**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**

**13 de Diciembre del 2017**

**HISTORIAL DE REVISIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Descripción** |
| 13/12/2017 | 1.0 | Yhan Fransua Mandros Poblet | Versión inicial |

**Tabla de Contenido**

[**1.** **DIAGRAMA DE ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA SOFTWARE A HABILITADA** 4](#_Toc500659871)

[**1.1.** **SERVICIOS PARA EL ANALISIS ESTATICO DE CODIGO FUENTE** 4](#_Toc500659872)

[**1.2.** **SERVICIOS PARA LA GESTIÓN DE REPOSITORIOS** 5](#_Toc500659873)

[**2.** **INSTRUCCIONES DE INSTALACIÓN Y/O CONFIGURACIÓN DE LOS PRODUCTOS SOFTWARE’S HABILITADOS** 6](#_Toc500659874)

[**2.1.** **INSTALACIÓN BASE** 6](#_Toc500659875)

[**2.1.1.** **Agregar repositorios a sistema operativo** 6](#_Toc500659876)

[**2.1.2.** **Instalación de docker** 6](#_Toc500659877)

[**2.1.3.** **Instalación de docker-compose** 7](#_Toc500659878)

[**2.2.** **SERVIDOR CIDFSW1** 7](#_Toc500659879)

[**2.2.1.** **Configurar sistema operativo** 7](#_Toc500659880)

[**2.2.2.** **Instalar HAProxy 1.7.9** 8](#_Toc500659881)

[**2.2.3.** **Instalar SonarQube 6.7** 9](#_Toc500659882)

[**2.3.** **SERVIDOR REPFSW1** 11](#_Toc500659883)

[**2.3.1.** **Configurar sistema operativo** 11](#_Toc500659884)

[**2.3.2.** **Instalar HAProxy 1.7.9** 11](#_Toc500659885)

[**2.3.3.** **Instalar Subversion 1.9.7** 12](#_Toc500659886)

[**2.3.4.** **Instalar Sonatype Nexus OSS 3.6.2** 13](#_Toc500659887)

[**3.** **INSTRUCCIONES DE OPERACIONES DE LOS PRODUCTOS SOFTWARE’S HABILITADOS** 17](#_Toc500659888)

[**3.1.** **SERVIDOR CIDFSW1** 17](#_Toc500659889)

[**3.1.1.** **Operaciones con Contenedores** 17](#_Toc500659890)

[**3.1.2.** **Operaciones con HAProxy 1.7.9** 17](#_Toc500659891)

[**3.1.3.** **Operaciones con SonarQube 6.7** 17](#_Toc500659892)

[**3.2.** **SERVIDOR REPFSW1** 18](#_Toc500659893)

[**3.2.1.** **Operaciones con Contenedores** 18](#_Toc500659894)

[**3.2.2.** **Operaciones con HAProxy 1.7.9** 18](#_Toc500659895)

[**3.2.3.** **Operaciones con Subversion 1.9.7** 18](#_Toc500659896)

[**3.2.4.** **Operaciones con Nexus OSS 3.6.2** 19](#_Toc500659897)

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

1. **DIAGRAMA DE ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA SOFTWARE A HABILITADA** 
   1. **SERVICIOS PARA EL ANALISIS ESTATICO DE CODIGO FUENTE**

El servidor de análisis estático de código fuente se usara el siguiente diagrama de despliegue:



La estructura de directorios a utilizar:

|  |  |
| --- | --- |
| **Directorio** | **Descripción** |
| /fssvc/bin | Contiene los archivos de comandos para crear e iniciar y detener los contenedores docker a través del cual se ejecutan los servicios HAProxy y SonarQube |
| /fssvc/haproxy/1.7.9-alpine/conf | Usado para almacenar las configuraciones que utiliza HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/haproxy/1.7.9-alpine/logs | Usado para almacenar los archivos log’s que genera HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/haproxy/1.7.9-alpine/ssl\_certs | Usado para almacenar los certificados SSL que utiliza HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/sonarqube/6.7-alpine/conf | Usado para almacenar las configuraciones que utiliza HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/sonarqube/6.7-alpine/data | Usado para almacenar la data que se gestiona en SonarQube |
| /fssvc/sonarqube/6.7-alpine/extensions | Usado para almacenar la data que colocar las extensiones que se requieren agregar a SonarQube en su versión docker 6.7-alpine |
| /fssvc/sonarqube/6.7-alpine/bundle-plugins | Contiene los archivos que utilizara SonarQube para habilitar diferentes funcionalidades, sea que estos archivos jar sean copiados directamente o instalados desde la consola de administración de SonarQube en su versión docker 6.7-alpine. |
| /fssvc/sonarqube/6.7-alpine/logs | Usado para almacenar los archivos log’s que genera SonarQube en su versión docker 6.7-alpine. |

