|  |  |  |  |
| --- | --- | --- | --- |
| Tech **Mahindra**  DOCUMENTO DEL PROCESO DE INSTALACIÓN DE CERTIFICADOS  DEVOPS PRODUCTION PLATFORM    **PresentADO POR:**  Alberto Velasco, Christopher Joel carrillo, Daniel Torres Velasco  **PresentADO PaRa**: SCOTIABANK México  **ESTADO**: Draft  July 14, 2017  Versión 1.0 |  |  |  |

**CONTROL DE CAMBIOS**

**REGISTRO DE CAMBIOS**

|  |  |  |  |
| --- | --- | --- | --- |
| **AUTOR** | **FECHA** | **VERSIÓN** | **CAMBIO** |
| Alberto Velasco | 14-Julio-2017 | 1.0 | Documento inicial |
| Alberto Velasco | 17-Julio-2017 | 1.1 | Modificación de Documento |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**REVISORES**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE** | **FECHA** | **VERSIÓN** | **CARGO** |
| Humberto Ramírez | 17-Julio-2017 | 1.1 | DevOps Technical Lead |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**PROPOSITO DEL DOCUMENTO**

|  |  |
| --- | --- |
| **AUDIENCIA** | **OBEJTIVO** |
| Centro de Excelencia DevOps Scotiabank México | El presente documento tiene como objetivo documentar paso a paso la instalación de certificados ssl para **DEVOPS PRODUCTION PLATFORM**, realizada para SCOTIABANK México. |

**NOTA DE CONFIDENCIALIDAD**

Toda la información contenida en este documento es proporcionada con confidencialidad con el único propósito de acercar posturas comerciales con la entidad receptora prohibiendo expresamente la publicación, divulgación y utilización de su contenido para fines propios o de terceros no autorizados sin el previo consentimiento expreso y por escrito de Tech Mahindra.

<http://www.techmahindra.com/General/privacy_policy.aspx>

**CONTENIDO**

[INSTALACIÓN CAPA DE DATOS 4](#_Toc488084923)

[POSTGRESQL 4](#_Toc488084924)

[MICROSOFT SQLSEVER 7](#_Toc488084925)

[INSTALACIÓN CAPA DE APLICACIÓN 8](#_Toc488084926)

[JENKINS 8](#_Toc488084927)

[ATLASSIAN CROWD 10](#_Toc488084928)

[ATLASSIAN LDAP CROWD 12](#_Toc488084929)

[ATLASSIAN JIRA 13](#_Toc488084930)

[ATLASSIAN CONFLUENCE 17](#_Toc488084931)

[ATLASSIAN BITBUCKET 20](#_Toc488084932)

[SONARQUBE 23](#_Toc488084933)

[ARTIFACTORY 24](#_Toc488084934)

[CA SERVICE VIRTUALIZATION 27](#_Toc488084935)

[CONEXIÓN SERVIDOR ESCLAVO 1 (LINUX) CON EL SERVIDOR JENKINS MAESTRO 29](#_Toc488084936)

[XL RELEASE 30](#_Toc488084937)

[XL DEPLOY 31](#_Toc488084938)

[CONEXIÓN SERVIDOR ESCLAVO 1 (LINUX) CON EL SERVIDOR JENKINS MAESTRO 32](#_Toc488084939)

[CONEXIÓN SERVIDOR ESCLAVO 1 (GIT) CON EL SERVIDOR BITBUCKET 33](#_Toc488084940)

[CONEXIÓN ESCLAVO 2 Y 3 (WINDOWS SERVER) CON EL SERVIDOR JENKINS MAESTRO 34](#_Toc488084941)

[CONEXIÓN JENKINS ESCLAVO 2 (GIT) CON EL SERVIDOR BITBUCKET 36](#_Toc488084942)

[ÍNDICE DE TABLAS 38](#_Toc488084943)

# INSTALACIÓN CAPA DE DATOS

Los procedimientos descritos en esta sección deberán realizarse en todos los servidores que formen parte de la **capa de datos** del sistema **DEVOPS PRODUCTION PLATFORM**.

## POSTGRESQL

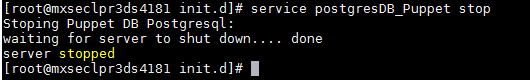
La instalación fue realizada de manera similar para todos los servidores que contienen una base de datos en **PostgreSql,** enseguida se muestra la tabla.

|  |  |  |
| --- | --- | --- |
| **SERVIDOR** | **APLICACIÓN** | **INSTANCIA** |
| SERVIDOR 15 | **PostgreSQL** | **Atlassian Crowd**  **Atlassian JIRA** |
| SERVIDOR 16 | **PostgreSQL** | **Atlassian Confluence**  **SonarQube** |
| SERVIDOR 17 | **PostgreSQL** | **Atlassian BitBucket** |
| SERVIDOR 18 | **PostgreSQL** | **Artifactory** |
| SERVIDOR 19 | **PostgreSQL** | **Puppet Enterprise** |

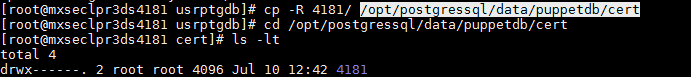
Tabla 1 Instancias Postgresql

Se mostrara como ejemplo la instalación del certificado realizado para la instancia del componente **Puppet Enterprise**, considerando que la generación de certificados fue entregada previamente y depositados en el servidor a instalar.

