



MANUAL USO KIT ANSIBLE

Release Notes	2
Compatibilidad de versiones	2
Upgrade kit de Ansible a versión 4.a4	2
Introducción Ansible	3
¿Qué es un rol?	3
¿Qué es un inventario?	3
¿Qué es un playbook?	3
Instalación de Ansible	4
Cómo instalar Ansible	4
Software requerido	4
Conectividad requerida	5
Playbooks disponibles	6
Roles disponibles	6
Estructura de los roles	8
Uso del kit	11
Generación de inventario	11
Generación de playbook	11
Ajuste de variables	11
Ejecución del playbook	12
Log kit de Ansible	12
Funcionalidades extras	13

Descargar artefactos de Anjana	13
Actualizar Anjana	13
Backup de los datos de Anjana.	13
Sampledata	13
Data delete	14
Data reset	14
Exportación e importación de datos	14
Instalar/Actualizar Anjana desde dentro de la propia máquina	15
Balanceador/HA para MinIO	16
Actualización del kit	17
Paso 1 Respaldo de configuración	17
Paso 2 Ajuste de versión + Actualización	17
Paso 3 Ajustar ips y credenciales de inventarios y variables de roles	18
Paso 4 Renombrar inventario y playbook	19
Ejemplos de uso	20
Despliegue single node + RDS	20
Despliegue single node	25
Despliegue en HA + resiliencia	27
Generación de rol para plugins	32
Generación de nueva instancia de un plugin ya existente	37

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Dpto DS	Responsable DS	Creación del documento

Release Notes

- Se han optimizado los roles de common y common-microservices. A partir de ahora es necesario especificar en el archivo hosts.yml los nodos en los cuales se despliega para common y common-microservices. Más detalles en [Despliegue en HA + resiliencia](#)
- Se han parametrizado el host, user, password, etc de ldap en su archivo de variables correspondiente para que pueda ser modificado si es necesario
- Se han simplificado los tags necesarios para las funcionalidades de datos de ejemplo. Más información en [Funcionalidades extras](#)
- Se añade a la configuración de Apache la posibilidad de tener un Balanceador HA para MinIO [Balanceador/HA para MinIO](#)
- Se añade a la configuración de Apache un proxy para el acceso a Swagger
- Se ha cambiado el nombre del tag Overwrite por el de Insert
- Se ha mejorado la información que muestra el log del kit de Ansible. Más detalles en [Log kit de Ansible](#)

Compatibilidad de versiones

La presente versión del kit puede ser usada para desplegar versiones de Anjana Data:

- 4.4

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 y ajustar las versiones de los elementos a desplegar a la última versión de bugfix disponible.

Upgrade kit de Ansible a versión 4.a4

Está detallado en [Actualización del kit](#)

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.

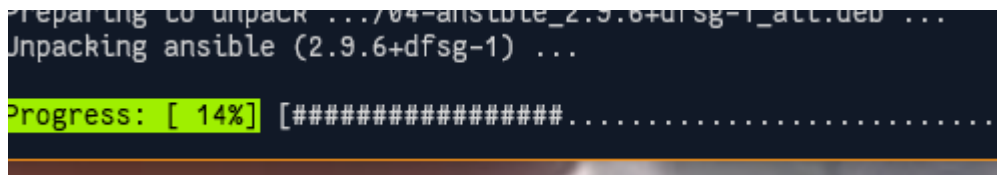
```
.
├── anjana.yml
├── ansible-inventories
│   └── sample
└── roles
    ├── apacheds
    ├── common
    ├── common-microservices
    ├── drittesta
    ├── edusa
    ├── hecate
    ├── hermes
    ├── kerno
    ├── minerva
    ├── minio
    ├── portal
    ├── portuno
    ├── postgresql
    ├── solr
    ├── tot
    ├── viator
    ├── zeus
    └── zookeeper
```

Instalación de Ansible

Cómo instalar Ansible

Para poder instalar ansible, tenemos que lanzar el siguiente comando en Ubuntu 20.04:

```
sudo apt update
sudo apt install ansible
ansible-galaxy collection install community.general
```



También instalará la Python3 como dependencia de Ansible.

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
 - POSTGRESQL-CLIENT >= 10
 - python-pip
 - python3
 - lsof
 - acl
 - sudo
 - jq
 - iproute2
 - ldap-utils
- Resto de nodos
 - python-pip

- python3
- lsof
- acl
- sudo
- jq
- iproute2
- ldap-utils
- Adicional en nodos frontales
 - apache2
- Adicional en nodos de persistencia RDBMS (si no se usa servicio externo)
 - POSTGRESQL >= 10

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 8443 contra el servidor artifactory.anjanadata.org, por tanto es necesario tener conectividad desde los servidores a provisionar con dicho servicio.

En modo local es necesario descargar un paquete zip 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.

Playbooks disponibles

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

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

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

- Fontales de Anjana
 - Portal Anjana
 - Portal Admin
 - Apache2

- 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

Tanto SOLR como Zookeeper podemos instalarlo como clusters o para un nodo en específico

- 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)

- Plugins

- Tot-plugin-aws-iam
- Tot-plugin-aws-s3
- Tot-plugin-aws-glue
- 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-oracle
- Tot-plugin-jdbc-denodo
- Tot-plugin-jdbc-redshift
- Tot-plugin-jdbc-sqlserver
- Tot-plugin-ldap
- Tot-plugin-powerbi

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.
 - Importar rol common
 - Se encarga de importar las variables
 - Actualiza el archivos hosts de la máquina
 - Instala los pre-requisitos necesarios para Anjana
 - Importar rol common-microservices
 - Instalar OpenJDK 8
 - Instalar OpenJDK 11
 - 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

- **Variables específicas** → Variables para cada microservicio necesarias para su funcionamiento, las cuales pueden ser encontradas en `ansible-inventories/<nombre_inventario>/group_vars/`
 - `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

```

○ All.yml:

- **versión.<microservicio>** → Aquí pondremos la versión que quiere ser desplegada en el servidor de los microservicios.

```

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.a4
  sampledata: 4.4-sample
  zeus: 4.4.0
  portal: 4.4.0
  admin: 4.4.0
  drittesta: 4.4.0

```

- **anjana.folder** → Permite especificar la carpeta donde se despliega el software de Anjana.

```

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

```

- **installation.startup** → Es systemd para la mayoría de los casos. Se puede cambiar a none para instalaciones en WSL o donde no esté disponible el systemd

```

41 installation:
42   # startup: none|systemd
43   startup: systemd

```

- **installation.type** → Para que descargue el binario launcher y el jar hay que poner external.

```
41 installation:
42   # startup: none|systemd
43   startup: systemd
44   # type: internal|external
45   type: external
```

- **installation.mode:** → Actualmente están contemplados 3 modos de despliegue dentro del kit

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

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

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

Uso 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.

Generación de inventario

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 sample y localhost serán sobrescritos por el kit al actualizar el propio kit.

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).

Es necesario copiar y renombrar el playbook anjana.yml, y posteriormente modificarlo para eliminar o comentar los roles que no queremos desplegar, típicamente se suele utilizar un servicio postgres provisto por tercero por ejemplo.

```
#En un single-node
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml

#En un nodo de Ansible donde el despliegue sea en otra u otras máquinas
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml
```

El playbook anjana.yml será sobrescrito por el kit al actualizar el propio kit.

Ajuste de variables

Es necesario ajustar las variables contenidas en el fichero global de variables del inventario (all.yml), adicionalmente se pueden ajustar resto de ficheros de variables, los cuales están segmentados por rol.

Ejecución del playbook

Para ello, necesitaremos una máquina Ubuntu con Ansible instalado en versión 2.9.6 (se explica más arriba como instalarlo y como hacer downgrade en caso de necesitarlo). En la máquina lanzaremos los siguientes comandos.

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

```
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

Log kit de Ansible

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
-----
```

Funcionalidades extras

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 inventories/sample/hosts.yml anjana.yml --tags download
```

En caso de que la intención sea descargar todos los artefactos (microservicios de anjana, servicios externos y kits de datos) habría que usar el siguiente comando:

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

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

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


Actualizar 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 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 inventories/sample/hosts.yml anjana.yml  
--tags update-edusa
```

NOTA: El cdn del front NO se actualiza automáticamente si está en un servicio externo como AWS S3.

Backup de los datos de Anjana.

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

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

Sampledata

Despliega un set de datos de ejemplo de uso de Anjana

