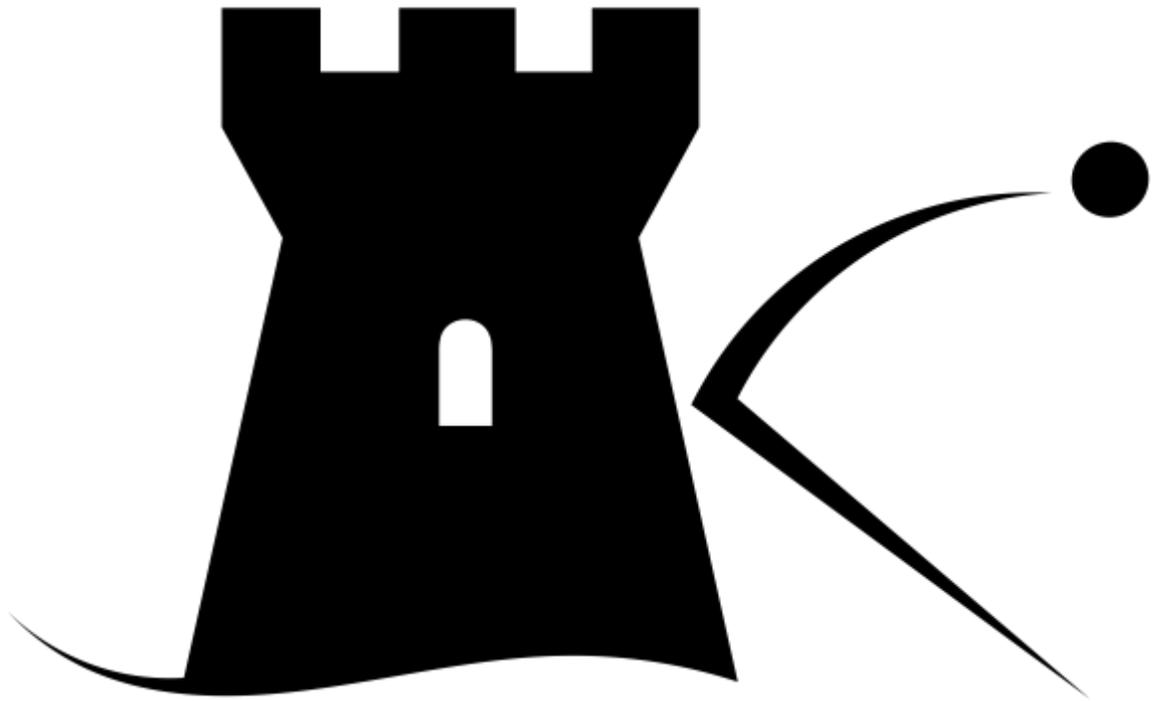


1- INSTALACIÓN FOREMAN/KATELLO Y PUPPET



KATELLO™

2- ÍNDICE

1- INSTALACIÓN FOREMAN/KATELLO Y PUPPET.....	1
2- ÍNDICE.....	2
3. OBJETIVOS A CONSEGUIR Y CONSEGUIDOS.....	3
4 - ESCENARIO NECESARIO PARA LA REALIZACIÓN DEL PROYECTO.....	4
5. FUNDAMENTOS TEÓRICOS Y CONCEPTOS.....	5
5.1. FOREMAN.....	5
5.2. KATELLO.....	6
5.3. PUPPET.....	9
6. DESCRIPCIÓN DETALLADA DE LO QUE SE HA REALIZADO.....	11
6.1. INSTALACIÓN DE FOREMAN/KATELLO Y PUPPET.....	11
6.1.1 PARTICIONADO.....	12
6.1.2 INSTALACIÓN PAQUETERÍA.....	13
6.1.3 MIGRACIÓN CENTOS7 A ORACLE LINUX.....	17
6.2 CREACIÓN Y CONFIGURACIÓN DE USUARIOS.....	19
6.3 REGISTRAR UNA MÁQUINA A KATELLO.....	24
6.4 CREACIÓN DE ORGANIZACIONES Y UBICACIONES.....	26
6.5 CREACIÓN DE REPOSITORIOS.....	29
6.6 IMPORTAR Y EJECUTAR MÓDULOS DE PUPPET.....	32
6.6.1 OPENSAN.....	32
6.6.2 REMOTE EXECUTION.....	35
6.7 CREACIÓN DE CONTENEDORES.....	36
7. CONCLUSIONES Y PROPUESTAS.....	37
8. DIFICULTADES QUE SE HAN ENCONTRADO.....	38
9. BIBLIOGRAFÍA, ENLACES, RESEÑAS.....	39

3. OBJETIVOS A CONSEGUIR Y CONSEGUIDOS

En este proyecto debemos conseguir los siguientes objetivos establecidos:

- Instalación de Foreman/Katello.
- Instalación de Puppet.
- Configuración de Foreman/Katello.
- Configuración de Puppet.

Mi intención en los dos primeros apartados es realizar la instalación en foreman/katello y puppet en un sistema Oracle Linux 7.9 Este apartado esta conseguido ya que más adelante explicaré los pasos de instalación que he seguido.

Dentro del apartado de configuración de Foreman/Katello detallaré ciertos puntos como la suscripción de una máquina a katello, la creación de roles y usuarios...

Dentro del apartado de configuración de Puppet detallaré la ejecución de los módulos de puppet dentro de katello.

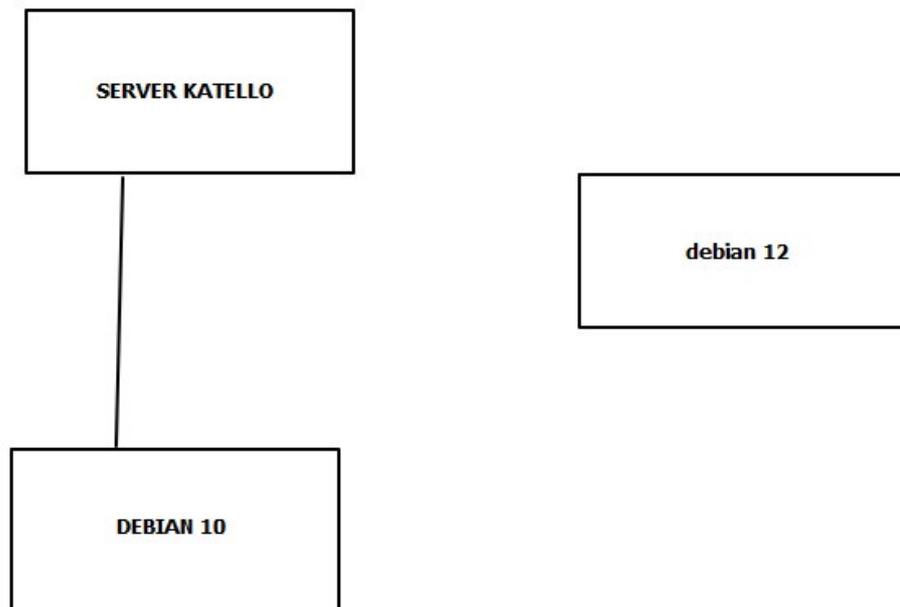
En cuanto al repositorio en GitHub lo crearé nada más entregar este documento de memoria que subiré a dicho repositorio.

4 - ESCENARIO NECESARIO PARA LA REALIZACIÓN DEL PROYECTO

Este escenario constará de dos servidores, uno con un servidor Debian 11, y otro con Rocky Linux, dichos servidores serán suscritos a Katello, y que a su vez serán configurados algunos parámetros de prueba con módulos de Puppet.

También, desde katello, se ejecutarán una serie de

El escenario se forma de la siguiente manera:



Como podemos ver, las máquinas Debian y Rocky serán dos máquinas virtuales dentro de SERVER-KATELLO las cuales funcionarán como ejemplo para realizar las configuraciones.

Ya instalado Foreman/Katello e implementado el escenario, se van a realizar las siguientes prácticas a demostrar:

El hipervisor de virtualización que utilizaré para la creación de ambos servidores será Virtualbox.

Para la demostración mi idea es configurar un usuario y demostrar los roles para registrar un debian 12 a katello

Con la máquina debian10 realizaré las pruebas para realizar los puntos que detallaré en la memoria.

5. FUNDAMENTOS TEÓRICOS Y CONCEPTOS

Este apartado procederemos a definir los conceptos teóricos de:

- Foreman
- Katello
- Puppet

5.1. FOREMAN

Foreman es una herramienta de gestión de infraestructura de código abierto que te permite automatizar y administrar tus servidores físicos y virtuales. Proporciona una interfaz gráfica para gestionar y controlar tus entornos de infraestructura, lo que facilita la gestión de recursos y la configuración de servicios.

Aquí hay algunas características clave de Foreman:

- **Provisionamiento de servidores:** Foreman te permite automatizar el proceso de aprovisionamiento de servidores físicos y virtuales. Puedes definir plantillas de configuración que se utilizarán para instalar y configurar sistemas operativos en los servidores.
- **Gestión de configuración:** Foreman puede integrarse con herramientas de gestión de configuración como Puppet, Chef o Ansible. Esto te permite gestionar y aplicar configuraciones consistentes en tus servidores, lo que facilita la administración y el mantenimiento.
- **Descubrimiento de hardware:** Foreman puede descubrir automáticamente los servidores físicos en tu red y agregarlos a su inventario. Esto te ayuda a tener un control centralizado sobre todos tus servidores y simplifica la gestión de la infraestructura.
- **Monitoreo y generación de informes:** Foreman ofrece capacidades de monitoreo y generación de informes que te permiten realizar un seguimiento del estado de tus servidores y recursos. Puedes obtener información sobre el rendimiento, la utilización de recursos y otros datos relevantes.
- **Integración con sistemas de virtualización:** Foreman es compatible con diferentes sistemas de virtualización, como VMware, oVirt, Libvirt y más. Esto te permite gestionar tanto servidores físicos como virtuales desde una única interfaz.

- **Roles y permisos:** Foreman cuenta con un sistema de roles y permisos que te permite definir y gestionar los niveles de acceso de los usuarios. Puedes asignar diferentes roles a los usuarios y controlar qué funcionalidades y recursos pueden acceder.
- **Extensibilidad:** Foreman es altamente extensible y personalizable. Puedes agregar extensiones, complementos e integraciones adicionales para adaptar la herramienta a tus necesidades específicas.

En cuanto a las configuraciones y funcionalidades, Foreman ofrece una amplia gama de opciones de configuración. Puedes personalizar la configuración de red, almacenamiento, seguridad, usuarios y otros aspectos de tu infraestructura. Además, puedes definir configuraciones específicas para cada servidor o grupo de servidores.

En resumen, Foreman es una herramienta poderosa para la gestión de infraestructura que automatiza y simplifica las tareas de aprovisionamiento, configuración y monitoreo de servidores físicos y virtuales. Con su interfaz intuitiva y sus características robustas, te ayuda a optimizar la administración de tu infraestructura y a mantener un control centralizado sobre tus recursos.

5.2. KATELLO

Katello es una herramienta de gestión de contenido de código abierto que se integra con Foreman para proporcionar capacidades de gestión de ciclo de vida completo para sistemas operativos y aplicaciones. Proporciona una solución centralizada para administrar y distribuir contenido como paquetes, erratas y configuraciones a través de una interfaz web fácil de usar.

Estas son algunas de las características y funcionalidades clave de Katello:

- **Gestión de contenido:** Katello te permite gestionar y distribuir contenido de software, como paquetes de software, erratas y módulos, desde múltiples fuentes. Puedes crear repositorios de software personalizados, sincronizarlos con fuentes externas y controlar las versiones y las actualizaciones del contenido.
- **Gestión de erratas y parches:** Katello facilita la gestión de erratas y parches de seguridad para tus sistemas. Puedes aplicar erratas automáticamente en tus sistemas y programar actualizaciones periódicas para mantener tus sistemas actualizados y seguros.
- **Provisionamiento de sistemas:** Katello se integra con Foreman para ofrecer capacidades de aprovisionamiento de sistemas. Puedes definir plantillas de configuración para instalar y configurar sistemas operativos en servidores físicos y virtuales, lo que facilita la implementación rápida y consistente de nuevos sistemas.
- **Administración de contenido del sistema operativo:** Katello permite gestionar y distribuir contenido específico del sistema operativo, como configuraciones de red, particiones y configuraciones de inicio. Puedes definir perfiles de configuración y aplicarlos a sistemas individuales o grupos de sistemas.

- **Auditoría y cumplimiento normativo:** Katello registra y registra todas las acciones realizadas en la plataforma, lo que facilita la auditoría y el cumplimiento normativo. Puedes rastrear quién hizo qué cambios y cuándo, lo que ayuda a mantener un historial de cambios y a cumplir con los requisitos de seguridad y auditoría.
- **Integración con herramientas de gestión de configuración:** Katello se puede integrar con herramientas de gestión de configuración como Puppet, Ansible o Chef. Esto te permite gestionar y aplicar configuraciones consistentes en tus sistemas, lo que simplifica la administración y el mantenimiento.
- **Gestión de contenido del ciclo de vida:** Katello ofrece capacidades de gestión de ciclo de vida completo para el contenido de software. Puedes definir entornos de ciclo de vida, como desarrollo, pruebas y producción, y promover contenido a través de estos entornos para garantizar la coherencia y la estabilidad de tus sistemas.

A su vez, Katello cuenta con una API donde se puede recopilar información si es necesario:

<https://theforeman.org/plugins/katello/3.17/api/apidoc/v2.html>

Roles en katello:

- **Ansible Roles Manager:** Encargado de administrar y gestionar los roles de Ansible.
- **Ansible Tower Inventory Reader:** Permisos requeridos para el usuario utilizado por el elemento de Inventario Dinámico de Ansible Tower.
- **Api_Role:** Rol relacionado con el acceso y permisos para utilizar la API de Ansible.
- **Auditor:** Rol que otorga permisos para ver únicamente el registro de auditoría y nada más.
- **Bookmarks manager:** Rol que otorga permisos para gestionar los marcadores de búsqueda. Por lo general, es útil en combinación con el rol de Visor. Este rol también permite actualizar todos los marcadores públicos.
- **Compliance manager:** Rol que otorga todos los permisos relacionados con las características de cumplimiento a usuarios que no son administradores.
- **Compliance viewer:** Rol que otorga permisos de lectura para la configuración de políticas, resultados de escaneo y descarga de informes relacionados con el cumplimiento.
- **Content Exporter:** Rol que otorga permisos para exportar vistas de contenido en una organización.

