**Configuracion JBOSS en dominio.**

**Fuente: https://docs.jboss.org/author/display/AS71/AS7+Cluster+Howto**

Partimos del hecho que jboss se encuentra instalado en la ruta /opt/jboss/jboss-eap-6.2. Existe archivo de inicio del servicio en /etc/init.d/jboss-eap.

1.- Modificamos el archivo host.xml en el servidor que será el master. Para este caso colocamos IP\_SERVIDOR .

Este paso lo repetimos en el servidor que será el slave.

|  |
| --- |
| <interfaces>  <interface name="management">  <inet-address value="${jboss.bind.address.management:IP\_SERVIDOR}"/>  </interface>  <interface name="public">  <inet-address value="${jboss.bind.address: IP\_SERVIDOR }"/>  </interface>  <interface name="unsecure">  <!-- Used for IIOP sockets in the standard configuration.  To secure JacORB you need to setup SSL -->  <inet-address value="${jboss.bind.address.unsecure: IP\_SERVIDOR }"/>  </interface>  </interfaces> |

2.- En el servidor slave debemos renombrar el archivo domain.xml ya que este equipo no debe comportarse como un controlador.

|  |
| --- |
| Mv /opt/jboss/jboss-eap-6.2/domain/configuration/domain.xml /opt/jboss/jboss-eap-6.2/domain/configuration/domain.xml.ORI |

3.- En el Servidor slave debemos hacer una configuración adicional en el archivo host.xml

|  |
| --- |
| Cambiar:  <host name="master" xmlns="urn:jboss:domain:1.1">  A  <host name="slave" xmlns="urn:jboss:domain:1.1"> |

En el archive host.xml modificar la sección domain-controller:

|  |
| --- |
| <domain-controller>  <remote host="192.168.56.10" port="9999" security-realm="ManagementRealm" username="slave"/>  </domain-controller> |

4.- En el servidor master debemos crear el usuario que utilizará el slave para conectarse, en estos momentos ya tenemos creado el usuario admin para la administración desde la consola.

|  |
| --- |
| /opt/jboss/jboss-eap-6.2/bin/add-user.sh  What type of user do you wish to add?  a) Management User (mgmt-users.properties)  b) Application User (application-users.properties)  (a):  Enter the details of the new user to add.  Using realm 'ManagementRealm' as discovered from the existing property files.  Username : slave  Password :  Re-enter Password :  What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:  About to add user 'usrslave' for realm 'ManagementRealm'  Is this correct yes/no? yes  Added user 'usrslave' to file '/opt/jboss/jboss-eap-6.2/domain/configuration/mgmt-users.properties'  Added user 'usrslave' with groups to file '/opt/jboss/jboss-eap-6.2/domain/configuration/mgmt-groups.properties'  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  To represent the user add the following to the server-identities definition <secret value="U2xhdmUxMjM0JA==" />  NOTA: Como password colocamos Slave1234$ |

5.- En el slave debemos configurar host.xml para autenticarse, para ello debemos cambiar la sección security-realms:

|  |
| --- |
| <security-realms>  <security-realm name="ManagementRealm">  <server-identities>  <secret value="**U2xhdmUxMjM0JA**==" />  </server-identities>  <authentication>  <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>  </authentication>  </security-realm>  </security-realms> |

Se ha agregado server-identities en la sección security-realms, la cual es usada para la autenticación cuando el slave intente conectarse al master. Debido a que el hostname de servidor slave se colocar “slave” debemos utilizar el usuario y password de slave que se creó en el master. En la etiqueta secret se colocó U2xhdmUxMjM0JA== lo cual es el código base64 para Slave1234$. Podemos utilizar la página <http://www.webutils.pl/index.php?idx=base64> para generar otros valores.

6.- En el slave debemos comentar las siguientes líneas:

|  |
| --- |
| <!-- <http-interface security-realm="ManagementRealm">  <socket interface="management" port="${jboss.management.http.port:9990}"/>  </http-interface> --> |

7.- Iniciamos el servicio de jboss en el servidor master y luego en el slave, al momento de iniciar en el slave, debe aparecer en los logs del master lo siguiente:

|  |
| --- |
| [Host Controller] 15:30:07,180 INFO [org.jboss.as.domain] (slave-request-threads - 1) JBAS010918: Registered remote slave host "slave", JBoss EAP 6.2.0.GA (AS 7.3.0.Final-redhat-14) |

8.- Para verificar podemos ejecutar el siguiente comando ls –l /host en el servidor master y debería mostrar los equipos registrados en este dominio, en este caso master y slave:

|  |
| --- |
| jboss-cli.sh controller=192.168.56.10 –connect  ls -l /host  master  slave |

9.- En el archivo $JBOSS\_HOME/domain/configuration/domain.xml modificar el nivel de log para cada profile, el mismo debe quedar de la siguiente manera:

|  |
| --- |
| </periodic-rotating-file-handler>  <logger category="com.arjuna">  <level name="WARN"/>  </logger>  <logger category="org.apache.tomcat.util.modeler">  <level name="WARN"/>  </logger>  <logger category="org.jboss.as.config">  <level name="WARN"/>  </logger>  <logger category="sun.rmi">  <level name="WARN"/>  </logger>  <logger category="jacorb">  <level name="WARN"/>  </logger>  <logger category="jacorb.config">  <level name="ERROR"/>  </logger>  <root-logger>  <level name="WARN"/>  <handlers>  <handler name="CONSOLE"/>  <handler name="FILE"/>  </handlers>  </root-logger> |

10.-Modificar el archivo $JBOSS\_HOME/domain/configuration/logging.properties en la sección Console handler configuration, colocar nivel de log en WARN para que genere menos escritura en disco en el archivo de log de la consola.

|  |
| --- |
| handler.CONSOLE.level=${jboss.boot.server.log.console.level:WARN} |

