**SERVICIO DE CONSULTORÍA PARA HABILITAR PLATAFORMA SOFTWARE RELACIONADA AL MODELO DE INTEGRACIÓN CONTINUA PARA REALIZAR ANÁLISIS ESTÁTICO DEL CÓDIGO FUENTE DE LOS SISTEMAS DEL OSCE BASADO EN TECNOLOGÍA JAVA**

**Orden de Servicio Nº 0000849-2017-OSCE**

**INFORME DE INSTALACIÓN DE LA PLATAFORMA SOFTWARE HABILITADA**

**Elaborado por: Ing. Yhan Fransua Mandros Poblet**

**10 de Julio del 2017**

**HISTORIAL DE REVISIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Descripción** |
| 28/04/2017 | 1.0 | Yhan Fransua Mandros Poblet | Versión inicial |

**Tabla de Contenido**

[**1.** **DIAGRAMA DE ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA SOFTWARE HABILITADA** 4](#_Toc498919140)

[**2.** **INSTRUCCIONES DE INSTALACIÓN Y/O CONFIGURACIÓN DE LOS PRODUCTOS SOFTWARE’S HABILITADOS** 5](#_Toc498919141)

[**2.1.** **Gestor de Contenedores** 5](#_Toc498919142)

[**2.1.1.** **Instalación de docker** 5](#_Toc498919143)

[**2.1.2.** **Instalación de docker-compose** 6](#_Toc498919144)

[**2.2.** **HAProxy 1.7.9** 6](#_Toc498919145)

[**2.2.1.** **Instalación de software base** 6](#_Toc498919146)

[**2.2.2.** **Integración base** 6](#_Toc498919147)

[**2.3.** **SonarQube 6.7** 7](#_Toc498919148)

[**2.3.1.** **Habilitación de base de datos** 7](#_Toc498919149)

[**2.3.2.** **Instalación de software base** 7](#_Toc498919150)

[**2.3.3.** **Habilitación en HAProxy** 7](#_Toc498919151)

[**2.4.** **Apache Subversión 1.9.7** 7](#_Toc498919152)

[**2.4.1.** **Instalación de software base** 7](#_Toc498919153)

[**2.4.2.** **Habilitación en HAProxy** 7](#_Toc498919154)

[**2.5.** **Sonatype Nexus OSS 3.6.1** 7](#_Toc498919155)

[**2.5.1.** **Instalación de software base** 7](#_Toc498919156)

[**2.5.2.** **Habilitación en HAProxy** 7](#_Toc498919157)

[**3.** **INSTRUCCIONES DE OPERACIONES DE LOS PRODUCTOS SOFTWARE’S HABILITADOS** 7](#_Toc498919158)

[**3.1.** **HAProxy 1.7.9** 7](#_Toc498919159)

[**3.1.1.** **Iniciar servicios** 7](#_Toc498919160)

[**3.1.2.** **Detener servicios** 7](#_Toc498919161)

[**3.1.3.** **Visualizar logs** 7](#_Toc498919162)

[**3.1.4.** **Verificación de disponibilidad** 7](#_Toc498919163)

[**3.2.** **SonarQube 6.7** 7](#_Toc498919164)

[**3.2.1.** **Iniciar servicios** 7](#_Toc498919165)

[**3.2.2.** **Detener servicios** 7](#_Toc498919166)

[**3.2.3.** **Visualizar logs** 7](#_Toc498919167)

[**3.2.4.** **Verificación de disponibilidad** 7](#_Toc498919168)

[**3.3.** **Apache Subversión 1.9.7** 7](#_Toc498919169)

[**3.3.1.** **Iniciar servicios** 7](#_Toc498919170)

[**3.3.2.** **Detener servicios** 8](#_Toc498919171)

[**3.3.3.** **Visualizar logs** 8](#_Toc498919172)

[**3.3.4.** **Verificación de disponibilidad** 8](#_Toc498919173)

[**3.4.** **Sonatype Nexus OSS 3.6.1** 8](#_Toc498919174)

[**3.4.1.** **Iniciar servicios** 8](#_Toc498919175)