- **Content Importer:** Rol que otorga permisos para importar vistas de contenido en una organización.
- **Create ARF report:** Crear informes ARF (Automatic Remediation Framework).
- **Default role:** Rol que se asigna automáticamente a cada usuario en el sistema. Agregar un permiso lo otorga a todos los usuarios.
- **Edit hosts:** Rol que otorga permisos para actualizar hosts. Para características proporcionadas por complementos, es posible que necesites combinar este rol con otros roles proporcionados por esos complementos.
- **Edit partition tables:** Rol que otorga permisos requeridos para administrar tablas de partición.
- **Manager:** Rol que otorga todos los permisos disponibles. Con este rol, el usuario puede hacer todo lo que un administrador puede hacer, excepto cambiar la configuración.
- **Organization admin:** Rol que otorga todos los permisos, excepto la gestión de organizaciones. Puede ser utilizado para delegar la administración de una organización específica a un usuario. Para crear este rol, se debe clonar este rol y asignar las organizaciones deseadas.
- **PuppetDB Node Manager:** Encargado de administrar los nodos de PuppetDB.
- **PuppetDB Node Viewer:** Rol que permite visualizar los nodos de PuppetDB.
- **Register hosts:** Rol que permite registrar hosts.
- **Remote Execution Manager:** Rol con permisos para gestionar plantillas de trabajo, características de ejecución remota, cancelar trabajos y ver registros de auditoría.
- **Remote Execution User:** Rol con permisos para ejecutar trabajos de ejecución remota en hosts.
- **Site manager:** Rol que otorga principalmente permisos de visualización, pero también permisos necesarios para administrar hosts en la infraestructura. Los usuarios con este rol pueden actualizar parámetros de Puppet, crear y editar hosts, administrar medios de instalación, subredes, grupos de usuarios y editar usuarios existentes.
- **System admin:** Rol que otorga permisos para administrar organizaciones, ubicaciones, usuarios, grupos de usuarios, fuentes de autenticación, roles, filtros y configuraciones. Este es un rol muy poderoso que potencialmente puede obtener acceso a todos los recursos.

- **Tasks Manager:** Rol que otorga permisos para inspeccionar, cancelar, reanudar y desbloquear tareas.
- **Tasks Reader:** Rol que otorga permisos para inspeccionar tareas.
- **Viewer:** Rol que otorga acceso de solo lectura. Los usuarios con este rol pueden ver todos los datos pero no pueden realizar modificaciones.
- **View hosts:** Rol que otorga permisos únicamente para ver hosts.

En resumen, Katello es una herramienta integral de gestión de contenido que se integra con Foreman para ofrecer una solución completa de gestión de ciclo de vida para sistemas operativos y aplicaciones. Con sus características de gestión de contenido, aprovisionamiento de sistemas y auditoría, Katello te ayuda a administrar y distribuir contenido de manera eficiente y segura en tu infraestructura.

5.3. PUPPET

Puppet es una herramienta de gestión de configuración de código abierto que te permite automatizar y administrar la configuración de tus sistemas de forma eficiente y escalable. Con Puppet, puedes definir y mantener la configuración de tus sistemas de manera declarativa, lo que significa que especificas el estado deseado de los sistemas y Puppet se encarga de aplicar y mantener ese estado.

Estas son algunas de las características y funcionalidades clave de Puppet:

- **Declarativo y basado en recursos:** Puppet utiliza un enfoque declarativo para la configuración, lo que significa que defines el estado deseado de tus sistemas en términos de recursos individuales, como archivos, servicios, paquetes, usuarios, entre otros. Puppet se encarga de resolver las dependencias y aplicar los cambios necesarios para alcanzar el estado deseado.
- **Lenguaje de dominio específico:** Puppet utiliza su propio lenguaje de dominio específico (DSL) para definir la configuración. El DSL de Puppet es intuitivo y legible, lo que facilita la definición de la configuración de tus sistemas.
- **Gestión de recursos:** Puppet proporciona una amplia biblioteca de recursos predefinidos que representan elementos comunes de configuración de sistemas, como archivos, servicios, paquetes, usuarios, grupos, entre otros. Puedes utilizar estos recursos para definir y gestionar la configuración de tus sistemas de manera eficiente.

- **Modularidad y reusabilidad:** Puppet fomenta la modularidad y la reusabilidad al permitirte organizar tu configuración en módulos. Los módulos son unidades lógicas que contienen recursos y se pueden reutilizar en diferentes sistemas y entornos, lo que facilita la gestión y el mantenimiento de la configuración.
- **Orquestación y gestión de aplicaciones:** Puppet no solo se limita a la configuración de sistemas individuales, sino que también te permite orquestar y gestionar la configuración de aplicaciones completas. Puedes definir perfiles y roles que agrupen recursos y configuraciones relacionados, lo que facilita la gestión de aplicaciones complejas y distribuidas.
- **Descubrimiento y remediación automáticos:** Puppet puede descubrir automáticamente los sistemas en tu infraestructura y aplicar la configuración apropiada. Además, Puppet es capaz de detectar y corregir desviaciones de la configuración deseada, lo que garantiza que tus sistemas se mantengan en el estado correcto de manera continua.
- **Comunidad y ecosistema:** Puppet cuenta con una comunidad activa y un ecosistema sólido de módulos y herramientas complementarias. Puedes aprovechar estos recursos para obtener soporte, compartir conocimientos y aprovechar soluciones predefinidas para casos de uso comunes.

En resumen, Puppet es una potente herramienta de gestión de configuración que te permite automatizar y administrar la configuración de tus sistemas de manera eficiente y escalable. Con su enfoque declarativo, modularidad y capacidad de gestión de recursos, Puppet te ayuda a mantener la configuración de tus sistemas de forma coherente y confiable.

6. DESCRIPCIÓN DETALLADA DE LO QUE SE HA REALIZADO

En este apartado indicaremos los pasos para la instalación y configuración de foreman/katello y puppet.

En el primer punto indicaremos los pasos de instalación a seguir para realizar la instalación de foreman/katello y puppet.

Seguidamente, una vez ya instalado, accederemos al sitio web con las credenciales que se generan automáticamente y crearemos nuestro usuario al cual asignaremos los roles de admin.

Crearemos una organización, en donde se registrará nuestro servidor. También crearemos una localización para indicar si es developer, integration, continuousintegration, preproduction, o production, en mi caso sera developer.

Posteriormente, explicaremos el proceso de suscripción de una máquina a katello y seguidamente realizaremos un script que automatice la suscripción de una máquina a katello.

Finalmente haré una demostración de como funciona puppet dentro de katello.

6.1. INSTALACIÓN DE FOREMAN/KATELLO Y PUPPET

Para la instalación de foreman/katello y puppet seguiremos una serie de puntos:

Primero realizaremos un particionado con lvm en el cuál irán los archivos de instalación necesarios para foreman/katello.

Seguidamente, instalaremos los repositorios necesarios para la instalación e instalaremos foreman/katello.

Hecho esto nos queda la última parte, ya que el sistema operativo en el cual vamos a instalar foreman/katello será centos7, en el cual posteriormente realizaremos una migración a Oracle Linux, esto se hace porque en Oracle Linux hay repositorios que no funcionan y cuando se intenta instalar foreman/katello no se instalan bien las dependencias.

6.1.1 PARTICIONADO

El particionado que realizaremos en nuestro sistema constará de un grupo de volúmenes lvm en donde se realizarán las siguientes particiones:

Particiones de datos:

NOMBRE	MOUNTPOINT	TAMAÑO	FORMATO
centos-home	/home	20G	xf
centos-lv_pulp	/var/lib/pulp	30G	xf
centos-lv_squid	/var/spool/squid	10G	xf
centos-lv_mongodb	/var/lib/mongodb	10G	xf
centos-lv_pgsql	/var/lib/pgsql	10G	xf

Particiones de Sistema

NOMBRE	MOUNTPOINT	TAMAÑO	FORMATO
centos-root	/	50G	xf
centos-var	/var	250G	xf
centos-swap	Swap	6G	swap
sda1	/boot/efi	1G	vf
sda2	/boot	1G	Xfs

Aquí una captura de las particiones realizadas:

```
[root@joseju ~]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sr0                  11:0    1  1024M  0 rom
sda                   8:0     0 447,1G  0 disk
├─sda2                 8:2     0    1G  0 part /boot
├─sda3                 8:3     0  386G  0 part
│   ├─centos-swap      252:1   0    6G  0 lvm  [SWAP]
│   ├─centos-lv_pulp   252:6   0   30G  0 lvm  /var/lib/pulp
│   ├─centos-lv_squid  252:4   0   10G  0 lvm  /var/spool/squid
│   ├─centos-home     252:2   0   20G  0 lvm  /home
│   ├─centos-root     252:0   0   50G  0 lvm  /
│   ├─centos-lv_pgsql  252:7   0   10G  0 lvm  /var/lib/pgsql
│   ├─centos-lv_mongodb 252:5   0   10G  0 lvm  /var/lib/mongodb
│   └─centos-var       252:3   0 250G  0 lvm  /var
└─sda1                 8:1     0    1G  0 part /boot/efi
```

6.1.2 INSTALACIÓN PAQUETERÍA

Una vez hemos realizado las pertinentes particiones, procederemos a instalar los repositorios de foreman, katello y puppet. Los pasos a seguir son los siguientes:

Primero instalamos los repositorios y paquetes necesarios para la instalación de katello(en este paso se instalará foreman):

```
yum -y localinstall
http://fedorapeople.org/groups/katello/releases/yum/3.8/katello/el7/x86\_64/katello-repos-latest.rpm

yum -y localinstall
http://yum.theforeman.org/releases/1.19/el7/x86\_64/foreman-release.rpm

yum -y localinstall https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm

yum -y localinstall http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

yum -y install foreman-release-scl python-django
```

Captura Repositorio Katello:

```
Instalando:
katello-repos          noarch          3.8.1-1.el7          /katello-repos-latest          2.9 k
Resumen de la transacción
=====
Instalar 1 Paquete
Tamaño total: 2.9 k
Tamaño instalado: 2.9 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando      : katello-repos-3.8.1-1.el7.noarch          1/1
  Comprobando    : katello-repos-3.8.1-1.el7.noarch          1/1
Instalado:
katello-repos.noarch 0:3.8.1-1.el7
¡Listo!
[root@joseju ~]# █
```

Captura Repositorio Foreman-release:

```
=====  
Package           Arquitectura Versión           Repositorio           Tamaño  
=====  
Instalando:  
foreman-release   noarch         1.19.1-1.el7         /foreman-release     5.1 k  
Resumen de la transacción  
=====  
Instalar 1 Paquete  
Tamaño total: 5.1 k  
Tamaño instalado: 5.1 k  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Instalando      : foreman-release-1.19.1-1.el7.noarch      1/1  
  Comprobando     : foreman-release-1.19.1-1.el7.noarch      1/1  
Instalado:  
foreman-release.noarch 0:1.19.1-1.el7  
¡Listo!  
[root@joseju ~]# █
```

Captura repositorio puppetlabs:

```
=====  
Instalando:  
puppetlabs-release-pc1 noarch 1.1.0-5.el7 /puppetlabs-release-pc1-el-7.noarch 11 k  
Resumen de la transacción  
=====  
Instalar 1 Paquete  
Tamaño total: 11 k  
Tamaño instalado: 11 k  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Instalando      : puppetlabs-release-pc1-1.1.0-5.el7.noarch 1/1  
  Comprobando     : puppetlabs-release-pc1-1.1.0-5.el7.noarch 1/1  
Instalado:  
puppetlabs-release-pc1.noarch 0:1.1.0-5.el7  
¡Listo!  
[root@joseju ~]# █
```

Captura repositorio epel-release:

```
=====  
Package           Arquitectura Versión           Repositorio           Tamaño  
=====  
Instalando:  
epel-release      noarch         7-14                /epel-release-latest-7.noarch     25 k  
Resumen de la transacción  
=====  
Instalar 1 Paquete  
Tamaño total: 25 k  
Tamaño instalado: 25 k  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Instalando      : epel-release-7-14.noarch                 1/1  
  Comprobando     : epel-release-7-14.noarch                 1/1  
Instalado:  
epel-release.noarch 0:7-14  
¡Listo!  
[root@joseju ~]# █
```

Captura instalación paquetes foreman-release-scl y python-django:

```
Running transaction
  Instalando      : python-django-bash-completion-1.11.27-1.el7.noarch      1/5
  Instalando      : centos-release-scl-rh-2-3.el7.centos.noarch            2/5
  Instalando      : centos-release-scl-2-3.el7.centos.noarch              3/5
  Instalando      : foreman-release-scl-7-1.el7.noarch                    4/5
  Instalando      : python2-django-1.11.27-1.el7.noarch                  5/5
  Comprobando     : python2-django-1.11.27-1.el7.noarch                  1/5
  Comprobando     : centos-release-scl-2-3.el7.centos.noarch            2/5
  Comprobando     : centos-release-scl-rh-2-3.el7.centos.noarch          3/5
  Comprobando     : python-django-bash-completion-1.11.27-1.el7.noarch    4/5
  Comprobando     : foreman-release-scl-7-1.el7.noarch                  5/5

Instalado:
  foreman-release-scl.noarch 0:7-1.el7          python2-django.noarch 0:1.11.27-1.el7

Dependencia(s) instalada(s):
  centos-release-scl.noarch 0:2-3.el7.centos
  centos-release-scl-rh.noarch 0:2-3.el7.centos
  python-django-bash-completion.noarch 0:1.11.27-1.el7

¡Listo!
[root@joseju ~]# █
```

Ya instalados los repositorios y paquetes necesarios para la instalación, procederemos actualizar los repositorios del sistema y a instalar katello, ejecutamos el siguiente comando:

```
yum update -y && yum install -y katello
```

Captura katello instalado:

```
tomcat.noarch 0:7.0.76-16.el7_9
tomcat-el-2.2-api.noarch 0:7.0.76-16.el7_9
tomcat-jsp-2.2-api.noarch 0:7.0.76-16.el7_9
tomcat-lib.noarch 0:7.0.76-16.el7_9
tomcat-servlet-3.0-api.noarch 0:7.0.76-16.el7_9
xalan-j2.noarch 0:2.7.1-23.el7
xerces-j2.noarch 0:2.11.0-17.el7_0
xml-commons-apis.noarch 0:1.4.01-16.el7
xml-commons-resolver.noarch 0:1.2-15.el7

¡Listo!
[root@joseju ~]# █
```

Y ya finalizada la instalación, procederemos a crear el escenario:

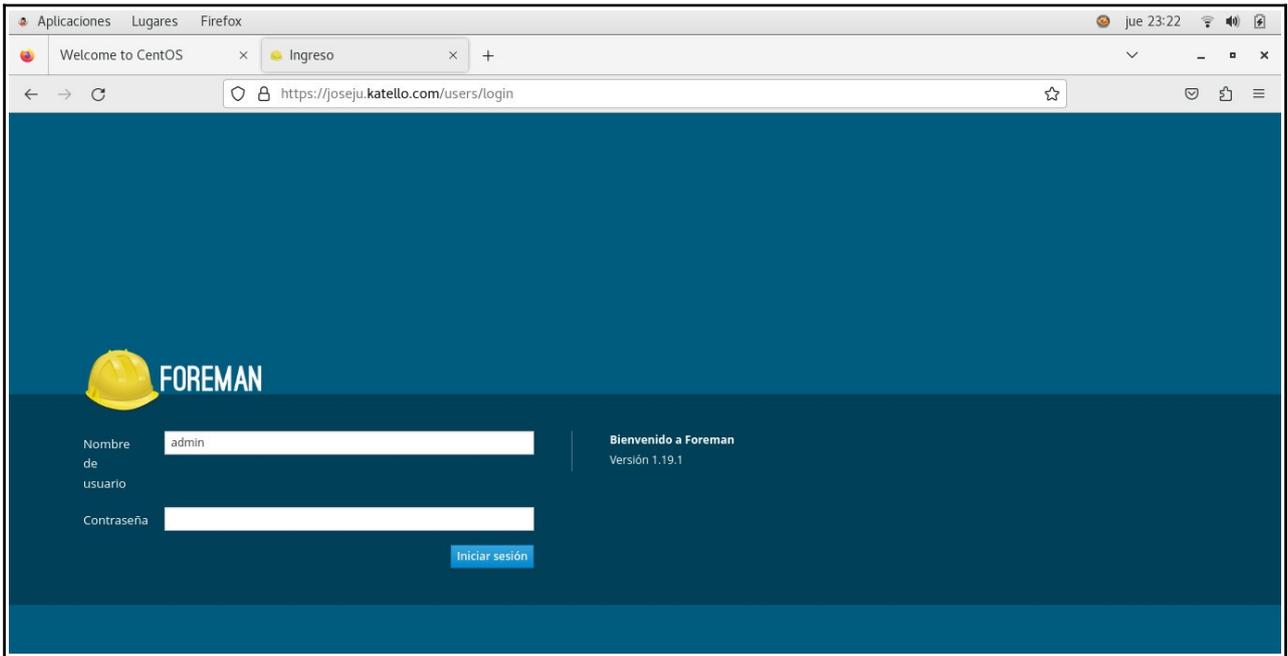
```
foreman-installer --scenario katello\
  --foreman-initial-location "Sevilla"\
  --foreman-initial-organization "Nazareno"
```

La localización inicial indicaremos que es en Sevilla y nuestra organización sera Nazareno, aquí la demostración de que el escenario se ha creado correctamente y nos da un usuario admin con unas credenciales:

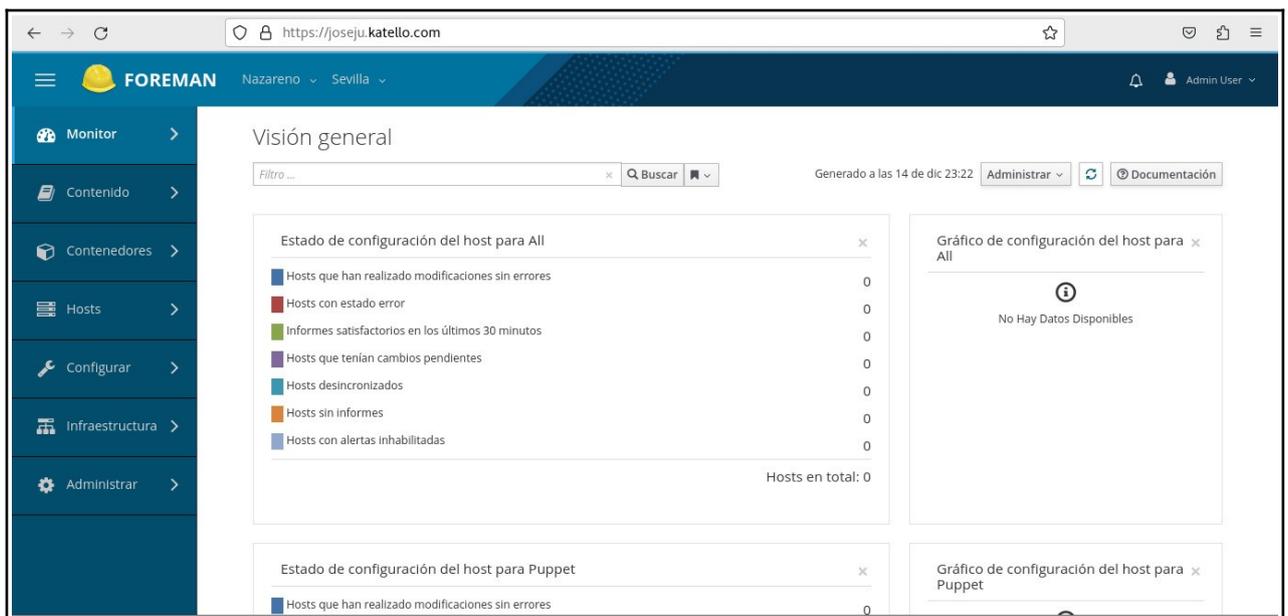
```
[root@joseju joseju]# foreman-installer --scenario katello --foreman-initial-location "Sevilla" --foreman-initial-organization "Nazareno"
Resetting puppet server version param...
Installing puppet server version param... [100%]
Success!
* Katello is running at https://joseju.katello.com
  Initial credentials are admin / MAdvtepGXoSiJCix
* To install an additional Foreman proxy on separate machine continue by running:

  foreman-proxy-certs-generate --foreman-proxy-fqdn "$FOREMAN_PROXY" --certs-tar "/root/$FOREMAN_PROXY-certs.tar"
The full log is at /var/log/foreman-installer/katello.log
Upgrade Step: remove_legacy_mongo...
yum install -y -q rh-mongodb34-syspaths finished successfully!
```

Y accedemos a la página de inicio:



Donde iniciamos sesión con las credenciales generadas anteriormente:



6.1.3 MIGRACIÓN CENTOS7 A ORACLE LINUX

En este punto realizaremos la migración de nuestro sistema actual, que es centos7, a oracle linux 8, ya que como se sabe centos7 ya no recibe soporte y eso conlleva a que tenemos que realizar la migración de nuestro sistema para no tener fallos de seguridad.

Primero comprobamos el sistema operativo instalado:

```
hostnamectl
```

Captura sistema actual:

```
[root@joseju ~]# hostnamectl
  Static hostname: joseju.katello.com
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 00aff61c9d1e584c86c4cfd9523a6990
        Boot ID: d77633d63418434f96bf852b7e639c7a
  Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
        CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-1160.71.1.el7.x86_64
  Architecture: x86_64
[root@joseju ~]#
```

Descargamos el script de migración a oracle:

```
curl -o
https://raw.githubusercontent.com/oracle/centos2ol/main/centos2ol.sh
```

Comprobamos que se ha descargado:

```
[root@joseju joseju]# ls
centos2ol.sh Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos
```

Ya descargado, lo ejecutamos, esto tardará un buen rato ya que se tienen que eliminar todos los paquetes de centos e instalar los de oracle:

```
bash centos2ol.sh
```

Al finalizar reiniciamos:

```
reboot
```

Comprobamos que el sistema se ha migrado:

```
[root@joseju ~]# hostnamectl
  Static hostname: joseju.katello.com
        Icon name: computer-laptop
        Chassis: laptop
        Machine ID: ceb3ad9e4c27416faf84e27874320b9b
        Boot ID: 204af58f2c394803a101beb4e58b30fc
  Operating System: Oracle Linux Server 7.9
        CPE OS Name: cpe:/o:oracle:linux:7:9:server
        Kernel: Linux 5.4.17-2136.326.6.el7uek.x86_64
        Architecture: x86-64
[root@joseju ~]# █
```

Y comprobamos que foreman sigue activo y que no ha habido ningún problema con la migración:

```
[root@joseju ~]# systemctl status foreman
● foreman.service - Foreman
   Loaded: loaded (/usr/lib/systemd/system/foreman.service; disabled; vendor preset: disabled)
   Active: active (running) since jue 2023-12-14 21:54:03 CET; 20s ago
     Docs: https://theforeman.org
  Main PID: 6675 (scl)
    Tasks: 4
   CGroup: /system.slice/foreman.service
           └─6675 /usr/bin/scl enable tfm rails server -e $FOREMAN_ENV -p $FO...
             └─6703 /bin/bash /var/tmp/sclcws6iC
               └─6758 /opt/rh/rh-ruby24/root/usr/bin/ruby bin/rails server -e pro...

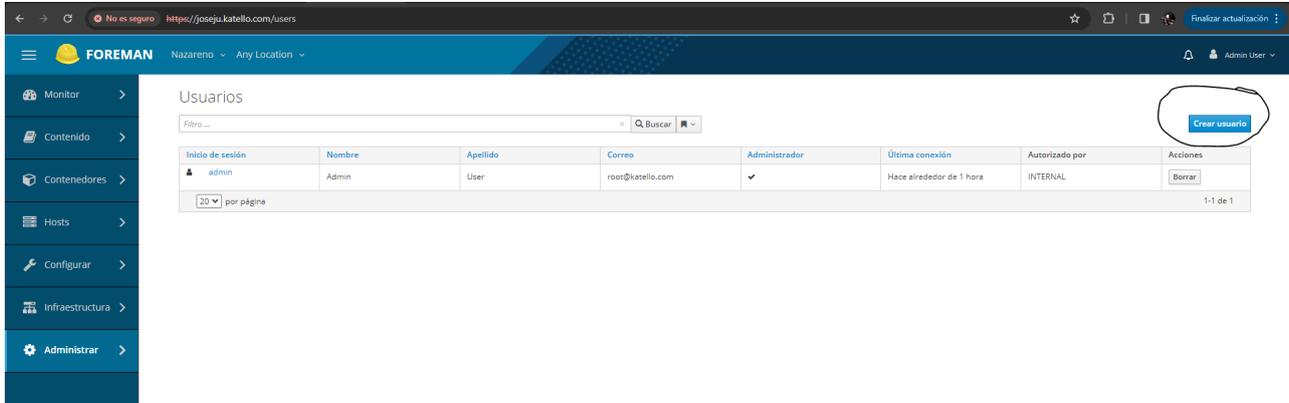
dic 14 21:54:03 joseju.katello.com systemd[1]: Started Foreman.
dic 14 21:54:20 joseju.katello.com scl[6675]: /usr/share/foreman/lib/foreman...P
dic 14 21:54:20 joseju.katello.com scl[6675]: /usr/share/foreman/lib/foreman...e
dic 14 21:54:20 joseju.katello.com scl[6675]: /usr/share/foreman/lib/core_ex...R
dic 14 21:54:20 joseju.katello.com scl[6675]: /opt/theforeman/tfm-ror51/root...e
Hint: Some lines were ellipsized, use -l to show in full.
```

De esta forma, habríamos instalado foreman/katello y puppet en Oracle Linux 7.9.

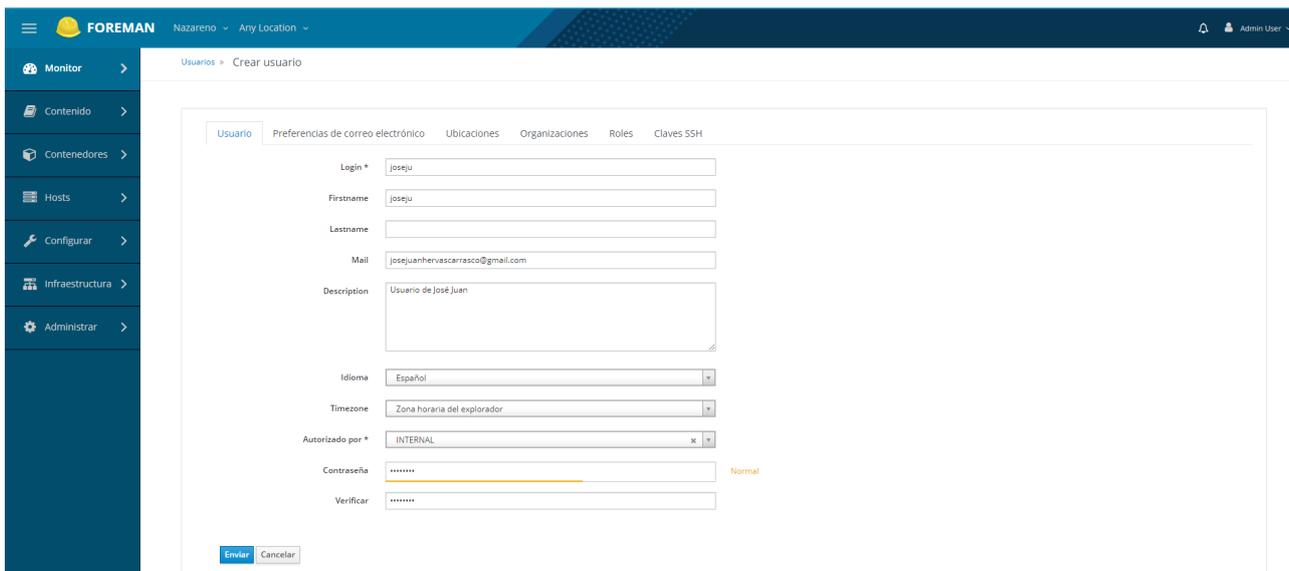
6.2 CREACIÓN Y CONFIGURACIÓN DE USUARIOS

Para la creación de usuarios, haremos lo siguiente:

Accedemos a Administrar-Usuarios-Crear cuenta de Usuarios:



Hecho esto tenemos estas opciones:

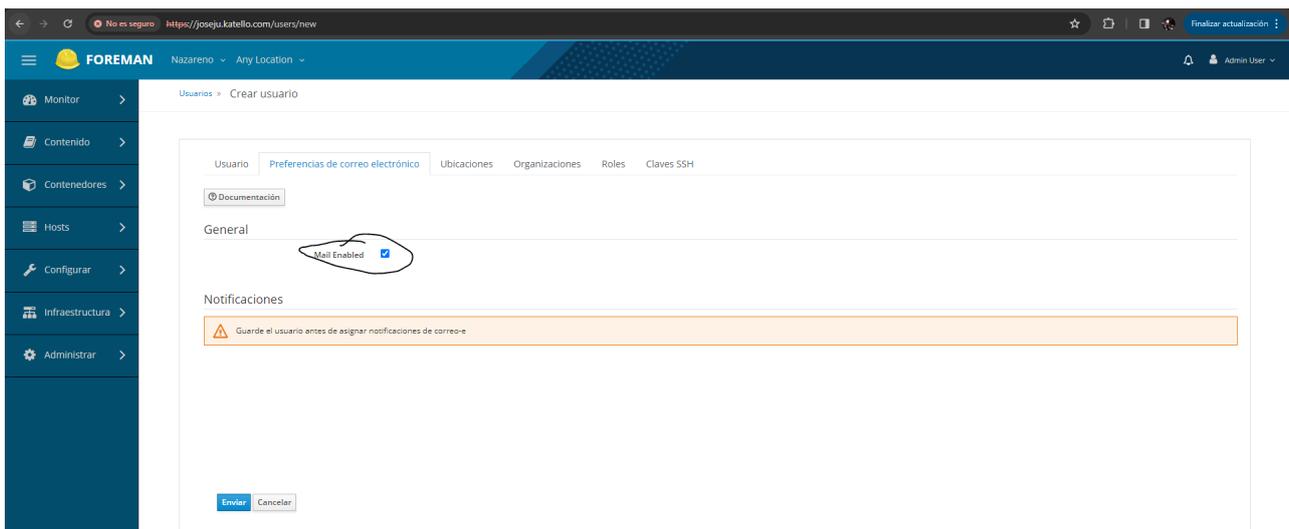


Dentro del apartado de usuario tenemos estas opciones:

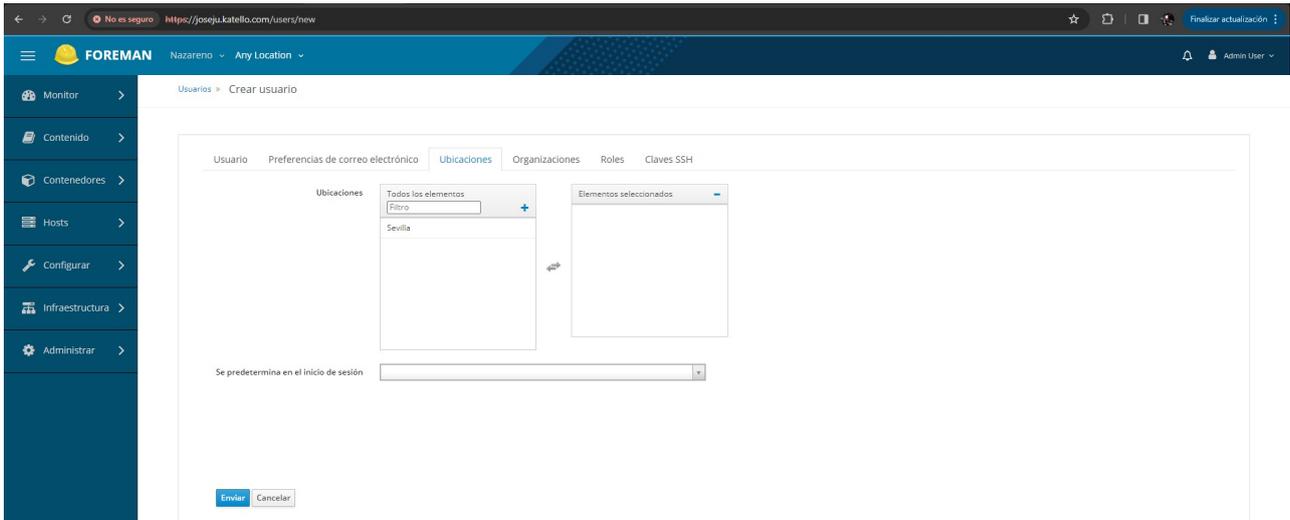
- Login: Este es el nombre que debemos introducir para iniciar sesión.
- Firstname: El nombre de la persona, en mi caso pondré joseju aunque lo suyo es poner José Juan, no es obligatorio introducir este dato.
- Lastname: Aquí debe ir el apellido.