**Creando JAVA KeyStore**

1.- Creamos un directorio para guardar el almacén de java /opt/jboss/keyStore

Keystore Password KCr3d1c4rd

|  |
| --- |
| /opt/jboss/keyStore> keytool -genkey -alias keyStore -keyalg RSA -keysize 2048 -keystore keyStore.keystore -validity 30000 -dname "CN=Jboss Admin,OU=Admin Applications,O=Credicard,L=Caracas,ST=DC,C=VE"  Enter keystore password:  Re-enter new password:  Enter key password for <keyStore>  (RETURN if same as keystore password):  [root@srv-vccs-portalapp-qa-01]:/opt/jboss/keyStore>  /opt/jboss/keyStore>ls -l  total 4  -rw-r--r-- 1 root root 2259 Apr 10 10:41 **keyStore.keystore** |

2.- Enmascarando el keystore password e inicializando la bóveda

|  |
| --- |
| /opt/jboss/jboss-eap-6.2/bin>./vault.sh  =========================================================================  JBoss Vault  JBOSS\_HOME: /opt/jboss/jboss-eap-6.2  JAVA: /usr/java/jdk1.7.0\_51/bin/java  =========================================================================  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\* JBoss Vault \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Please enter a Digit:: 0: Start Interactive Session 1: Remove Interactive Session 2: Exit  0  Starting an interactive session  Enter directory to store encrypted files:/opt/jboss/keyStore/  Enter Keystore URL: /opt/jboss/keyStore/keyStore.keystore  Enter Keystore password:  Enter Keystore password again:  Values match  Enter 8 character salt:credi678  Enter iteration count as a number (Eg: 44):50  Enter Keystore Alias:keyStore  Initializing Vault  Apr 10, 2014 10:51:20 AM org.picketbox.plugins.vault.PicketBoxSecurityVault init  INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready  Vault Configuration in AS7 config file:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ...  </extensions>  <vault>  <vault-option name="KEYSTORE\_URL" value="/opt/jboss/keyStore/keyStore.keystore"/>  <vault-option name="KEYSTORE\_PASSWORD" value="MASK-28k/CVQoXZSqzGZLbCSSoV"/>  <vault-option name="KEYSTORE\_ALIAS" value="keyStore"/>  <vault-option name="SALT" value="credi678"/>  <vault-option name="ITERATION\_COUNT" value="50"/>  <vault-option name="ENC\_FILE\_DIR" value="/"/>t/jboss/keyStore/  </vault><management> ...  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Vault is initialized and ready for use  Handshake with Vault complete  Please enter a Digit:: 0: Store a secured attribute 1: Check whether a secured attribute exists 2: Exit |

3.- Configurar JBOSS para usar el Bóveda de password.

En modo dominio

|  |
| --- |
| /jboss-cli.sh  You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for the list of supported commands.  [disconnected /] connect 10.134.0.83:9999  [domain@10.134.0.83:9999 /] /host=portalappqa01/core-service=vault:add(vault-options=[("KEYSTORE\_URL" =>"/opt/jboss/keyStore/keyStore.keystore"),("KEYSTORE\_PASSWORD" =>"MASK-28k/CVQoXZSqzGZLbCSSoV"),("KEYSTORE\_ALIAS" => "keyStore"),("SALT" =>"credi678"),("ITERATION\_COUNT" => "50"),("ENC\_FILE\_DIR" =>"/opt/jboss/keyStore/")])  {  "outcome" => "success",  "result" => undefined,  "server-groups" => undefined  }  [domain@10.134.0.83:9999 /] |

**4.-** **Almacenar y recuperar strings encriptados en el Java KeyStore**

|  |
| --- |
| /opt/jboss/jboss-eap-6.2/bin>./vault.sh  =========================================================================  JBoss Vault  JBOSS\_HOME: /opt/jboss/jboss-eap-6.2  JAVA: /usr/java/jdk1.7.0\_51/bin/java  =========================================================================  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\* JBoss Vault \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Please enter a Digit:: 0: Start Interactive Session 1: Remove Interactive Session 2: Exit  0  Starting an interactive session  Enter directory to store encrypted files: /opt/jboss/keyStore/  Enter Keystore URL: /opt/jboss/keyStore/keyStore.keystore  Enter Keystore password:  Enter Keystore password again:  Values match  Enter 8 character salt:credi678  Enter iteration count as a number (Eg: 44):50  Enter Keystore Alias:keyStore  Initializing Vault  Apr 10, 2014 11:39:38 AM org.picketbox.plugins.vault.PicketBoxSecurityVault init  INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready  Vault Configuration in AS7 config file:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ...  </extensions>  <vault>  <vault-option name="KEYSTORE\_URL" value="/opt/jboss/keyStore/keyStore.keystore"/>  <vault-option name="KEYSTORE\_PASSWORD" value="MASK-28k/CVQoXZSqzGZLbCSSoV"/>  <vault-option name="KEYSTORE\_ALIAS" value="keyStore"/>  <vault-option name="SALT" value="credi678"/>  <vault-option name="ITERATION\_COUNT" value="50"/>  <vault-option name="ENC\_FILE\_DIR" value="/opt/jboss/keyStore/"/>  </vault><management> ...  Vault is initialized and ready for use  Handshake with Vault complete  Please enter a Digit:: 0: Store a secured attribute 1: Check whether a secured attribute exists 2: Exit  0  Task: Store a secured attribute  Please enter secured attribute value (such as password):  Please enter secured attribute value (such as password) again:  Values match  Enter Vault Block:datasourceOracle  Enter Attribute Name:password  Secured attribute value has been stored in vault.  Please make note of the following:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Vault Block:datasourceOracle  Attribute Name:password  Configuration should be done as follows:  **VAULT::datasourceOracle::password::1**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Please enter a Digit:: 0: Store a secured attribute 1: Check whether a secured attribute exists 2: Exit |

