

Curso de Nginx

Laboratorio 5

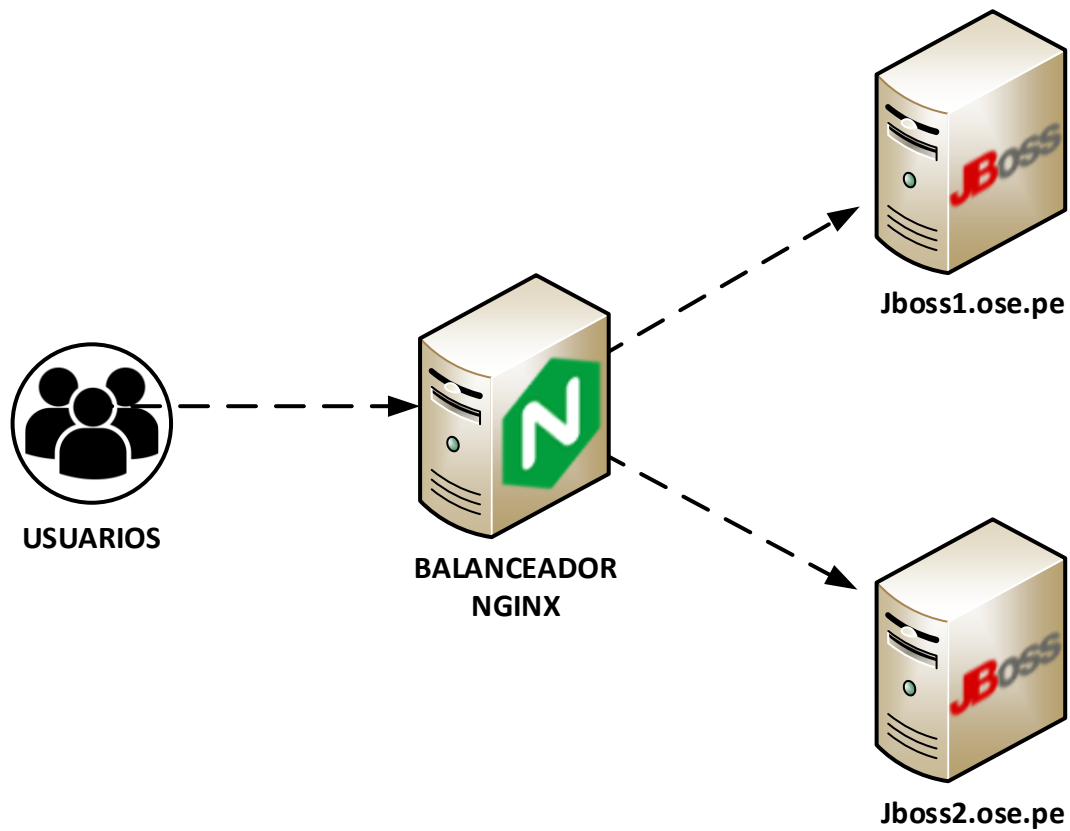
“Nginx como balanceador de carga”



LABORATORIO 5

“Balanceo de Carga”

Arquitectura de laboratorio:



Instalación de servidores Jboss Standalone

Configuraciones previas (Servidor jboss1.ose.pe)

1. Hay que asignarle un nombre al servidor Jboss

```
# hostnamectl set-hostname jboss1.ose.pe
```
2. Desactivamos el firewall.

```
# systemctl stop firewalld  
# systemctl disable firewalld
```

Configuraciones previas (Servidor jboss2.ose.pe)

3. Hay que asignarle un nombre al servidor Jboss

```
# hostnamectl set-hostname jboss2.ose.pe
```
4. Desactivamos el firewall.

```
# systemctl stop firewalld  
# systemctl disable firewalld
```

Instalación de JAVA (En Ambos Servidores)

Para este laboratorio usaremos la versión de OpenJDK, viene incluida en el repositorio de red hat o centos que estemos usando.