- **Mail:** Aquí introducimos el correo electrónico.
- **Description:** Aquí podemos introducir una breve descripción sobre la función de este usuario o algún dato significativo.
- **Idioma:** Seleccionamos el idioma que va a utilizar el usuario.
- **Timezone:** Seleccionamos la zona horaria.
- **Autorizado por:** Introducimos si el usuario será registrado a nivel Interno o también podemos indicar si es un usuario LDAP, en mi caso como no tengo configurado LDAP, será un usuario interno.
- **Contraseña:** Introducimos la contraseña para acceder a la página.

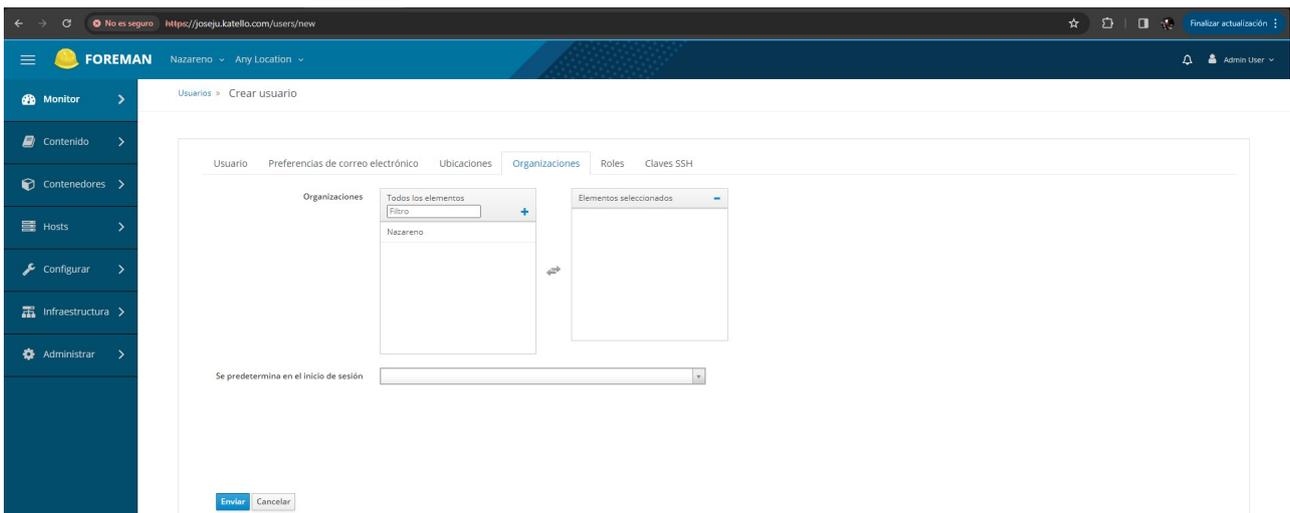
Dentro del apartado de preferencias de correo electrónico, podemos habilitar o deshabilitar el email:



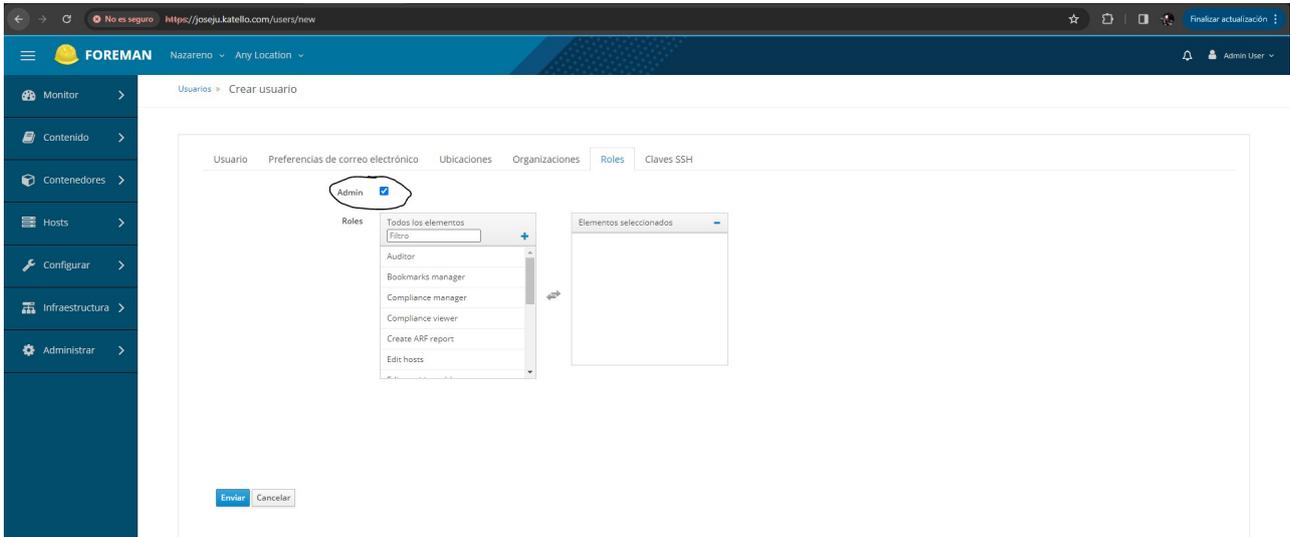
En el apartado de Ubicaciones, podemos indicar las ubicaciones que queremos añadir dentro del usuario:



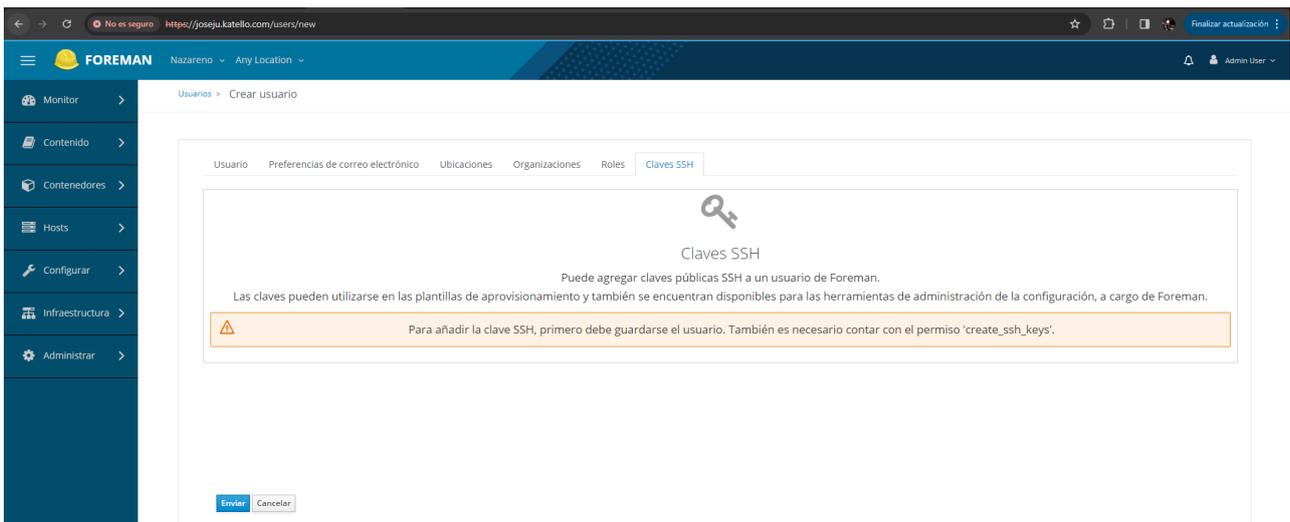
En el apartado de Organizaciones, podemos indicar las organizaciones que queremos añadir dentro del usuario:



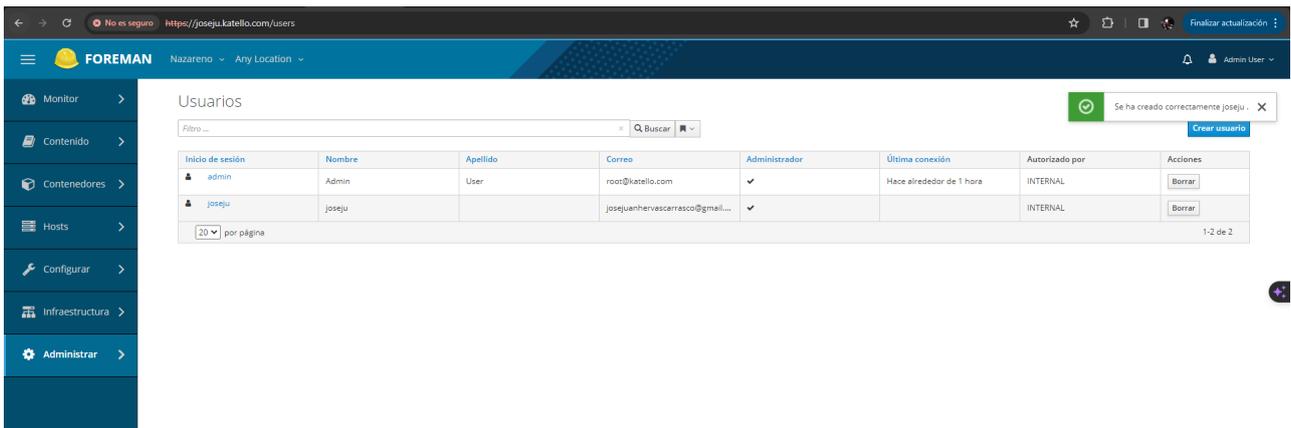
Dentro del apartado de Roles podemos establecer los roles que queremos que tenga nuestro usuario, en nuestro caso pondremos que sea un usuario admin y de esta forma no es necesario ir añadiendo roles, ya que no es necesario:



También se pueden añadir claves ssh a nuestro usuario de katello:



Creamos el usuario:

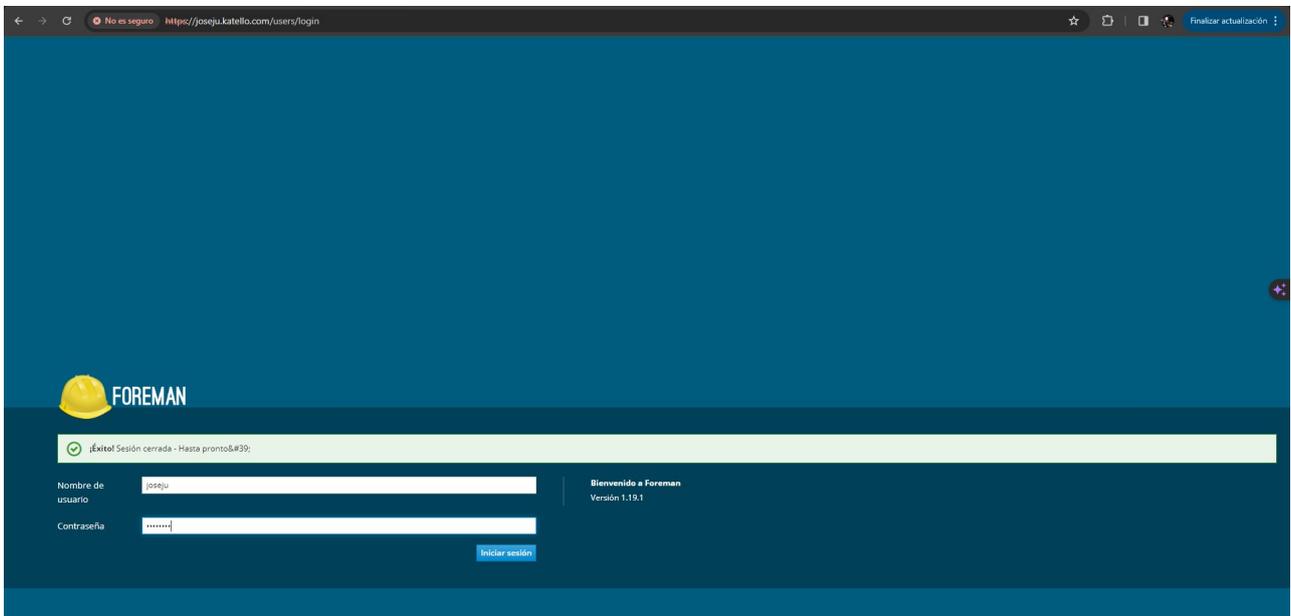


The screenshot shows the FOREMAN web interface at the URL `https://joseju.katello.com/users`. The page title is "Usuarios". A notification at the top right states "Se ha creado correctamente joseju". Below the notification is a table of users:

Inicio de sesión	Nombre	Apellido	Correo	Administrador	Última conexión	Autorizado por	Acciones
admin	Admin	User	root@katello.com	✓	Hace alrededor de 1 hora	INTERNAL	Borrar
joseju	joseju		josejuanherrascarrasco@gmail...	✓		INTERNAL	Borrar

At the bottom of the table, it indicates "20 por página" and "1-2 de 2".