```
## DEPLOY ANJANA WITH ALL SAMPLEDATA  
  
sudo ansible-playbook -i ansible-inventories/sample/hosts.yml anjana.yml -v  
--tags anjana-sample
```

```
## DEPLOY ALL SAMPLEDATA  
  
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

```
## 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 escogidos en el entorno y posteriormente carga todos los kit de datos disponibles para la versión y tipo de kit seleccionados en all.yml

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

```
--tags reset
```

Exportación e importación de datos

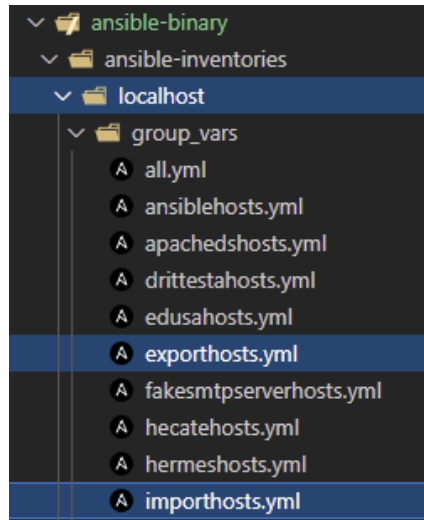
Es posible exportar e importar las persistencias para una transferencia entre máquinas/entornos con los roles de export e import respectivamente.

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 así como otros datos existentes como puertos, host, contraseñas, entre otros, para bases de datos y solr:

```
import:
  path: //{{anjana.folder}}/export-import
  utility: '{{installation.tmpdir}}/export-import-utility'
minio:
  datapath: //{{anjana.folder}}/data/minio
  path: //{{anjana.folder}}/minio
  imp_tmpdir: '{{installation.tmpdir}}/minio_import'
  import: '{{import.path}}/minio'
  artifact_url: /minio/mc
  ● host: http://localhost
  ● port: 9000
  ● user: anjana
  ● pass: anjanadata
bbdd:
  import: '{{import.path}}/bbdd'
  ● user: anjana
  ● pass: anjana
  ● host: localhost
  ● port: 5432
  ● version: 13
  ● database: anjana
solr:
  path: //{{anjana.folder}}/solr
  import: '{{import.path}}/solr'
  ● host: http://localhost
  ● port: 8983
  artifact_url: /solr/solr-utility/
  artifact: solr-utility.tgz
  ● collections:
    - snapshots
    - kerno
    - audit_logs
    - recommendations
    - business_tags

export:
  path: //{{anjana.folder}}/export-import
  utility: '{{installation.tmpdir}}/export-import-utility'
minio:
  datapath: //{{anjana.folder}}/data/minio
  path: //{{anjana.folder}}/minio
  exp_tmpdir: '{{installation.tmpdir}}/minio_export'
  export: '{{export.path}}/minio'
  artifact_url: /minio/mc
  ● host: http://localhost
  ● port: 9000
  ● user: anjana
  ● pass: anjanadata
bbdd:
  export: '{{export.path}}/bbdd'
  ● host: localhost
  ● port: 5432
  ● version: 13
  ● database: anjana
  ● user: anjana
  ● pass: anjana
solr:
  path: //{{anjana.folder}}/solr
  export: '{{export.path}}/solr'
  ● host: http://localhost
  ● port: 8983
  artifact_url: /solr/solr-utility/
  artifact: solr-utility.tgz
  ● collections:
    - snapshots
    - kerno
    - audit_logs
    - recommendations
    - business_tags
```



Estos ficheros pueden ser encontrados en ansible-inventories, bajo la carpeta de inventario que se esté utilizando para el despliegue, por ejemplo:

ansible-inventories/localhost/group_vars/exporthosts.yml

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
exporthosts:
  hosts:
    export1:
      # Nodo en el cual se instalan y ejecutan los clientes de exportación (minio-mc, postgresql-client, solr-utility, etc)
      # La exportación es almacenada en este nodo
      ansible_host: <server_ip>
      ansible_port: 22
      service_hostname: localhost
importhosts:
  hosts:
    import1:
      # Nodo desde el cual se ejecutan los clientes de importación (se instalan minio-mc, postgresql-client, solr-utility, etc)
      # 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
```

Instalar/Actualizar Anjana 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: | <artifactory_user>
  password: <artifactory_user>
  internal:
```

Balancedor/HA para MinIO

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

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

- `port.minioProxyPass` para indicar el puerto que tendrá el balanceador que luego habrá que tenerlo en cuenta para la configuración por ejemplo de Kerno, Hermes 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

<pre> 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 </pre>	<pre> backup: {{{anjana.folder}}}/backup/portal port: http: 80 https: 443 minioProxyPass: 9100 balancer: swagger: BalancerMember http://viatorserver_node1:8085/swagger timeout=5 swaggerapi: BalancerMember http://viatorserver_node1:8085/v3/api-docs timeout=5 gateway: BalancerMember http://viatorserver_node1:8085 timeout=5 portuno: BalancerMember http://portunoserver:8998/portuno timeout=5 cdn: BalancerMember http://s3service_node1:9000/cdn timeout=5 dsa: BalancerMember http://s3service_node1:9000/dsa timeout=5 imports: BalancerMember http://s3service_node1:9000/imports timeout=5 minioProxyPass: BalancerMember http://s3service_node1:9000 timeout=5 </pre>
--	--

```
databaseSchema: 'h2mms'  
minio:  
  url: http://<front_node>:9100  
  accessKey: anjana  
  secretKey: anjanadata  
# Mail connection properties
```

Actualización del kit

Paso 1 Respaldo de configuración

Lo primero que habría que comprobar sería la existencia del siguiente inventario y playbook en la instalación existente (los nombres son orientativos).

```
/opt# ll ansible/*
-rw-r--r--  1 anjana anjana 2774 Oct  7 06:53 ansible/README.md
-rwx-----  1 anjana anjana  147 Oct 10 12:54 ansible/actualizacion-sinbackup.sh*
-rwx-----  1 anjana anjana  123 Oct 10 12:54 ansible/actualizacion.sh*
-rw-r--r--  1 anjana anjana 9215 Oct  7 06:53 ansible/anjana.yml
-rw-r--r--  1 anjana anjana 9684 Oct 10 12:53 ansible/anjana_xxx.yml
-rw-r--r--  1 anjana anjana  172 Oct  7 06:53 ansible/k8s.yml

ansible/ansible-inventories:
total 24
drwxr-xr-x  6 anjana anjana 4096 Oct 10 12:52 ./
drwx-----  4 anjana anjana 4096 Oct 10 12:54 ../
drwxr-xr-x  3 anjana anjana 4096 Oct 10 12:52 anjana_xxx/
drwxr-xr-x  3 anjana anjana 4096 Oct  7 06:53 k8s/
drwxr-xr-x  3 anjana anjana 4096 Oct  7 06:53 localhost/
drwxr-xr-x  3 anjana anjana 4096 Oct  7 06:53 sample/
```

En caso de no existir es necesario duplicar y renombrar el inventario sample o localhost y el playbook anjana.yml.

```
#En un single-node
cp -r /opt/ansible/ansible-inventories/localhost /opt/ansible/ansible-inventories/anjana_pro_localhost
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml

#En un nodo de Ansible con despliegue distribuido
cp -r /opt/ansible/ansible-inventories/sample /opt/ansible/ansible-inventories/anjana_pro
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml
```

Los inventarios sample y localhost serán sobrescritos por el kit al actualizar el propio kit.

Paso 2 Ajuste de versión + Actualización

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, anjana_xxx o anjana_xxx_localhost.

```
/opt/ansible/ansible-inventories# ll anjana_xxx/group_vars/  
total 224  
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 drittestahosts.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: 4.3.1  
  config_local: 4.3.1  
  hecate: 4.3.1  
  hermes: 4.3.2  
  kerno: 4.3.2  
  minerva: 4.3.1  
  portuno: 4.3.1  
  tot: 4.3.2  
  viator: 4.3.1  
  ansible: 4.a4  
  sampledata: 4.3.1-sample  
  zow: 4.3.1
```

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).

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

En cuanto se haya completado ya dispondremos del kit en la versión seleccionada.

Paso 3 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.