5. Para descargar Open JDK, utilizaremos el siguiente comando:

```
# yum install java
```

```
Dependency Updated:
 freetype.x86_64 0:2.8-12.el7_6.1
 glib2.x86_64 0:2.56.1-4.el7_6
 nspr.x86_64 0:4.19.0-1.el7_5
 nss.x86_64 0:3.36.0-7.1.el7_6
 nss-softokn.x86_64 0:3.36.0-5.el7_5
 nss-softokn-freebl.x86_64 0:3.36.0-5.el7_5
 nss-sysinit.x86_64 0:3.36.0-7.1.el7_6
 nss-tools.x86_64 0:3.36.0-7.1.el7_6
 nss-util.x86_64 0:3.36.0-1.1.el7_6
Complete!
```

6. Verifique la versión de Java que tiene instalada con el siguiente comando:

```
# java -version
```

```
[root@jboss ~]# java -version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
[root@jboss ~]#
```

Instalación del servicio de JBoss (En Ambos Servidores)

1. Creamos el usuario jbossadm, con su home directory como "/opt/jbosseap".

```
# useradd jbosseap -d /opt/jbosseap
```
2. Descargue el comprimido de JBoss. **(El instructor proveerá la URL que contendrá el comprimido)**
3. Copiamos el archivo comprimido de Jboss hacia el directorio /opt/jbosseap y ejecutamos los siguientes comandos:

```
# cp jboss-eap-7.2.0.zip /opt/jbosseap/  
# cd /opt/jbosseap/  
# unzip jboss-eap-7.2.0.zip
```
4. Asignamos los permisos del usuario y grupo jbosseap.

```
# chown -R jbosseap.jbosseap /opt/jbosseap/
```
5. Debe crear un usuario para el portal de Web de Jboss, para ello requiere ser el usuario jbosseap y ejecutar lo siguiente:

```
# su - jbosseap  
$ sh jboss-eap-7.2/bin/add-user.sh
```

Esto será para crear el siguiente usuario jbossAdmin, sus datos serán:

Username: jbossAdmin

Password: jbossadmin@2019

Nos aparecerá el siguiente setup, hágalo como muestra las líneas marcadas en rojo.

```
What type of user do you wish to add?  
a) Management User (mgmt-users.properties)  
b) Application User (application-users.properties)  
(a): a  
Enter the details of the new user to add.  
Using realm 'ManagementRealm' as discovered from the existing property  
files.  
Username : jbossAdmin  
Password : jbossadmin@2019  
Re-enter Password : jbossadmin@2019  
What groups do you want this user to belong to? (Please enter a comma  
separated list, or leave blank for none)[ ]: ENTER  
About to add user 'jbossAdmin' for realm 'ManagementRealm'  
Is this correct yes/no? yes  
Is this new user going to be used for one AS process to connect to  
another AS process?  
e.g. for a slave host controller connecting to the master or for a  
Remoting connection for server to server EJB calls.  
yes/no? yes
```

Regresamos como el usuario root.

```
$ exit
```

Scripts de inicio de JBoss (En Ambos Servidores)

1. Para configurar como servicio Jboss, vamos a copiar los scripts de inicio que se encuentra en:

```
# cp /opt/jbosseap/jboss-eap-7.2/bin/init.d/jboss-eap-rhel.sh
/etc/init.d/jboss
# chmod +x /etc/init.d/jboss
```

2. Ahora editamos el script de inicio

```
# vim /etc/init.d/jboss
```

Las líneas marcadas simbolizan los cambios realizados, el archivo de configuración deberá quedar de la siguiente manera:

Modificamos la línea 25 del archivo de configuración:

```
if [ -z "$JBOSS_CONF" ]; then
    JBOSS_CONF="/etc/default/jboss-eap.conf"
fi
```

Modificamos la línea 33 del archivo de configuración:

```
if [ -z "$JBOSS_HOME" ]; then
    JBOSS_HOME="/opt/jbosseap/jboss-eap-7.2"
fi
```

3. Después realizamos una copia de un archivo de configuración de ejemplo a la ruta de /etc/defaults/jboss-eap.conf, este archivo será usado por el script de inicio de jbos.

```
# cp /opt/jbosseap/jboss-eap-7.2/bin/init.d/jboss-eap.conf
/etc/default/jboss-eap.conf
```