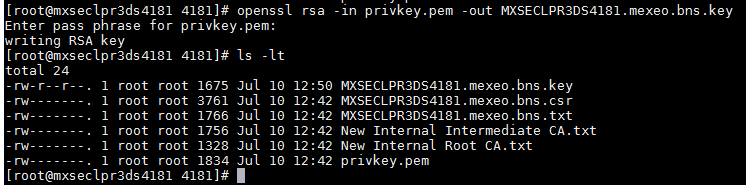
* El primer paso es detener el servicio de la instancia.



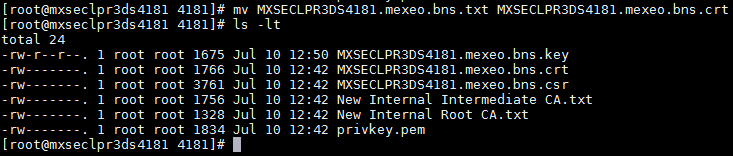
* Después recurrimos a copiar la carpeta que contiene los certificados que nos fueron entregados.



* Lo siguiente es crear el archivo **.key** desde el archivo **.pem** la contraseña que se solicita es con la que fue creado el archivo **.csr**.



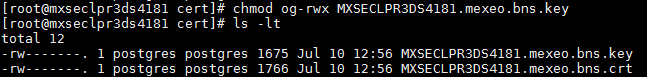
* Cambiaremos la extensión del certificado por **.crt** ya que fue entregado como **.txt**.



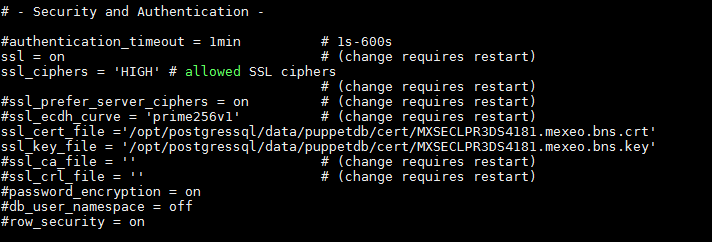
* Después cambiaremos el usuario y el grupo de la carpeta que contiene los certificados para que la instancia los pueda leer.

C:\Users\s3368663\Pictures\Screenpresso\Produccion\CertificadosProd\bit\PostgresPuppet\2017-07-10_12h57_40.png

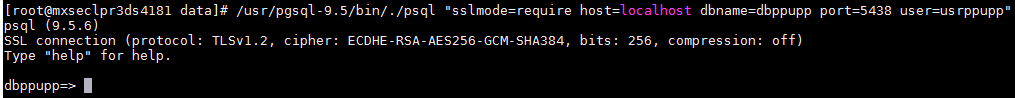
* Lo siguiente será cambiar los privilegios para el archivo **.key** creado anteriormente.



* El siguiente paso es modificar el archivo **postgresql.conf** para a indicarle a la instancia donde está ubicado nuestro archivo **.key** y el archivo **.crt**.



* El último paso será comprobar la conexión exitosa, para esto iniciaremos el servicio de la instancia y nos conectaremos a la base, verificando el tipo de conexión.



Todos los pasos descritos anteriormente se tienen que realizar repetidamente para todas las instancias mencionadas en la Tabla Intancias Postgresql.

## MICROSOFT SQLSEVER

La instalación fue realizada solo con un certificado ya que se ocupo solo una instancia para tres herramentas.

|  |  |  |
| --- | --- | --- |
| **SERVIDOR** | **APLICACIÓN** | **HERRAMIENTA** |
| SERVIDOR 20 | **PostgreSQL** | **XL Deploy**  **XL Release**  **CA Service Virtualization** |

Tabla 2 Instancias Microsoft Sql

# INSTALACIÓN CAPA DE APLICACIÓN

Los procedimientos descritos en esta sección deberán realizarse en todos los servidores que formen parte de la capa de **aplicación del sistema** **DEVOPS PRODUCTION PLATFORM**.

Asimismo es importante destacar que los procedimientos descritos en el presente documento parten considerando que la generación de certificados fue entregada previamente y depositados en el servidor a instalar.

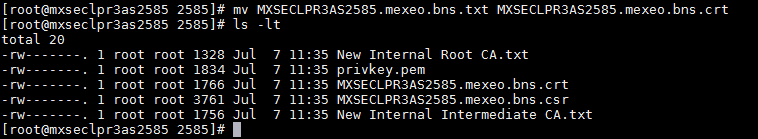
## JENKINS

Los siguientes pasos son los que se deben de realizar para hacer la instalación de certificado **ssl** en la herramienta que utilizaremos para la orquestación.

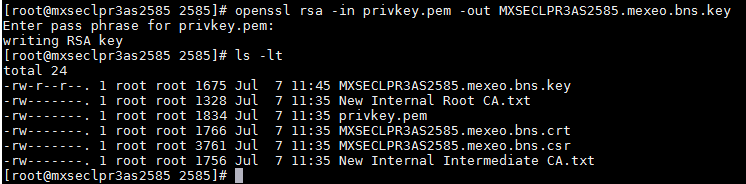
* El primer paso es detener el servicio.