```

all:
  children:
    commonhosts:
      hosts:
        common1:
          # Hay que duplicar este host (common1, common2, etc) tantas veces como máquinas haya en el despliegue
          # Cuentan las máquinas de Anjana, persistencias, ansible, etc
          ansible_host: <server_ip>
          ansible_port: 22
          service_hostname: localhost
        commonmicroserviceshosts:
          hosts:
            commonmicroservices1:
              # Hay que duplicar este host (commonmicroservices1, commonmicroservices2, etc ) tantas veces como máquinas usen Java 8
              # Cuentan las máquinas de backend de Anjana, export, import, ansible, Solr, Zookeeper, export, etc
              ansible_host: <server_ip>
              ansible_port: 22
              service_hostname: localhost
    edusahosts:
      hosts:
        edusa1:
  
```

```

total 200K
ls -lh /opt/ansible/ansible-inventories/sample/group_vars/
-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 apachedshosts.yml
-rw-r--r-- 1 anjana anjana 756 Oct 10 13:09 dritttestahosts.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 fakesmtpserverhosts.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 kernohosts.yml
-rw-r--r-- 1 anjana anjana 694 Oct 10 13:09 minervahosts.yml
-rw-r--r-- 1 anjana anjana 707 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 tothosts.yml
-rw-r--r-- 1 anjana anjana 885 Oct 10 13:09 totpluginaws gluehosts.yml
-rw-r--r-- 1 anjana anjana 871 Oct 10 13:09 totpluginaws iamhosts.yml
-rw-r--r-- 1 anjana anjana 857 Oct 10 13:09 totpluginaws s3hosts.yml
-rw-r--r-- 1 anjana anjana 885 Oct 10 13:09 totpluginazure adhosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluginazure fileshosts.yml
-rw-r--r-- 1 anjana anjana 955 Oct 10 13:09 totpluginazure storagehosts.yml
-rw-r--r-- 1 anjana anjana 941 Oct 10 13:09 totpluginazcpbigqueryhosts.yml
-rw-r--r-- 1 anjana anjana 871 Oct 10 13:09 totpluginazcp iamhosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluginazcp storagehosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluginhdfs hosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluginhive hosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluginjdbc coraclehosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluginjdbc chosts.yml
-rw-r--r-- 1 anjana anjana 927 Oct 10 13:09 totpluginjdbc coraclehosts.yml
-rw-r--r-- 1 anjana anjana 955 Oct 10 13:09 totpluginjdbc credshift hosts.yml
-rw-r--r-- 1 anjana anjana 969 Oct 10 13:09 totpluginjdbc sqlserver hosts.yml
-rw-r--r-- 1 anjana anjana 834 Oct 10 13:09 totpluginldap hosts.yml
-rw-r--r-- 1 anjana anjana 876 Oct 10 13:09 totpluginpowerbi hosts.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
  
```

Paso 4 Renombrar inventario y playbook

Una vez ajustado sample y anjana.yaml con los datos/credenciales del entorno lo renombramos o copiamos para identificar a dónde apunta. Como tenemos ya un inventario y playbook del anterior kit debemos antes hacerle un backup por si hemos olvidado alguna credencial o dato necesario.

#Backup

```

mv /opt/ansible/ansible-inventories/anjana_pro /opt/ansible/ansible-inventories/anjana_pro_old
mv /opt/ansible/anjana_pro.yml /opt/ansible/anjana_pro.old
  
```

#En un single-node

```

cp -r /opt/ansible/ansible-inventories/localhost
  
```



```
/opt/ansible/ansible-inventories/anjana_pro_localhost  
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml
```

```
#En un nodo de Ansible con despliegue distribuido  
cp -r /opt/ansible/ansible-inventories/sample /opt/ansible/ansible-inventories/anjana_pro  
cp /opt/ansible/anjana.yml /opt/ansible/anjana_pro.yml
```

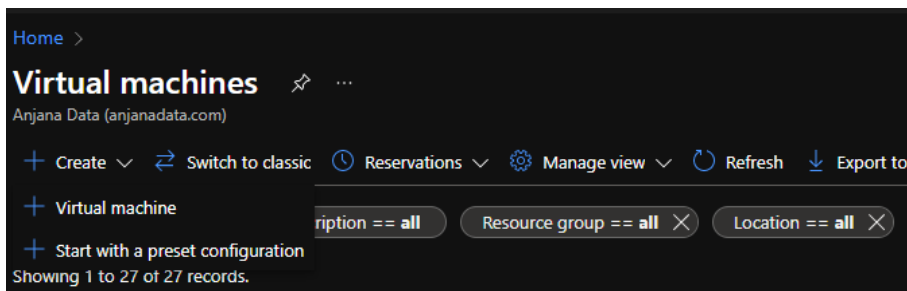
El no revisar y portar los cambios de los archivos actualizados a los inventarios personalizados puede causar que el kit no funcione correctamente.

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 AWS.

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.

```

EXPLORER
ANSIBLE-IN...  singlenode_&_rds > hosts.yml
sample
  group_vars
    hosts.yml
singlenode_&_rds
  group_vars
    hosts.yml
.gitignore

162 apacheexporterhosts:
163   hosts:
164     apacheexporter1:
165       ansible_host: 10.200.20.4
166       ansible_port: 22
167       service_hostname: localhost
168   monithosts:
169     hosts:
170       monit1:
171         ansible_host: 10.200.20.4
172         ansible_port: 22
173         service_hostname: localhost
174   vars:
175     ansible_connection: smart
176     ansible_user: azureuser
177     ansible_ssh_private_key_file: /root/.ssh/dev.pem
178     ansible_become: true
179     force_service_mgr: systemd # systemd or initd
180

```

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

```

ANSIBLE-IN...  singlenode_&_rds > group_vars > all.yml
sample
  group_vars
    hosts.yml
singlenode_&_rds
  group_vars
    all.yml
    apachedshosts.yml
    apacheexporterh...
    azuredeployhosts...
    dritttestahosts.yml
    edusahosts.yml
    grafana-simpleho...
    grafanahosts.yml
    hecatehosts.yml

1 version:
2   edusa: 4.2.7
3   config_local: 4.2.10
4   hecate: 4.2.7
5   hermes: 4.2.8
6   kerno: 4.2.10
7   minerva: 4.2.7
8   portuno: 4.2.10
9   tot: 4.2.7
10  viator: 4.2.10
11  zeus: 4.2.10
12  portal: 4.2.10
13  admin: 4.2.10
14  dritttesta: 4.2.7
15  #NOTA: RECORDAD AJUSTAR LA RAMA DE CON
16

```

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

```

ANSIBLE-IN...  singlenode_&_rds > group_vars > all.yml
sample
  group_vars
    hosts.yml
singlenode_&_rds
  group_vars
    all.yml
    apachedshosts.yml
    apacheexporterh...
    azuredeployhosts...
    dritttestahosts.yml
    edusahosts.yml
    grafana-simpleho...
    grafanahosts.yml

27 artifactory:
28   user: <artifactory_user>
29   password: <artifactory_password>
30   internal:
31     mavenurl: internal
32     mvnrepo: internal
33     npmurl: internal
34     rawurl: internal
35   external:
36     mavenurl: https://artifactory.anjanadata.org:8443/repository/anjanareleases
37     mvnrepo: anjanareleases
38     npmurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesnpm
39     rawurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesraw
40
41