* 1. **SERVICIOS PARA LA GESTIÓN DE REPOSITORIOS**

El servidor de repositorios usara el siguiente diagrama de despliegue:



La estructura de directorios a utilizar:

|  |  |
| --- | --- |
| **Directorio** | **Descripción** |
| /fssvc/bin | Contiene los archivos de comandos para crear e iniciar y detener los contenedores docker a través del cual se ejecutan los servicios HAProxy y SonarQube |
| /fssvc/bin/image-svn | Contiene el archivo dockerfile y la configuración inicial con la cual se crea la imagen docker personalizada de subversión. |
| /fssvc/haproxy/1.7.9-alpine/conf | Usado para almacenar las configuraciones que utiliza HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/haproxy/1.7.9-alpine/logs | Usado para almacenar los archivos log’s que genera HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/haproxy/1.7.9-alpine/ssl\_certs | Usado para almacenar los certificados SSL que utiliza HAProxy en su versión docker 1.7.9-alpine. |
| /fssvc/nexus/3.6.2/data | Usado para almacenar los archivos log’s que gestiona Nexus en su versión docker 3.6.2. |
| /fssvc/nexus/3.6.2/logs | Usado para almacenar los archivos log’s que genera Nexus en su versión docker 3.6.2. |
| /fssvc/svn/1.9.7/httpd-conf | Usado para almacenar los archivos de configuración que usa el Apache HHTP Server donde se despliega el servicio SVN. En estos archivos de configuración se realiza la configuración de LDAP |
| /fssvc/svn/1.9.7/httpd-logs | Usado para almacenar los archivos log’s que genera el Apache HTTP Server que está ejecutando el servicio SVN. |
| /fssvc/svn/1.9.7/httpd-www | Usado para publicar los archivos web en Apache HTTP Server. |
| /fssvc/svn/1.9.7/svn-conf | Usado para almacenar los archivos de configuración del módulo SVN que está desplegado sobre Apache HTTP Server. En este directorio se encuentra el archivo de control de accesos a los repositorios creados en SVN |
| /fssvc/svn/1.9.7/svn-data | Usado para almacenar los repositorios gestionados por el módulo SVN que está desplegado sobre Apache HTTP Server. |

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

Para habilitar los software’s representados en los diagramas de arquitectura anteriores se deberá realizar las siguientes tareas:

* 1. **INSTALACIÓN BASE**
     1. **Agregar repositorios a sistema operativo**
* 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. **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.

* 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:

$ sudo chown operadorsvc -R /fssvc

$ mkdir -p /fssvc/bin

$ mkdir -p /fssvc/haproxy/1.7.9-alpine/{conf, logs,ssl\_certs}

$ mkdir -p /fssvc/sonarqube/6.7-alpine/{bundle-plugins,conf,data,extensions,logs}

$ chmod o=wrx -R /fssvc

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

version: '2'

services:

cidfsw-haproxy1:

image: haproxy:1.7.9-alpine

container\_name: cidfsw-haproxy1

environment:

- NAME=name\_cidfsw-haproxy1

volumes:

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

ports:

- "80:80"

- "443:443"

expose:

- "80"

- "443"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/11-cidfsw-haproxy-images.sh” con el siguiente contenido:

#!/bin/sh

clear

echo ""

echo ""

echo "###### Construyendo CIDFSW-HAPROXY ...."