4. Ahora editamos el archivo de configuración

```
# vim /etc/default/jboss-eap.conf
```

Las líneas marcadas simbolizan los cambios realizados, el archivo de configuración deberá quedar de la siguiente manera:

```
# General configuration for the init.d scripts,
# not necessarily for JBoss EAP itself.
# default location: /etc/default/jboss-eap

## Location of JDK
# JAVA_HOME="/usr/lib/jvm/default-java"

## Location of JBoss EAP
JBOSS_HOME="/opt/jbosseap/jboss-eap-7.2"

## The username who should own the process.
JBOSS_USER=jbosseap

## The mode JBoss EAP should start, standalone or domain
JBOSS_MODE=standalone
```

```
## Configuration for standalone mode
JBOSS_CONFIG=standalone.xml

## Configuration for domain mode
# JBOSS_DOMAIN_CONFIG=domain.xml
# JBOSS_HOST_CONFIG=host-master.xml

## The amount of time to wait for startup
# STARTUP_WAIT=60

## The amount of time to wait for shutdown
# SHUTDOWN_WAIT=60

## Location to keep the console log
JBOSS_CONSOLE_LOG="/var/log/jboss-eap/console.log"

## Additional args to include in startup
# JBOSS_OPTS="--admin-only -b 127.0.0.1"

JBOSS_ADDR=192.168.1.X

JBOSS_OPTS=" -Djboss.bind.address=$JBOSS_ADDR -
Djboss.bind.address.management=$JBOSS_ADDR"
```

NOTA:

El parámetro de **JBOSS_ADDR** debe hacer referencia a la IP del servidor JBoss.

5. Iniciamos el servicio de JBoss como host Controller

```
# chkconfig jboss on
# service jboss start
```

6. Verifique el servicio de Jboss ingresando a las siguiente URL:
http:IP_DEL_SERVIDOR:9990

Ingrese el usuario y password:

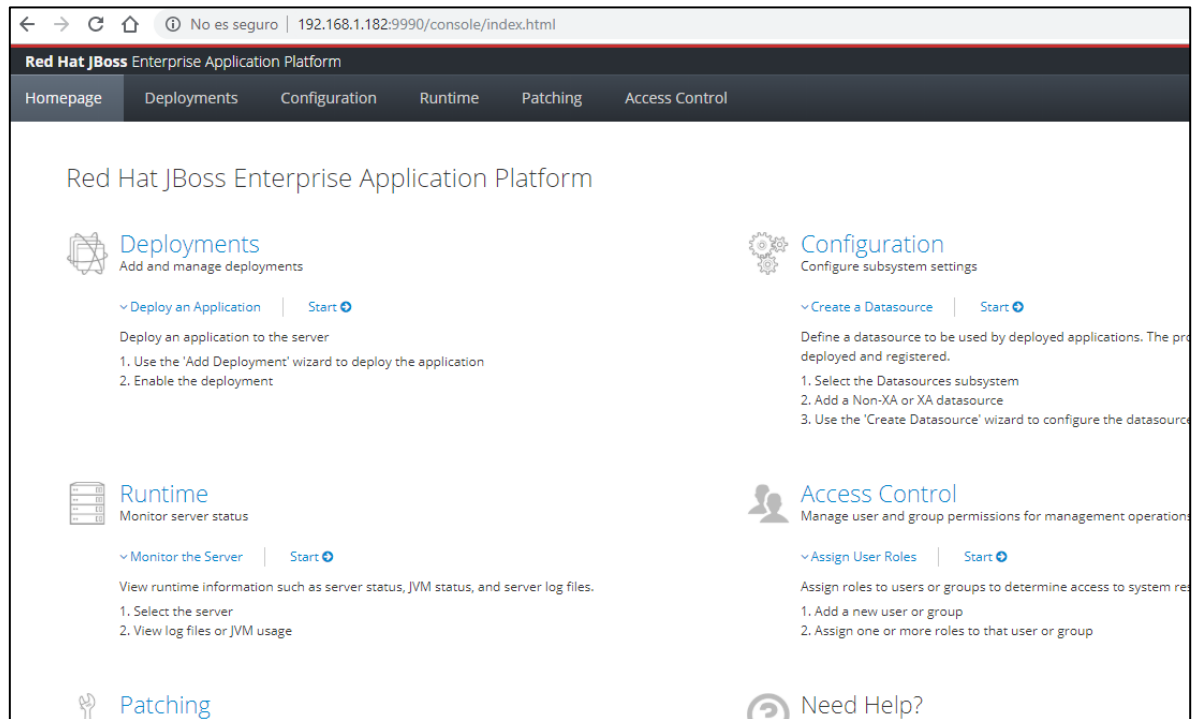
Iniciar sesión

http://192.168.1.182:9990

Tu conexión con este sitio web no es privada

Nombre de usuario

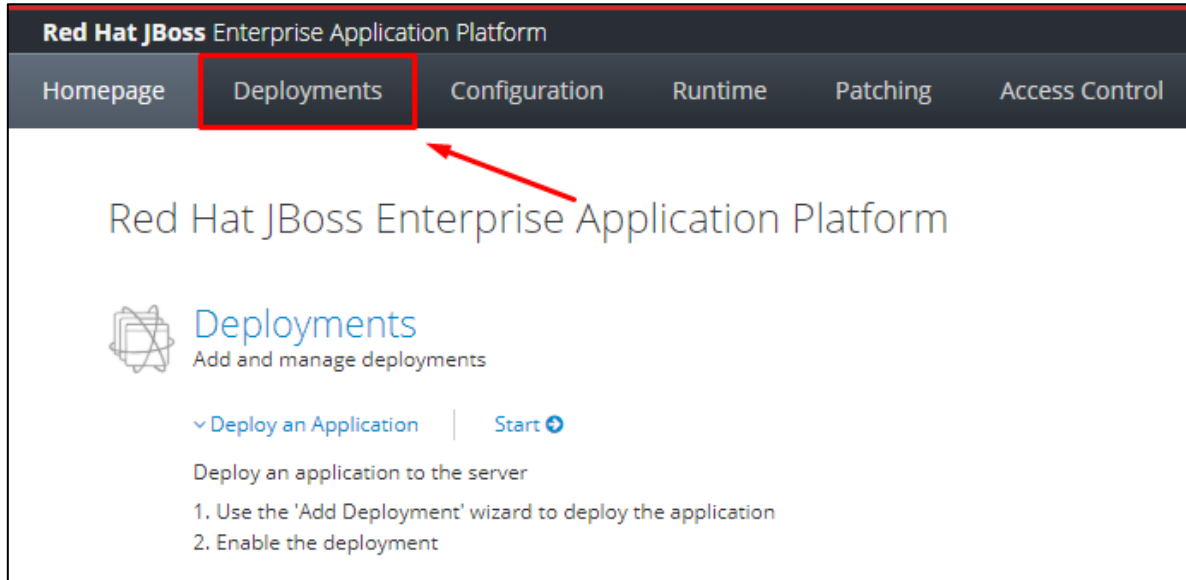
Contraseña




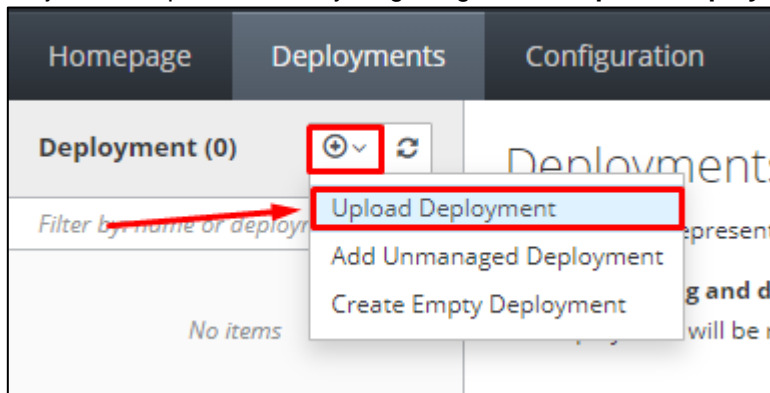
7. **Ahora que realizo el procedimiento de instalación, realice las configuraciones anteriores en el segundo servidor Jboss.**

Despliegue de Wars

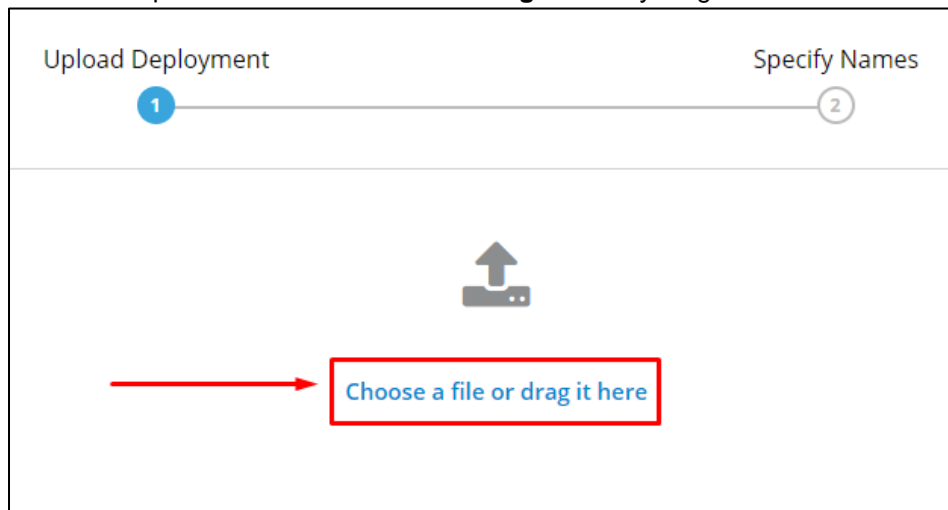
1. Para desplegar dirijase a la opción de Deploiments