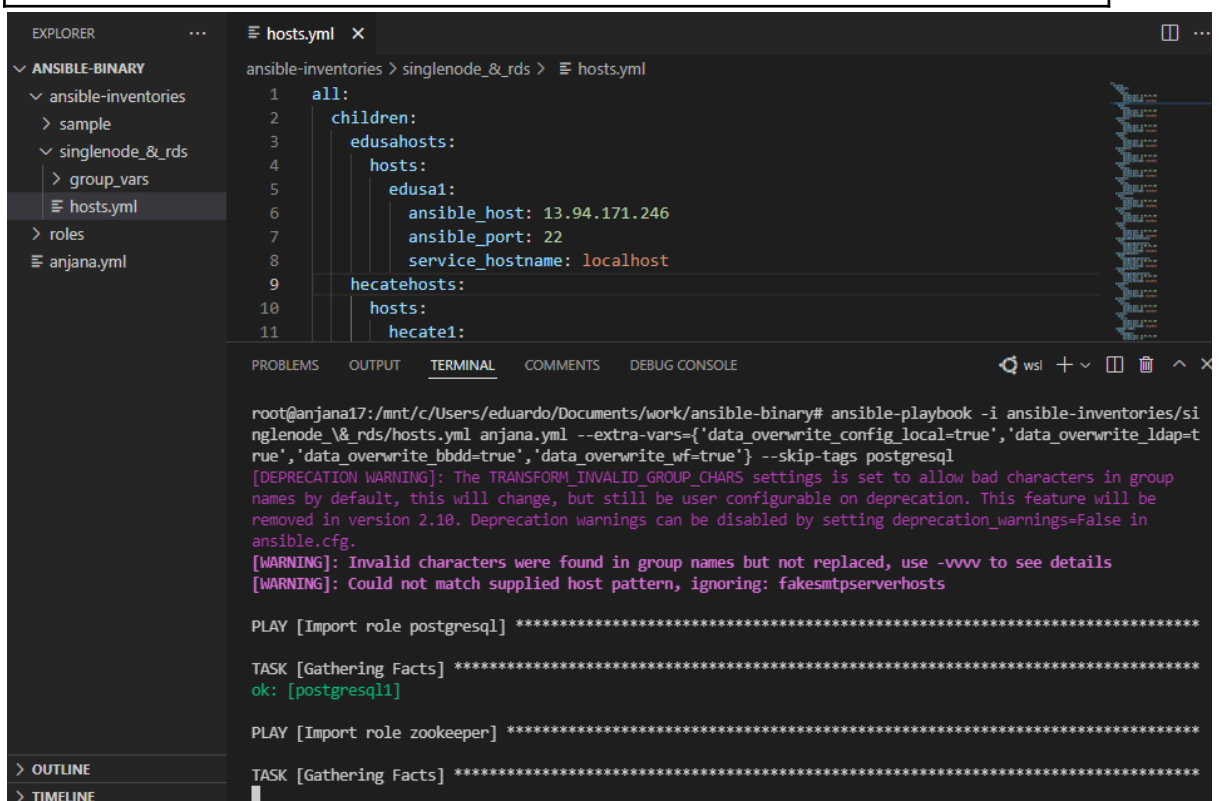
```

6. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos (puerto 8443).

7. 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 ejemplos PERO vamos a añadir `--skip-tags postgresql` para que no lo instale ya que vamos a usar RDS:

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



The screenshot shows a code editor with the following content in the `hosts.yml` file:

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

The terminal output shows the execution of the Ansible playbook with the following commands and warnings:

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

PLAY [Import role postgresql] *****

TASK [Gathering Facts] *****
ok: [postgresql1]

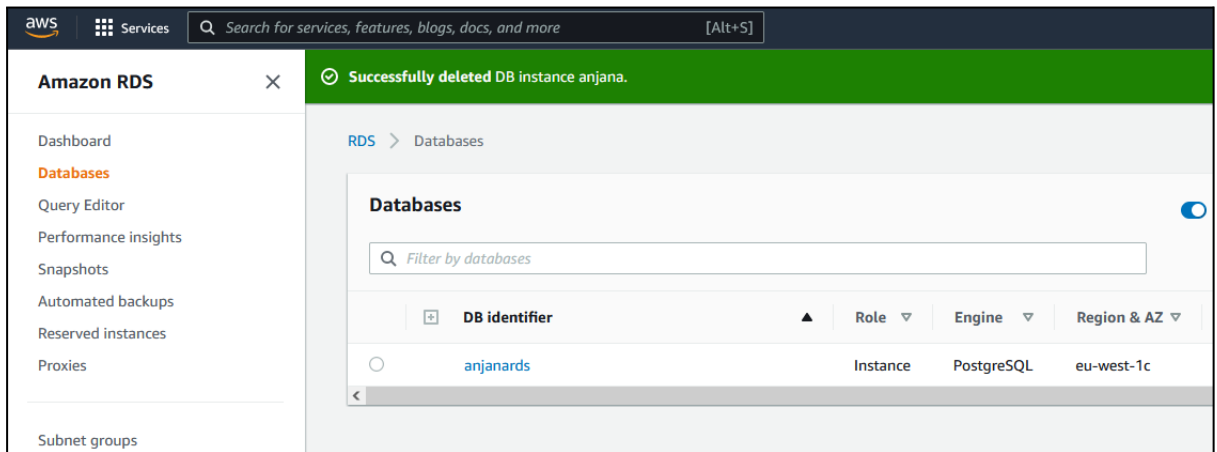
PLAY [Import role zookeeper] *****

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

8. 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
fakesmtpserver1 : ok=10  changed=6   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
hecate1       : ok=26  changed=11  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
hermes1       : ok=26  changed=9   unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
kerno1        : ok=26  changed=9   unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
minerva1      : ok=26  changed=9   unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
minio1        : ok=26  changed=17  unreachable=0    failed=0    skipped=5    rescued=0    ignored=1
monit1        : ok=7   changed=5   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
portal1       : ok=34  changed=23  unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
portuno1      : ok=26  changed=9   unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
postgresql1   : ok=1   changed=0   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
solr1         : ok=26  changed=11  unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
viator1       : ok=24  changed=9   unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
zeus1         : ok=26  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
zookeeper1    : ok=27  changed=11  unreachable=0    failed=0    skipped=6    rescued=0    ignored=0
```

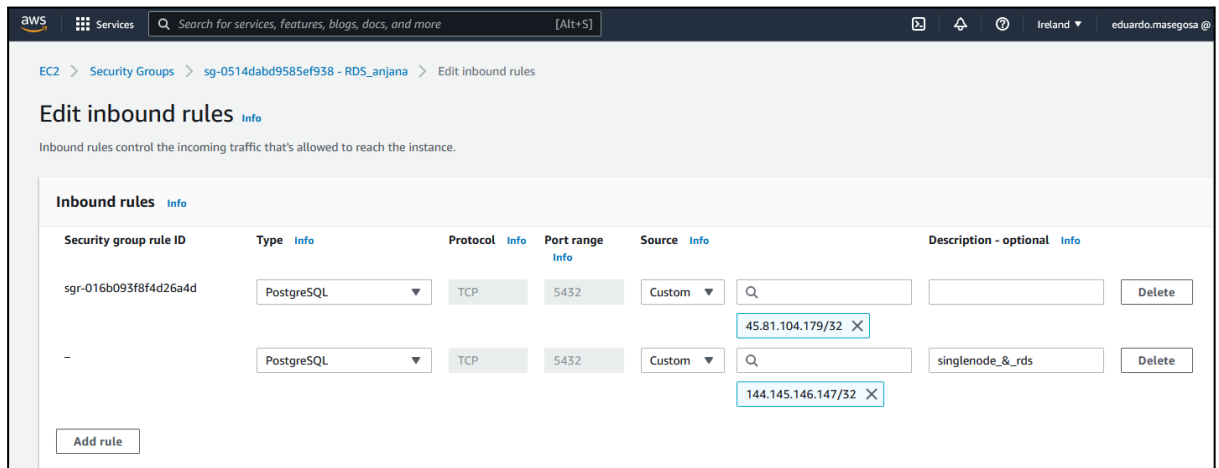
9. Creamos la base de datos en RDS



10. Creamos los esquemas de Anjana

```
CREATE SCHEMA anjana AUTHORIZATION anjanouser;
CREATE SCHEMA hermes AUTHORIZATION anjanouser;
CREATE SCHEMA minerva AUTHORIZATION anjanouser;
CREATE SCHEMA portuno AUTHORIZATION anjanouser;
CREATE SCHEMA zeus AUTHORIZATION anjanouser;
```

11. Habilitamos el acceso desde la máquina desplegada



- Sustituimos la ruta y credenciales de la conexión a la base de datos en la configuración de los microservicios `/opt/data/configrepo`.

```

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

- Una vez que los microservicios conecten crearán en los esquemas anteriormente creados las tablas de Anjana