[**3.4.2.** **Detener servicios** 8](#_Toc498919176)

[**3.4.3.** **Visualizar logs** 8](#_Toc498919177)

[**3.4.4.** **Verificación de disponibilidad** 8](#_Toc498919178)

**INFORME DE INSTALACIÓN DE LA PLATAFORMA SOFTWARE HABILITADA**

1. **DIAGRAMA DE ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA SOFTWARE HABILITADA**





* 1. En cumplimiento con los requerimientos establecidos en los términos de referencia se elaboró los siguientes entregables:
* Plan de Trabajo
* Documento de Diseño Funcional
* Documento de Diseño Técnico
* Componentes de software desarrollados
* Informe de Pruebas
* Manual de Usuario
* Informe transferencia de conocimientos.
* Documento de informe final de actividades.
  1. Se realizaron reuniones de trabajo con personal del área usuaria para precisar los requerimientos planteados en los términos de referencia y comunicar las propuestas de implementación para tales requerimientos.
  2. Los Componentes de Carga y Descarga de archivos han sido desarrollados con las tecnologías solicitadas en los términos de referencia, en reuniones de trabajo se planteó el uso del frameworks y/o librerías como: Spring Boot 1.5, Spring Data 1.2, AngularJS 1.5, bootstrap 3.3, JQuery 1.11, Socket IO 2.0.
  3. El despliegue de los componentes de carga y descarga de archivos han sido desarrollados y desplegados bajo las tecnologías solicitadas en los términos de referencias y las reuniones de trabajo se planteó el uso de las herramientas y productos: NodeJS 7.7.3, Redis 3.2.2, HAProxy 1.5.18 y contenedores Docker.
  4. La aplicación web para la prueba de concepto desarrollada en la presente consultoría ha usado un diseño de presentación derivados de los diseños utilizados actualmente en las aplicaciones web del OSCE al cual se le ha incluido características de uso diseño web adaptativo, precisando que el propósito principal de la consultoría es el desarrollo de componentes de carga y descarga de archivos.
  5. Se desarrolló parte de las funcionalidades en la prueba de Unitarias para se realizaron con JUnit y Selenium para validar los componentes de carga y descarga de archivos.

1. **INSTRUCCIONES DE INSTALACIÓN Y/O CONFIGURACIÓN DE LOS PRODUCTOS SOFTWARE’S HABILITADOS**
   1. **Sistema operativo**
      1. **Agregar repositorios**

* Proporcionar acceso a internet a las siguientes URL’s
* [http://dl.fedoraproject.org/\*](http://dl.fedoraproject.org/*)
* https://mirrors.fedoraproject.org/ \*
* https://download.docker.com/linux/\*
* [https://hub.docker.com/\*](https://hub.docker.com/*)
* [http://\*.docker.io/\*](http://*.docker.io/*)
* [https://\*.docker.io/\*](https://*.docker.io/*)
* [https://github.com/\*](https://github.com/*)
* Agregar repositorio docker

$ sudo yum-config-manager \

--add-repo \

<https://download.docker.com/linux/centos/docker-ce.repo>

* Agregar repositorio EPEL

$ cd /tmp

$ wget http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

$ ls \*.rpm

$ sudo yum install epel-release-latest-7.noarch.rpm

* Actualizar yum package index

$ sudo yum makecache fast

* 1. **Gestor de Contenedores**
     1. **Instalación de docker**
* Listar repositorios y paquetes a utilizar:

$ sudo yum repolist

$ sudo yum --showduplicates list “docker\*” | sort -r

$ sudo yum --showduplicates list docker-ce.x86\_64 | sort -r

* Instalar paquetes mínimos requeridos

$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2

* Desinstalar cualquier versión de docker existente:

$ sudo yum remove docker \

docker-common \

docker-selinux \

docker-engine

**Nota:** docker enviará un fingerprint el cual permite validar con la web del repositorio que se instalara algo correcto