2. Dirijase a la opción de  y luego haga clic en **Upload Deployment**.



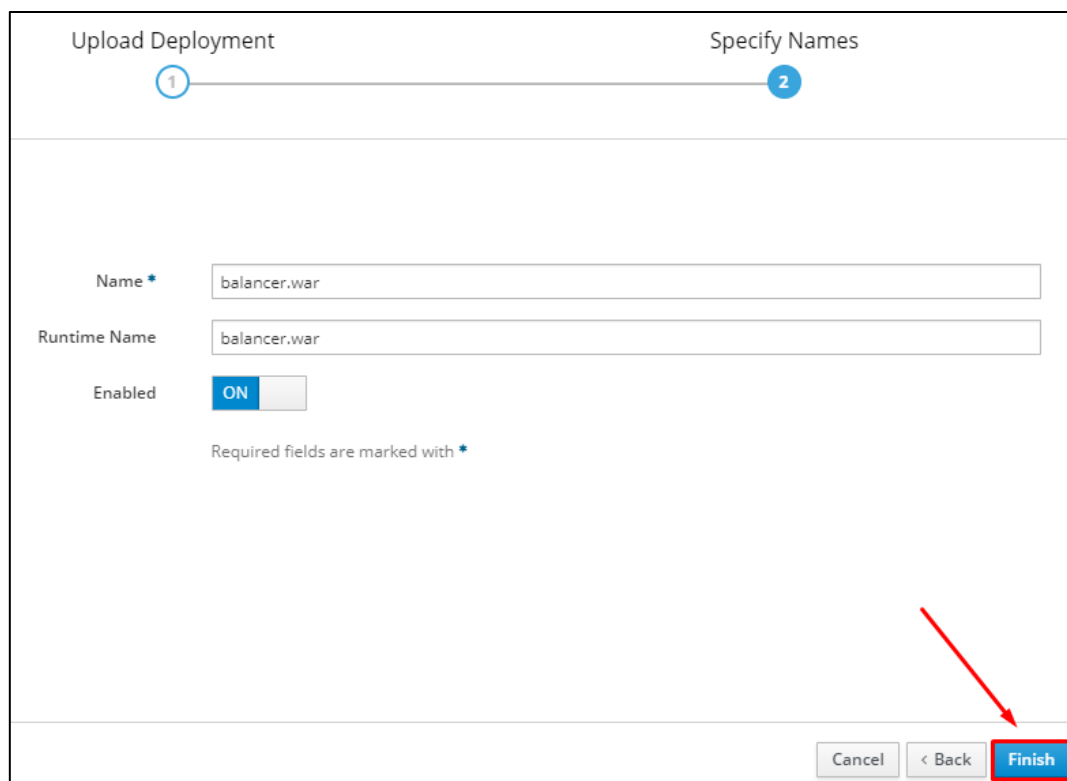
3. Haga clic sobre la opción de “**Choose a file or drag it here**” y cargue el war **balancer.war**.



