd_Alias APT            = /usr/bin/apt *
Cmnd_Alias ANSIBLE        = /usr/bin/nano /opt/ansible/anjana.yml, /usr/bin/nano
/opt/ansible/ansible-inventories/*
Cmnd_Alias ANSIBLE_PLAYBOOK = /usr/bin/ansible-playbook *
Cmnd_Alias ANSIBLE_VIM   = /usr/bin/vim /opt/ansible/anjana.yml, /usr/bin/vim
/opt/ansible/ansible-inventories/*

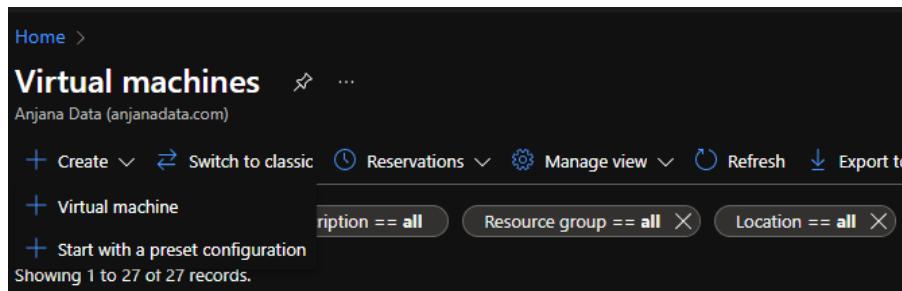
Cmnd_Alias ANJANA_ALL    = SERVICES, READ, NETWORK, APT, ANSIBLE, ANSIBLE_VIM,
ANSIBLE_PLAYBOOK
%<grupo_técnicos> ALL=ANJANA_ALL
<usuario_técnico> ALL=ANJANA_ALL
```

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
```

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

```
ansible-inventories > sample > group_vars > all.yml
  1  #-----#
  2  # Ansible Kit 4.a5 #
  3  #-----#
  4  version:
  5  # MICROSERVICES -
  6  edusa: 4.4.0
  7  hecate: 4.4.0
  8  horus: 4.4.0
  9  hermes: 4.4.0
 10 kerno: 4.4.0
 11 minerva: 4.4.0
 12 portuno: 4.4.0
 13 tot: 4.4.0
 14 viator: 4.4.0
 15 zeus: 4.4.0
 16 portal: 4.4.0
 17 admin: 4.4.0
 18 drittesta: 4.4.0
 19 # UTILITIES -
```

5. Editar group_vars/all.yml

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

```
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
```

b. Url y credenciales de RDS

```
persistences:
  s3:
    type: aws_s3 # minio|aws_s3
    access_key: AAAABBBBBCCCCDDDD
    secret_key:aaaaaaaaabbbbbbbbbbccccccccddddd
    # MinIO Only -----
    host: http://s3service # Ex. http://s3service or http://10.10.10.10
    port: 9000
    dump_bucket: datadump
    # AWS S3 Only -----
    region: eu-central-1
    buckets:
      - { bucket: cdn, name: cdn-pre-env1 }
      - { bucket: dsa, name: dsa-pre-env1 }
      - { bucket: imports, name: imports-pre-env1 }
      - { bucket: textarea, name: textarea-pre-env1 }
      - { bucket: workflows, name: workflows-pre-env1 }
      - { bucket: datadump, name: datadump-pre-env1 }
  bbdd:
    host: rdbservice # Ex. rdbservice or 10.10.10.10
    port: 5432
    database: anjana
    user: anjana
    pass: anjana
  solr:
    host: http://indexservice # Ex. http://indexservice or http://10.10.10.10
```

6. Sustituimos la ruta y credenciales de la conexión a la base de datos en los templates de la configuración de los microservicios en
“ansible-inventories/<inventario>/templates/config_core/xxxx”