* Instalar última versión de docker-ce

$ sudo yum install docker-ce-<version>-<release>

* Instalar versión especifica de docker-ce-<version>-<release>

$ sudo yum install docker-ce-17.09.0.ce-1.el7.centos.x86\_64

**Nota:** En una instalación con versión puede obviarse el sufijo .x86\_64

* Si hay problemas container-selinux >= 2.9 se debe instalar desde los repositorios de RHEL o como “workaround” descargar estos archivos desde repositorios CentOS:

container-selinux-2.19.2.1.el7.noarch.rpm

policycoreutils-2.5-11.el7\_3.x86\_64.rpm (dependencia de container-selinux)

* Validar que RHEL/CentOS 6 y 7 para los casos de usar 3.10 kernel con los siguientes sistemas de ficheros overlayfs/ext4/xfs algunos contenedores no funcionan. La solución es realizar los siguientes pasos:

$ sudo systemctl stop docker

$ sudo mkdir -p /etc/docker/daemon

$ sudo touch /etc/docker/daemon/daemon.json

$ sudo vi /etc/docker/daemon.json

Adicionar { "storage-driver": "devicemapper" } en el archivo “daemon.json” y guardar.

En el caso que haya un registry a utilizar en las PC clientes agregar

{ "insecure-registries":[”docker-registry:5000”] }

* Configurar para iniciar docker cuando se inicie el sistema operativo

$ sudo systemctl enable docker.service

* Iniciar el servicio docker

$ sudo systemctl start docker

* Verificar el estado del servicio

$ sudo systemctl status docker

* Detener el servicio docker

$ sudo systemctl stop docker

* Validar servicio docker

$ sudo docker run hello-world

* Después de la instalación dar permisos al usuario del sistema para ejecutar docker

$ sudo usermod -aG docker <usuario>

O

$ sudo usermod -a -G docker **$USER**

O

$ sudo usermod -aG docker **operadorsvc**

**Nota:** no aplica de manera inmediata debe volverse a loguear

$ docker run hello-world

* + 1. **Instalación de docker-compose**
* Instalación docker-compose como contenedor:

$ sudo curl -L --fail https://github.com/docker/compose/releases/download/1.16.0/run.sh -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose

$ docker-compose -version

* Desinstalar la instalación docker-compose como contenedor:

$ sudo rm /usr/local/bin/docker-compose

* 1. **Servidor cidfsw1**
     1. **Configurar sistema operativo**
* Crear red docker:

$ sudo docker network create **fsw-red**

* Crear estructura de carpetas requeridas:

$ mkdir -p /fssvc/bin

$ mkdir -p /fssvc/haproxy/1.7.9-alpine/conf

$ mkdir -p /fssvc/haproxy/1.7.9-alpine/logs

$ mkdir -p /fssvc/haproxy/1.7.9-alpine/ssl\_certs

$ mkdir -p /fssvc/sonarqube/6.7-alpine/bundle-plugins

$ mkdir -p /fssvc/sonarqube/6.7-alpine/conf

$ mkdir -p /fssvc/sonarqube/6.7-alpine/data

$ mkdir -p /fssvc/sonarqube/6.7-alpine/extensions

$ mkdir -p /fssvc/sonarqube/6.7-alpine/logs

$ chown operadorsvc -R /fssvc

* + 1. **Instalar HAProxy 1.7.9**
* Crear el archivo “/fssvc/bin/cidfsw-haproxy.yml” con el siguiente contenido

version: '2'

services:

haproxy:

image: haproxy:1.7.9-alpine

mem\_limit: 512MB

volumes:

- /fssvc/haproxy/1.7.9-alpine/conf:/usr/local/etc/haproxy

ports:

- "80:80"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/12-cidfsw-haproxy-redeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO CIDFSW-HAPROXY ...."

docker-compose -f ./cidfsw-haproxy.yml stop

docker-compose -f ./cidfsw-haproxy.yml rm -f

docker-compose -f ./cidfsw-haproxy.yml pull

docker-compose -f ./cidfsw-haproxy.yml up –d

* Crear el archivo “/fssvc/bin/12-cidfsw-haproxy-undeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### DESINSTALANDO CIDFSW-HAPROXY ...."