5.- Usar el string encriptado en la configuración. En el archivo domain.xml

Passw Actual es: credicard

|  |
| --- |
| <datasource jta="false" jndi-name="java:jboss/jdbc/Oracle" pool-name="datasourceOracle" enabled="true" use-ccm="false">  <connection-url>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=10.124.0.53)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=10.124.0.54)(PORT=1521))(LOAD\_BALANCE=yes)(FAILOVER=on)(CONNECT\_DATA=(SERVER=DEDICATED)(SERVICE\_NAME=SATDESA))(FAILOVER\_MODE=(TYPE=session)(METHOD=basic))))</connection-url>  <driver-class>oracle.jdbc.OracleDriver</driver-class>  <driver>oracle</driver>  <pool>  <min-pool-size>3</min-pool-size>  <max-pool-size>10</max-pool-size>  <flush-strategy>IdleConnections</flush-strategy>  </pool>  <security>  <user-name>usrportal</user-name>  <password> **VAULT::datasourceOracle::password::1**</password>  </security>  <validation>  <validate-on-match>false</validate-on-match>  <background-validation>false</background-validation>  </validation>  <timeout>  <idle-timeout-minutes>120</idle-timeout-minutes>  </timeout>  <statement>  <share-prepared-statements>false</share-prepared-statements>  </statement>  </datasource> |

Configurar password encriptado para DB2.

Passw Actual es: ZAQ1XSW2CD

Keystore Password KCr3d1c4rd

|  |
| --- |
| /opt/jboss/jboss-eap-6.2/bin>./vault.sh  =========================================================================  JBoss Vault  JBOSS\_HOME: /opt/jboss/jboss-eap-6.2  JAVA: /usr/java/jdk1.7.0\_51/bin/java  =========================================================================  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\* JBoss Vault \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Please enter a Digit:: 0: Start Interactive Session 1: Remove Interactive Session 2: Exit  0  Starting an interactive session  Enter directory to store encrypted files:/opt/jboss/keyStore/  Enter Keystore URL: /opt/jboss/keyStore/keyStore.keystore  Enter Keystore password:  Enter Keystore password again:  Values match  Enter 8 character salt:credi678  Enter iteration count as a number (Eg: 44):50  Enter Keystore Alias:keyStore  Initializing Vault  Apr 10, 2014 12:41:01 PM org.picketbox.plugins.vault.PicketBoxSecurityVault init  INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready  Vault Configuration in AS7 config file:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ...  </extensions>  <vault>  <vault-option name="KEYSTORE\_URL" value="/opt/jboss/keyStore/keyStore.keystore"/>  <vault-option name="KEYSTORE\_PASSWORD" value="MASK-28k/CVQoXZSqzGZLbCSSoV"/>  <vault-option name="KEYSTORE\_ALIAS" value="keyStore"/>  <vault-option name="SALT" value="credi678"/>  <vault-option name="ITERATION\_COUNT" value="50"/>  <vault-option name="ENC\_FILE\_DIR" value="/opt/jboss/keyStore/"/>  </vault><management> ...  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Vault is initialized and ready for use  Handshake with Vault complete  Please enter a Digit:: 0: Store a secured attribute 1: Check whether a secured attribute exists 2: Exit  0  Task: Store a secured attribute  Please enter secured attribute value (such as password):  Please enter secured attribute value (such as password) again:  Values match  Enter Vault Block:datasourceDB2  Enter Attribute Name:password  Secured attribute value has been stored in vault.  Please make note of the following:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Vault Block:datasourceDB2  Attribute Name:password  Configuration should be done as follows:  **VAULT::datasourceDB2::password::1**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Please enter a Digit:: 0: Store a secured attribute 1: Check whether a secured attribute exists 2: Exit |

2.- Usar el string encriptado en la configuración. En el archivo domain.xml

|  |
| --- |
| <datasource jta="false" jndi-name="java:jboss/jdbc/DB2" pool-name="datasourceDB2" enabled="true" use-ccm="false">  <connection-url>jdbc:as400://10.134.0.36</connection-url>  <driver-class>com.ibm.as400.access.AS400JDBCDriver</driver-class>  <driver>db2AS400</driver>  <pool>  <min-pool-size>3</min-pool-size>  <max-pool-size>10</max-pool-size>  <flush-strategy>IdleConnections</flush-strategy>  </pool>  <security>  <user-name>usrportal</user-name>  <password>**VAULT::datasourceDB2::password::1**</password>  </security>  <validation>  <validate-on-match>false</validate-on-match>  <background-validation>false</background-validation>  </validation>  <timeout>  <idle-timeout-minutes>120</idle-timeout-minutes>  </timeout>  <statement>  <share-prepared-statements>false</share-prepared-statements>  </statement>  </datasource> |

**Configuración de la consola sobre https (modo dominio).**

1.- Creamos el certificado:

|  |
| --- |
| keytool -genkeypair -alias console -keyalg RSA -keystore console.keystore -storepass console --dname "CN=srv-vccs-portalapp-qa-01,OU=Infraestructura,O=credicard.com.ve,L=Caracas,S=DC,C=VE" -validity 30000 |

Estos nos creó un archivo llamado console.keystore para este caso lo guardamos en la ruta /opt/jboss/jboss-eap-6.2/domain/configuration, que es lo mismo que ${jboss.domain.config.dir}. Esto debemos hacerlo en ambos nodos, tanto el master como en el slave, claro genera el archivo en uno y luego lo copias en el otro.

NOTA: Cuando activamos la administración vía HTTPS, la comunicación entre el master y el slave realizada por el puerto 9999 también se va a realizar con el mismo certificado, es decir también se encripta esta comunicación.