EXPLORER

- ANSIBLE-BINARY
 - ansible-inventories
 - > k8s
 - > localhost
 - > sample
 - > group_vars
 - > templates
 - config_core
 - configserver-gitfiles.yaml.j2
 - configserver-localfiles.yaml.j2
 - drittesta.config.yaml.j2
 - hecate.config.yaml.j2
 - hermes.config.yaml.j2
 - horus.config.yaml.j2
 - kemo.config.yaml.j2
 - minerva.config.yaml.j2
 - portuno.config.yaml.j2
 - tot.config.yaml.j2

hosts.yml all.yml kemo.config.yaml.j2

```

password: anjana
jpa:
  properties:
    hibernate:
      default_schema: anjana
eureka:
  client:
    serviceUrl:
      defaultZone: http://hecatesserver:50761/eureka
anjana:
  scheduling:
    datasource:
      .url: jdbc:postgresql://rdbservice:5432/anjana?currentSchema=portuno
      .username: anjana
      .password: anjana
  security:
    oauth2:
      resource:

```

```

anjana:
  datasource:
    driver-class-name: org.postgresql.Driver
    url: jdbc:postgresql://anjanards.cgikxvfi33gk.eu-west-1.rds.amazonaws.com:5432/anjanardsdatabase
    password: V4xq7Fc4edfUPhD
    username: anjanauer
  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. seteadas 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 ansible-inventories/sample/hosts.yml anjana.yml -tags anjana-sample
```

EXPLORER ...

ANSIBLE-BINARY

- ansible-inventories
 - sample
 - singlenode_&_rds
 - > group_vars
 - hosts.yml
 - roles
- anjanayml

hosts.yml X

```

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@anjanaya17:/mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&_rds/hosts.yml anjanayml --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 setting 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 -vvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtserverhosts

PLAY [Import role postgresql] ****
TASK [Gathering Facts] ****
ok: [postgresql1]

PLAY [Import role zookeeper] ****
TASK [Gathering Facts] ****

```

> OUTLINE > TIMELINE

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

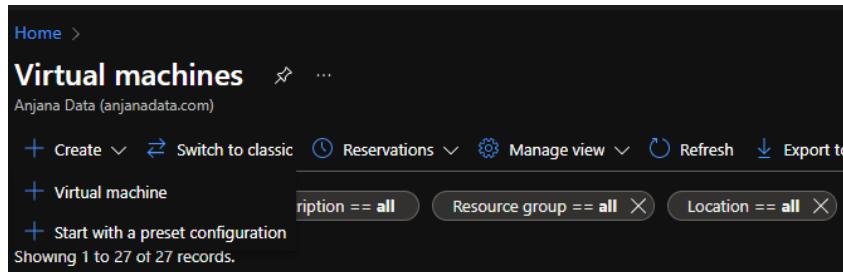
```

PLAY RECAP ****
apacheds1           : ok=27   changed=18   unreachable=0   failed=0    skipped=5   rescued=0   ignored=0
drittesta1          : 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
fakesmtserver1      : 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