```

anjanardsdatabase=> SELECT *
FROM pg_catalog.pg_tables where tableowner like 'anjanouser';

```

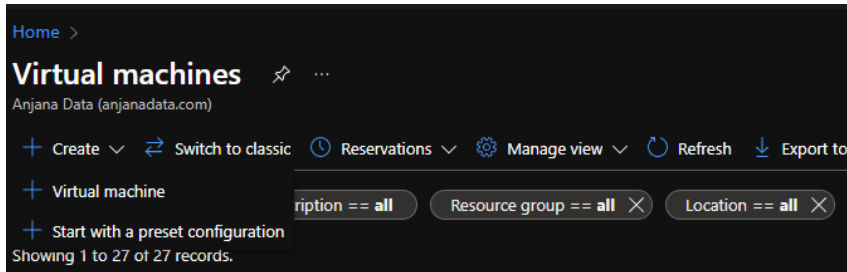
schemaname	tablename	tableowner	tablespace	hasindexes	hasrules	hastriggers	rowsecurity
hermes	task_step	anjanouser		t	f	t	f
hermes	workflow_ou	anjanouser		t	f	t	f
hermes	workflow_execution	anjanouser		t	f	t	f
hermes	act_ru_deadletter_job	anjanouser		t	f	t	f
hermes	workflow_task	anjanouser		t	f	t	f
hermes	workflowtask_groupcandidates	anjanouser		f	f	t	f
hermes	act_ru_suspended_job	anjanouser		t	f	t	f
hermes	act_ru_timer_job	anjanouser		t	f	t	f
hermes	act_ru_task	anjanouser		t	f	t	f
hermes	act_ru_job	anjanouser		t	f	t	f
minerva	solr_filter_conf	anjanouser		t	f	f	f
hermes	act_ge_property	anjanouser		t	f	f	f
zeus	parameters	anjanouser		t	f	f	f
zeus	languages	anjanouser		t	f	t	f
zeus	user_language	anjanouser		t	f	t	f
zeus	user_ou_role	anjanouser		t	f	t	f
zeus	organizational_unit	anjanouser		t	f	t	f
zeus	role	anjanouser		t	f	t	f
zeus	permission	anjanouser		t	f	t	f
zeus	user_config	anjanouser		t	f	t	f
zeus	users	anjanouser		t	f	t	f
hermes	act_re_model	anjanouser		t	f	t	f
hermes	notification	anjanouser		t	f	f	f
hermes	sent_notification	anjanouser		t	f	f	f
hermes	act_re_deployment	anjanouser		t	f	t	f
hermes	act_evt_log	anjanouser		t	f	f	f
hermes	act_re_procdef	anjanouser		t	f	t	f
hermes	act_ru_event_subscr	anjanouser		t	f	t	f
hermes	act_ge_bytearray	anjanouser		t	f	t	f
hermes	act_ru_identitylink	anjanouser		t	f	t	f
hermes	act_ru_integration	anjanouser		t	f	t	f
hermes	act_ru_variable	anjanouser		t	f	t	f
hermes	act_hi_actinst	anjanouser		t	f	f	f
hermes	act_procdef_info	anjanouser		t	f	t	f
hermes	act_ru_execution	anjanouser		t	f	t	f
hermes	act_hi_procinstant	anjanouser		t	f	f	f
hermes	act_hi_taskinst	anjanouser		t	f	f	f
hermes	act_hi_varinst	anjanouser		t	f	f	f
hermes	act_hi_detail	anjanouser		t	f	f	f
hermes	act_hi_comment	anjanouser		t	f	f	f
hermes	act_hi_attachment	anjanouser		t	f	f	f
hermes	act_hi_identitylink	anjanouser		t	f	f	f
anjana	adherence_cart	anjanouser		t	f	f	f
anjana	attribute_relationships	anjanouser		t	f	t	f
anjana	edge	anjanouser		t	f	f	f
anjana	object_subtype	anjanouser		t	f	t	f
anjana	entity	anjanouser		t	f	t	f
anjana	imported_metadata	anjanouser		t	f	f	f
anjana	menu	anjanouser		t	f	t	f
anjana	node	anjanouser		t	f	f	f
anjana	sections	anjanouser		t	f	t	f
anjana	relationship	anjanouser		t	f	f	f
anjana	pending_adherence_event	anjanouser		t	f	t	f
anjana	edition_configuration	anjanouser		t	f	t	f
anjana	workflow_info	anjanouser		t	f	f	f
anjana	dsa_ds_combination	anjanouser		t	f	t	f
anjana	attribute_definition_value	anjanouser		t	f	t	f
anjana	combination	anjanouser		t	f	t	f
anjana	attribute_value	anjanouser		t	f	t	f
anjana	template_attribute	anjanouser		t	f	t	f
anjana	attribute_definition	anjanouser		t	f	t	f
anjana	template_attribute_validation	anjanouser		t	f	t	f

(62 rows)

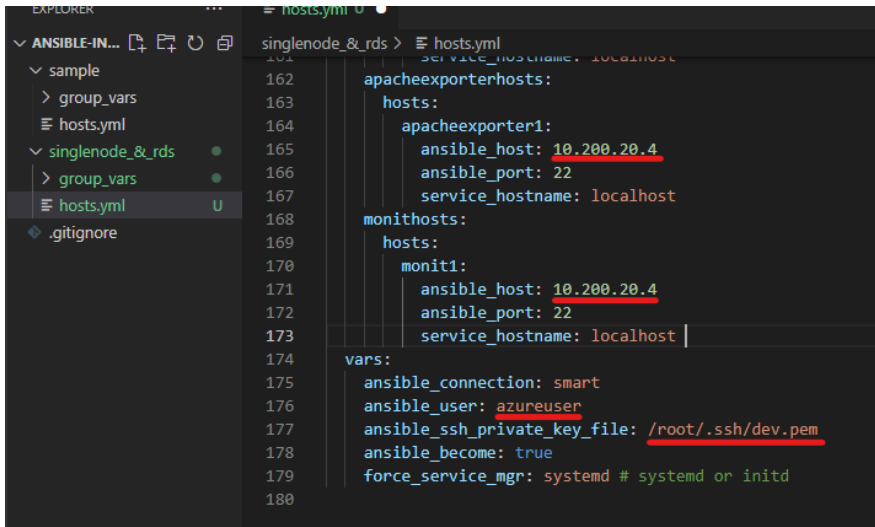
En este punto el despliegue por parte de sistemas está finalizado, solo falta la carga de datos de configuración de la aplicación en base de datos y ajuste del árbol LDAP.

Despliegue single node

1. Creamos una máquina virtual



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



```

162 apacheexporterhosts:
163   hosts:
164     apacheexporter1:
165       ansible_host: 10.200.20.4
166       ansible_port: 22
167       service_hostname: localhost
168 monithosts:
169   hosts:
170     monit1:
171       ansible_host: 10.200.20.4
172       ansible_port: 22
173       service_hostname: localhost
174 vars:
175   ansible_connection: smart
176   ansible_user: azureuser
177   ansible_ssh_private_key_file: /root/.ssh/dev.pem
178   ansible_become: true
179   force_service_mgr: systemd # systemd or initd
180
  
```

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

```

ANSIBLE-IN...  singlenode_&rds > group_vars > all.yml
├── sample
│   └── group_vars
│       ├── hosts.yml
│       └── singlenode_&rds
│           └── group_vars
│               ├── all.yml
│               ├── apachedshosts.yml
│               ├── apacheexporterh...
│               ├── azuredeployhosts...
│               ├── drittestahosts.yml
│               ├── edusahosts.yml
│               ├── grafana-simpleho...
│               ├── grafanahosts.yml
│               └── hecatehosts.yml
└── version:
    1
    2 edusa: 4.2.7
    3 config_local: 4.2.10
    4 hecate: 4.2.7
    5 hermes: 4.2.8
    6 kerno: 4.2.10
    7 minerva: 4.2.7
    8 portuno: 4.2.10
    9 tot: 4.2.7
   10 viator: 4.2.10
   11 zeus: 4.2.10
   12 portal: 4.2.10
   13 admin: 4.2.10
   14 drittesta: 4.2.7
   15 #NOTA: RECORDAD AJUSTAR LA RAMA DE CON
   16

```

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

```

ANSIBLE-IN...  singlenode_&rds > group_vars > all.yml
├── sample
│   └── group_vars
│       ├── hosts.yml
│       └── singlenode_&rds
│           └── group_vars
│               ├── all.yml
│               ├── apachedshosts.yml
│               ├── apacheexporterh...
│               ├── azuredeployhosts...
│               ├── drittestahosts.yml
│               ├── edusahosts.yml
│               ├── grafana-simpleho...
│               ├── grafanahosts.yml
│               └── hecatehosts.yml
└── artifactory:
    28 user: <artifactory_user>
    29 password: <artifactory_password>
    30 internal:
    31   mavenurl: internal
    32   mvnrepo: internal
    33   npmurl: internal
    34   rawurl: internal
    35 external:
    36   mavenurl: https://artifactory.anjanadata.org:8443/repository/anjanareleases
    37   mvnrepo: anjanareleases
    38   npmurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesnpm
    39   rawurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesraw
    40
    41

```

6. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos (puerto 8443).
7. 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 ejemplos