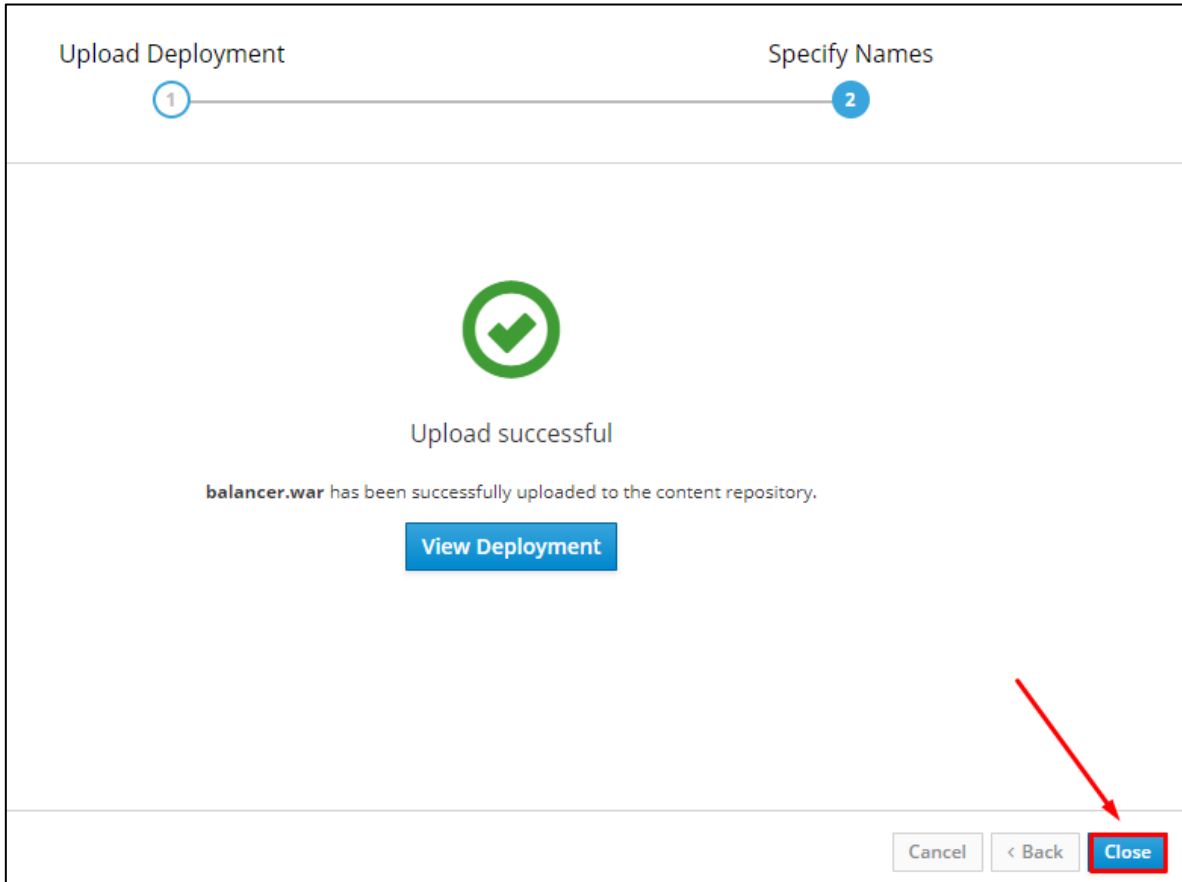
Nota:

El instructor proveerá el war para su descarga desde un servidor Web

4. En la siguiente ventana haga clic en Finish para terminar de subir el War.




5. El war fue desplegado exitosamente, haga clic sobre la opción de Close.



Upload Deployment

Specify Names

1 2



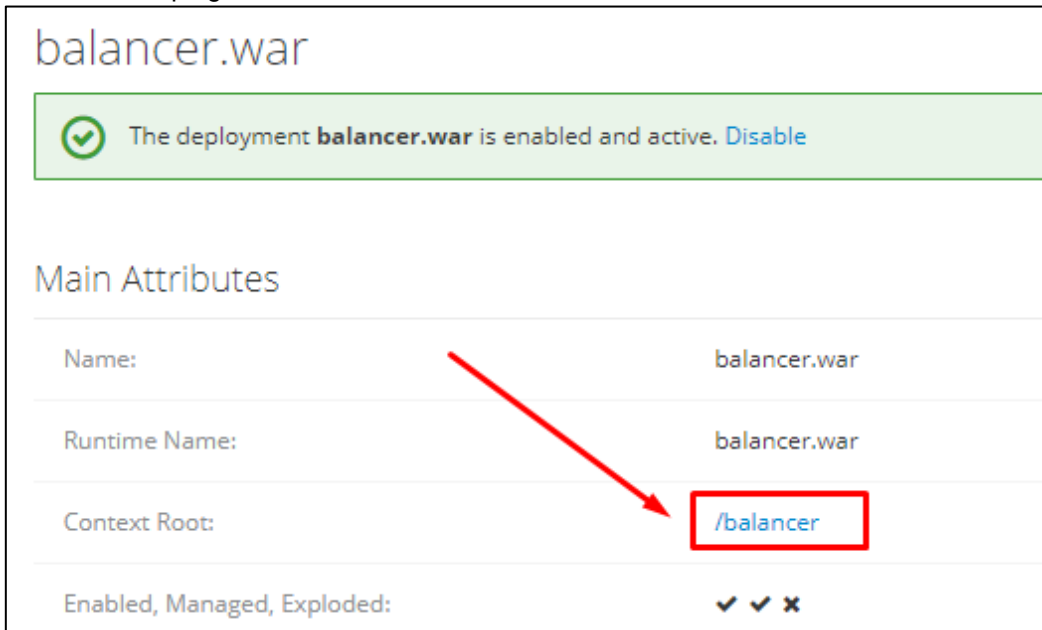
Upload successful

balancer.war has been successfully uploaded to the content repository.


[View Deployment](#)

[Cancel](#) [< Back](#) [Close](#)

6. El war fue desplegado en el contexto de /balancer.



balancer.war

 The deployment **balancer.war** is enabled and active. [Disable](#)

Main Attributes

Name:	balancer.war
Runtime Name:	balancer.war
Context Root:	/balancer
Enabled, Managed, Exploded:	✓ ✓ ✗

Para ver la aplicación desplegada nos dirigimos a la URL:

http:IP_DEL_SERVIDOR:8080/balancer



Nota:

La ip mostrada en la imagen es solo de ejemplo, use la IP de su servidor JBoss.

7. Ahora que realizo el procedimiento para el despliegue de una aplicación de ejemplo, realice las configuraciones anteriores en el segundo servidor Jboss.

Hasta este punto debemos tener configurado 2 servidores Jboss Standalone con el despliegue del War “balancer.war”.

Configuración de Balanceador Nginx

El equilibrio de carga con nginx usa un algoritmo round-robin de forma predeterminada si no se define ningún otro método. Con el esquema round-robin, cada servidor se selecciona por turnos de acuerdo con el orden en que los configuró en el archivo load-balancer.conf . Esto equilibra el número de solicitudes por igual.

1. Vamos a crear el siguiente archivo de configuración, donde crearemos un server block para el balanceo de carga.

```
# vim /etc/nginx/conf.d/app-balancer.conf
```

Ese debe tener el siguiente contenido:

```
upstream apps_jboss {  
    server 192.168.1.X:8080;  
    server 192.168.1.Z:8080;  
}  
  
server {  
    listen 80;  
    server_name localhost;  
  
    location / {  
        proxy_pass http://apps_jboss/balancer/;  
    }  
}
```

Nota:

La IP'S mostradas en son solo de ejemplo, use las IP'S de sus servidores JBoss.

2. Cambiamos de nombre al archivo **default.conf** por **default.conf.disabled**, ya que este archivo está usando el puerto 80.

```
# mv /etc/nginx/conf.d/default.conf  
/etc/nginx/conf.d/default.conf.disabled
```

3. Validamos la configuración usando el siguiente comando:

```
# nginx -t
```

4. Reiniciamos el servicio de Nginx usando:

```
# nginx -s reload
```

5. Ingresamos a la siguiente URL:

```
http://192.168.1.X/
```

Nota:

La ip mostrada en la imagen es solo de ejemplo, use la IP de su servidor NGINX.



Realice un refresco de la página Web usando la tecla F5.



Observara que ahora la consulta al balanceador nos derivó al segundo servidor Jboss

Configuración de Balanceador por PESOS - Nginx

En una configuración de servidor donde los recursos disponibles entre diferentes hosts no son iguales, puede ser conveniente favorecer algunos servidores sobre otros. La definición de los pesos del servidor le permite ajustar aún más el equilibrio de carga con nginx. El servidor con el mayor peso en el equilibrador de carga se selecciona con mayor frecuencia

1. Edite el archivo app-balancer.conf

```
# vim /etc/nginx/conf.d/app-balancer.conf
```

Vamos a modificar las siguientes líneas marcadas en el archivo de configuración:

```
upstream apps_jboss {  
    server 192.168.1.X:8080 weight=2;  
    server 192.168.1.Z:8080 weight=1;  
}  
  
server {  
    listen 80;  
    server_name localhost;  
    location / {  
        proxy_pass http://apps_jboss/balancer/;  
    }  
}
```

2. Reiniciamos el servicio de NGINX

```
# nginx -s reload
```

3. Validamos la configuración ingresando a la IP del balanceador en el navegador.

<http://192.168.1.X/>



Refresque la página:



Observara que seguimos en el mismo servidor jboss1.