```

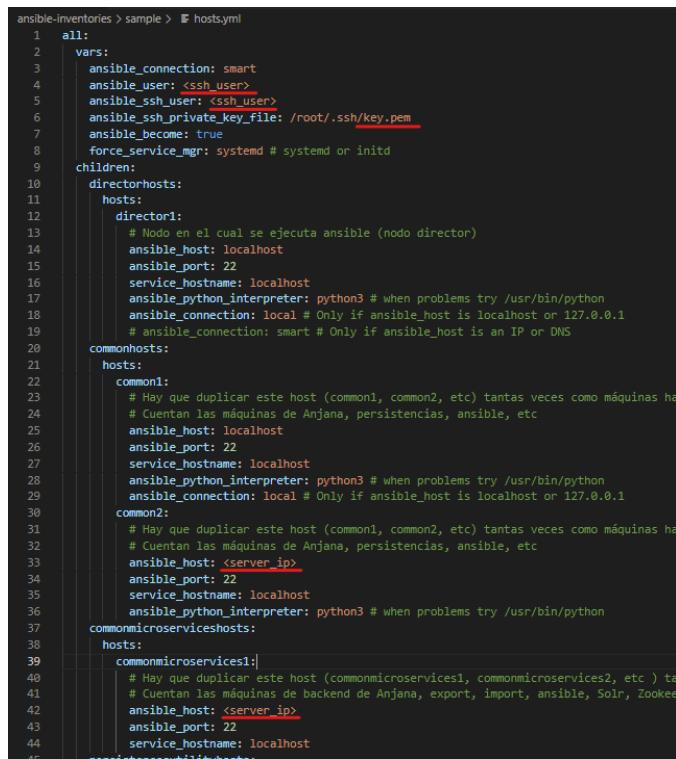
Despliegue single node

1. Creamos una máquina virtual



The screenshot shows the 'Virtual machines' section of the Anjana Data interface. At the top, there are navigation links for 'Home' and 'Virtual machines'. Below that is a search bar with filters: 'Subscription == all', 'Resource group == all', and 'Location == all'. The main area displays a list of 27 virtual machines.

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 # systemctl 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              ansible_python_interpreter: python3 # when problems try /usr/bin/python

```

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

```
ansible-inventories > sample > group_vars > all.yml
 1  #-----#
 2  # Ansible Kit 4.a5 #
 3  #-----#
 4  version:
 5  # MICROSERVICES -----
 6  edusa: 4.4.0
 7  hecate: 4.4.0
 8  horus: 4.4.0
 9  hermes: 4.4.0
10  kerno: 4.4.0
11  minerva: 4.4.0
12  portuno: 4.4.0
13  tot: 4.4.0
14  viator: 4.4.0
15  zeus: 4.4.0
16  portal: 4.4.0
17  admin: 4.4.0
18  drittesta: 4.4.0
19  # UTILITIES -----
```

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    external:
64      mavenurl: https://artifactory.anjanadata.org:8443/repository/anjanareleases
65      mvnrepo: anjanareleases
66      npmurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesnpm
67      rawurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesraw
68
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:
- acceso por ssh a la máquina
 - seteadas las variables con los datos correctos
 - 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 ansible-inventories/sample/hosts.yml anjana.yml -v -tags
anjana-sample
```

EXPLORER ...

ANSIBLE-BINARY

- ansible-inventories
 - sample
 - singlenode_&_rds
 - > group_vars
 - hosts.yml
 - roles
- anjanayml

hosts.yml X

```

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@anjanaya17:/mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&_rds/hosts.yml anjanayml --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 -vvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtserverhosts

PLAY [Import role postgresql] ****
TASK [Gathering Facts] ****
ok: [postgresql1]

PLAY [Import role zookeeper] ****
TASK [Gathering Facts] ****

```

> OUTLINE > TIMELINE

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
fakesmtserver1 : 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
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=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
- 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:

```
root, 10 seconds ago | 5 authors (ajgomezjanja and others)
all:
  children:
    commonhosts:
      hosts:
        common1:
          ansible_host: <server_ip>
          ansible_port: 22
          service_hostname: localhost
        common2:
          ansible_host: <server_ip>
          ansible_port: 22
          service_hostname: localhost
        common3:
          ansible_host: <server_ip>
          ansible_port: 22
          service_hostname: localhost
        common4:
          ansible_host: <server_ip>
          ansible_port: 22
          service_hostname: localhost
```

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:
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost
    commonmicroservices2:
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost
    commonmicroservices3:
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost
    commonmicroservices4:
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost

```

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 MinIO por ejemplo quedarían excluidos.

Ejemplo de back:

```

hosts.yml U ●
ansible-inventories > ha_resiliencia > hosts.yml
1   all:
2     children:
3       edusahosts:
4         hosts:
5           edusa1:
6             ansible_host: <IP_servidor>
7             ansible_port: 22
8             service_hostname: localhost
9           edusa2:
10          ansible_host: <IP_servidor>
11          ansible_port: 22
12          service_hostname: localhost
13       hecatehosts:
14         hosts:
15           hecate1.

```

Ejemplo de front