```

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

```



```

EXPLORER  ...  hosts.yml  X
ANSIBLE-BINARY
├── ansible-inventories
│   └── sample
│       └── singlenode_&_rds
│           ├── group_vars
│           └── hosts.yml
└── roles
    └── anjana.yml
hosts.yml
1  all:
2  |   children:
3  |     edusahosts:
4  |       hosts:
5  |         edusa1:
6  |           ansible_host: 13.94.171.246
7  |           ansible_port: 22
8  |           service_hostname: localhost
9  |   hecatehosts:
10 |   hosts:
11 |     hecate1:
PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE
root@anjana17: /mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/singlenode_&_rds/hosts.yml anjana.yml --extra-vars={'data_overwrite_config_local=true','data_overwrite_ldap=true','data_overwrite_bbdd=true','data_overwrite_wf=true'} --skip-tags postgresql
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group names by default, this will change, but still be user configurable on deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtpserverhosts

PLAY [Import role postgresql] *****

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

PLAY [Import role zookeeper] *****

TASK [Gathering Facts] *****

```

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
edusa1        : ok=33  changed=18  unreachable=0  failed=0  skipped=10  rescued=0  ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
hecate1       : ok=26  changed=11  unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
hermes1       : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
kerno1        : ok=30  changed=11  unreachable=0  failed=0  skipped=4  rescued=0  ignored=0
minerval      : ok=26  changed=9   unreachable=0  failed=0  skipped=8  rescued=0  ignored=0
minio1        : ok=26  changed=17  unreachable=0  failed=0  skipped=5  rescued=0  ignored=1
monit1        : ok=7   changed=5   unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
portall       : 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

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

- 2 vm de front
- 2 vm de back + apacheds (solo en el primer nodo)
- 3 vm de solr+zookeeper
- 1 vm de minio

- 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:

```
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
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.

```

hermeshosts.yml U 158
kernohosts.yml U 159
minervahosts.yml U 160
miniohosts.yml U 161
monithosts.yml U 163
portalhosts.yml U 164
portunohosts.yml U 165
postgresqlhosts.yml U 166
solr_clusterhosts.yml U 168
solrhosts.yml U 169
tothosts.yml U 170
viatorhosts.yml U 171
zeushosts.yml U 172
zookeeper_clusterhosts.yml U 173
zookeeper-nodes.yml U 174
hosts.yml U 176
> sample 177
> circe 178
    service_hostname: localhost
apachedshosts:
hosts:
  apacheds1:
    ansible_host: 10.0.0.6
    ansible_port: 22
    service_hostname: localhost
postgresqlhosts:
hosts:
  postgresql1:
    ansible_host: <IP_servidor>
    ansible_port: 22
    service_hostname: localhost
    ansible_python_interpreter: python3
vars:
  ansible_connection: smart
  ansible_user: azureuser
  ansible_ssh_private_key_file: /root/.ssh/dev.pem
  ansible_become: yes
  force_service_mgr: systemd # systemd or initd

```

4. Solr y Zookeeper en modo cluster requieren de los siguientes ajustes:
 - a. 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 ---
2 #host: not used
3 #ip: used in solr.service in Environment=ZK_HOST
4 zookeeper_hosts:
5   - host: "zknode1"
6     ip: 10.0.0.12 # localhost on singlenode, ip/hostname of zk_node1 on cluster
7   - host: "zknode2"
8     ip: 10.0.0.13 # localhost on singlenode, ip/hostname of zk_node1 on cluster
9   - host: "zknode2"
10    ip: 10.0.0.14 # localhost on singlenode, ip/hostname of zk_node1 on cluster
11
12 installation:
13   startup: system
  
```

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

```

142     ansible_host: 20.123.247.161
143     ansible_port: 22
144     service_hostname: localhost
145
146   zookeeper-nodes:
147     hosts:
148       zookeeper1:
149         ansible_host: 20.123.247.115
150         nodeip: 10.0.0.12 # localhost on singlenode, ip/hostname on cluster; used in ../conf/zoo.cfg
151         myid: 1 # used in ../data/zookeeper/myid and ../conf/zoo.cfg
152         ansible_port: 22
153         service_hostname: localhost
154       zookeeper2:
155         ansible_host: 20.123.247.162
156         nodeip: 10.0.0.13 # localhost on singlenode, ip/hostname on cluster; used in ../conf/zoo.cfg
157         myid: 2 # used in ../data/zookeeper/myid and ../conf/zoo.cfg
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. Si Minio está en un nodo separado de Edusa hay que poner la propiedad symboliclink a “false”

```

17     # UNCOMMENT IF EDUSA USES con
18     path: /opt/data/configrepo
19     # If true creates configrepo
20     symboliclink: false
21     symboliclink_path: /opt/data/
  
```

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

```

1  version:
2  edusa: 4.2.7
3  config_local: 4.2.10
4  hecate: 4.2.7
5  hermes: 4.2.8
6  kerno: 4.2.10
7  minerva: 4.2.7
8  portuno: 4.2.10
9  tot: 4.2.7
10 viator: 4.2.10
11 zeus: 4.2.10
12 portal: 4.2.10
13 admin: 4.2.10
14 drittesta: 4.2.7
15 #NOTA: RECORDAD AJUSTAR LA RAMA DE CON
16

```

7. Editar group_vars/all.yml con el usuario y contraseña proporcionado por Anjana para el acceso al repositorio de artefactos

```

27
28 artifactory:
29   user: <artifactory_user>
30   password: <artifactory_password>
31   internal:
32     mavenurl: internal
33     mvnrepo: internal
34     npmurl: internal
35     rawurl: internal
36   external:
37     mavenurl: https://artifactory.anjanadata.org:8443/repository/anjanareleases
38     mvnrepo: anjanareleases
39     npmurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesnpm
40     rawurl: https://artifactory.anjanadata.org:8443/repository/anjanareleasesraw
41

```

8. Solicitar acceso desde la ip pública del servidor que se está desplegando en el firewall del repositorio de artefactos (puerto 8443).
9. Una vez los siguientes puntos preparados ya podemos lanzar el despliegue:
 - a. acceso por ssh a la máquina
 - b. seteadas las variables con los datos correctos
 - c. acceso al repositorio de artefactos desde la máquina

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

```
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 -vvvv to see details
viator1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hecate2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hecate1 | SUCCESS => {
  "ansible_facts": {