Cerramos sesión:



The screenshot shows the FOREMAN login page at the URL `https://joseju.katello.com/users/login`. A green notification bar at the top says "¡Éxito! Sesión cerrada - Hasta pronto!@39:". Below the notification, there is a login form with fields for "Nombre de usuario" (containing "joseju") and "Contraseña" (masked with dots). A "Bienvenido a Foreman" message is visible on the right, stating "Versión 1.19.1". A "Iniciar sesión" button is located at the bottom right of the form.

E iniciamos sesión con nuestro nuevo usuario:



The screenshot shows the FOREMAN web interface at the URL `https://joseju.katello.com/hosts`. The page title is "Hosts". A notification at the top right says "0 of 2 selected". Below the notification is a table of hosts:

Energía	Nombre	Sistema operativo	Entorno de Puppet	Modelo	Grupo de hosts	Último informe	Acciones
	debian10.linuexamimages-local	Debian 10	production	VirtualBox		Hace alrededor de 2 h...	Editar
	joseju.katello.com	OracleLinux 7.9	production	SATELLITE PRO C50-A...		Hace 18 minutos	Editar

At the bottom of the table, it indicates "20 por página" and "1-2 de 2".

6.3 REGISTRAR UNA MÁQUINA A KATELLO

En nuestro caso, procederemos a registrar nuestro servidor debian de la siguiente forma:

Debian:

En debian primero actualizamos a la última versión e instalamos puppet:

```
apt update && apt upgrade -y && apt install puppet -y
```

Hecho esto, añadimos el host de nuestro servidor katello a *etc/hosts* ya que no contamos con un servidor DNS:

```
GNU nano 3.2 /etc/hosts Modified
127.0.0.1    localhost
127.0.1.1    debian10.linuxvmimages.local  debian10
192.168.34   joseju.katello.com_

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Seguidamente, configuramos el archivo */etc/puppet/puppet.conf*:

```
[main]
logdir=/var/log/puppet
vardir=/var/lib/puppet
ssldir=/var/lib/puppet/ssl
rundir=/var/run/puppet
factpath=$vardir/lib/facter
server = joseju.katello.com
report = true
pluginsync = true

[agent]
certname = debian.katello.com
default_schedules = false
environment = production
```

Ejecutamos el siguiente comando:

```
puppet agent -t
```

Captura de pantalla puppet agent:

```
root@debian10:~# puppet agent -t
Warning: Setting 'pluginsync' is deprecated.
(location: /usr/lib/ruby/vendor_ruby/puppet/defaults.rb:1879:in `block in <module:Puppet>')
Warning: Downgrading to PSON for future requests
Info: Caching certificate for ca
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for debian.katello.com
Info: Certificate Request fingerprint (SHA256): 8A:AE:57:85:E1:FA:F8:F8:D5:13:F3:2D:7C:EF:49:6F:27:7B:62:B3:78:62:04:BA:25:95:6E:05:EF:08:90:19
Info: Caching certificate for ca
Exiting; no certificate found and waitforcert is disabled
root@debian10:~#
```

Comprobamos desde el server que se ha registrado el nuevo certificado:

```
puppet cert list -a
```

Captura de pantalla puppet cert list:

```
[root@joseju joseju]# puppet cert list -a
"debian.katello.com" (SHA256) 8A:AE:57:85:E1:FA:F8:F8:D5:13:F3:2D:7C:EF:49:6F:27:7B:62:B3:78:62:04:BA:25:95:6E:05:EF:08:90:19
+ "joseju.katello.com" (SHA256) B2:EF:3B:EB:63:EF:12:7A:D4:BC:A3:28:6A:90:1E:E2:1B:DB:A6:CF:E2:DC:22:6F:D1:C5:B3:0C:B9:A2:45:68
(alt names: "DNS:joseju.katello.com", "DNS:puppet", "DNS:puppet.katello.com")
[root@joseju joseju]#
```

Y firmamos el certificado:

```
puppet cert sign debian.katello.com
```

Captura de pantalla firmando el certificado:

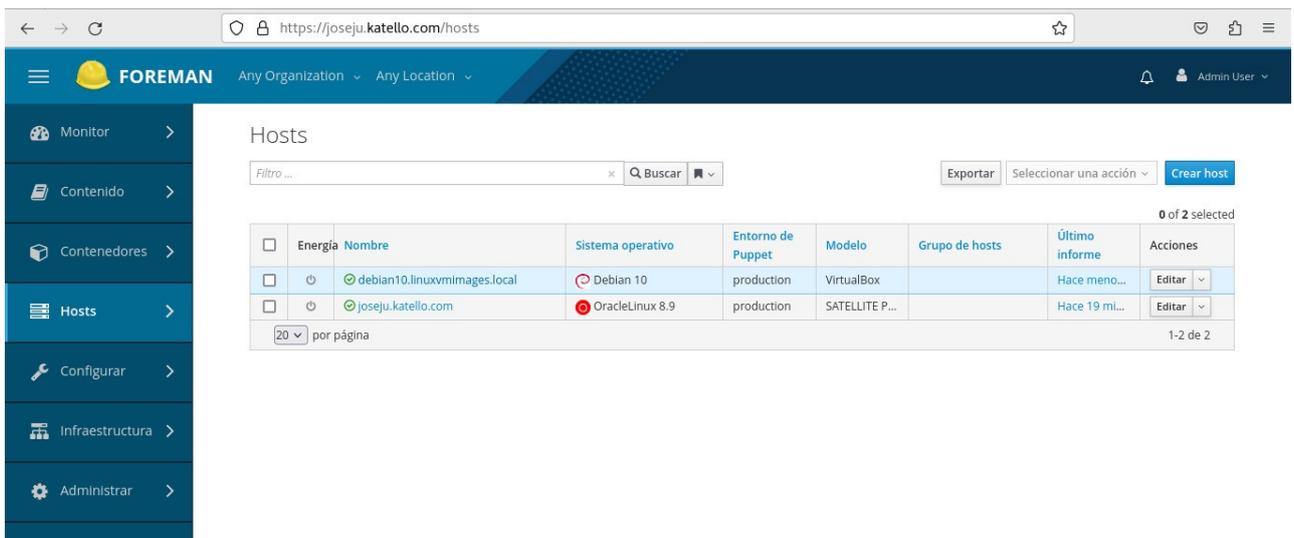
```
[root@joseju joseju]# puppet cert sign debian.katello.com
Signing Certificate Request for:
"debian.katello.com" (SHA256) 8A:AE:57:85:E1:FA:F8:F8:D5:13:F3:2D:7C:EF:49:6F:27:7B:62:B3:78:62:04:BA:25:95:6E:05:EF:08:90:19
Notice: Signed certificate request for debian.katello.com
Notice: Removing file Puppet::SSL::CertificateRequest debian.katello.com at '/etc/puppetlabs/puppet/ssl/ca/requests/debian.katello.com.pem'
[root@joseju joseju]#
```

Para explicarlo de una mejor forma, al instalar katello se nos crea una CA y esta se encargará de firmar todos los certificados que se generen desde un servidor cliente para poder registrar dicho servidor en katello.

Hecho esto, volvemos a ejecutar un agent puppet -t:

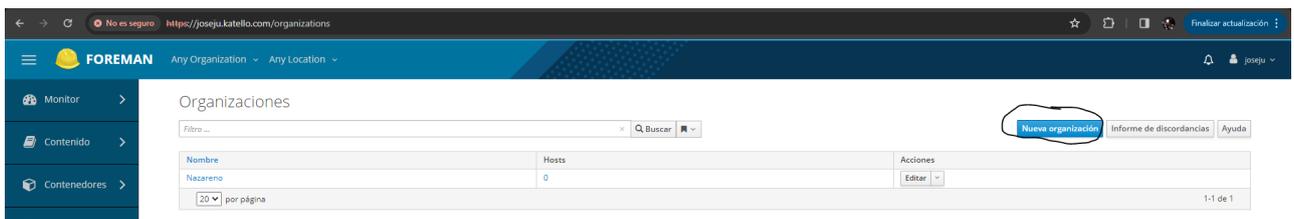
```
root@debian10:~# puppet agent -t
Warning: Setting 'pluginsync' is deprecated.
(location: /usr/lib/ruby/vendor_ruby/puppet/defaults.rb:1879:in `block in <module:Puppet>')
Warning: Downgrading to PSON for future requests
Info: Caching certificate for debian.katello.com
Info: Caching certificate_revocation_list for ca
Info: Caching certificate for debian.katello.com
Warning: Unable to fetch my node definition, but the agent run will continue:
Warning: Error 500 on SERVER: Server Error: Failed to find debian.katello.com via exec: Execution of '/etc/puppetlabs/puppet/node.rb debian.katello.com' returned 1:
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for debian.katello.com
Info: Applying configuration version '1702857828'
Info: Creating state file /var/lib/puppet/state/state.yaml
Notice: Applied catalog in 0.03 seconds
root@debian10:~#
```

Y de esta forma hemos registrado nuestro servidor debian en katello:

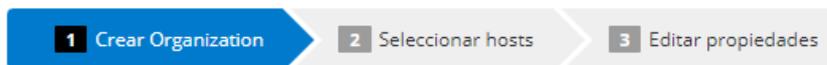


6.4 CREACIÓN DE ORGANIZACIONES Y UBICACIONES

En este punto crearemos una organización y una ubicación, las cuales asignaremos al hosts debian que acabamos de registrar para ello accedemos a Administrar-Organizaciones-Nueva organización:



Ahora, creamos nuestra organización la cual se llamará debians:

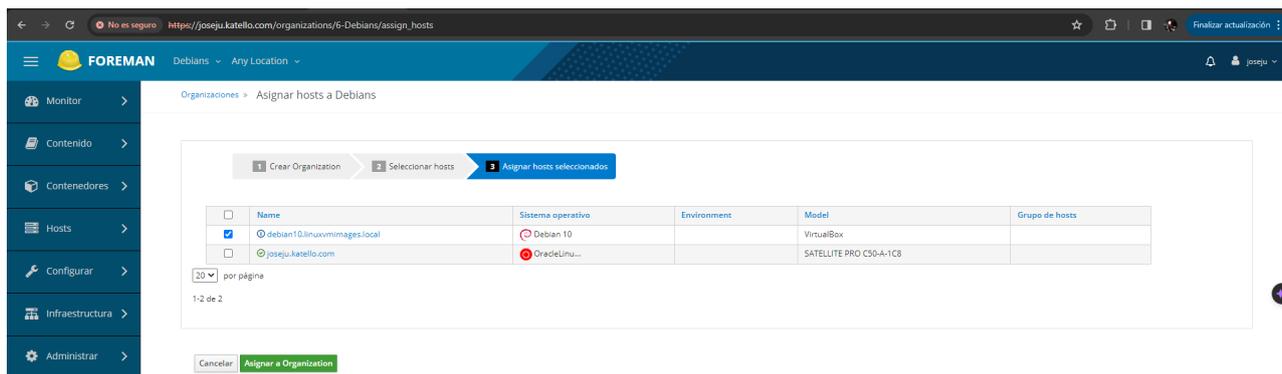


Name *

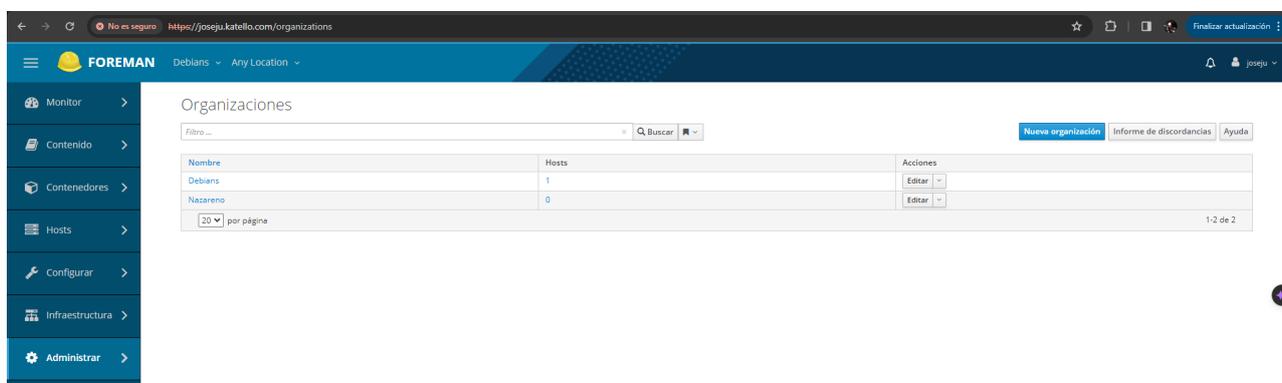
Label *

Description

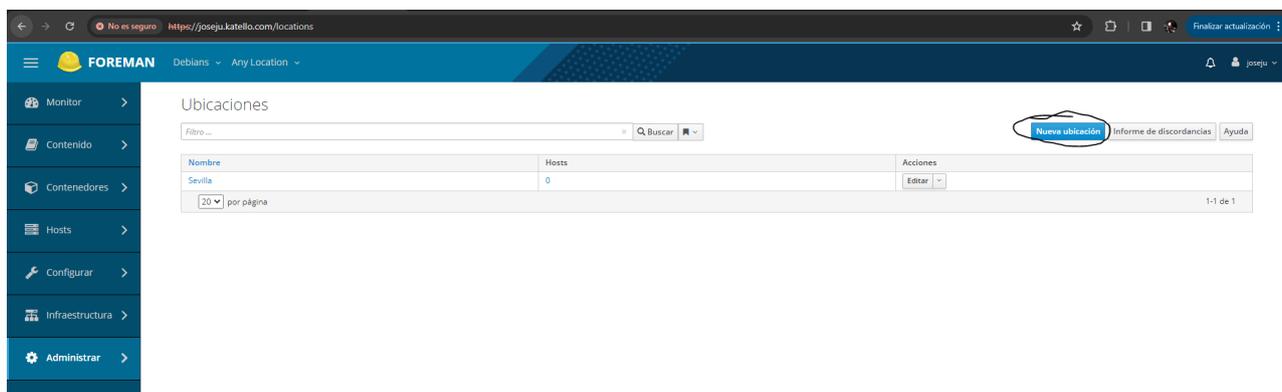
Seguidamente, seleccionamos el hosts que acabamos de registrar:



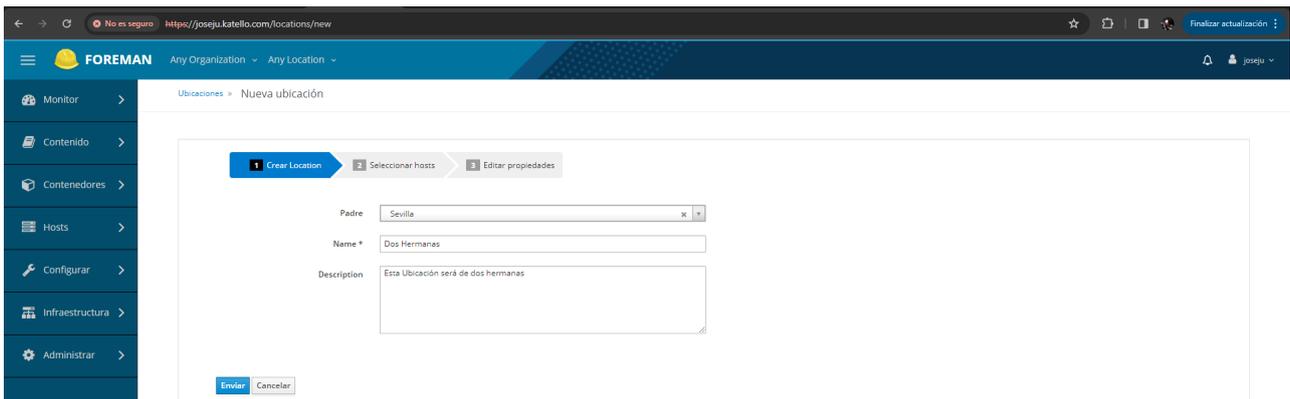
Finalmente hemos creado y registrado el host en nuestra nueva organización:



Con las Ubicaciones, accedemos a Administrar-Ubicaciones-Nueva Ubicación:

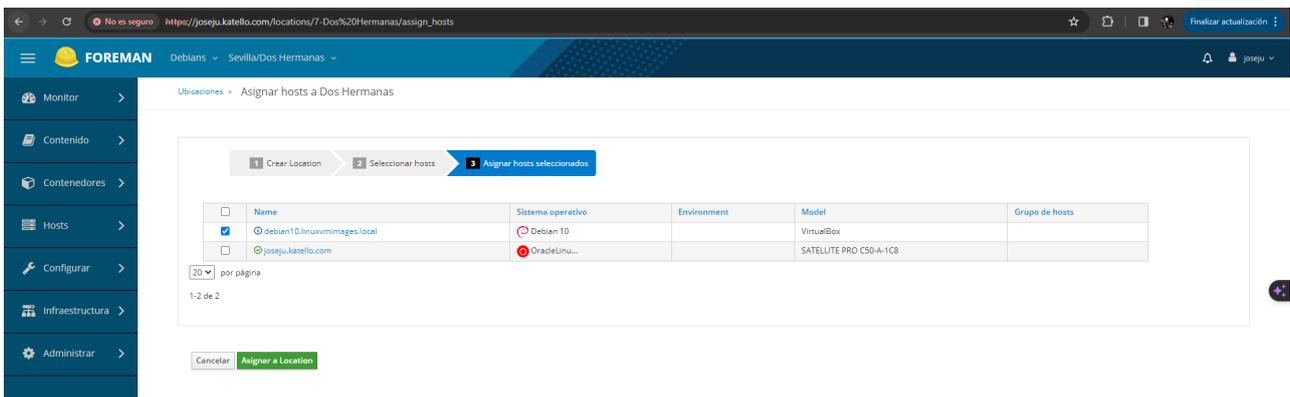


Crearemos nuestra nueva ubicación Dos Hermanas:

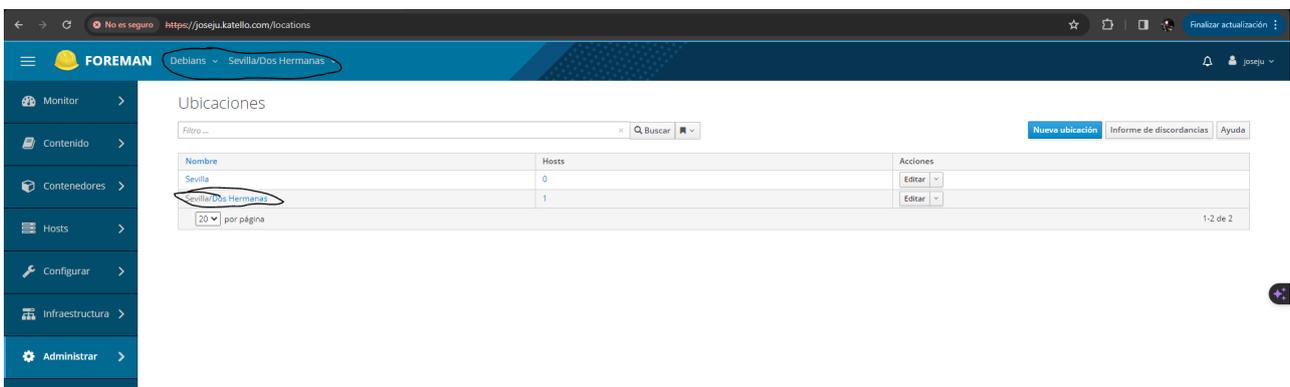


Como padre seleccionamos la ubicación de Sevilla, ya que dos hermanas será la nueva ubicación que crearemos.

Hecho esto, seleccionamos el nuevo host que hemos creado:

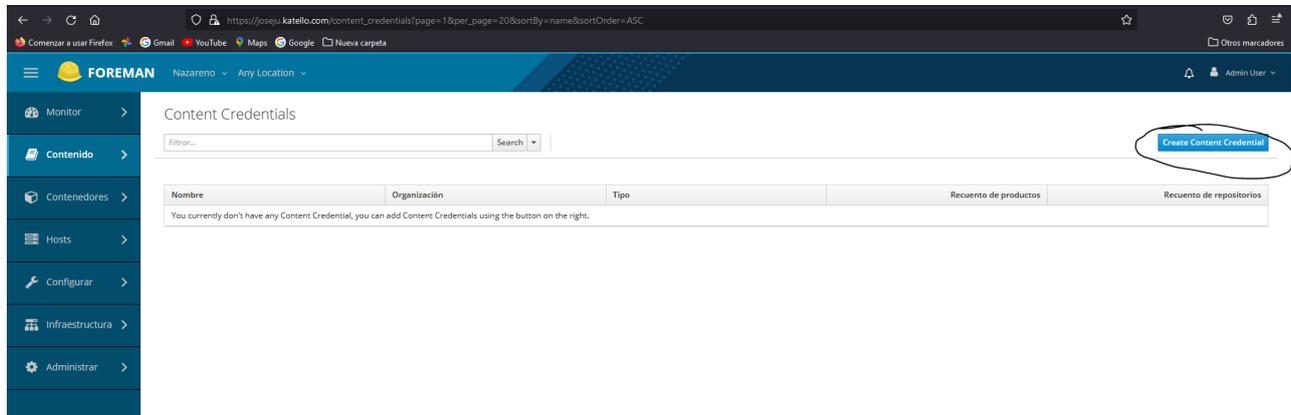


Hecho esto, ya hemos añadido a nuestro hosts `debian.katello.org` a nuestra nueva organización creada Debians y nuestra nueva ubicación Dos Hermanas:



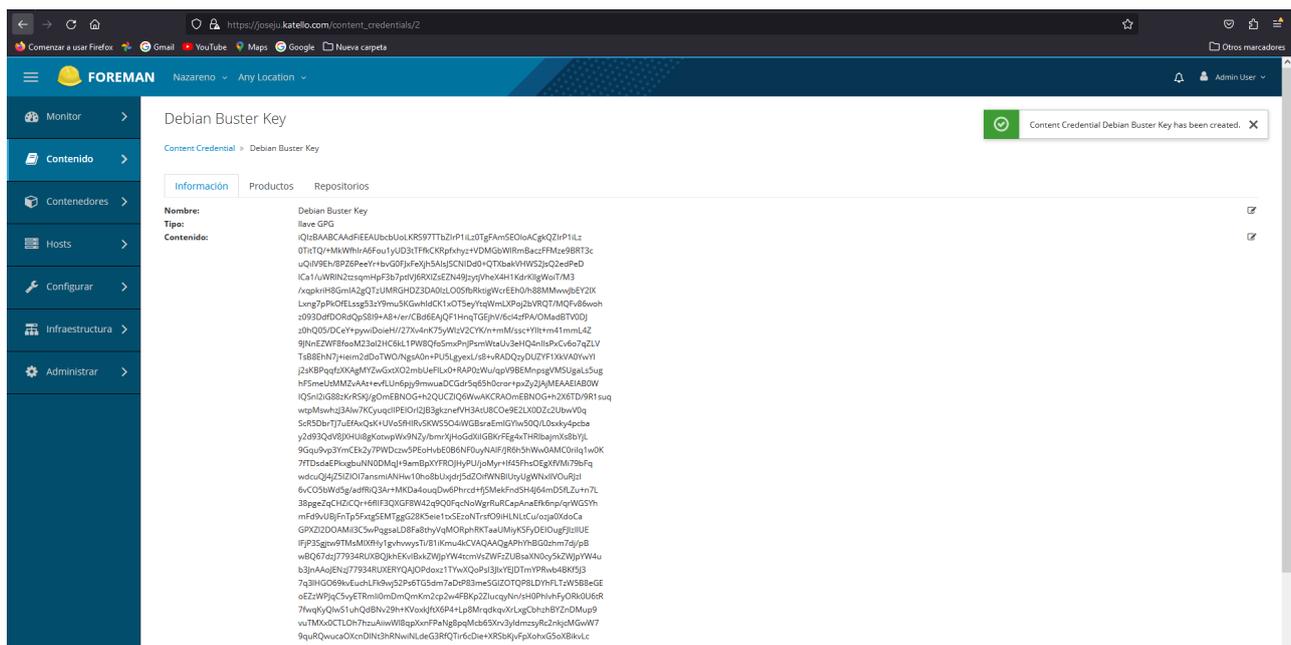
6.5 CREACIÓN DE REPOSITORIOS

Para crear los repositorios para la actualización de paquetería de los servidores debían primero debemos acceder a Contenido-Content Credentials-Create Content Credential:

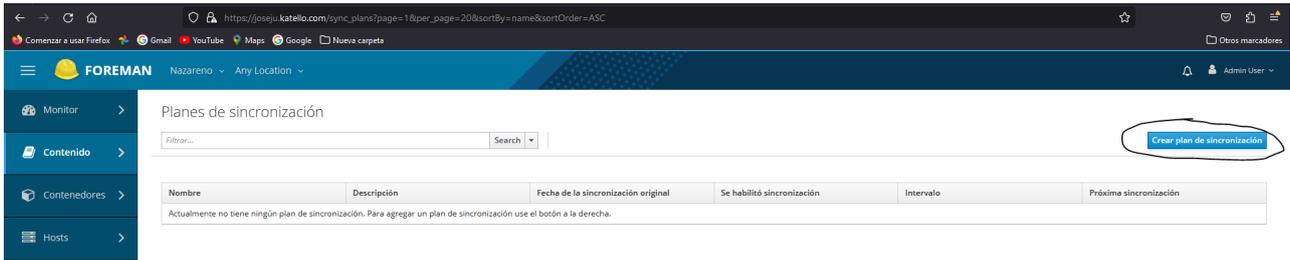


Hecho esto debemos acceder al siguiente enlace

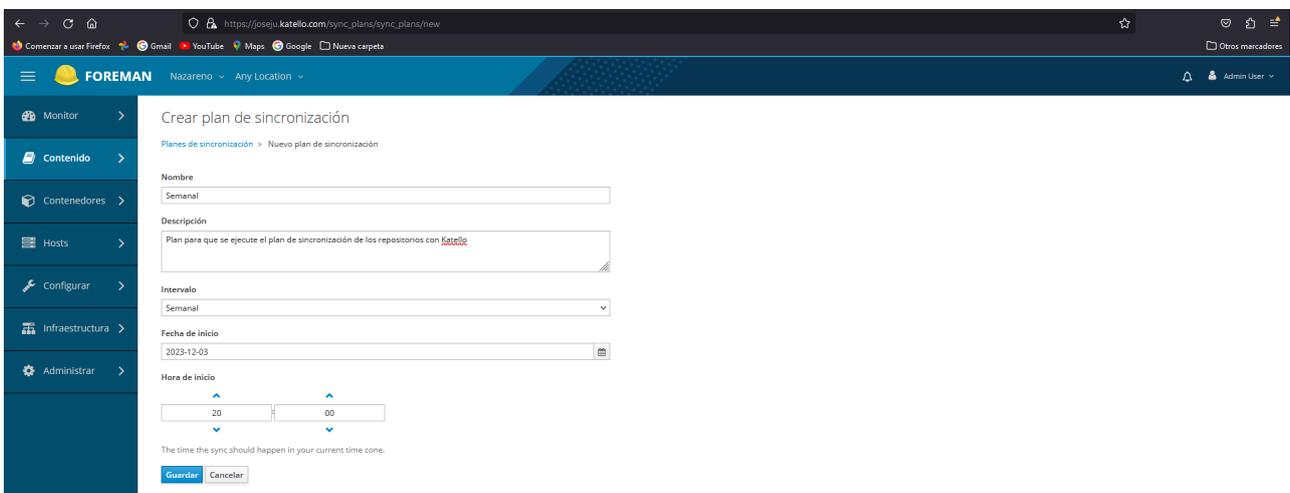
<http://ftp.debian.org/debian/dists/buster/Release.gpg> para obtener la key GPG de debian buster en nuestro caso y la introducimos:



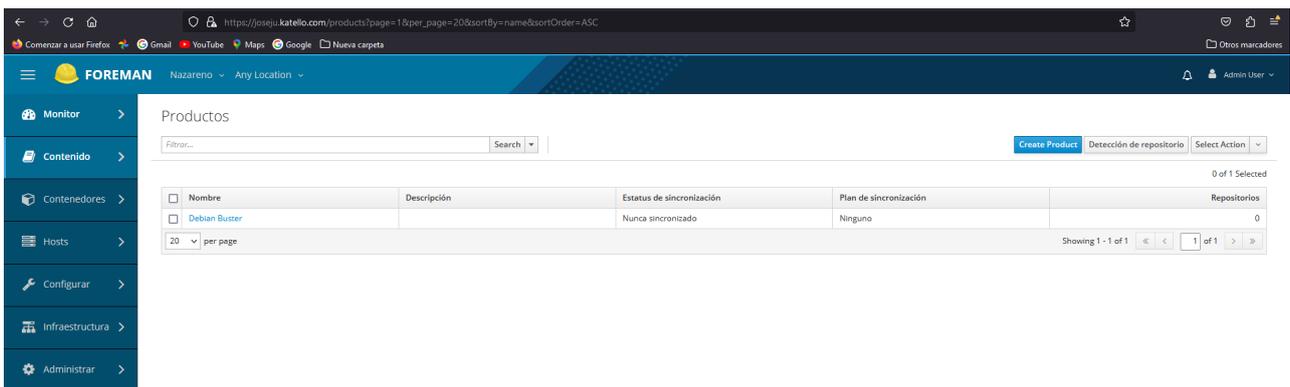
Hecho esto, en la misma página, accedemos Contenido-Planes de Sincronización-Crear plan de sincronización:



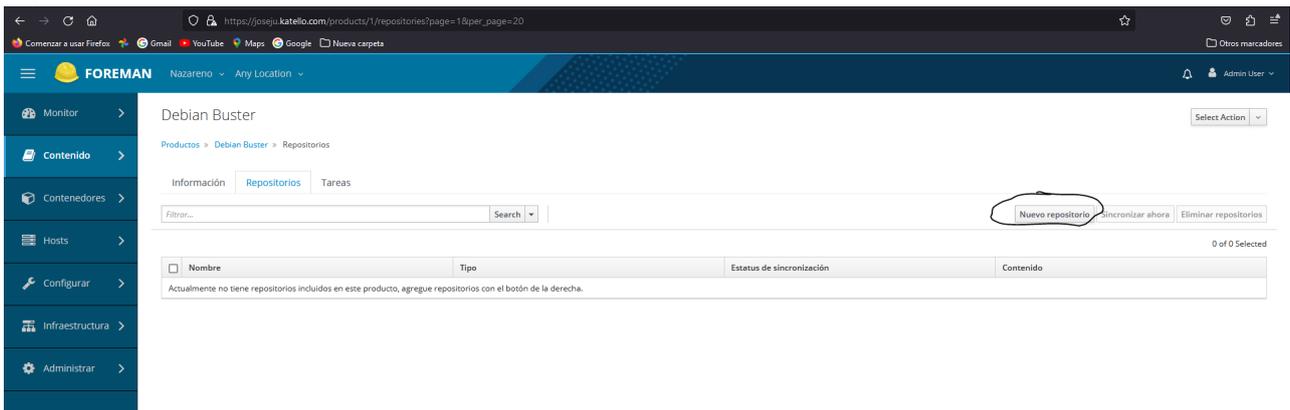
Seguidamente, realizamos las configuraciones oportunas para que este plan de sincronización se ejecuten todos los lunes empezando por el de la semana que viene, día 3/12/2023, a las 20:00PM:



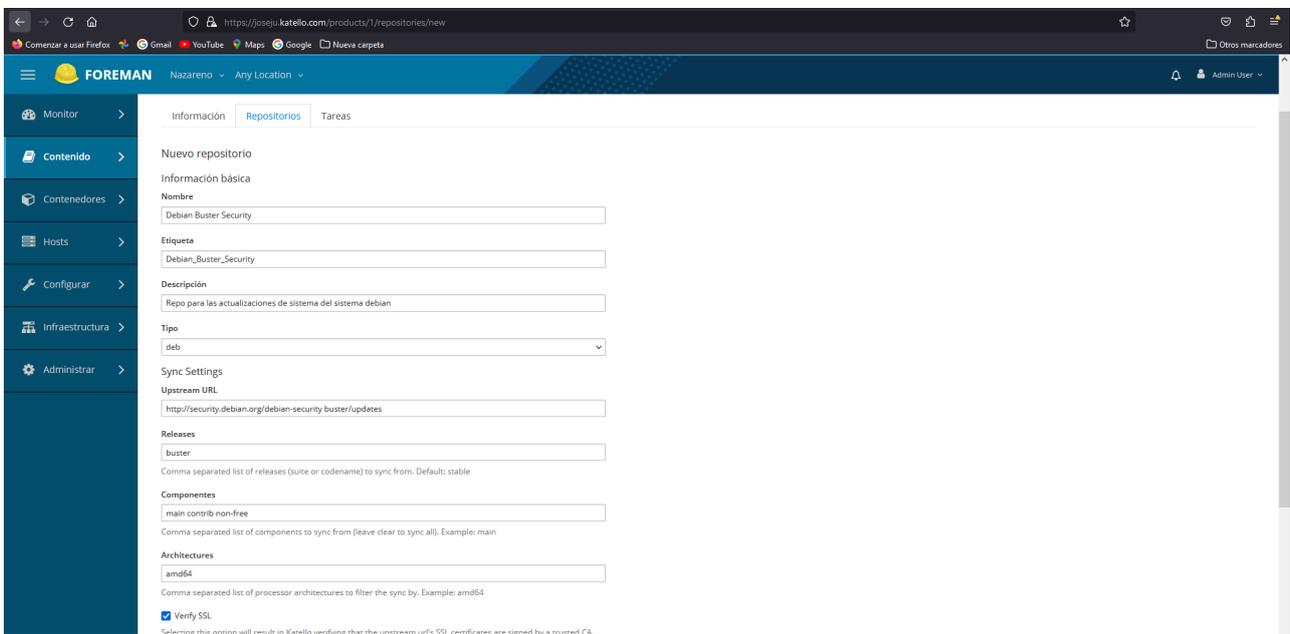
Hecho esto, accedemos a Contenido-Productos-Crear Producto, donde accederemos a Debian Buster:



En donde ya dentro, accederemos al apartado de Repositorios y crearemos un nuevo repositorio:

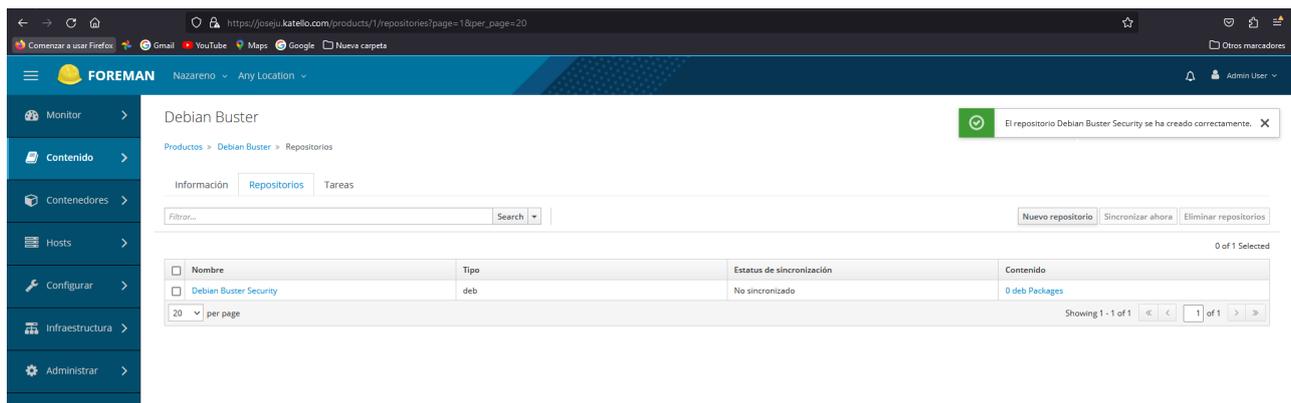


Y configuraremos los siguientes parámetros:



- **Nombre:** Aquí Introduciremos el nombre del repositorio.
- **Descripción:** Breve explicación en donde indicaremos que va a realizar dicho repositorio.
- **Tipo:** Tipo de repositorio que se va a introducir.
- **Upstream URL:** URL del repositorio.
- **Releases:** Versión del repositorio que se va a introducir.
- **Componentes:** Información como si es main, contrib, non-free...
- **Architectures:** Aquí especificamos la arquitectura de nuestra CPU.

Hecho esto, se nos debe haber creado un nuevo repositorio:



El cual se sincronizará el día Lunes día 3 de diciembre de 2023 a las 20:00PM como indicamos en nuestro plan de sincronización semanal.

6.6 IMPORTAR Y EJECUTAR MÓDULOS DE PUPPET

En este punto importaremos y ejecutaremos módulos de puppet, en mi caso los sacaré de dos plugins que tiene katello:

- OpenScan: Este plugin nos permitirá conocer el estado de vulnerabilidad de actual de nuestros servidores.
- Remote Execution: Este plugin nos permitirá ejecutar comandos o scripts desde katello.

6.6.1 OPENSCAN

Para la instalación de dicho plugin, primero updatearemos el sistema e instalaremos el plugin:

```
yum install -y tfm-rubygem-foreman_openscap.noarch rubygem-smart_proxy_openscap.noarch
```

```
Resumen de la transacción
-----
Instalar 2 Paquetes (+4 Paquetes dependientes)

Tamaño total de la descarga: 13 M
Tamaño instalado: 331 M
Downloading packages:
(1/6): openscap-scanner-1.2.17-15.0.3.el7_9.x86_64.rpm
(2/6): rubygem-openscap-0.4.7-1.el7.noarch.rpm
(3/6): rubygem-smart_proxy_openscap-0.6.10-1.el7.noarch.rpm
(4/6): tfm-rubygem-foreman_openscap-0.10.2-1.fm1_19.el7.noarch.rpm
(5/6): openscap-1.2.17-15.0.3.el7_9.x86_64.rpm
(6/6): scap-security-guide-0.1.69-1.0.1.el7_9.noarch.rpm
-----
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando   : openscap-1.2.17-15.0.3.el7_9.x86_64
  Instalando   : rubygem-openscap-0.4.7-1.el7.noarch
  Instalando   : openscap-scanner-1.2.17-15.0.3.el7_9.x86_64
  Instalando   : scap-security-guide-0.1.69-1.0.1.el7_9.noarch
  Instalando   : tfm-rubygem-foreman_openscap-0.10.2-1.fm1_19.el7.noarch
  Instalando   : rubygem-smart_proxy_openscap-0.6.10-1.el7.noarch
  Comprobando  : rubygem-openscap-0.4.7-1.el7.noarch
  Comprobando  : scap-security-guide-0.1.69-1.0.1.el7_9.noarch
  Comprobando  : tfm-rubygem-foreman_openscap-0.10.2-1.fm1_19.el7.noarch
  Comprobando  : openscap-1.2.17-15.0.3.el7_9.x86_64
  Comprobando  : openscap-scanner-1.2.17-15.0.3.el7_9.x86_64
  Comprobando  : rubygem-smart_proxy_openscap-0.6.10-1.el7.noarch

Instalado:
  rubygem-smart_proxy_openscap.noarch 0:0.6.10-1.el7          tfm-rubygem-foreman_openscap.noarch 0:0.10.2-1.fm1_19.el7

Dependencia(s) instalada(s):
  openscap.x86_64 0:1.2.17-15.0.3.el7_9          openscap-scanner.x86_64 0:1.2.17-15.0.3.el7_9          rubygem-openscap.noarch 0:0.4.7-1.el7

!list!
[root@joseju ~]#
```

Seguidamente, migramos y aplicamos los cambios en la BBDD:

```
foreman-rake db:migrate
foreman-rake db:seed
```

```
[root@joseju ~]# foreman-rake db:migrate
/usr/share/foreman/lib/foreman.rb:8: warning: already initialized constant Foreman::UUID_REGEX
/usr/share/foreman/lib/foreman.rb:8: warning: previous definition of UUID_REGEX was here
/usr/share/foreman/lib/core_extensions.rb:182: warning: already initialized constant ActiveSupport::MessageEncryptor::DEFAULT_CIPHER
/opt/the_foreman/tfm-ror51/root/usr/share/gems/gems/activerecord-5.1.6/lib/active_record/message_encryptor.rb:22: warning: previous definition of DEFAULT_CIPHER was here
[root@joseju ~]# foreman-rake db:seed
/usr/share/foreman/lib/foreman.rb:8: warning: already initialized constant Foreman::UUID_REGEX
/usr/share/foreman/lib/foreman.rb:8: warning: previous definition of UUID_REGEX was here
/usr/share/foreman/lib/core_extensions.rb:182: warning: already initialized constant ActiveSupport::MessageEncryptor::DEFAULT_CIPHER
/opt/the_foreman/tfm-ror51/root/usr/share/gems/gems/activerecord-5.1.6/lib/active_record/message_encryptor.rb:22: warning: previous definition of DEFAULT_CIPHER was here
Seeding /usr/share/foreman/db/seeds.d/020-partition_tables_list.rb
Seeding /usr/share/foreman/db/seeds.d/020-permissions_list.rb
Seeding /usr/share/foreman/db/seeds.d/020-provisioning_templates_list.rb
Seeding /usr/share/foreman/db/seeds.d/020-roles_list.rb
Seeding /usr/share/foreman/db/seeds.d/030-auth_sources.rb
Seeding /usr/share/foreman/db/seeds.d/030-permissions.rb
Seeding /usr/share/foreman/db/seeds.d/035-admin.rb
Seeding /usr/share/foreman/db/seeds.d/040-roles.rb
Seeding /usr/share/foreman/db/seeds.d/050-taxonomies.rb
Seeding /usr/share/foreman/db/seeds.d/060-architectures.rb
Seeding /usr/share/foreman/db/seeds.d/070-provisioning_templates.rb
Seeding /usr/share/foreman/db/seeds.d/080-partition_tables.rb
Seeding /usr/share/foreman/db/seeds.d/100-installation_media.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/101-locations.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/102-organizations.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/104-proxy.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/106-mail_notifications.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/107-enable_dynflow.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/108-subscription-bookmarks.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/109-katello-notification-blueprints.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/110-content-view-autopublish.rb
Seeding /usr/share/foreman/db/seeds.d/110-smart_proxy_features.rb
Seeding /usr/share/foreman/db/seeds.d/130-compute_profiles.rb
Seeding /usr/share/foreman/db/seeds.d/150-bookmarks.rb
Seeding /usr/share/foreman/db/seeds.d/160-mail_notifications.rb
Seeding /usr/share/foreman/db/seeds.d/170-notification_blueprints.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman-tasks-0.13.4/db/seeds.d/20-foreman_tasks_permissions.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman-tasks-0.13.4/db/seeds.d/60-dynflow_proxy_feature.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman-tasks-0.13.4/db/seeds.d/61-foreman_tasks_bookmarks.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/katello-3.8.1/db/seeds.d/75-job_templates.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman_openscap-0.10.2/db/seeds.d/75-job_templates.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman_openscap-0.10.2/db/seeds.d/openscap_feature.rb
Seeding /opt/the_foreman/tfm/root/usr/share/gems/gems/foreman_openscap-0.10.2/db/seeds.d/openscap_policy_notification.rb
All seed files executed
[root@joseju ~]#
```

Y reiniciamos el servicio de Apache:

```
systemctl restart httpd
```

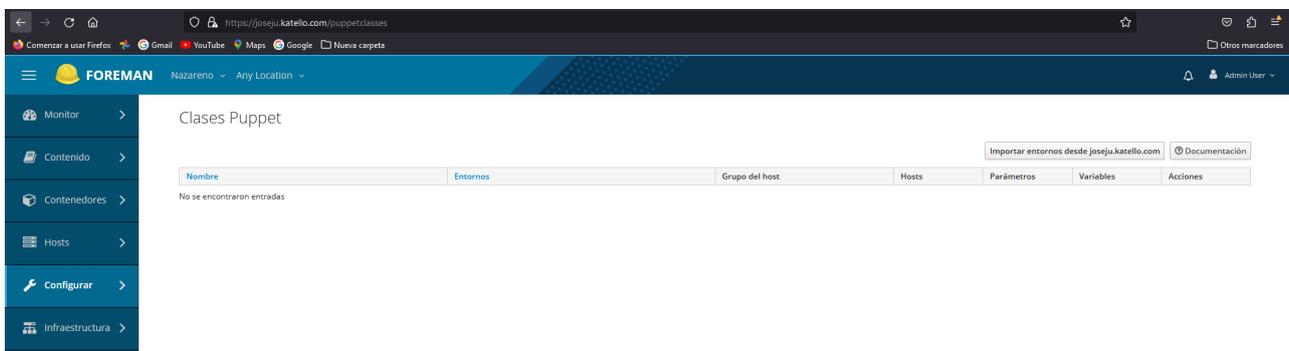
```
[root@joseju ~]# systemctl restart httpd
[root@joseju ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since mar 2023-12-19 18:17:46 CET; 8s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 9747 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
  Process: 12339 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
 Main PID: 9795 (httpd)
   Status: "Processing requests..."
    Tasks: 187
   CGroup: /system.slice/httpd.service
           └─ 9795 /usr/sbin/httpd -DFOREGROUND
              └─ 9799 PassengerHelperAgent
                 └─ 9815 PassengerWatchdog (cleaning up...)
```

Instalamos el siguiente módulo de puppet:

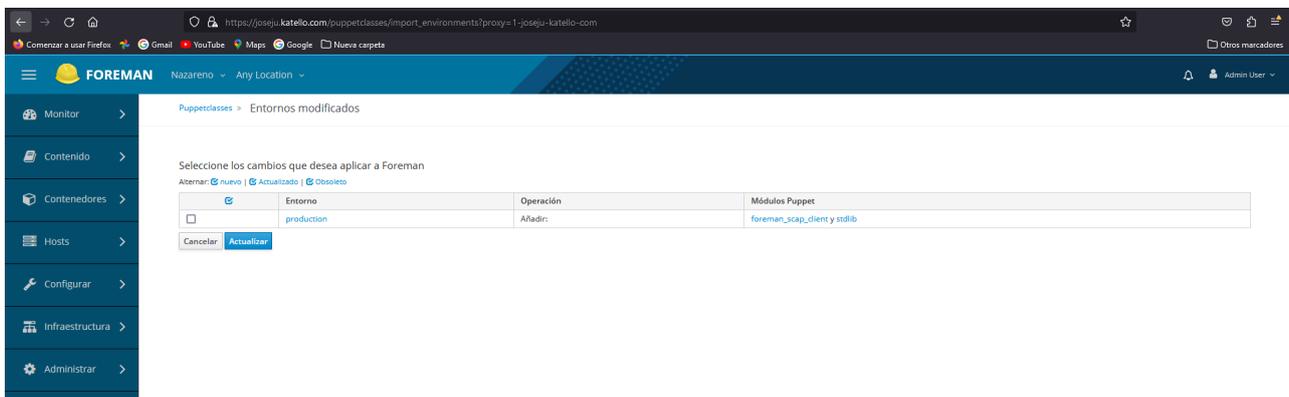
```
puppet module install theforeman-foreman_scap_client
```

```
[root@joseju ~]# puppet module install theforeman-foreman_scap_client
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/modules ...
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
├── theforeman-foreman_scap_client (v1.0.0)
└── puppetlabs-stdlib (v8.6.0)
[root@joseju ~]#
```

Ahora para comprobar que se ha instalado correctamente, debemos entrar en Configurar-Puppet-Clases-Importar Entornos desde joseju.katello.com, de principio no nos debería aparecer nada:



Y al darle a importar nos debería aparecer esto:



Esto quiere decir que el módulo se ha instalado correctamente.

6.6.2 REMOTE EXECUTION

En el paso anterior realizamos una instalación manual de el plugin, en esta la haremos de manera automática para centrarnos en la ejecución del módulo. Ejecutamos el siguiente comando:

```
foreman-installer --enable-foreman-plugin-remote-execution --enable-foreman-proxy-plugin-remote-execution-ssh
```

```
[root@joseju ~]# foreman-installer --enable-foreman-plugin-remote-execution --enable-foreman-proxy-plugin-remote-execution-ssh
Resetting puppet server version param...
Installing Done [100%] [.....]
Success!
* Katello is running at https://joseju.katello.com
* To install an additional Foreman proxy on separate machine continue by running:
    foreman-proxy-certs-generate --foreman-proxy-fqdn "$FOREMAN_PROXY" --certs-tar "/root/$FOREMAN_PROXY-certs.tar"
The full log is at /var/log/foreman-installer/katello.log
[root@joseju ~]#
```

Ahora para la ejecución de este plugin, compartimos nuestra clave pública ssh en el servidor que registramos anteriormente:

```
ssh-copy-id -i
/usr/share/foreman-proxy/.ssh/id_rsa_foreman_proxy.pub
debian@192.168.1.20
```

```
[root@joseju ~]# ssh-copy-id -i /usr/share/foreman-proxy/.ssh/id_rsa_foreman_proxy.pub debian@192.168.1.20
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/usr/share/foreman-proxy/.ssh/id_rsa_foreman_proxy.pub"
The authenticity of host '192.168.1.20 (192.168.1.20)' can't be established.
ECDSA key fingerprint is SHA256:IdRzUpzJr5N0aBwKfxFag0xrZsQ0+kJppoI6yKYM1ac.
ECDSA key fingerprint is MD5:61:64:0b:85:68:c7:49:15:cd:d1:e2:64:d2:bc:94:da.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

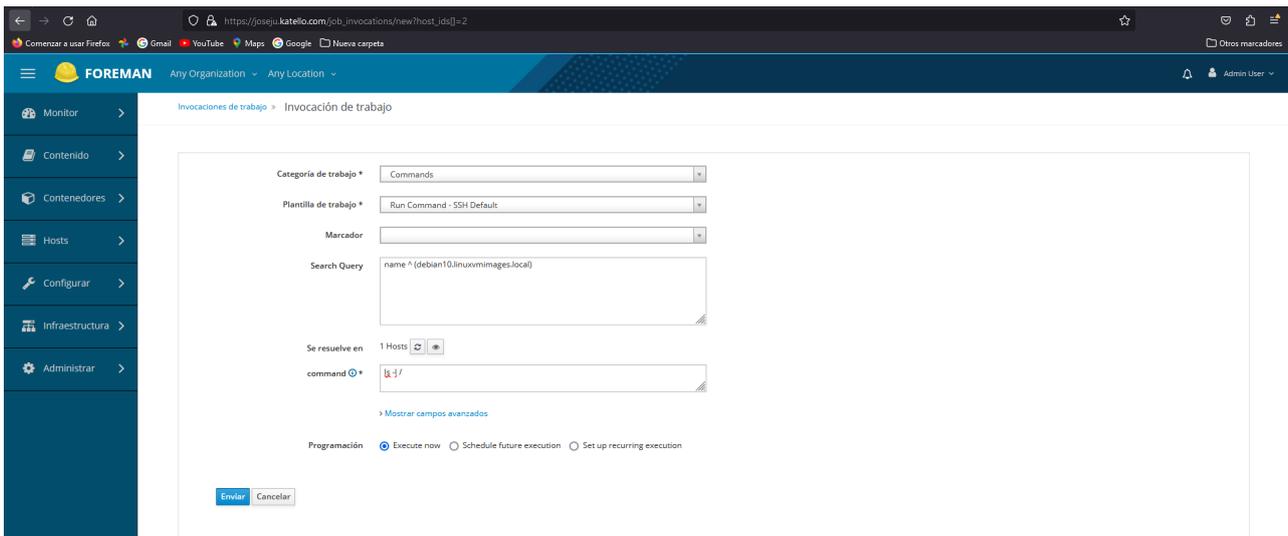
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
+++++
LINUXVMIMAGES.COM
+++++
User Name: debian
Password: debian (sudo su -)
debian@192.168.1.20's password:

Number of key(s) added: 1

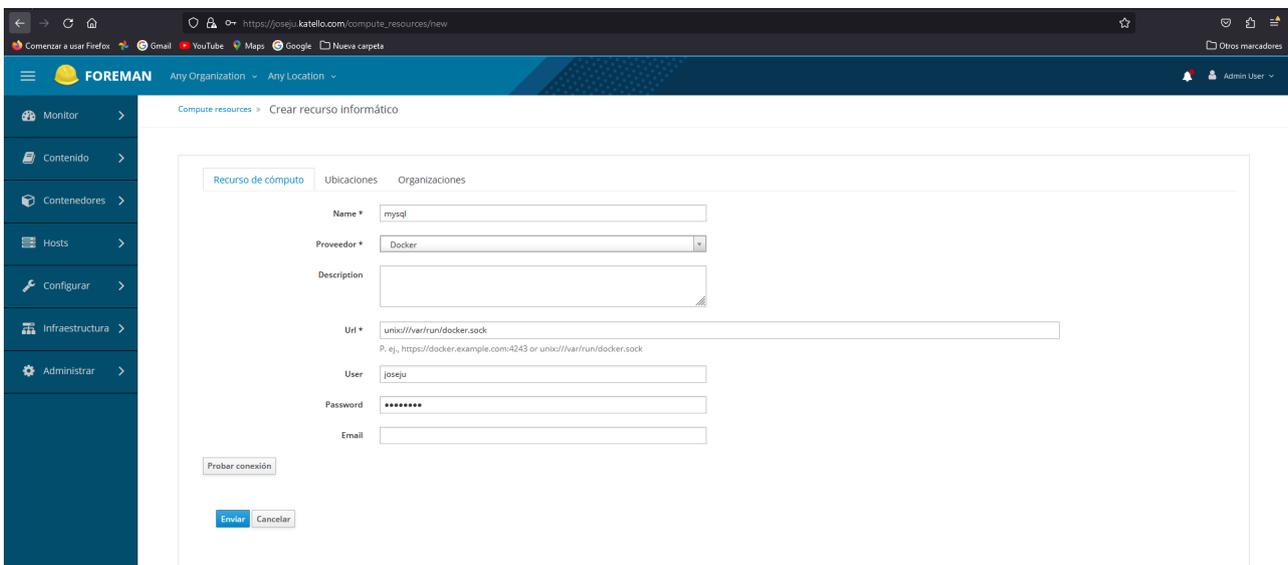
Now try logging into the machine, with: "ssh 'debian@192.168.1.20'"
and check to make sure that only the key(s) you wanted were added.
```

Finalmente realizamos la prueba de ejecución del módulo ejecutando el comando ls -l:



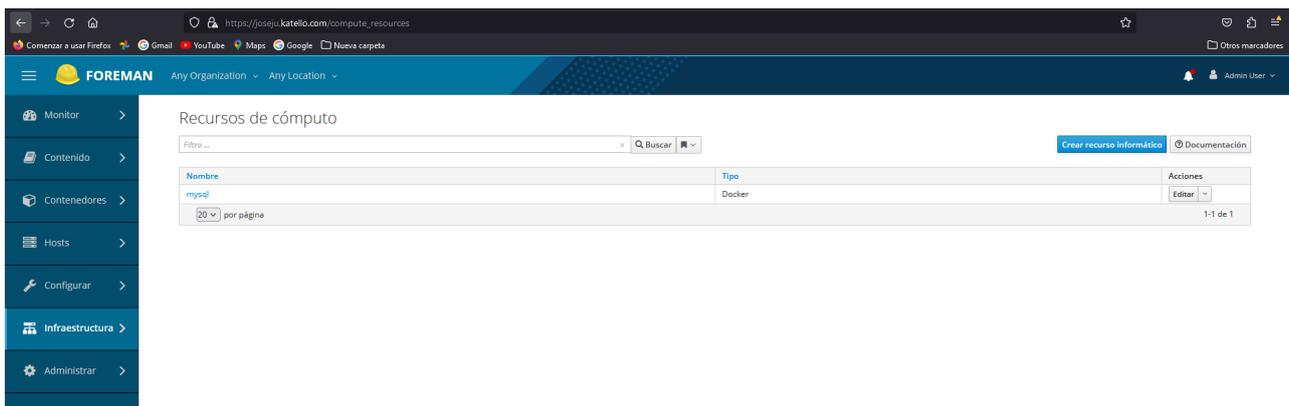
6.7 CREACIÓN DE CONTENEDORES

En katello, también podemos crear contenedores, en mi caso crearé un nuevo contenedor que se llamará mysql donde habrá una base de datos. Para ello accedemos a Contenedores-Create Container:



- Name: Indicamos el nombre del contenedor.
- Proveedor: Indicamos el proveedor que utilizará el contenedor, en nuestro caso es docker.
- URL: La URL para la ejecución de docker.
- User: Nombre del usuario para la ejecución.

Le damos a crear y comprobamos que se ha creado:



7. CONCLUSIONES Y PROPUESTAS

De este proyecto puedo sacar las siguientes conclusiones:

- Debo mejorar en la previsión a la hora de realizar una configuración previa, ya que por esto pude perfectamente no haber conseguido instalar katello.
- También he de decir que esta herramienta es muy extensa pero está muy poco trabajada a mi parecer. Se podría mejorar los errores de repositorios que conllevan a que en la instalación no se instalen bien ciertas dependencias.
- Que solucionen el problema que tienen con la instalación de certificados firmados de lets encrypt.
- También he aprendido bastante en paquetería yum y sistemas basados en centos.

En este proyecto podría llevarme mucho tiempo explicando cada una de las dificultades que me he encontrado, desde que la versión de katello no es la correcta para los requisitos del sistema, errores de dependencias debido a enlaces que ya están deprecados, o incluso en la instalación de ciertos módulos de Puppet, cosas que iré detallando ahora, pero lo que debo de decir es que sin ninguna duda es el trabajo que más me ha costado de todo el ciclo sin ningún tipo de duda.

8. DIFICULTADES QUE SE HAN ENCONTRADO

Voy a detallar los errores que he tenido durante el proyecto:

Errores en los prerequisites del sistema: Este error vino gracias a que hice una mala previsualización del proyecto, con lo que a la hora de realizar la instalación de katello en un Oracle Linux 9 tal como tenía previsto de primeras, me dio error en la ejecución de katello ya que mis recursos no cumplen con los requisitos mínimos para las versiones mas actuales.

La solución por la que opté es instalar katello en una versión mas antigua que corresponda con los recursos que tiene mi máquina.

Errores de dependencias: También he de indicar que a la hora de instalar los paquetes de katello en Oracle Linux estaba constantemente dando error con ciertos paquetes que no se instalaban correctamente, ya que hay algunos repositorios de katello que no están disponibles para este sistema.

La solución por la que opté es instalar katello en un CentOS 7 que es una versión compatible con todos los repositorios y realizar una migración a Oracle Linux 7.9.

Errores de compatibilidad de paquetes: A la vez que me surgían errores de dependencias también me surgían errores de incompatibilidad de versiones.

Opté por la misma solución que arriba.

Errores de Certificado HTTPS: Para que katello tuviera un certificado de seguridad válido y firmado por una CA opté por sacarlo de Let's Encrypts, pero por algún motivo aparente no funciona.

He intentado buscar alguna solución, inclusive fue un problema que tuve en mi empresa de prácticas, pero es un problema que tiene el propio katello.

Errores de Red: A veces Katello tiene una carga de tráfico de red importante, con lo que hizo que la propia conexión de mi casa fallara.

La solución que tuve aquí fue paciencia y poco a poco ir sacandolo.

Errores de documentación: Aquí me desahogo en parte, no viene prácticamente en ningún foro/documentación oficial inclusive que sirva explícitamente, no indican formas de instalarlo en sistemas basados en Centos como pueden ser Oracle Linux, Rocky, AlmaLinux etc, tampoco explican bien que repositorios no están deprecados y cuáles si, e incluso tampoco te viene detallado los requisitos mínimos que necesita el sistema para la instalación de katello.

Sinceramente esta herramienta me parece increíble por la cantidad de posibilidades que te puede ofrecer, pero esta muy mal gestionada por parte de los desarrolladores.

9. BIBLIOGRAFÍA, ENLACES, RESEÑAS...

<https://www.google.es>

<https://www.theforeman.org>

<https://www.gmail.com>

<https://web.whatsapp.com>

<https://www.debian.org>

<https://mirror.centos.org>

<https://www.oracle.org>