```

hosts.yml U ●

ansible-inventories > ha_resiliencia > hosts.yml
106     ansible_host: <IP_servidor>
107     ansible_port: 22
108     service_hostname: localhost
109     portalhosts:
110         hosts:
111             portal1:
112                 ansible_host: <IP_servidor>
113                 ansible_port: 22
114                 service_hostname: localhost
115             portal2:
116                 ansible_host: <IP_servidor>
117                 ansible_port: 22
118                 service_hostname: localhost
119     monithosts:

```

Ejemplo de Solr

```

hosts.yml U ●

ansible-inventories > ha_resiliencia > hosts.yml
129     ansible_port: 22
130     service_hostname: localhost
131     solrhosts:
132         hosts:
133             solr1:
134                 ansible_host: <IP_servidor>
135                 ansible_port: 22
136                 service_hostname: localhost
137             solr2:
138                 ansible_host: <IP_servidor>
139                 ansible_port: 22
140                 service_hostname: localhost
141             solr3:
142                 ansible_host: <IP_servidor>
143                 ansible_port: 22
144                 service_hostname: localhost
145     zookeeper-nodes:
146         hosts:

```

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
10  children:
11    directorhosts:
12      hosts:
13        director1:
14          # Nodo en el cual se ejecuta ansible (nodo director)
15          ansible_host: localhost
16          ansible_port: 22
17          service_hostname: localhost
18          ansible_python_interpreter: python3 # when problems try /usr/bin/python
19          ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
20          # ansible_connection: smart # Only if ansible_host is an IP or DNS
21    commonhosts:
22      hosts:
23        common1:
24          # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
25          # Cuentan las máquinas de Anjana, persistencias, ansible, etc
26          ansible_host: localhost
27          ansible_port: 22
28          service_hostname: localhost
29          ansible_python_interpreter: python3 # when problems try /usr/bin/python
30          ansible_connection: local # Only if ansible_host is localhost or 127.0.0.1
31        common2:
32          # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas ha
33          # Cuentan las máquinas de Anjana, persistencias, ansible, etc
34          ansible_host: <server_ip>
35          ansible_port: 22
36          service_hostname: localhost
37          ansible_python_interpreter: python3 # when problems try /usr/bin/python

```

4. Solr y Zookeeper en modo cluster requieren de los siguientes ajustes:

- En “sample/group_vars/solrhosts.yml” ajustar la ip de conexión con cada nodo de zookeeper desde los nodos solr (dejar localhost en singlenode)

```

 1  You, seconds ago | 3 authors (ajgomezdanja and others)
 2  ---
 3  #host: not used
 4  #ip: used in solr.service in Environment=ZK_HOST
 5  zookeeper_hosts:
 6    - host: "zknode1"
 7      ip: 10.0.0.12 # localhost on singlenode, ip/hostname of zk_node1 on cluster
 8    - host: "zknode2"
 9      ip: 10.0.0.13 # localhost on singlenode, ip/hostname of zk_node1 on cluster
10    - host: "zknode2"
11      ip: 10.0.0.14 # localhost on singlenode, ip/hostname of zk_node1 on cluster
12
13  installation:
14    startup: systemd

```

Más abajo en el mismo solrhosts.yml ajustar los hosts de zookeeper (localhost:2181 en singlenode)

```

 18  solr_version: "8.6.0"
 19  solr_mirror: "https://archive.apache.org/dist"
20  solr_remove_crfu: false
21
22  solr_service_manage: true
23  solr_service_name: solr
24  solr_service_state: started
25
26  solr_zookeeper_client_path: {{anjana.folder}}/zookeeper/bin
27  ## IMPORTANT > hostname or alias reachable/accessible from solr nodes
28  solr_zookeeper_hosts: 10.0.0.12:2181,10.0.0.13:2181,10.0.0.14:2181
29  solr_zookeeper_hosts_solr_path: solrcloud
30