```

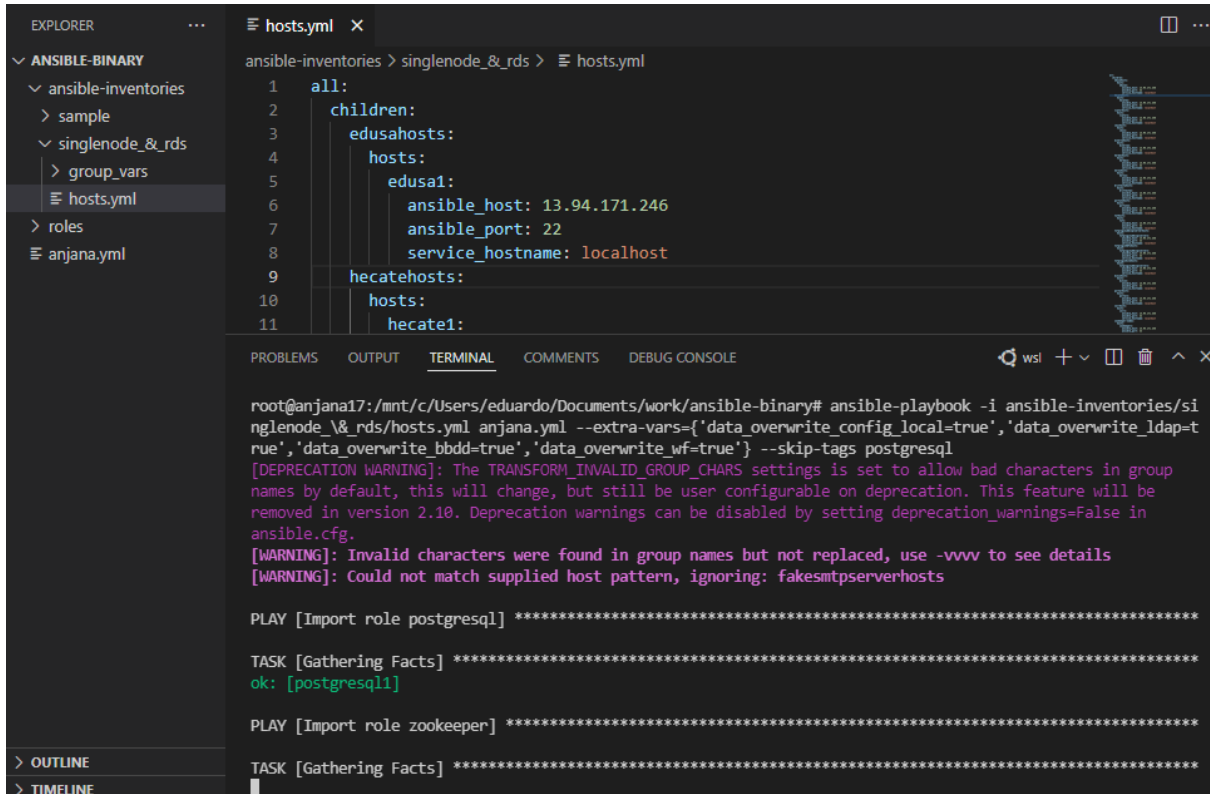
10. Lanzamos el despliegue

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

```



```

EXPLORER  ...  hosts.yml X
ANSIBLE-BINARY
├── ansible-inventories
│   └── sample
│       ├── singlenode_&_rds
│       │   ├── group_vars
│       │   └── hosts.yml
│       └── roles
└── anjana.yml
ansible-inventories > singlenode_&_rds > hosts.yml
1  all:
2  |   children:
3  |     edusahosts:
4  |       hosts:
5  |         edusa1:
6  |           ansible_host: 13.94.171.246
7  |           ansible_port: 22
8  |           service_hostname: localhost
9  |   hecatehosts:
10 |     hosts:
11 |       hecate1:
PROBLEMS  OUTPUT  TERMINAL  COMMENTS  DEBUG CONSOLE
root@anjana17:/mnt/c/Users/eduardo/Documents/work/ansible-binary# ansible-playbook -i ansible-inventories/si
nglenode_&_rds/hosts.yml anjana.yml --extra-vars={'data_overwrite_config_local=true','data_overwrite_ldap=t
rue','data_overwrite_bbdd=true','data_overwrite_wf=true'} --skip-tags postgresql
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to allow bad characters in group
names by default, this will change, but still be user configurable on deprecation. This feature will be
removed in version 2.10. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Could not match supplied host pattern, ignoring: fakesmtpserverhosts

PLAY [Import role postgresql] *****

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

PLAY [Import role zookeeper] *****

TASK [Gathering Facts] *****

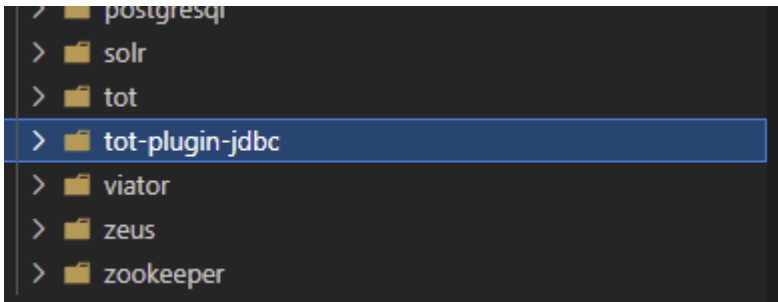
```

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

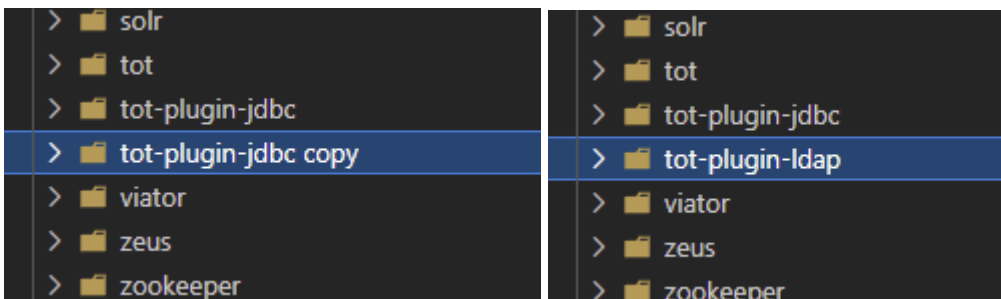
```
PLAY RECAP *****
*****
apacheds1      : ok=23  changed=10  unreachable=0    failed=0    skipped=10   rescued=0    ignored=0
drittista1    : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
drittista2    : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
edusa1        : ok=31  changed=14  unreachable=0    failed=0    skipped=13   rescued=0    ignored=0
edusa2        : ok=31  changed=14  unreachable=0    failed=0    skipped=13   rescued=0    ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
hecate1       : ok=27  changed=12  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
edusa2        : ok=31  changed=14  unreachable=0    failed=0    skipped=13   rescued=0    ignored=0
fakesmtpserver1 : ok=10  changed=7   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
hecate1       : ok=27  changed=12  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
hecate2       : ok=27  changed=12  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
hermes1       : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
hermes2       : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
kerno1        : ok=31  changed=12  unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
kerno2        : ok=31  changed=12  unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
minerva1      : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
minerva2      : ok=27  changed=10  unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
minio1        : ok=25  changed=14  unreachable=0    failed=0    skipped=7    rescued=0    ignored=1
monit1        : ok=7   changed=5   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
portall        : 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 rol para plugins

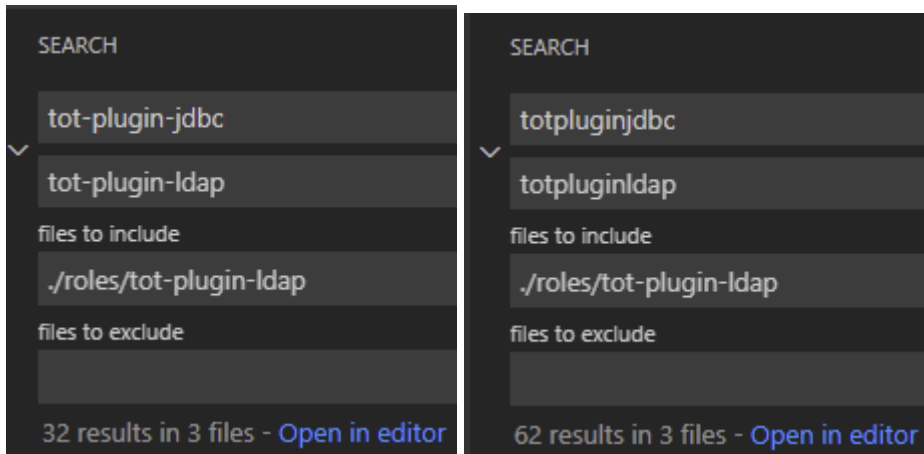
En el kit provisto está contenido un rol de ejemplo para un plugin de tot como se puede ver a continuación:



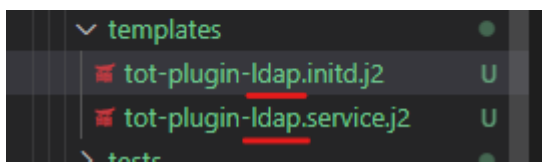
1. Para la generación de un nuevo rol de plugin simplemente duplicamos el directorio y renombramos para hacerlo coincidir con la implementación del nuevo plugin:



2. A continuación habrá que renombrar todas las variables y valores que apunten al nombre del plugin como se puede ver la siguiente imagen:



3. Renombrar los ficheros de la carpeta templates.



4. Duplicamos el fichero de variables para el nuevo plugin en "groups_vars/totpluginXXXhosts.yaml". Recordando ajustar las variables o valores apuntando al nuevo plugin

Las variables a ajustar, dependiendo del escenario, son las siguientes:

- Si cambiamos a otro tipo de plugin (Tot-plugin-jdbc a Tot-plugin-LDAP) debemos de cambiar todas las variables seleccionadas en la imagen.