2.- Modificar el archivo host.xml del master.

|  |
| --- |
| <security-realms>  <security-realm name="ManagementRealm">  **<server-identities>**  **<ssl protocol="TLSv1">**  **<keystore path="${jboss.domain.config.dir}/console.keystore" keystore-password="console" alias="console"/>**  **</ssl>**  **</server-identities>**  <authentication>  <local default-user="$local"/>  <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>  **<truststore path="${jboss.domain.config.dir}/console.keystore" keystore-password="console" alias="console"/>**  </authentication>  …….  …….  …….  </security-realms> |

|  |
| --- |
| <management-interfaces>  <native-interface security-realm="ManagementRealm">  <socket interface="management" port="${jboss.management.native.port:9999}"/>  </native-interface>  <!--  <http-interface security-realm="ManagementRealm">  <socket interface="management" port="${jboss.management.http.port:9990}"/>  </http-interface>  -->  **<http-interface security-realm="ManagementRealm">**  **<socket interface="management" secure-port="9990"/>**  **</http-interface>**  </management-interfaces> |

3.- Modificamos el archivo host.xml del slave

|  |
| --- |
| <security-realms>  <security-realm name="ManagementRealm">  <server-identities>  <secret value="U2xhdmUxMjM0JA=="/>  **<ssl protocol="TLSv1">**  **<keystore path="${jboss.domain.config.dir}/console.keystore" keystore-password="console" alias="console"/>**  **</ssl>**  </server-identities>  <authentication>  <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>  **<truststore path="${jboss.domain.config.dir}/console.keystore" keystore-password="console" alias="console"/>**  </authentication>  ….  ….  ….  </security-realms> |

|  |
| --- |
| <management-interfaces>  <native-interface security-realm="ManagementRealm">  <socket interface="management" port="${jboss.management.native.port:9999}"/>  </native-interface>  <!--  <http-interface security-realm="ManagementRealm">  <socket interface="management" port="${jboss.management.http.port:9990}"/>  </http-interface>  -->  </management-interfaces> |

**Configuración de la consola sobre https (modo standalone).**

1.- Creamos el certificado:

|  |
| --- |
| keytool -genkeypair -alias console -keyalg RSA -keystore console.keystore -storepass console --dname "CN=srv-vccs-portalapp-qa-01,OU=Infraestructura,O=credicard.com.ve,L=Caracas,S=DC,C=VE" -validity 30000 |

NOTA: CN debe ser igual al hostname del equipo al cual se le está realizando la configuración.

2.- Realizamos las siguientes modificaciones al archivo standalone.xml:

2.1 Agregamos el tag server-identities.

/core-service=management/security-realm=ManagementRealm/server-identity=ssl:add(protocol="TLSv1",keystore-path="/opt/jboss/jboss-eap-6.2/ssl/jboss\_console.keystore",keystore-password="console",alias="jboss\_console")

|  |
| --- |
| <security-realms>  <security-realm name="ManagementRealm">  **<server-identities>**  **<ssl protocol="TLSv1">**  **<keystore path="${jboss.server.config.dir}/console.keystore" keystore-password="console" alias="console"/>**  **</ssl>**  **</server-identities>**  <authentication>  <local default-user="$local"/>  <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>  </authentication>  <authorization map-groups-to-roles="false">  <properties path="mgmt-groups.properties" relative-to="jboss.server.config.dir"/>  </authorization>  </security-realm>  <security-realm name="ApplicationRealm">  <authentication>  <local default-user="$local" allowed-users="\*"/>  <properties path="application-users.properties" relative-to="jboss.server.config.dir"/>  </authentication>  <authorization>  <properties path="application-roles.properties" relative-to="jboss.server.config.dir"/>  </authorization>  </security-realm>  </security-realms> |

/core-service=management/security-realm=ManagementRealm/authentication=truststore:add(keystore-password="console",keystore-path="/opt/jboss/jboss-eap-6.2/ssl/jboss\_console.keystore")

2.2 Comentamos el tag socket-binding por http y agregadmos socket-binding por https.

/core-service=management/management-interface=http-interface:remove()

/core-service=management/management-interface=http-interface:add(secure-socket-binding="management-https",security-realm="ManagementRealm")

|  |
| --- |
| <management-interfaces>  <native-interface security-realm="ManagementRealm">  <socket-binding native="management-native"/>  </native-interface>  <http-interface security-realm="ManagementRealm">  <!--  <socket-binding http="management-http"/>  -->  **<socket-binding https="management-https"/>**  </http-interface>  </management-interfaces> |

2.3 En el tag de socket-binding-group comentamos las entradas de management-http y management-https, el primero porque es por donde se está prestando servicio actualmente, el segundo por https tiene definido el puerto 9443 y no es por ese puerto que deseamos prestar servicio. Agregamos la entrada que nos va a permitir la consola de administración por el puerto 9990 mediante https.

|  |
| --- |
| <socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">  <socket-binding name="management-native" interface="management" port="${jboss.management.native.port:9999}"/>  <!--  <socket-binding name="management-http" interface="management" port="${jboss.management.http.port:9990}"/>  <socket-binding name="management-https" interface="management" port="${jboss.management.https.port:9443}"/>  -->  **<socket-binding name="management-https" interface="management" port="${jboss.management.https.port:9990}"/>**  <socket-binding name="ajp" port="8009"/>  <socket-binding name="http" port="8080"/>  <socket-binding name="https" port="8443"/>  <socket-binding name="remoting" port="4447"/>  <socket-binding name="txn-recovery-environment" port="4712"/>  <socket-binding name="txn-status-manager" port="4713"/>  <outbound-socket-binding name="mail-smtp">  <remote-destination host="localhost" port="25"/>  </outbound-socket-binding>  </socket-binding-group> |