```

- En el archivo sample/hosts.yml ajustar la ip de conexión entre los nodos de zookeeper y el myid (dejar localhost en singlenode)

```

141     sources:
142         ansible_host: 20.123.247.161
143         ansible_port: 22
144         service_hostname: localhost
145     zookeeper-nodes:
146         hosts:
147             zookeeper1:
148                 ansible_host: 20.123.247.115
149                 nodeip: 10.0.0.12 # localhost on singlenode, ip/hostname on cluster; used in /.../conf/zoo.cfg
150                 myid: 1           # used in /.../data/zookeeper/myid and /.../conf/zoo.cfg
151                 ansible_port: 22
152                 service_hostname: localhost
153             zookeeper2:
154                 ansible_host: 20.123.247.162
155                 nodeip: 10.0.0.13 # localhost on singlenode, ip/hostname on cluster; used in /.../conf/zoo.cfg
156                 myid: 2           # used in /.../data/zookeeper/myid and /.../conf/zoo.cfg
157                 ansible_port: 22
158                 service_hostname: localhost
159             zookeeper3:
160                 ansible_host: 20.123.247.161
161                 nodeip: 10.0.0.14 # localhost on singlenode, ip/hostname on cluster; used in /.../conf/zoo.cfg
162                 myid: 3           # used in /.../data/zookeeper/myid and /.../conf/zoo.cfg

```

5. Para MinIO cluster habrá que editar el miniohosts.yml en el inventario para ajustar los volúmenes a los cuatro nodos que hay disponibles.

Se comentará la opción de volumes para el modo Standalone y se descomenta y ajusta la línea de volumes para MinIO cluster

| | |
|---|--|
| <pre> A hecatehosts.yml A hermeshosts.yml A horushosts.yml A kernohosts.yml A lokihosts.yml A minervahosts.yml A miniohosts.yml M A monithosts.yml A nodeexporterhosts.yml A persistencesutilityhosts.yml A portalhosts.yml A portunohosts.yml A postgrexporterhosts.yml A postgresqlhosts.yml A prometheushosts.yml A promtailhosts.yml </pre> | <pre> You, 12 seconds ago 3 authors (eduardomasegos and others) minio_version: RELEASE.2022-05-26T05-48-41Z minio: consoleaddr: ':9001' auth_metrics: public # MinIO Standalone Only # MinIO Cluster / Distributed Volumes volumes: http://<minio_host_1>/{{anjana.folder}}/data/minio http://<minio_host_2>/{{anjana.folder}}/data/minio http://<minio_host_3>/{{anjana.folder}}/data/minio http://<minio_host_4>/{{anjana.folder}}/data/minio path:/{{anjana.folder}}/minio nas: false datapath:/{{anjana.folder}}/data/minio install: artifact: /minio/minio.{{ minio_version }} artifactclient: /minio/mc </pre> |
|---|--|

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

```

ansible-inventories > sample > group_vars > all.yml
1  #-----#
2  # Ansible Kit 4.a5 #
3  #-----#
4  version:
5  # MICROSERVICES -----
6  edusa: 4.4.0
7  hecate: 4.4.0
8  horus: 4.4.0
9  hermes: 4.4.0
10 kerno: 4.4.0
11 minerva: 4.4.0
12 portuno: 4.4.0
13 tot: 4.4.0
14 viator: 4.4.0
15 zeus: 4.4.0
16 portal: 4.4.0
17 admin: 4.4.0
18 drittesta: 4.4.0
19 # UTILITIES -----

```

7. 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

```

8. En `group_vars/portalhosts.yml` hay que ajustar los endpoints de los proxypass con los nuevos nodos:

```

modsecurity: v2.1.0
balancer:
  swagger: |
    BalancerMember http://viatorserver_node1:8085/swagger
  swaggerapi: |
    BalancerMember http://viatorserver_node1:8085/v3/api-docs
  gateway: |
    BalancerMember http://viatorserver_node1:8085
    BalancerMember http://viatorserver_node2:8085
  portuno: |
    BalancerMember http://portunoserver:8998/portuno
    BalancerMember http://portunoserver_node2:8998/portuno
  cdn: |
    BalancerMember http://s3service_node1:9000/cdn
    BalancerMember http://s3service_node2:9000/cdn
    BalancerMember http://s3service_node3:9000/cdn
    BalancerMember http://s3service_node4:9000/cdn
  dsa: |
    BalancerMember http://s3service_node1:9000/dsa
    BalancerMember http://s3service_node2:9000/dsa
    BalancerMember http://s3service_node3:9000/dsa

```

9. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos.

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

- acceso por ssh a la máquina
- seteadas las variables con los datos correctos
- 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 ansible-inventories/sample/hosts.yml all -m ping
```

```

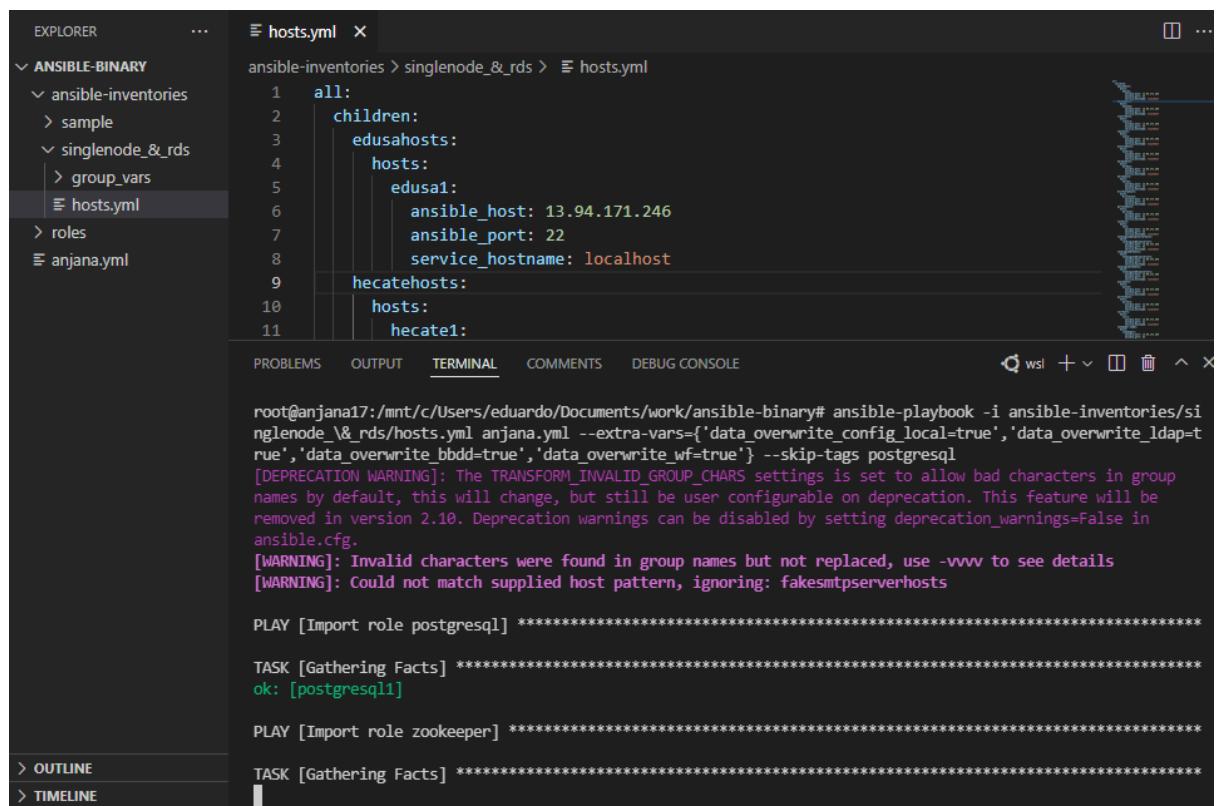
root@anjana17:/mnt/c/Users/eduardo/Documents/work/ansible-binary-Copy# ansible -i ansible-inventories/sample/hosts_ha.yml all -m ping
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change deprecation. This feature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