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

* 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/13-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:

$ sudo chmod +x /fssvc/bin/\*.sh

* Para instalar por primera vez las imágenes docker a utilizar ejecutar:

$ /fssvc/bin/11-cidfsw-haproxy-images.sh

* Para iniciar el servicio ejecutar:

$ /fssvc/bin/12-cidfsw-haproxy-redeploy.sh

* Para acceder al servicio dirigirse a URL <http://192.168.32.40> y habrá un servidor respondiendo.
  + 1. **Instalar SonarQube 6.7**
* Crear el archivo “/fssvc/bin/cidfsw-sonarqube.yml” con el siguiente contenido

version: '2'

services:

cidfsw-sonarqube1:

image: sonarqube:6.7-alpine

container\_name: cidfsw-sonarqube1

environment:

- NAME=svr\_cidfsw\_sonarqube1

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

expose:

- "9000"

- "9092"

ports:

- "9000:9000"

- "9092:9092"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/21-cidfsw-sonarqube-images.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo ""

echo "###### Construyendo CIDFSW-SONARQUBE ...."

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

* 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-sonarqube-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

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

$ chmod +x /fssvc/bin/\*.sh

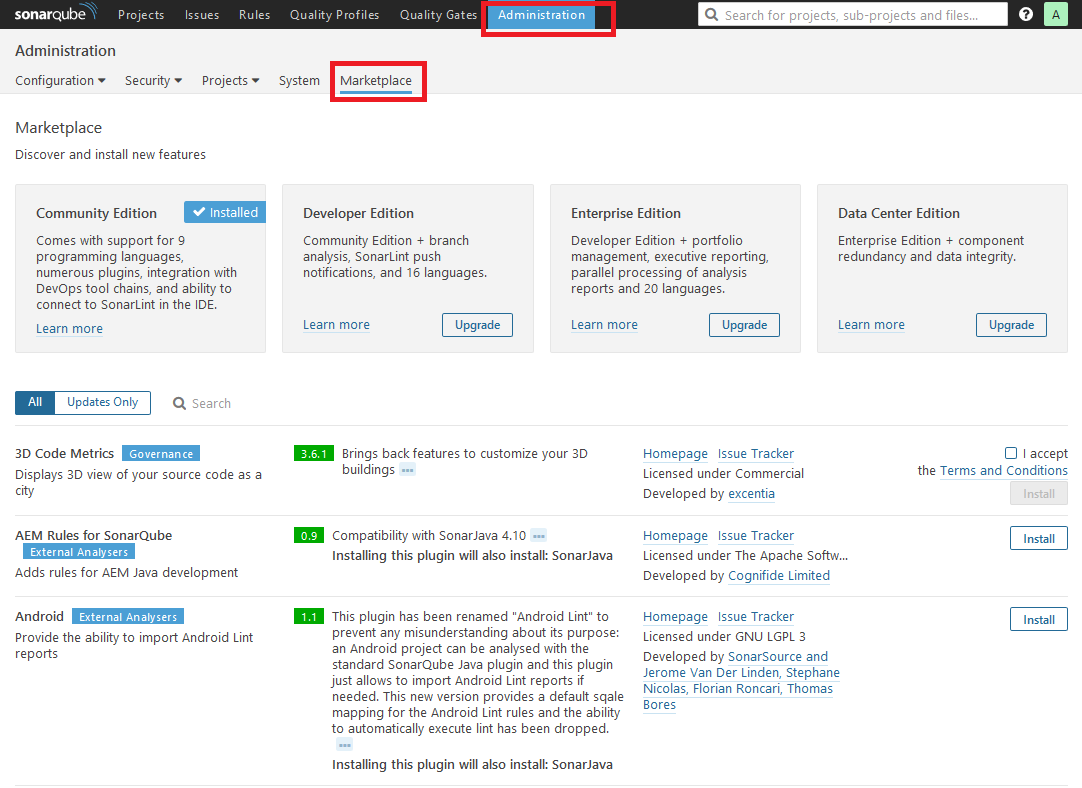
* Para instalar por primera vez las imágenes docker a utilizar ejecutar:

$ /fssvc/bin/21-cidfsw-sonarqube-images.sh

* Para iniciar el servicio ejecutar:

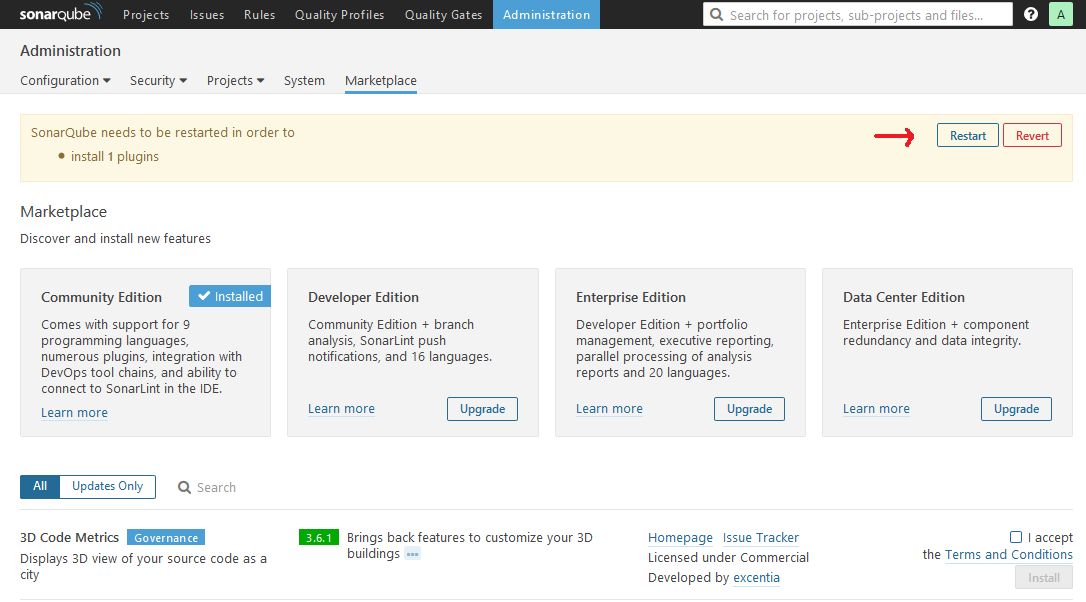
$ /fssvc/bin/22-cidfsw-sonarqube-redeploy.sh

* Para acceder al servicio dirigirse a URL <http://192.168.32.40/sonarqube67> e ingresar las credenciales de un usuario con privilegios de administración.
* Habilitado el servicio agregar plugins a SonarQube a través del siguiente formulario Marketplace



Dar **Install** a los siguientes plugins y luego dar clic a la opción reiniciar:

* + Bitbucket Authentication for SonarQube
  + CSS / SCSS / Less
  + Checkstyle
  + Code Smells
  + GitHub
  + GitLab
  + LDAP
  + SonarC#
  + SonarJS
  + SonarJava
  + SonarPHP
  + SonarPython
  + SonarQube :: Plugins :: SCM :: Git
  + SonarQube :: Plugins :: SCM :: SVN
  + SonarXML
  + Sonargraph
  + Sonargraph Integration
  + Spanish Pack
  + Web
  + jDepend



* 1. **SERVIDOR REPFSW1**
     1. **Configurar sistema operativo**
* Crear la red docker:

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

* Crear estructura de carpetas requeridas:

$ sudo chown operadorsvc -R /fssvc

$ mkdir -p /fssvc/bin

$ mkdir -p /fssvc/haproxy/1.7.9-alpine/{conf,logs,ssl\_certs}

$ mkdir -p /fssvc/nexus/3.6.2/{data,logs}

$ mkdir -p /fssvc/svn/1.9.7/{httpd-conf,httpd-logs,httpd-www,svn-conf,svn-data}

$ sudo chown 200:200 -R /fssvc/nexus/3.6.2/

**Nota:** el servicio de Nexus que corre dentro del contenedor docker usa el UID 200

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

version: '2'

services:

repfsw-haproxy1:

image: haproxy:1.7.9-alpine

container\_name: repfsw-haproxy1

environment:

- NAME=svr\_repfsw-haproxy1

volumes:

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

ports:

- "80:80"

- "443:443"

expose:

- "80"

- "443"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/11-repfsw-haproxy-images.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo ""

echo "###### Construyendo REPFSW-HAPROXY ...."