Refresque la página nuevamente:



Observara que ahora estamos el servidor jboss2 con una nueva sesión.

Mantener las sesiones de los usuarios

Hasta el momento hemos probado las configuraciones de balanceo de carga, pero no hemos podido mantener las sesiones de los usuarios. Para ello vamos a usar IP HASH

1. Nos dirigimos al archivo de configuración app-balancer.conf

```
# vim /etc/nginx/conf.d/app-balancer.conf
```

Agregaremos la siguiente línea:

```
upstream apps_jboss {  
    ip_hash ;  
    server 192.168.1.X:8080 ;  
    server 192.168.1.Z:8080 ;  
}  
  
server {  
    listen 80;  
    server_name localhost;  
    location / {  
        proxy_pass http://apps_jboss/balancer/;  
    }  
}
```

2. Reiniciamos el servicio de Nginx

```
# nginx -s reload
```

3. Validamos si ahora ingresando al navegador nuestra sesión cambia de servidor JBoss

<http://192.168.1.X/>



4. Refresque el navegador usando la tecla F5



Observe que ahora el servidor jboss1 es el que responde permanentemente.

Configuración SSL para el balanceo de carga

Hasta el momento hemos probado las configuraciones de balanceo de carga y hemos mantenido la sesión de un usuario en un servidor en específico. Ahora vamos a agregarle certificado SSL.

Creación de certificado autofirmado (Servidor Nginx)

Para crear el certificado autofirmado vamos a necesitar de OpenSSL. Para ello ejecute el siguiente comando:

1. Vamos a crear un directorio que albergara los certificados

```
# mkdir /certs
```

2. Ahora vamos a crear los certificados ose.key y ose.csr

```
# cd /certs
# openssl req -new -nodes -keyout ose.key -out ose.csr -days 3650
```

Nos aparecerá el siguiente SETUP, utilice las siguientes opciones

```
Generating a 2048 bit RSA private key
writing new private key to 'ose.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:PE
State or Province Name (full name) []:LIMA
Locality Name (eg, city) [Default City]:LIMA
Organization Name (eg, company) [Default Company Ltd]:OSE
Organizational Unit Name (eg, section) []:TI
Common Name (eg, your name or your server's hostname) []:www.ose.pe
Email Address []: [ENTER]

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: [ENTER]
An optional company name []: [ENTER]
```

3. Ahora vamos a auto firmar el certificado para obtener el certificado ose.crt.

```
# openssl x509 -req -days 3650 -in ose.csr -signkey ose.key -out
ose.crt
```

```
[root@nginx certs]# openssl x509 -req -days 3650 -in ose.csr -signkey ose.key -
out ose.crt
Signature ok
subject=/C=PE/ST=LIMA/L=LIMA/O=OSE/OU=TI/CN=www.ose.pe
Getting Private key
```


Configuración de balanceo SSL

Esta configuración del certificado SSL contempla el redireccionamiento de HTTP a HTTPS.

1. Ahora que ya tenemos nuestro certificados, debe editar el archivo app-balancer.conf

```
# vim /etc/nginx/conf.d/app-balancer.conf
```

El archivo de configuración debe ser de la siguiente manera:

```
upstream apps_jboss {
    server 192.168.1.X:8080 ;
    server 192.168.1.Z:8080 ;
}

server {
    listen 80;
    server_name 192.168.1.N;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name 192.168.1.N;
    ssl_certificate /certs/ose.crt;
    ssl_certificate_key /certs/ose.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    location / {
        proxy_pass http://apps_jboss/balancer/;
    }
}
```

NOTA:

Para el parámetro "server_name" use la dirección IP del servidor NGINX

2. Reiniciamos el servicio de Nginx

```
# nginx -s reload
```

3. Validamos si ahora ingresando al navegador, nos redirecciona de http a https

<http://192.168.1.X/>



Observe que ahora las aplicaciones de los servidores JBoss, ya contemplan su configuración del certificado SSL.

Agregue la configuración de IP HASH y observe que la sesión se mantendrá sin problema en el mismo servidor.