```

11. Lanzamos el despliegue

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

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



The screenshot shows the VS Code interface with the hosts.yml file open in the editor. The file contains configuration for Ansible, defining groups like 'all', 'edusahosts', and 'hecatehosts' with specific host definitions. Below the code editor is a terminal window showing the execution of an ansible-playbook command. The terminal output includes several warning messages about deprecated features and invalid group names, followed by the playbooks being imported and tasks being gathered for the 'postgresql' and 'zookeeper' roles.

```

ansible-inventories > singlenode_&_rds > hosts.yml
1  all:
2    children:
3      edusahosts:
4        hosts:
5          edusal1:
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 -vvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtpserverhosts

PLAY [Import role postgresql] ****
TASK [Gathering Facts] ****
ok: [postgresql1]

PLAY [Import role zookeeper] ****
TASK [Gathering Facts] ****

```

12. 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
dritttestal : ok=27 changed=10 unreachable=0 failed=0 skipped=8 rescued=0 ignored=0
dritttesta2 : ok=27 changed=10 unreachable=0 failed=0 skipped=8 rescued=0 ignored=0
edusal : 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
fakesmtpproxy1 : 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
fakesmtpproxy1 : 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
minerval : 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
portall1 : ok=35 changed=29 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
portall2 : 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:

- Deberemos copiar tantos templates como servicios queramos en el apartado de ansible-inventories/<inventario>/templates/service_plugins y dependiendo si es systemd o initd, entrar a esa ruta, copiar el archivo de ese plugin cambiando el nombre a tot-plugin-XXX2.service.j2

```

    <pre>
        <code>
            <div>
                <ul>
                    <li><span>ansible-inventories</span></li>
                    <li><span>k8s</span></li>
                    <li><span>localhost</span></li>
                        <ul>
                            <li><span>group_vars</span></li>
                            <li><span>templates</span></li>
                                <ul>
                                    <li><span>config_core</span></li>
                                    <li><span>config_plugins</span></li>
                                    <li><span>monit</span></li>
                                    <li><span>portal</span></li>
                                    <li><span>services_core</span></li>
                                    <li><span>services_plugins</span></li>
                                        <ul>
                                            <li><span>initd</span></li>
                                            <li><span>systemd</span></li>
                                                <ul>
                                                    <li><span>tot-plugin-aqtvia.service.j2</span></li>
                                                    <li><span>tot-plugin-aws-glue.service.j2</span></li>
                                                    <li><span>tot-plugin-aws-iam.service.j2</span></li>
                                                    <li><span>tot-plugin-aws-s3.service.j2</span></li>
                                                    <li><span>tot-plugin-azure-ad.service.j2</span></li>
                                                    <li><span>tot-plugin-azure-files.service.j2</span></li>
                                                    <li><span>tot-plugin-azure-storage.service.j2</span></li>
                                                    <li><span>tot-plugin-gcp-bigquery.service.j2</span></li>
                                                    <li><span>tot-plugin-gcp-iam.service.j2</span></li>
                                                    <li><span>tot-plugin-gcp-storage.service.j2</span></li>
                                                    <li><span>tot-plugin-hdfs.service.j2</span></li>
                                                    <li><span>tot-plugin-hive.service.j2</span></li>
                                                    <li><span>tot-plugin-jdbc-denodo.service.j2</span></li>
                                                    <li><span>tot-plugin-jdbc-oracle.service.j2</span></li>
                                                    <li><span>tot-plugin-jdbc-redshift.service.j2</span></li>
                                                    <li><span>tot-plugin-jdbc-sqlserver.service.j2</span></li>
                                                </ul>
                                                <li><span>tot-plugin-jdbc.service.j2</span></li>
                                                <li><span>tot-plugin-jdbc2.service.j2</span></li>
                                                <li><span>tot-plugin-ldap.service.j2</span></li>
                                            </ul>
                                        </li>
                                    </ul>
                                </li>
                            </ul>
                        </li>
                    </ul>
                </div>
            </code>
        </pre>
    
```

- En el archivo “groups_vars/totpluginXXXhosts.yaml” en el apartado totpluginXXXhosts.instances añadir al array los nombres de los templates creados anteriormente.

```

instances:
  - tot-plugin-jdbc.service.j2
  - tot-plugin-jdbc2.service.j2
  