/socket-binding-group=standard-sockets/socket-binding=management-http:remove()

/socket-binding-group=standard-sockets/socket-binding=management-https:remove()

/socket-binding-group=standard-sockets/socket-binding=management-https:add(port="${jboss.management.https.port:9990}",interface="management")

**Configuración para dar servicio HTTPS (Modo dominio)**

1.- Para el caso de credicard pedimos a la unidad de criptografía un certificado pero que lo coloquen dentro de un almacen de certificado de formato .keystore, acá deben colocar el clave privada, el certificado y la ca. Una vez se tenga el archivo (para este caso se llama portalappclusqa.keystore) lo guardamos en la ruta **/opt/jboss/jboss-eap-6.2/domain/configuration** y se procede a realizar la siguiente modificación en el archivo domain.xml, específicamente en la sección subsystem xmlns="urn:jboss:domain:web:1.5" default-virtual-server="default-host", de cada uno de los profiles.

/profile=full/subsystem=web/connector=https:add(name="https",protocol="HTTP/1.1",scheme="https",socket-binding="https",enable-lookups=false,secure=true)

|  |
| --- |
| <connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" enable-lookups="false" secure="true">  <ssl name="conf-ssl" password="852789//4145assd" certificate-key-file="${jboss.domain.config.dir}/portalappclusqa.keystore" ca-certificate-file="${jboss.domain.config.dir}/portalappclusqa.keystore"/>  </connector> |

**Nota**: Como esta configuración se realizó bajo dominio, para los demás nodos solo debes copiar el certificado (archivo .keystore) en la ruta definida anteriormente.

**Configuración para dar servicio HTTPS (Modo standalone)**

1.- Debemos tener el certificado en formato .keystore, se lo pedimos a seguridad, pero en este caso vamos a generar un certificado temporal.

|  |
| --- |
| keytool -genkeypair -alias **portalpre-qa-01** -keyalg RSA -keystore **portalpre-qa-01.keystore** -storepass **password** --dname "CN=srv-vccs-portalapp-qa-01,OU=Infraestructura,O=credicard.com.ve,L=Caracas,S=DC,C=VE" -validity 30000 |

2.- Realizar la siguiente configuración en el archivo standalone.xml en el tag “<subsystem xmlns="urn:jboss:domain:web:1.5" default-virtual-server="default-host"”

|  |
| --- |
| <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http" **enabled="false"**/>  **<connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" enable-lookups="false" secure="true">**  **<ssl name="conf-ssl" password="password" certificate-key-file="${jboss.server.config.dir}/portalpre-qa-01.keystore" ca-certificate-file="${jboss.server.config.dir}/portalpre-qa-01.keystore"/>**  **</connector>** |

**Deshabilitando Welcome-root de JBOSS 6.**

1.- Para evitar dar información sobre la plataforma a terceras personas, es recomendable deshabilitar la página de bienvenida de jboss, para esto debemos editar el archivo domain.xml y modificar dicha propiedad para cada uno de los perfiles.

|  |
| --- |
| <virtual-server name="default-host" enable-welcome-root="false"> |

**Configuración iptables para los servidores JBOSS**.

Esta configuración se debe realizar para mitigar la vulnerabilidad de **Slow  
Denial Of Service Attack. (https://access.redhat.com/security/cve/CVE-2012-5568)**

Editamos el archivo /etc/sysconfig/iptables

|  |
| --- |
| \*filter  :INPUT ACCEPT [0:0]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [0:0]  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  #-A INPUT -i lo -j ACCEPT  -A INPUT -s 10.136.0.75 -j ACCEPT  **-A INPUT -p tcp --syn --dport 8443 -m connlimit --connlimit-above 20 -j REJECT**  **-A INPUT -p tcp --syn --dport 9999 -m connlimit --connlimit-above 20 -j REJECT**  **-A INPUT -p tcp --syn --dport 9990 -m connlimit --connlimit-above 20 -j REJECT**  -A INPUT -p udp -m udp -m state --state NEW --dport 161 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 8443 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 9990 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 9999 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 10050 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 3800 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 3802 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 3803 -j ACCEPT  -A INPUT -s 10.124.0.245 -p tcp -m state --state NEW -m tcp --dport 1433 -j ACCEPT  -A INPUT -j REJECT --reject-with icmp-host-prohibited  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  COMMIT |

**Configuración de usuario y roles en jboss.**

**(https://access.redhat.com/documentation/en-US/JBoss\_Enterprise\_Application\_Platform/6.2/html/Administration\_and\_Configuration\_Guide/Enabling\_Role-Based\_Access\_Control.html)**

1. Agregar los usuarios que utilizaran la consola administrativa. (Cr3d1c4rd$)

Nelson Pérez = nelson.perez

Andreina Rivaldo = andreina.rivaldo

Wilmer Canchica = wilmer.canchica

Joan Mogrovejo = joan.mogrovejo

zabbix = Usuario para el monitoreo de la máquina virtual de java mediante Zabbix (El password de este usuario es: “Z4bb1xUs3r.” ) Ojo sin las comillas. Debemos configurarle los roles monitor y audit.

|  |
| --- |
| #/opt/jboss/jboss-eap -6.2/bin/ add-user.sh  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 : wilmer.canchica  Password :  Re-enter Password :  What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:  About to add user ' wilmer.canchica ' for realm 'ManagementRealm'  Is this correct yes/no? yes  Added user 'wilmer.canchica ' to file '/opt/jboss/jboss-eap-6.2/standalone/configuration/mgmt-users.properties'  Added user 'wilmer.canchica ' to file '/opt/jboss/jboss-eap-6.2/domain/configuration/mgmt-users.properties'  Added user 'wilmer.canchica ' with groups to file '/opt/jboss/jboss-eap-6.2/standalone/configuration/mgmt-groups.properties'  Added user 'wilmer.canchica ' with groups to file '/opt/jboss/jboss-eap-6.2/domain/configuration/mgmt-groups.properties'  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? no |