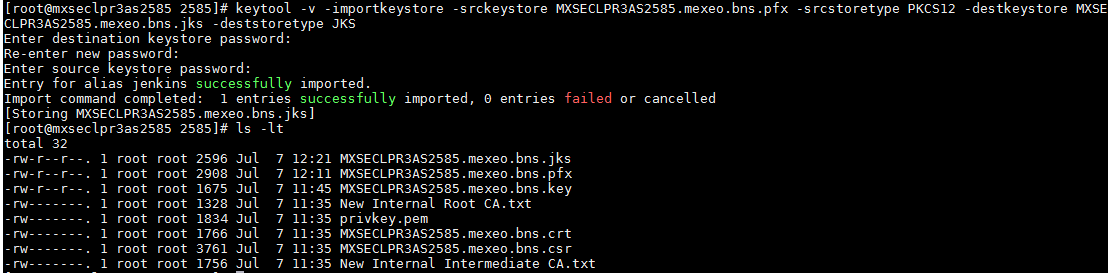
* Realizaremos el cambio de extensión al certificado por **.crt** ya que fue entregado con la extensión **.txt**.



* Después procedemos a generar el archivo **.key** desde el archivo **.pem**.



* El siguiente paso será crear el archivo **.pfx** para después generar el archivo **.jks** que se ocupara para indicar el certificado.



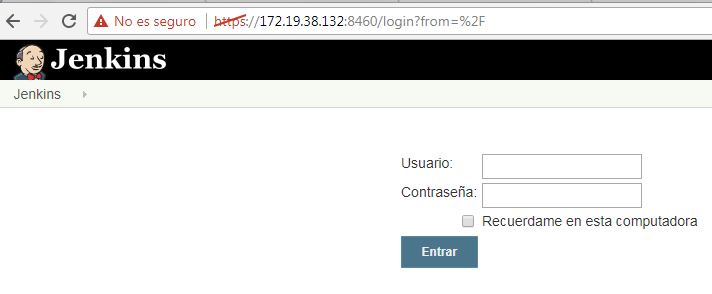
* Después indicaremos en el archivo **jenkins** ubicado **/etc/sysconfig** la ubicación del certificado y el puerto que se ocupara para el protocolo https.

C:\Users\s3368663\Pictures\Screenpresso\Produccion\CertificadosProd\bit\jenkins\2017-07-07_12h52_15.png

* Iniciaremos el servicio nuevamente.



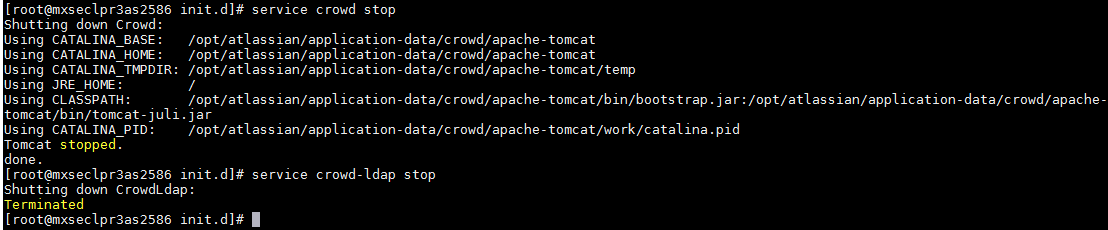
* Como último paso comprobaremos que el puerto se encuentre funcionando.



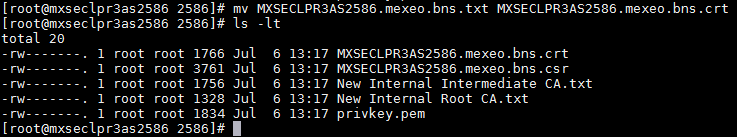
## ATLASSIAN CROWD

Los siguientes pasos son los que se deben de realizar para hacer la instalación de certificado **ssl** en la herramienta **Crowd**.

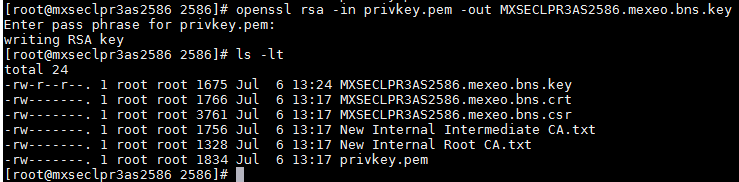
* El primer paso es detener el servicio.



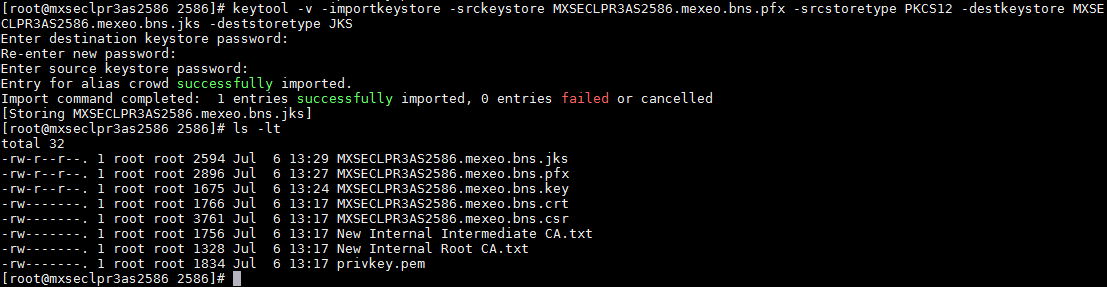
* Realizaremos el cambio de extensión al certificado por **.crt** ya que fue entregado con la extensión **.txt**.