docker-compose -f ./cidfsw-haproxy.yml stop

docker-compose -f ./cidfsw-haproxy.yml rm -f

docker-compose -f ./cidfsw-haproxy.yml down

* Conceder permisos de ejecución a todos los archivos \*.sh:

$ chmod +x \*.sh

* + 1. **Instalar** **SonarQube 6.7**
* Crear el archivo “/fssvc/bin/cidfsw-sonarqube.yml” con el siguiente contenido

version: '2'

services:

sonarqube:

image: sonarqube:6.7-alpine

volumes:

- /fssvc/sonarqube/6.7-alpine/conf:/opt/sonarqube/conf

- /fssvc/sonarqube/6.7-alpine/data:/opt/sonarqube/data

- /fssvc/sonarqube/6.7-alpine/extensions:/opt/sonarqube/extensions

- /fssvc/sonarqube/6.7-alpine/bundle-plugins:/opt/sonarqube/lib/bundled-plugins

ports:

- "9000:9000"

- "9092:9092"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/22-cidfsw-sonarqube-redeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO CIDFSW-SONARQUBE ...."

docker-compose -f ./cidfsw-sonarqube.yml stop

docker-compose -f ./cidfsw-sonarqube.yml rm -f

docker-compose -f ./cidfsw-sonarqube.yml pull

docker-compose -f ./cidfsw-sonarqube.yml up -d

* Crear el archivo “/fssvc/bin/23-cidfsw-haproxy-undeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### DESINSTALANDO CIDFSW-SONARQUBE ...."

docker-compose -f ./cidfsw-sonarqube.yml stop

docker-compose -f ./cidfsw-sonarqube.yml rm -f

docker-compose -f ./cidfsw-sonarqube.yml down

* 1. **Apache Subversión 1.9.7**
     1. **Instalación de software base**

Abc

* + 1. **Habilitación en HAProxy**

Abc

* 1. **Sonatype Nexus OSS 3.6.1**
     1. **Instalación de software base**
* Crear el archivo “/fssvc/bin/cidfsw-sonarqube.yml” con el siguiente contenido

version: '2'

services:

sonarqube:

image: sonarqube:6.7-alpine

volumes:

- /fssvc/sonarqube/6.7-alpine/conf:/opt/sonarqube/conf

- /fssvc/sonarqube/6.7-alpine/data:/opt/sonarqube/data

- /fssvc/sonarqube/6.7-alpine/extensions:/opt/sonarqube/extensions

- /fssvc/sonarqube/6.7-alpine/bundle-plugins:/opt/sonarqube/lib/bundled-plugins

ports:

- "9000:9000"

- "9092:9092"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/22-cidfsw-sonarqube-redeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO CIDFSW-SONARQUBE ...."

docker-compose -f ./cidfsw-sonarqube.yml stop

docker-compose -f ./cidfsw-sonarqube.yml rm -f

docker-compose -f ./cidfsw-sonarqube.yml pull

docker-compose -f ./cidfsw-sonarqube.yml up -d

* Crear el archivo “/fssvc/bin/23-cidfsw-haproxy-undeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

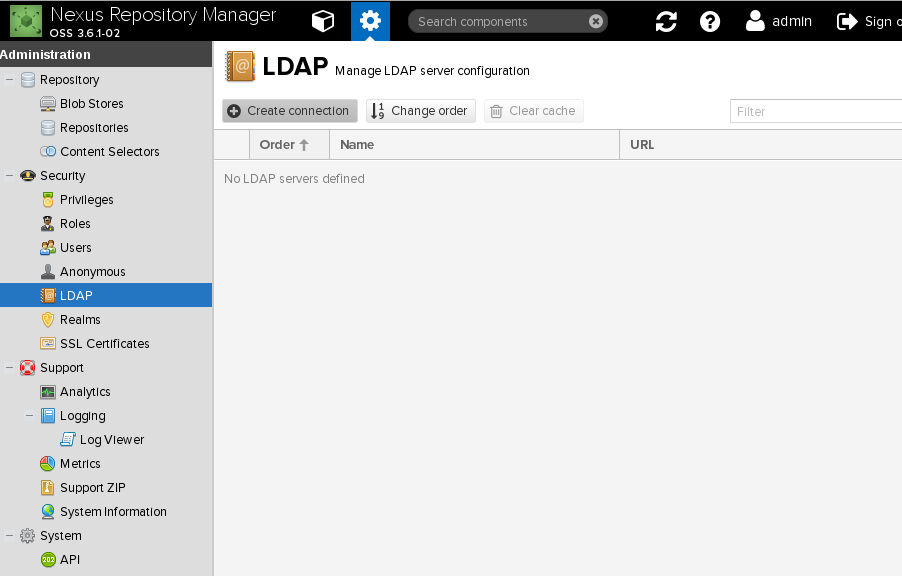
echo "###### DESINSTALANDO CIDFSW-SONARQUBE ...."