2.- Activar the Role-Based Access Control (RABC) en la autenticación de usuarios.

|  |
| --- |
| #./bin/jboss-cli.sh controller=192.168.56.10 --connect  [domain@192.168.56.10:9999 /] /core-service=management/access=authorization:write-attribute(name=provider,value=rbac)  {  "outcome" => "success",  "response-headers" => {  "operation-requires-reload" => true,  "process-state" => "reload-required"  },  "result" => undefined,  "server-groups" => {"other-server-group" => {"host" => {"master" => {"server-three" => {"response" => {  "outcome" => "success",  "response-headers" => {  "operation-requires-restart" => true,  "process-state" => "**restart-required**"  }  }}}}}}  }  [domain@192.168.56.10:9999 /] exit  # /etc/init.d/jboss-eap stop  # /etc/init.d/jboss-eap start |

3.- Listamos los roles por defecto. (Nos conectaos a la consola de jboss antes de ejecutar estos comando)

|  |
| --- |
| [domain@192.168.56.10:9999 /] /core-service=management/access=authorization:read-children-names(child-type=role-mapping)  {  "outcome" => "success",  "result" => ["SuperUser"]  } |

4.- Como solo tenemos el ROL de SuperUser debemos crear el ROL de Deployer, Administrator y Auditor y Monitor.

|  |
| --- |
| [domain@192.168.56.10:9999 /] /core-service=management/access=authorization/role-mapping=Deployer:add |

5.- Listamos nuevamente.

|  |
| --- |
| [domain@192.168.56.10:9999 /] /core-service=management/access=authorization:read-children-names(child-type=role-mapping)  {  "outcome" => "success",  "result" => [  **"Deployer",**  "SuperUser"  ]  } |

6.- Agregamos nuestros usuarios al ROL de Administrator y el usuario admin al Rol de SuperUser

mapping=SuperUser: Este es el nombre del ROL al cual deseamos asociar el usuario.

include=user-admin: Este es el alias para el usuario, RedHat recomienda que utilice user-xxxx.

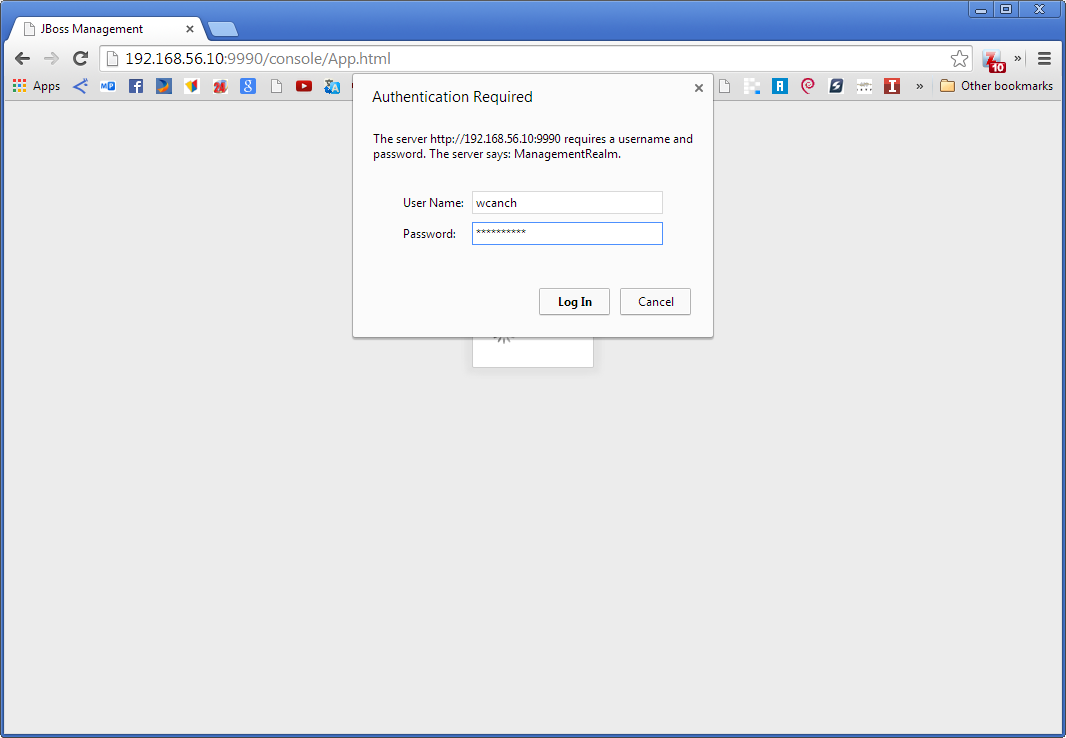
|  |
| --- |
| [domain@192.168.56.10:9999 /] /core-service=management/access=authorization/role-**mapping=SuperUser**/**include=user-admin**:add(name=admin,type=user)  {  "outcome" => "success",  "result" => undefined,  "server-groups" => {"other-server-group" => {"host" => {"master" => {"server-three" => {"response" => {"outcome" => "success"}}}}}}  } |

7.- Listamos los usuarios para verificar que se encuentra asociado al ROL que deseamos.

|  |
| --- |
| [domain@192.168.56.10:9999 /] /core-service=management/access=authorization/role-mapping=SuperUser:read-resource(recursive=true)  {  "outcome" => "success",  "result" => {  "include-all" => false,  "exclude" => undefined,  "include" => {  "user-$local" => {  "name" => "$local",  "realm" => undefined,  "type" => "USER"  },  "user-admin" => {  "name" => "admin",  "realm" => undefined,  "type" => "USER"  },  **"user-admin" => {**  **"name" => "admin",**  **"realm" => undefined,**  **"type" => "USER"**  **}**  }  }  } |

NOTA: Solo coloque el ejemplo del usuario admin, para los demás usuarios es exactamente igual, solo debemos cambiar el nombre del ROL y el usuario.