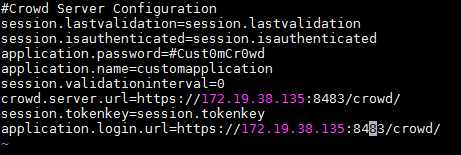
* Después procedemos a generar el archivo **.key** desde el archivo **.pem**.



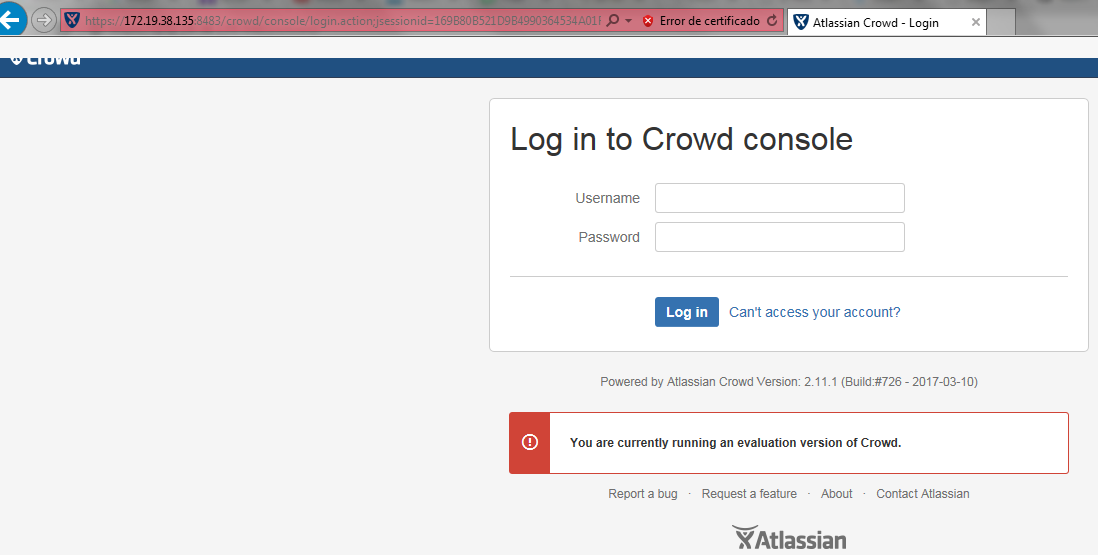
* El siguiente paso será crear el archivo **.pfx** para después generar el archivo **.jks** que se ocupara para indicar el certificado.



* Después indicaremos en el archivo **crowd.properties** la ubicación del certificado y el puerto que se ocupara para el protocolo https.



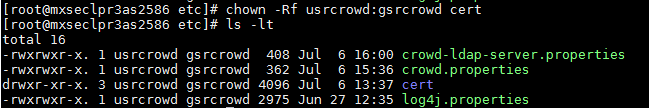
* Iniciaremos el servicio nuevamente.
* Como último paso comprobaremos que el puerto se encuentre funcionando.



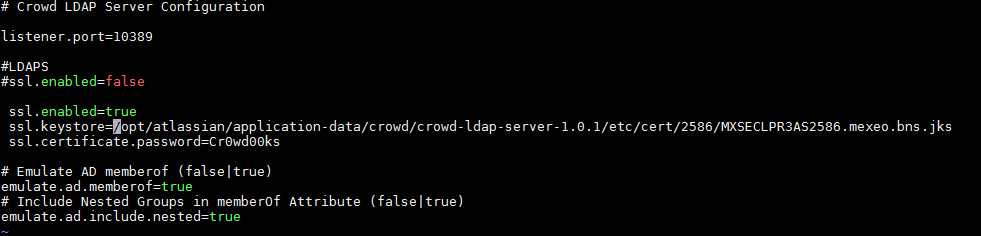
## ATLASSIAN LDAP CROWD

Los siguientes pasos son los que se deben de realizar para hacer la instalación de certificado **ssl** en la herramienta **LDAP** **Crowd**.

* El primer paso es detener el servicio.
* Como el certificado ya fue realizado anteriormente para **Crowd** solo copiaremos la carpeta **cert** en la Dirección donde está ubicado el LDAP.
* Cambiaremos el usuario y grupo de la carpeta.



* Después indicaremos en el archivo **crowd-ladap-server.properties** la ubicación del certificado.

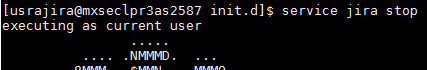


* Iniciaremos el servicio nuevamente.

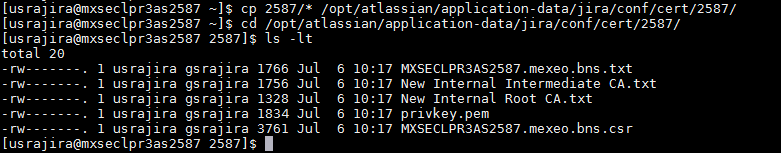
## ATLASSIAN JIRA

A continuación realizaremos la instalación del certificado para la herramienta **JIRA** para ello realizaremos los siguientes pasos.

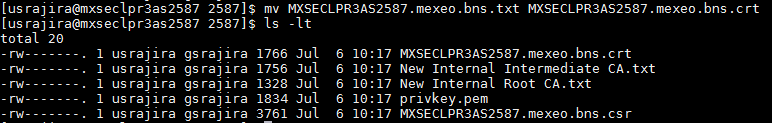
* El primer paso es detener el servicio.