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

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

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO REPFSW-HAPROXY ...."

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

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

#docker-compose -f ./repfsw-haproxy.yml pull

docker-compose -f ./repfsw-haproxy.yml up -d

* Crear el archivo “/fssvc/bin/13-repfsw-haproxy-undeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### DESINSTALANDO REPFSW-HAPROXY ...."

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

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

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

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

$ chmod +x /fssvc/bin/\*.sh

* Para instalar por primera vez las imágenes docker a utilizar ejecutar:

$ /fssvc/bin/11-repfsw-haproxy-images.sh

* Para iniciar el servicio ejecutar:

$ /fssvc/bin/12-repfsw-haproxy-redeploy.sh

* Para acceder al servicio dirigirse a URL <http://192.168.32.40> y habrá un servidor respondiendo.
  + 1. **Instalar Subversion 1.9.7**
* Crear el archivo “/fssvc/bin/repfsw-svn.yml” con el siguiente contenido

version: '2'

services:

repfsw-svn1:

image: apache2-subversion:1.9.7

container\_name: repfsw-svn1

environment:

- NAME=svr\_repfsw\_svn1

volumes:

- /fssvc/svn/1.9.7/httpd-www:/home/httpd/www

- /fssvc/svn/1.9.7/httpd-conf:/home/httpd/conf

- /fssvc/svn/1.9.7/httpd-logs:/home/httpd/logs

- /fssvc/svn/1.9.7/svn-conf:/home/svn/conf

- /fssvc/svn/1.9.7/svn-data:/home/svn/data

ports:

- "9480:80"

- "9443:443"

expose:

- "80"

- "443"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/21-repfsw-svn-images.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo ""

echo "###### Construyendo REPFSW-SVN ...."

cd image-svn

docker build . -t apache2-subversion:1.9.7

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

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO REPFSW-SVN ...."

docker-compose -f ./repfsw-svn.yml stop

docker-compose -f ./repfsw-svn.yml rm -f

#docker-compose -f ./repfsw-svn.yml pull

docker-compose -f ./repfsw-svn.yml up -d

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

#!/bin/sh

clear

echo ""

echo "###### DESINSTALANDO REPFSW-SVN ...."

docker-compose -f ./repfsw-svn.yml stop

docker-compose -f ./repfsw-svn.yml rm -f

docker-compose -f ./repfsw-svn.yml down

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

$ chmod +x /fssvc/bin/\*.sh

* Para instalar por primera vez las imágenes docker a utilizar ejecutar:

$ /fssvc/bin/21-repfsw-svn-images.sh

* Para iniciar el servicio ejecutar:

$ “/fssvc/bin/22-repfsw-svn-redeploy.sh

* Para acceder al servicio dirigirse a URL <http://192.168.32.40> y habrá un servidor respondiendo
* Ingresar a consola administrativa <http://192.168.32.40/svn> e ingresar las credenciales de un usuario con acceso a los repositorios SVN.
  + 1. **Instalar Sonatype Nexus OSS 3.6.2**
* Crear el archivo “/fssvc/bin/repfsw-nexus.yml” con el siguiente contenido

version: '2'

services:

repfsw-nexus1:

image: sonatype/nexus3:3.6.2

container\_name: repfsw-nexus1

environment:

- NAME=svr\_repfsw\_nexus1

- NEXUS\_CONTEXT=nexus3

- INSTALL4J\_ADD\_VM\_PARAMS=-Xms2g -Xmx2g -XX:MaxDirectMemorySize=3g

volumes:

- /fssvc/nexus/3.6.2/data:/nexus-data

ports:

- "8081:8081"

expose:

- "8081"

networks:

- fsw-red

networks:

fsw-red:

external: true

* Crear el archivo “/fssvc/bin/31-repfsw-nexus-images.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo ""

echo "###### Construyendo REPFSW-NEXUS ...."

docker-compose -f ./repfsw-nexus.yml pull

* Crear el archivo “/fssvc/bin/32-repfsw-nexus-redeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### REDESPLEGANDO REPFSW-NEXUS ...."