8.- Verificamos el acceso accediendo a la consola de administración.



Comandos Varios

|  |
| --- |
| **Connectarse a la consola**  jboss-cli.sh controller=192.168.56.10 --connect  **Habiliar RBAC Role-Based Access Control**  /core-service=management/access=authorization:write-attribute(name=provider, value=rbac)  **Deshabilitar RBAC Role-Based Access Control**  /core-service=management/access=authorization:write-attribute(name=provider, value=simple)    **Agregar el usuario admin al role SuperUser**  /core-service=management/access=authorization/role-mapping=SuperUser/include=user-admin:add(name=admin,type=user)  **Agregar el usuario desplegador al role Deployer**  /core-service=management/access=authorization/role-mapping=Deployer/include=user-desplegador:add(name=desplegador,type=user)    **Leear los usuarios asigandos al role SuperUser**  /core-service=management/access=authorization/role-mapping=SuperUser:read-resource(recursive=true)    **Leer los roles creados**  /core-service=management/access=authorization:read-children-names(child-type=role-mapping)      **Eliminar user from Role**  /core-service=management/access=authorization/role-mapping=SuperUser/include=user-nperez:remove |

**Configuración de Deep Copy Subject Mode**

Activar el parámetro Deep Copy Subject Mode, fue una solicitud del grupo de seguimiento y control debido a una investigación sobre los consejos de seguridad sobre JBOSS, este parámetro viene desactivado por defecto.

1.- Para verificar el status actual del parámetro. (Previamente conectado a la consola de administración de JBOSS CLI)

Domain mode

|  |
| --- |
| # /profile=full/subsystem=security:read-attribute(name=deep-copy-subject-mode)  {  "outcome" => "success",  **"result" => false**  } |

Standalone mode

|  |
| --- |
| # /subsystem=security:read-attribute(name=deep-copy-subject-mode)  {  "outcome" => "success",  **"result" => false**  } |

2.- Activamos Deep Copy Subjecdt Mode

Domain mode

|  |
| --- |
| # /profile=full/subsystem=security:write-attribute(name=deep-copy-subject-mode,value=true)  {  "outcome" => "success",  "result" => undefined,  "server-groups" => undefined  } |

Standalone mode

|  |
| --- |
| #/subsystem=security:write-attribute(name=deep-copy-subject-mode,value=true) |

Después de realizar esta actividad debemos reiniciar el servicio.

**Configuración Mail en jboss.**

1. Debemos editar el archivo standalone.xml si JBOSS está ejecutando en modo standalone, o domain.xml si JBOSS se ejecutando en modo dominio. En modo dominio se debe realizar la configuración para cada uno de los perfiles. (Los puntos se debe a que existe otra configuración que es obviada en estas notas)

/profile=default/subsystem=mail/mail-session=java\:\/Mail:add(jndi-name="java:/Mail",from="no-responder@credicard.com.ve",debug="false")

/profile=default/subsystem=mail/mail-session=java\:\/Mail/server=smtp:add(outbound-socket-binding-ref="mail-smtp")

---

/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=mail-smtp:add(host="10.133.0.63",port="25")

--

/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=mail-smtp:read-attribute(name=host)'

/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=mail-smtp:remove()

|  |
| --- |
| <subsystem xmlns="urn:jboss:domain:mail:1.1">  <mail-session jndi-name="java:/Mail" debug="true" from="no-responder@credicard.com.ve">  <smtp-server outbound-socket-binding-ref="mail-smtp"/>  </mail-session>  </subsystem>  .  .  .  .  .  <socket-binding-group name="standard-sockets" default-interface="public">  .  .  .  <outbound-socket-binding name="mail-smtp">  <remote-destination host="**hubtrannlb.credicard.com.ve**" port="25"/>  </outbound-socket-binding>  .  .  .  </socket-binding-group> |

**Configuración HTTP TimeOut**

Para cambiar el HTTP TimeOunt que viene por defecto 60000 ms debemos ejecutar:

|  |
| --- |
| jboss-cli.sh controller=$IP --connect --commands="/system-property=org.apache.coyote.http11.DEFAULT\_CONNECTION\_TIMEOUT:add(value=8000)" |

Esto agrega la siguiente entrada en el archive domain.xml (si estas ejecutando en domain mode) o en standalone.xml (si estas ejecutando en standalone mode)

Stanalone mode

|  |
| --- |
| <server….>  .  .  .  .  <system-properties>  <property name="org.apache.coyote.http11.DEFAULT\_CONNECTION\_TIMEOUT" value="8000" />  </system-properties>  .  .  .  </server> |

Domain mode

|  |
| --- |
| <domain….>  .  .  .  .  <system-properties>  <property name="org.apache.coyote.http11.DEFAULT\_CONNECTION\_TIMEOUT" value="8000" />  </system-properties>  .  .  .  </domain> |

Configuración Audit Loggin Management desde el CLI (standalone)

Fuente: <https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.2/html/Administration_and_Configuration_Guide/Enable_Management_Interface_Audit_Logging_from_the_Management_CLI.html>

1.- Ejecutar en el CLI:

/core-service=management/access=audit/logger=audit-log:write-attribute(name=enabled,value=true)

2.- Por defecto los logs de auditoria se guardan en EAP\_HOME/standalone/data/audit-log.log

Nota:

Podemos revisar la configuración en el archivo standalone.xml (o el xml que corresponda) con la siguiente información por defecto:

|  |
| --- |
| <audit-log>  <formatters>  <json-formatter name="json-formatter"/>  </formatters>  <handlers>  <file-handler name="file" formatter="json-formatter" path="audit-log.log" relative-to="jboss.server.data.dir"/>  </handlers>  <logger log-boot="true" log-read-only="false" enabled="true">  <handlers>  <handler name="file"/>  </handlers>  </logger>  </audit-log> |

También la podemos consultar via CLI con el comando:

|  |
| --- |
| **[standalone@192.168.156.10:9999 /] /core-service=management/access=audit:read-resource(recursive=true)**  {  "outcome" => "success",  "result" => {  "file-handler" => {"file" => {  "formatter" => "json-formatter",  "max-failure-count" => 10,  "path" => "audit-log.log",  "relative-to" => "jboss.server.data.dir"  }},  "json-formatter" => {"json-formatter" => {  "compact" => false,  "date-format" => "yyyy-MM-dd HH:mm:ss",  "date-separator" => " - ",  "escape-control-characters" => false,  "escape-new-line" => false,  "include-date" => true  }},  "logger" => {"audit-log" => {  "enabled" => true,  "log-boot" => true,  "log-read-only" => false,  "handler" => {"file" => {}}  }},  "syslog-handler" => undefined  }  } |

Configuración Audit Loggin Management desde el CLI (dominio)

1.- Para consultar si está habilitado podemos consultar:

/host=jboss-clust-1/core-service=management/access=audit/logger=audit-log:read-resource

|  |
| --- |
| {  "outcome" => "success",  "result" => {  **"enabled" => false,**  "log-boot" => true,  "log-read-only" => false,  "handler" => {"host-file" => undefined}  }  } |

2.- Para Habilitar.

[domain@192.168.156.10:9999 /] /host=jboss-clust-1/core-service=management/access=audit/logger=audit-log:write-attribute(name=enabled,value=true)

**NOTA**: Si el dominio de jboss tiene dos nodos estos pasos deben ejecutarse para ambos, desde la misma consola, solo debemos cambiar el valor del parámetro host.

|  |
| --- |
| {  "outcome" => "success",  "result" => undefined,  "server-groups" => undefined  } |

3.- Ver toda la configuración

[domain@192.168.156.10:9999 /] /host=jboss-clust-1/core-service=management/access=audit:read-resource(recursive=true)

|  |
| --- |
| {  "outcome" => "success",  "result" => {  "file-handler" => {  **"host-file" => {**  "formatter" => "json-formatter",  "max-failure-count" => 10,  **"path" => "audit-log.log",**  **"relative-to" => "jboss.domain.data.dir"**  },  **"server-file" => {**  "formatter" => "json-formatter",  "max-failure-count" => 10,  "path" => "audit-log.log",  "relative-to" => "jboss.server.data.dir"  }  },  "json-formatter" => {"json-formatter" => {  "compact" => false,  "date-format" => "yyyy-MM-dd HH:mm:ss",  "date-separator" => " - ",  "escape-control-characters" => false,  "escape-new-line" => false,  "include-date" => true  }},  **"logger" => {"audit-log" => {**  **"enabled" => true,**  **"log-boot" => true,**  "log-read-only" => false,  "handler" => {"host-file" => {}}  }},  "server-logger" => {"audit-log" => {  "enabled" => false,  "log-boot" => true,  "log-read-only" => false,  "handler" => {"server-file" => {}}  }},  "syslog-handler" => undefined  }  } |

4.- Los archivos de auditoria lo tienen guardado en: /opt/jboss/jboss-eap-6.2/domain/data/audit-log.log

**Remove Silent authentication with de Management CLI.**

Para que la consola de administración pida usuario y password en caso de que varias personas tengan acceso al a consola de Sistema Operativo.

1.- /core-service=management/security-realm=ManagementRealm/authentication=local:remove

/core-service=management/security-realm=ApplicationRealm/authentication=local:remove

Antes:

|  |
| --- |
| <authentication>  <local default-user="$local" />  <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>  <!--SSL\_CONFIG  <truststore path="/opt/jboss/jboss-eap-6.2/ssl/jboss\_console.keystore" keystore-password="console" alias="jboss\_console"/>  SSL\_CONFIG-->  </authentication> |

**Trabajando con server-groups y servers:**

**Para crear el grupo**

|  |
| --- |
| /server-group=portalappclusqa-group2:add(profile=default,socket-binding-group=standard-sockets,socket-binding-port-offset=0) |

Para configurar opciones de la máquina virtual de java a los servidores que pertenezcan a este grupo.

|  |
| --- |
| /server-group=portalappclusqa-group1/jvm=default:add(heap-size=768m,max-heap-size=768m,max-permgen-size=256m, permgen-size=256m) |

Cambiamos la configuración del server al nuevo grupo.

|  |
| --- |
| /host=srv-vccs-portalapp-qa-01/server-config=nodo-1:write-attribute(name=group,value=portalappclusqa-group1) |

Debemos reiniciar el server para que tome los cabios

|  |
| --- |
| reload --host=srv-vccs-portalapp-qa-01 |

**Agregando un servidor Nuevo:**

|  |
| --- |
| /host=srv-vccs-portalapp-qa-02/server-config=nodo-1:add(auto-start=true,socket-binding-group=standard-sockets,socket-binding-port-offset=100,group=portalappclusqa-group2) |

Deteniendo e Iniciando los servidores que pertenecen a un server group.

|  |
| --- |
| /server-group=portalappclusqa-group2:stop-servers  /server-group=portalappclusqa-group2:start-servers |

Iniciar/Detener el server de un grupo específico:

|  |
| --- |
| /host=srv-vccs-portalapp-qa-02/server-config=nodo-1:**start**(server=nodo-1,blocking=false)  /host=srv-vccs-portalapp-qa-02/server-config=nodo-1:**stop**(server=nodo1,blocking=false) |