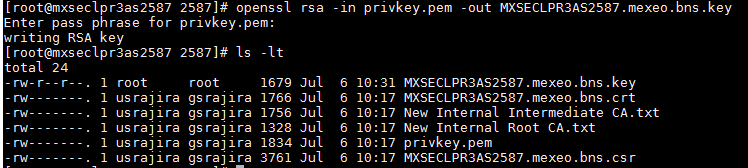
* Copiaremos los certificados previamente depositados en el servidor en la carpeta cert.



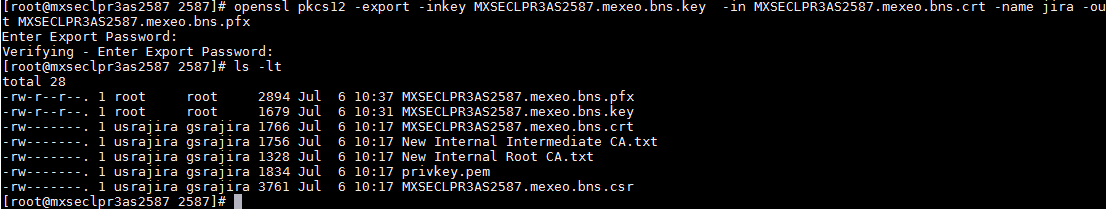
* Cambiamos la extensión **.txt** a **.crt** del certificado.



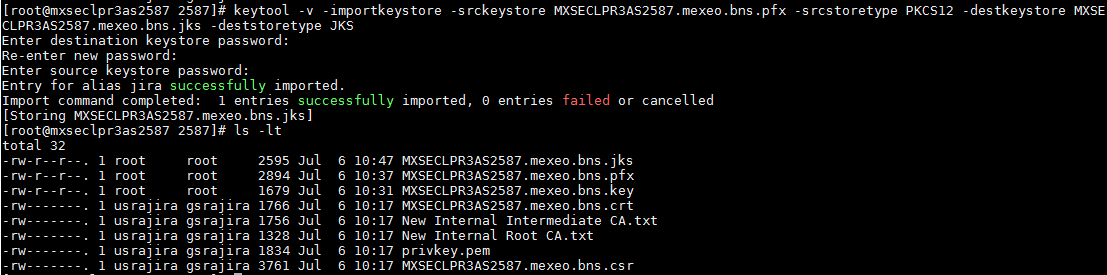
* Lo siguiente es crear el archivo **.key** desde el **.pem.**



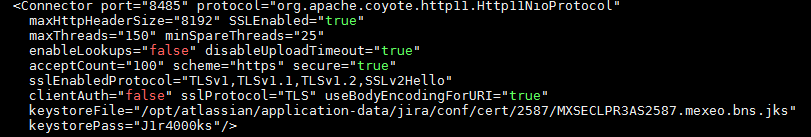
* Después creamos el archivo **.pfx** con el archivo **.key** y el archivo **.crt.**



* El siguiente paso es crear el archivo **.jks** desde el archivo **.pfx**.



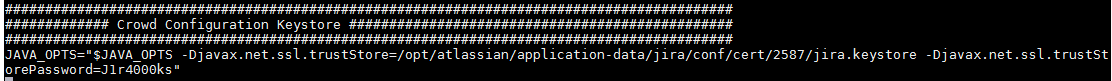
* Ahora modificaremos el archivo **server.xml** e indicaremos la ubicación del certificado y el puerto.



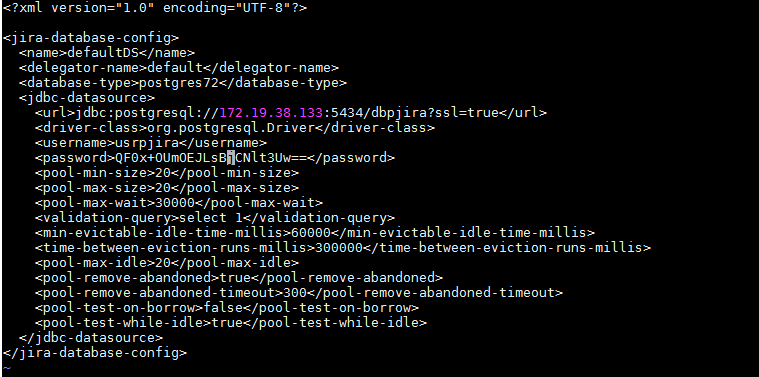
* El siguiente paso será la integración de la herramienta con su base de datos para esto tomaremos el **.crt** de la base de datos y lo depositaremos en el servidor. Despues crearemos el archivo **.der** con el comando **openssl x509 -in server.crt -out server.crt.der** -outform dery con el comando **keytool -v -importkeystore -srckeystore server.crt.der -srcstoretype JKS -destkeystore jira.keystore -deststoretype JKS** crearemos el archivo **.keystore** de la misma manera se integraran todas las herramientas con las que esté integrada la herramienta.