```

- Debemos ir al template del inventario de ansible, y modificamos el profile del descriptor de servicio duplicado por uno diferente.

```

ansible > ansible-inventories > localhost > templates > services_plugins > systemd > tot-plugin-jdbc2.service.j2
 1 [Unit]
 2 Description=Anjana tot plugin jdbc server
 3 Requires=network.target
 4 After=network.target tot.service
 5
 6 [Service]
 7 Environment=HOSTNAME={{ service_hostname }}
 8 {% if (ansible_pkg_mgr == "yum" or ansible_pkg_mgr == "dnf") -%}
 9     Environment=JAVA_HOME={{installation.javaPATH}}/jre-1.8.0-openjdk
10 {% else %}
11     Environment=JAVA_HOME={{installation.javaPATH}}/java-8-openjdk-amd64
12 {% endif %}
13 WorkingDirectory={{ totpluginjdbc.install.path }}
14 TimeoutStopSec=20
15 ExecStart=%{if (installation.type == "external") }{{ totpluginjdbc.install.path }}%{endif} %{{ if
16 (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 }} {{ totpluginjdbc.install.javaopts }} -jar {{ totpluginjdbc.
install.path }}/tot-plugin-jdbc.jar --spring.config.import={{ totpluginjdbc.install.edusaurl }} --spring.profiles.active=profile2
--spring.cloud.config.failFast={{ installation.failFast }} %{{ if not (installation.eurekaPreferIpAddress ) }}--eureka.instance.
preferIpAddress=false%{{ endif }}%
 16

```

- Para lanzar este nuevo role, debemos de ejecutar el siguiente comando:

```

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

```

- Como último paso, vamos a la carpeta de archivos de configuración de los microservicios, duplicamos el archivo de configuración que ya está, y cambiamos el nombre a application-<nombre_perfilnuevo>.yaml. Una vez creado, lo editamos y debemos de modificar los siguientes campos. Hay que tener precaución de que el nuevo fichero no tenga permisos distintos para que Edusa no tenga problemas de permisos para leerlo.

```

server:
  max-http-header-size: 20000
  port: 15111
  compression:
    enabled: false # Whether response compression is enabled.

totplugin:
  location: http://totpluginjdbcservice:15111/plugin/jdbc/api/v1
  server:
    url: http://totserver:15000/tot/
    keep-alive-seconds: 60
  connection:
    vendor: POSTGRESQL
    driver: org.postgresql.Driver
    url: jdbc:postgresql://rdbservice:5432/anjana
    user: anjana
    password: anjana
    path-separator: "/"
    using-catalogs: false
    using-schemas: true
    sampleRows: 15
    obfuscationString: "*****"
  aris:
    - ari: "anja:totplugin:sample:/jdbc/Postgresql2/postgresql2/"
    - ari: "anja:totplugin:extract:/jdbc/Postgresql2/postgresql2/"

spring:
  application:
    name: tot-plugin-jdbc
  cloud:
    config:
      enabled: true
    loadbalancer:
      ribbon:
        enabled: false
    profiles: default2

```

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+rwx-o-rwx'})}
failed: [portuno1 -> localhost] (item={'url': 'https://artifactory.anjanadata.org:8443/reposi
tory.type == "external" and installation.mode != "local", 'mode': 'u+rwx,g+rwx,o-rwx'}) =>
launcher", "mode": "u+rwx,g+rwx,o-rwx", "url": "https://artifactory.anjanadata.org:8443/reposi
tory.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.exis

```

Fallo remove-anjana

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

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

```
TASK [anjana-utility : Disabling now [minerva] service in [minerva1] 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 -> 10.150.100.55] => (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 -> 10.150.100.138] (item=Removed [/opt/data] dir) => {"ansible_loop_var": "item", "changed": false, "item": ["/opt/data", {"ip": "10.150.100.138"}], "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'"}
ok: [director1 -> 10.150.100.66] (item=Removed [/opt/data] dir) => (item=Removed [/opt/data] dir)
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:
```