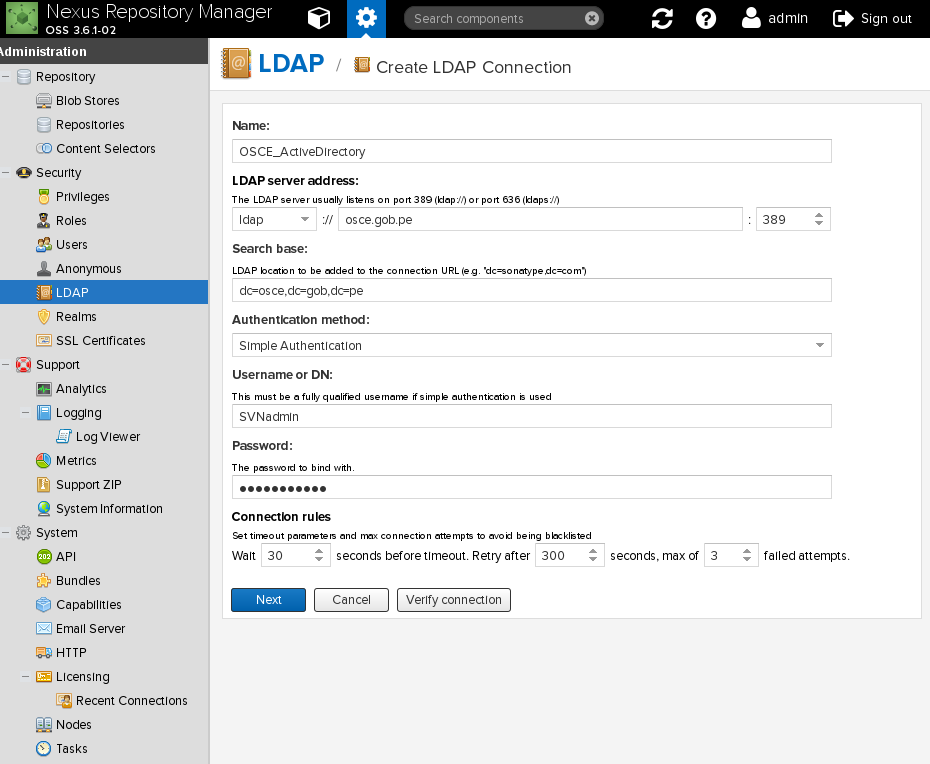
docker-compose -f ./cidfsw-sonarqube.yml stop