* Ahora indicaremos el lugar del certificado en el archivo catalina.sh

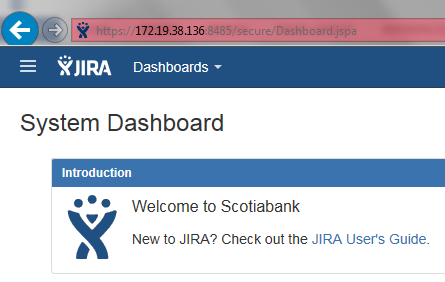


* Lo siguiente será indicar en el archivo dbconfig.xml



* Después iniciaremos el servicio y comprobaremos el funcionamiento de la herramienta.

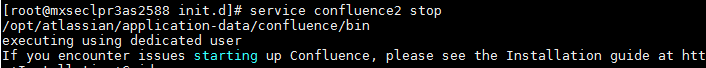




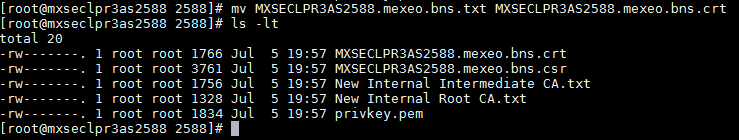
## ATLASSIAN CONFLUENCE

A continuación realizaremos la instalación del certificado.

* El primer paso es detener el servicio.



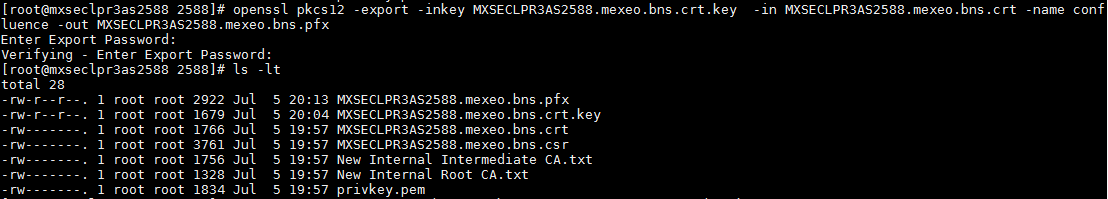
* Cambiamos la extensión **.txt** a **.crt** del certificado.



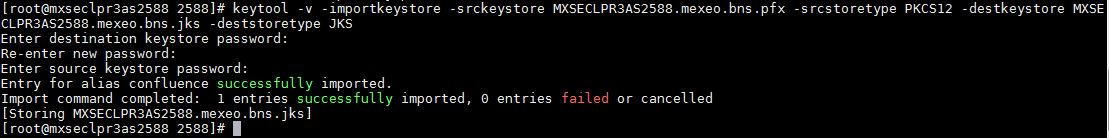
* Lo siguiente es crear el archivo **.key** desde el **.pem.**

C:\Users\s3368663\Pictures\Screenpresso\Produccion\CertificadosProd\bit\Confluence cert\2017-07-05_20h17_27.png

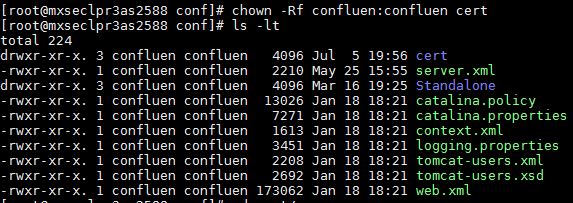
* Después creamos el archivo **.pfx** con el archivo **.key** y el archivo **.crt.**



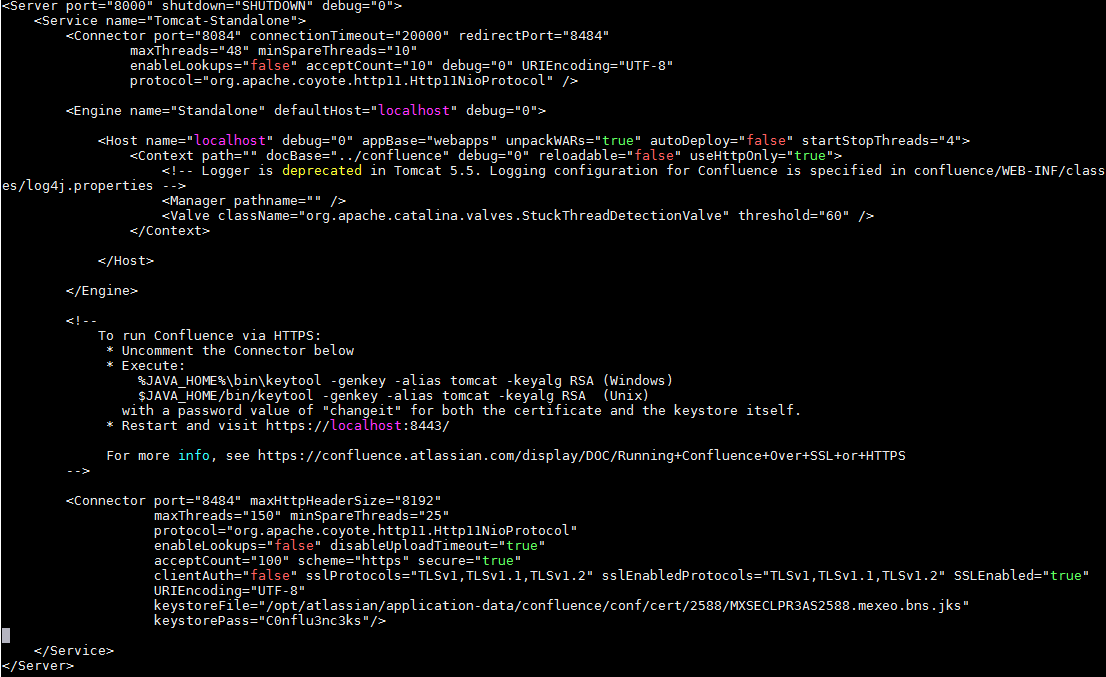
* El siguiente paso es crear el archivo **.jks** desde el archivo **.pfx**.