docker-compose -f ./repfsw-nexus.yml stop

docker-compose -f ./repfsw-nexus.yml rm -f

#docker-compose -f ./repfsw-nexus.yml pull

docker-compose -f ./repfsw-nexus.yml up –d

* Crear el archivo “/fssvc/bin/33-repfsw-nexus-undeploy.sh” con el siguiente contenido

#!/bin/sh

clear

echo ""

echo "###### DESINSTALANDO REPFSW-NEXUS ...."

docker-compose -f ./repfsw-nexus.yml stop

docker-compose -f ./repfsw-nexus.yml rm -f

docker-compose -f ./repfsw-nexus.yml down

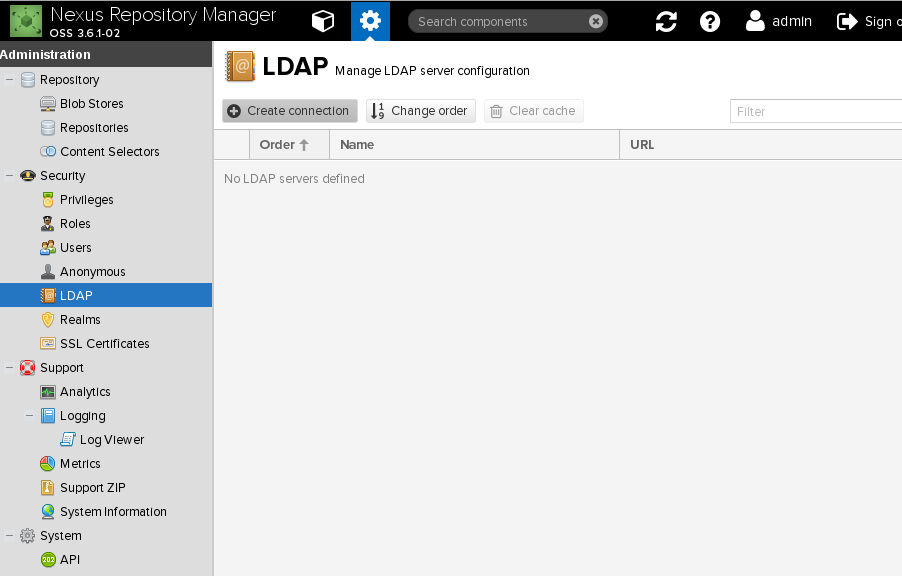
* Para instalar por primera vez las imágenes docker a utilizar ejecutar:

$ /fssvc/bin/31-repfsw-nexus-images.sh

* Para iniciar el servicio ejecutar:

$ /fssvc/bin/32-repfsw-nexus-redeploy.sh

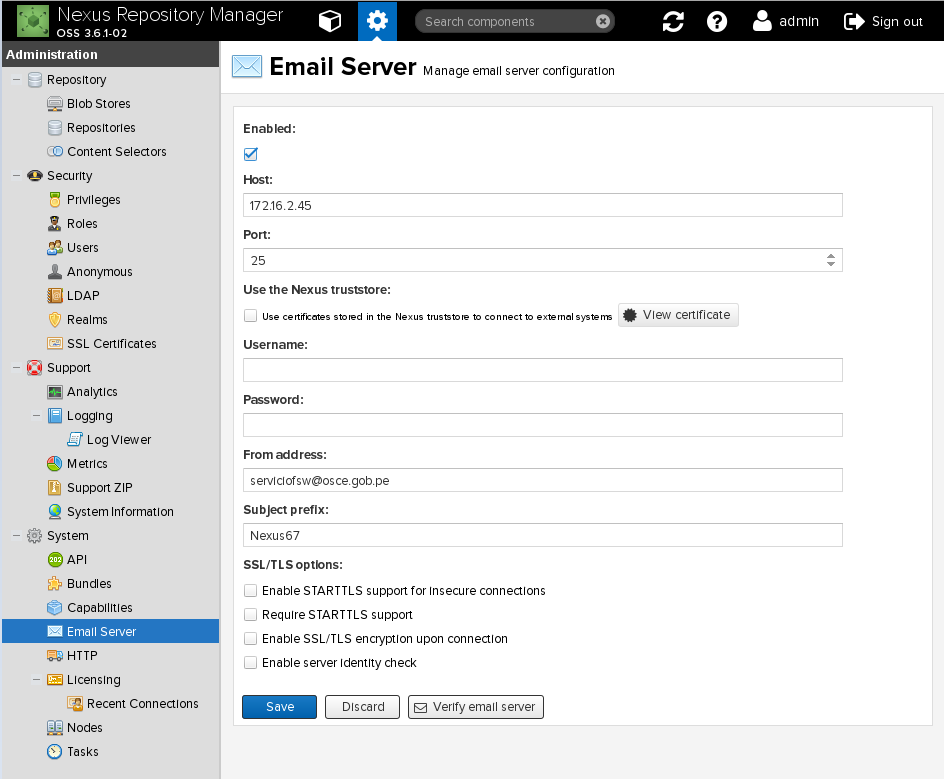
* Para acceder al servicio dirigirse a URL <http://192.168.32.40> y habrá un servidor respondiendo
* Ingresar a consola administrativa <http://192.168.32.40/nexus3> e ingresar las credenciales de un usuario con privilegios de administración.
* Crear configuración LDAP/Active Directory siguiendo los pasos de las siguientes imágenes:







* Para configurar el correo electrónico usar los valores de la siguiente imagen:

****

1. **INSTRUCCIONES DE OPERACIONES DE LOS PRODUCTOS SOFTWARE’S HABILITADOS**

Los comandos de operación deben ejecutarse en cada servidor donde está desplegado el servicio a operar, para ello autenticarse con el usuario de sistema operativo **operadorsvc**, a partir de ello ejecutar los comandos descritos a continuación para cada uno de los servicios:

* 1. **SERVIDOR CIDFSW1**
     1. **Operaciones con Contenedores**
* Para visualizar los contenedores que se están ejecutando:

$ docker ps

* Para visualizar todos los contenedores del host:

$ docker ps –a

* Para eliminar contenedor creado en host:

$ docker rm –f <container id>

* + 1. **Operaciones con HAProxy 1.7.9**
* Para **INICIAR** el contenedor de HAProxy:

$ cd /fssvc/bin/

$ ./12-cidfsw-haproxy-redeploy.sh

* Para **DETENER** el contenedor de HAProxy:

$ cd /fssvc/bin/

$ ./13-cidfsw-haproxy-undeploy.sh

* Para **VISUALIZAR LOGS** del contenedor de HAProxy:

$ docker logs -f cidfsw-haproxy1

* Para **VISUALIZAR LOGS** generados por el servicio HAProxy ubicarse en el directorio de logs:

$ cd /fssvc/haproxy/1.7.9-alpine/logs

* Para **VERIFICAR DISPONIBILIDAD** del servicio HAProxy:

|  |  |
| --- | --- |
| **Comandos** | **Resultado esperado** |
| $ docker ps | En lista resultado aparece que una imagen **haproxy:1.7.9-alpine** con nombre **cidfsw-haproxy1** está ejecutándose. |
| $ telnet localhost 80 | Se consigue repuesta del siguiente tipo:  Connected to localhost.  Escape character is '^]'. |
| $ cd /tmp  $ wget http://localhost:80 | Se recibe una respuesta HTTP. |

* + 1. **Operaciones con SonarQube 6.7**
* Para **INICIAR** el contenedor de SonarQube:

$ cd /fssvc/bin

$ ./22-cidfsw-sonarqube-redeploy.sh

* Para **DETENER** el contenedor de SonarQube:

$ cd /fssvc/bin

$ ./23-cidfsw-sonarqube-undeploy.sh

* Para **VISUALIZAR LOGS** del contenedor de SonarQube:

$ docker logs -f cidfsw-sonarqube1

* Para **VISUALIZAR LOGS** generados por el servicio SonarQube ubicarse en el directorio de logs:

$ cd /fssvc/sonarqube/6.7-alpine/logs

* Para **VERIFICAR DISPONIBILIDAD** del servicio SonarQube:

|  |  |
| --- | --- |
| **Comandos** | **Resultado esperado** |
| $ docker ps | En lista resultado aparece que una imagen **sonarqube:6.7-alpine** con nombre **cidfsw-sonarqube1** está ejecutándose. |
| $ telnet localhost 9000 | Se consigue repuesta del siguiente tipo:  Connected to localhost.  Escape character is '^]'. |
| $ cd /tmp  $ wget http://localhost:9000 | Se recibe una respuesta HTTP. |