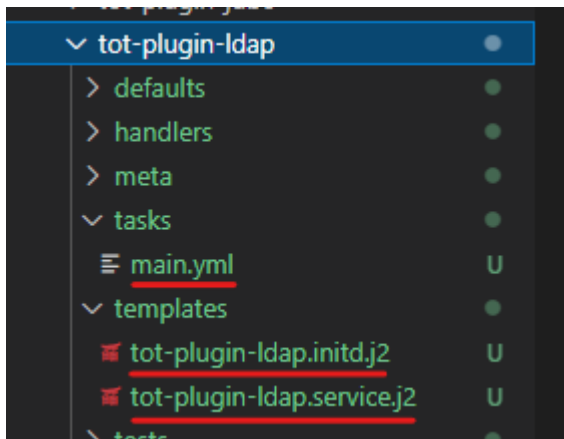
```

totpluginjdbc.yml M X
ansible-binary > ansible-inventories > sample > group_vars > totpluginjdbc.yml > {} totpluginjdbc > {} install > jar
You, 3 minutes ago | 2 authors (Antonio José Gómez Díaz and others) | Ansible Vars Schema (ansible-varsjson)
1 ---
2 totpluginjdbc:
3   profile: default
4   install:
5     jar: /com/anjana/tot-plugin-jdbc/{{ version.totpluginjdbc }}/tot-plugin-jdbc-{{ version.totpluginjdbc }}.jar
6     launcher: /com/anjana/tot-plugin-jdbclauncher/{{ version.totpluginjdbc }}/tot-plugin-jdbclauncher-{{ version.totpluginjdbc }}.bin
7     path: /opt/tot-plugin-jdbc
8     backup: /opt/backup/tot-plugin-jdbc
9     edusaur: localhost
10    edusaport: 8888
11    config: /opt/tot-plugin-jdbc
12    serviceuser: anjana
13    javaopts: '-Xmx256m'
14    groupUrl: 'com.anjana'
15    nameUrl: 'tot-plugin-jdbc'
16    hibernateagent:
17      jar: /com/anjana/xjar-agent-hibernate/30.2.0/xjar-agent-hibernate-30.2.0.jar
18      path: /opt/common
19  instances:
20  - tot-plugin-jdbc.service.j2
21  - tot-plugin-jdbc2.service.j2
22

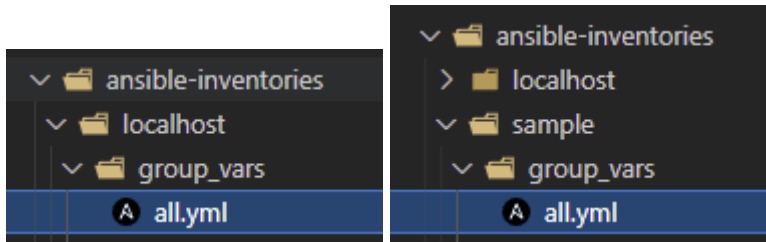
```

NOTA: Esto resultará muy sencillo en Visual Studio buscando directamente en la carpeta del nuevo plugin y reemplazando los valores y variables. Tienen que ser reemplazados ambos.

También se pueden editar los valores directamente en los ficheros correspondientes para las tareas y templates del descriptor de servicio de forma manual.

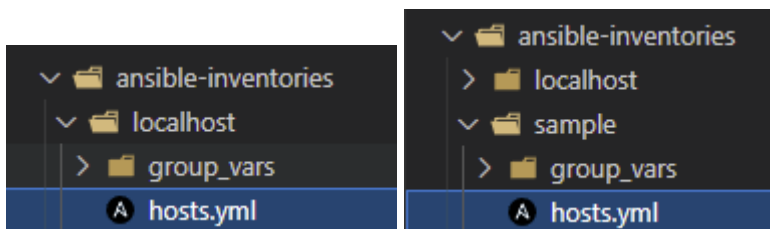


- Adicionalmente para los dos inventarios que aparecen a continuación también es necesaria la adición de la línea que indica la versión del nuevo plugin:



```
version:  
  edusa: 4.2.7  
  config_local: 4.2.11  
  hecate: 4.2.7  
  hermes: 4.2.11  
  kerno: 4.2.11  
  minerva: 4.2.11  
  portuno: 4.2.11  
  tot: 4.2.11  
  viator: 4.2.10  
  zeus: 4.2.11  
  portal: 4.2.11  
  admin: 4.2.11  
  drittesta: 4.2.7  
  totpluginjdbc: 4.2.11  
  totpluginldap: 4.2.11  
  ansible: 4.2.11.a1
```

6. A continuación será necesario replicar otra entrada cualquiera de los ficheros hosts.yml que se muestran:



Sustituyendo los valores por los que apliquen al nuevo plugin de igual manera que la mencionada anteriormente:

```
ansible_python_interpreter: python3
totpluginjdbchosts:
  hosts:
    totpluginjdbc1:
      ansible_host: <IP_servidor>
      ansible_port: 22
      service_hostname: localhost
totpluginldaphosts:
  hosts:
    totpluginldap1:
      ansible_host: <IP_servidor>
      ansible_port: 22
      service_hostname: localhost
```

7. El último paso ya sería actualizar el fichero anjana.yml para la incorporación del nuevo rol, de igual manera, replicando si se quiere una entrada existente de esta forma:

```
> ansible-inventories
> cicdcq
> roles
> .gitignore
A anjana.yml
README.md
```

```
- name: Import role tot-plugin-jdbc
hosts: totpluginjdbchosts
tasks:
  - import_role:
      name: tot-plugin-jdbc
      vars:
        dir: group_vars
      tags:
        - tot-plugin-jdbc
        - tot-plugins
- name: Import role tot-plugin-ldap
hosts: totpluginldaphosts
tasks:
  - import_role:
      name: tot-plugin-ldap
      vars:
        dir: group_vars
      tags:
        - tot-plugin-ldap
        - tot-plugins
```

Tras esto ya es posible el lanzamiento del comando para el despliegue del nuevo plugin de acuerdo a las indicaciones anteriores del kit:

Para su despliegue local:

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

Para el aprovisionamiento de otra máquina alcanzable en la red:

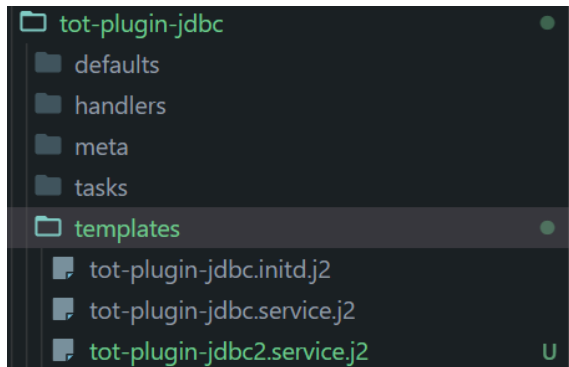
```
sudo ansible-playbook -i ansible-inventories/sample/hosts.yml anjana.yml --tags tot-plugin-ldap
```

NOTA: Para el correcto funcionamiento del nuevo plugin la plantilla de configuración tiene que ser ajustada siguiendo las indicaciones de la documentación existente para dicho plugin.

Generación de nueva instancia de un plugin ya existente

Los puntos a editar/crear son los siguientes:

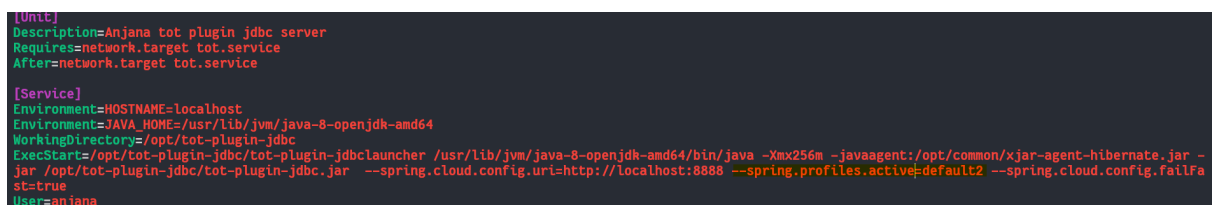
- Debemos copiar tantos templates como servicios queramos en el apartado de roles/totpluginXXX/templates cambiando el nombre a tot-plugin-XXX2.service.j2



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



- Una vez en la máquina, debemos irnos al servicio nuevo y modificar el profile por uno diferente.



- Como último paso, vamos a la carpeta de configuración del plugin, debemos de duplicar el archivo de configuración que ya está, y cambiarle el nombre a application-<nombre_perfilnuevo>.yaml. Una vez creado, lo editamos y debemos de modificar los siguientes campos.

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

totplugin:
  location: http://totpluginjdbcserver: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
```