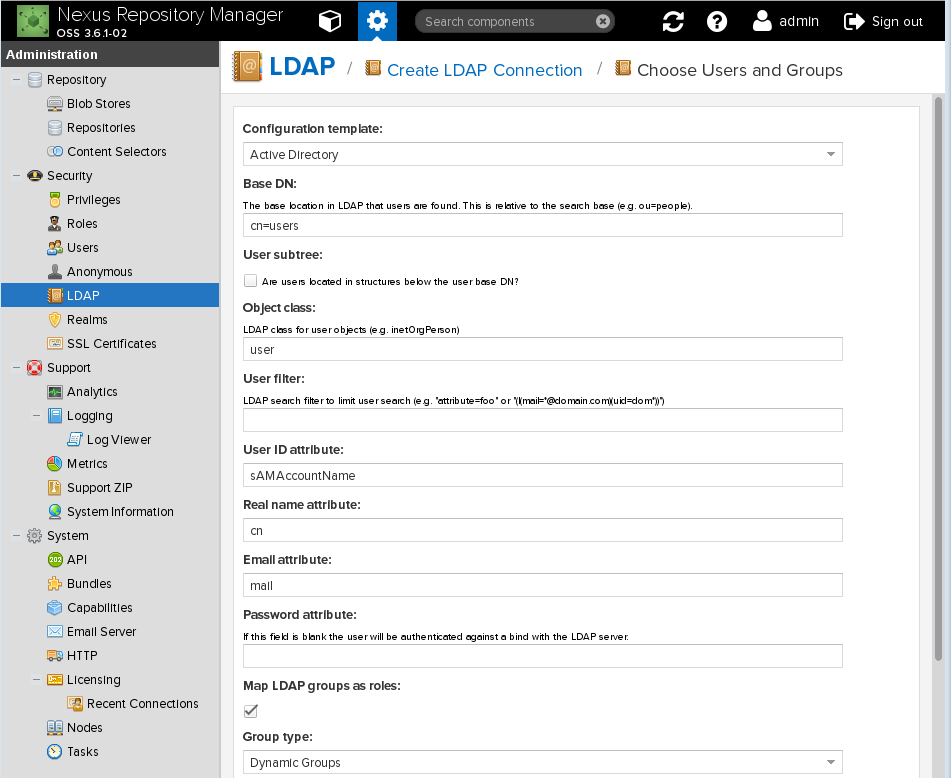
docker-compose -f ./cidfsw-sonarqube.yml rm -f

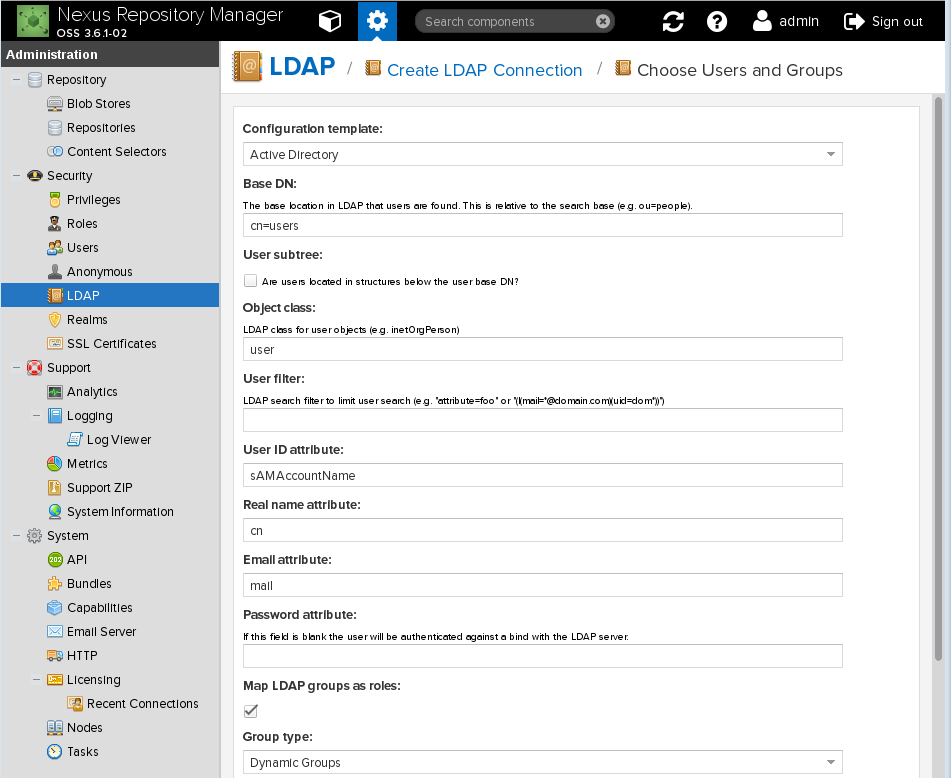
docker-compose -f ./cidfsw-sonarqube.yml down