* 1. **SERVIDOR REPFSW1**
     1. **Operaciones con Contenedores**
* Para visualizar los contenedores que se están ejecutando:

$ docker ps

* Para visualizar todos los contenedores del host:

$ docker ps –a

* Para eliminar contenedor creado en host:

$ docker rm –f <container id>

* + 1. **Operaciones con HAProxy 1.7.9**
* Para **INICIAR** el contenedor de HAProxy:

$ cd /fssvc/bin/

$ ./12-repfsw-haproxy-redeploy.sh

* Para **DETENER** el contenedor de HAProxy:

$ cd /fssvc/bin/

$ ./13-repfsw-haproxy-undeploy.sh

* Para **VISUALIZAR LOGS** del contenedor de HAProxy:

$ docker logs -f repfsw-haproxy1

* Para **VISUALIZAR LOGS** generados por el servicio HAProxy ubicarse en el directorio de logs:

$ cd /fssvc/haproxy/1.7.9-alpine/logs

* Para VERIFICAR DISPONIBILIDAD del servicio HAProxy:

|  |  |
| --- | --- |
| **Comandos** | **Resultado esperado** |
| $ docker ps | En lista resultado aparece que una imagen **haproxy:1.7.9-alpine** con nombre **repfsw-haproxy1** está ejecutándose. |
| $ telnet localhost 80 | Se consigue repuesta del siguiente tipo:  Connected to localhost.  Escape character is '^]'. |
| $ cd /tmp  $ wget http://localhost:80 | Se recibe una respuesta HTTP. |

* + 1. **Operaciones con Subversion 1.9.7**
* Para **INICIAR** el contenedor de Subversion:

$ cd /fssvc/bin/

$ ./22-repfsw-svn-redeploy.sh

* Para **DETENER** el contenedor de Subversion:

$ cd /fssvc/bin/

$ ./23-repfsw-svn-undeploy.sh

* Para **VISUALIZAR LOGS** del contenedor de Subversion:

$ docker logs -f repfsw-svn1

* Para **VISUALIZAR LOGS** generados por el servicio Subversion ubicarse en el directorio de logs:

$ cd /fssvc/svn/1.9.7/httpd-logs

* Para **VERIFICAR DISPONIBILIDAD** del servicio Subversion:

|  |  |
| --- | --- |
| **Comandos** | **Resultado esperado** |
| $ docker ps | En lista resultado aparece que una imagen  **apache2-subversion:1.9.7** con nombre **repfsw-svn1** está ejecutándose. |
| $ telnet localhost 9480 | Se consigue repuesta del siguiente tipo:  Connected to localhost.  Escape character is '^]'. |
| $ cd /tmp  $ wget http://localhost:9480 | Se recibe una respuesta HTTP. |

* + 1. **Operaciones con Nexus OSS 3.6.2**
* Para **INICIAR** el contenedor de Nexus:

$ cd /fssvc/bin/

$ ./32-repfsw-nexus-redeploy.sh

* Para **DETENER** el contenedor de Nexus:

$ cd /fssvc/bin/

$ ./33-repfsw-nexus-undeploy.sh

* Para **VISUALIZAR LOGS** del contenedor de Nexus:

$ docker logs -f repfsw-nexus1

* Para **VISUALIZAR LOGS** generados por el servicio Nexus ubicarse en el directorio de logs:

$ cd /fssvc/nexus/3.6.2/logs

* Para **VERIFICAR DISPONIBILIDAD** del servicio Nexus:

|  |  |
| --- | --- |
| **Comandos** | **Resultado esperado** |
| $ docker ps | En lista resultado aparece que una imagen  **sonatype/nexus3:3.6.2** con nombre **repfsw-nexus1** está ejecutándose. |
| $ telnet localhost 8081 | Se consigue repuesta del siguiente tipo:  Connected to localhost.  Escape character is '^]'. |
| $ cd /tmp  $ wget http://localhost:8081 | Se recibe una respuesta HTTP. |