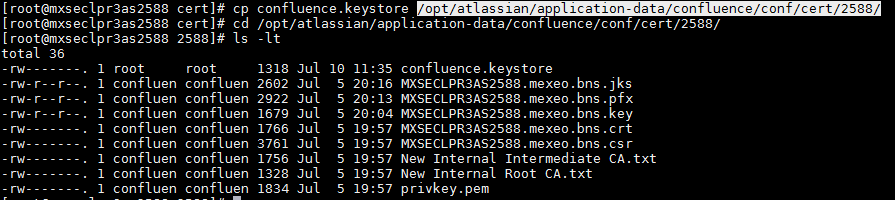
* Ahora cambiaremos los el usuario y grupo de la carpeta cert.



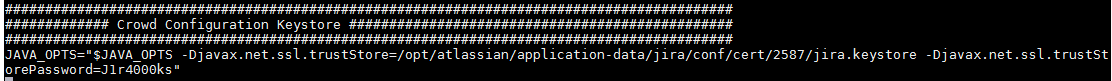
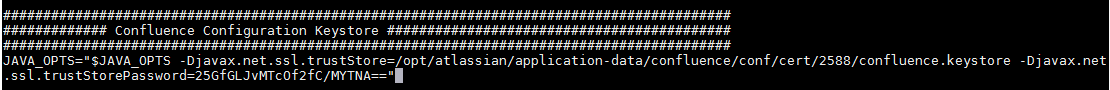
* Ahora modificaremos el archivo **server.xml** e indicaremos la ubicación del certificado y el puerto.



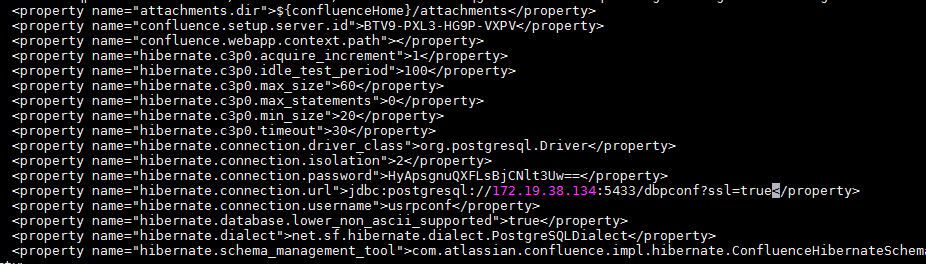
* El siguiente paso será la integración de la herramienta con su base de datos para esto tomaremos el **.crt** de la base de datos y lo depositaremos en el servidor. Despues crearemos el archivo **.der** con el comando **openssl x509 -in server.crt -out server.crt.der** -outform dery con el comando **keytool -v -importkeystore -srckeystore server.crt.der -srcstoretype JKS -destkeystore jira.keystore -deststoretype JKS** crearemos el archivo **.keystore** de la misma manera se integraran todas las herramientas con las que esté integrada la herramienta.



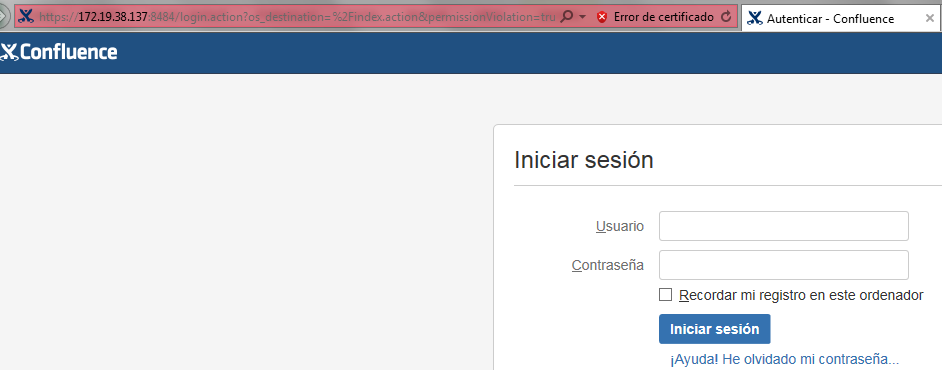
* Ahora indicaremos el lugar del certificado en el archivo **catalina.sh**



* Lo siguiente será indicar en el archivo **confluence.cfg.xml**



* Después iniciaremos el servicio y comprobaremos el funcionamiento de la herramienta.



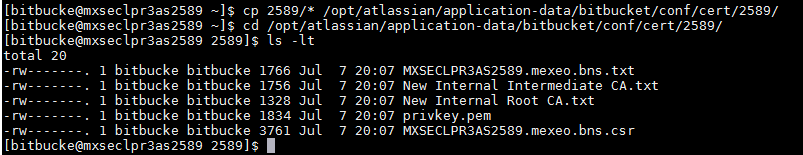
## ATLASSIAN BITBUCKET

A continuación realizaremos la instalación del certificado.