* Ingresar a consola administrativa <http://127.0.0.1:8081/> con las credenciales admin/admin123
* Crear configuración LDAP/Active Directory

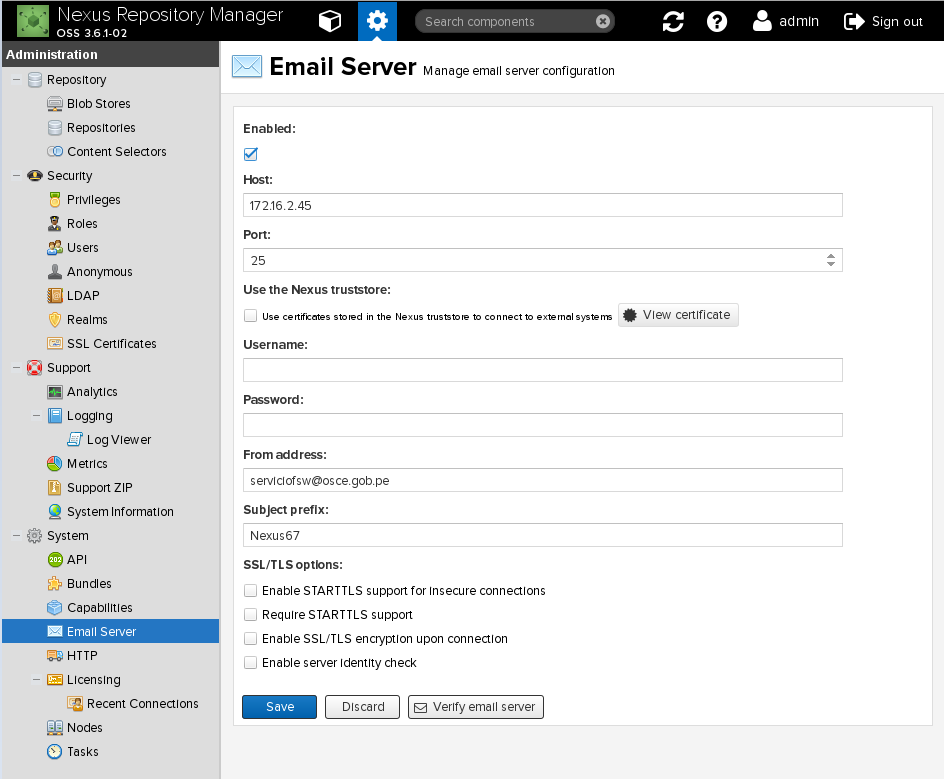


****

****

****

* Configurando correo electrónico

****

* + 1. **Habilitación en HAProxy**
* Abc

1. **INSTRUCCIONES DE OPERACIONES DE LOS PRODUCTOS SOFTWARE’S HABILITADOS**
   1. **HAProxy 1.7.9**
      1. **Iniciar servicios**

$

* + 1. **Detener servicios**

Abc

* + 1. **Visualizar logs**
* Mientras se esta en docker el servicio se puede visualizar de la siguiente manera:
  + 1. **Verificación de disponibilidad**

Abc

* 1. **SonarQube 6.7**
     1. **Iniciar servicios**

Abc

* + 1. **Detener servicios**

Abc

* + 1. **Visualizar logs**

Abc

* + 1. **Verificación de disponibilidad**

Abc

* 1. **Apache Subversión 1.9.7**
     1. **Iniciar servicios**

Abc

* + 1. **Detener servicios**

Abc

* + 1. **Visualizar logs**

Abc

* + 1. **Verificación de disponibilidad**

Abc

* 1. **Sonatype Nexus OSS 3.6.1**
     1. **Iniciar servicios**

Abc

* + 1. **Detener servicios**

Abc

* + 1. **Visualizar logs**

Abc

* + 1. **Verificación de disponibilidad**

Abc