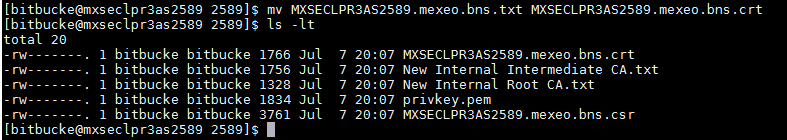
* El primer paso es detener el servicio.

C:\Users\s3368663\Pictures\Screenpresso\Produccion\CertificadosProd\bit\bitbuketkeystore\2017-07-10_13h42_11.png

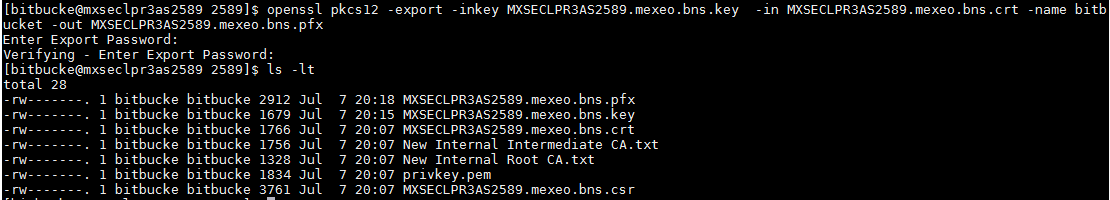
* El primer paso es copiar la carpeta que contiene el certificado.



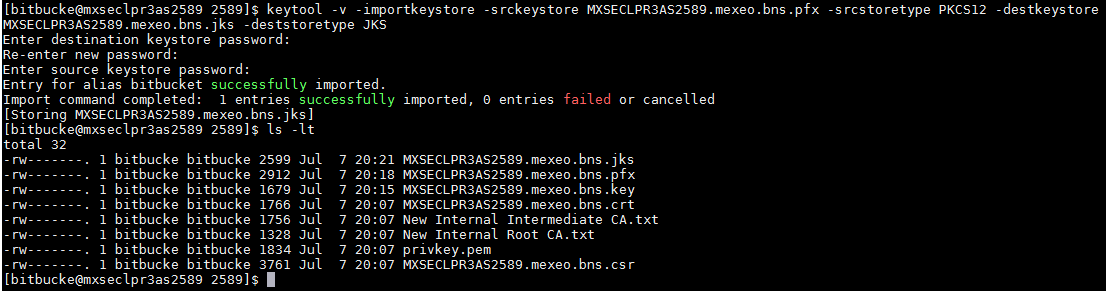
* Cambiamos la extensión **.txt** a **.crt** del certificado.



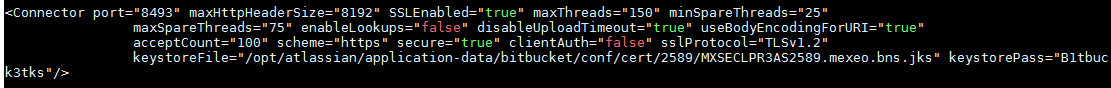
* Lo siguiente es crear el archivo **.key** desde el **.pem.** Con el comando **openssl rsa -in privkey.pem -out serverCA.key**
* Después creamos el archivo **.pfx** con el archivo **.key** y el archivo **.crt.**



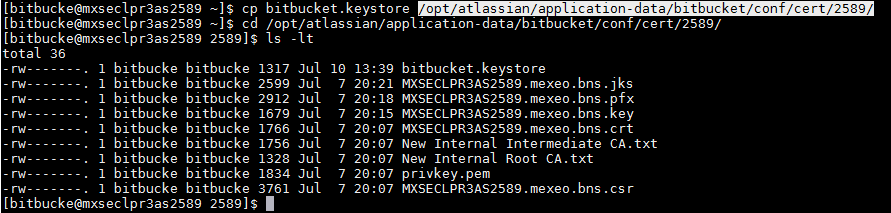
* El siguiente paso es crear el archivo **.jks** desde el archivo **.pfx**.



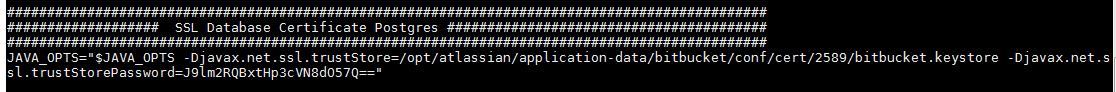
* Ahora modificaremos el archivo **server.xml** e indicaremos la ubicación del certificado y el puerto.



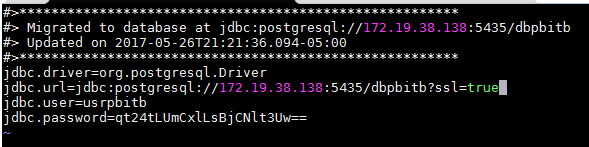
* El siguiente paso será la integración de la herramienta con su base de datos para esto tomaremos el **.crt** de la base de datos y lo depositaremos en el servidor. Despues crearemos el archivo **.der** con el comando **openssl x509 -in server.crt -out server.crt.der** -outform dery con el comando **keytool -v -importkeystore -srckeystore server.crt.der -srcstoretype JKS -destkeystore jira.keystore -deststoretype JKS** crearemos el archivo **.keystore** de la misma manera se integraran todas las herramientas con las que esté integrada la herramienta.



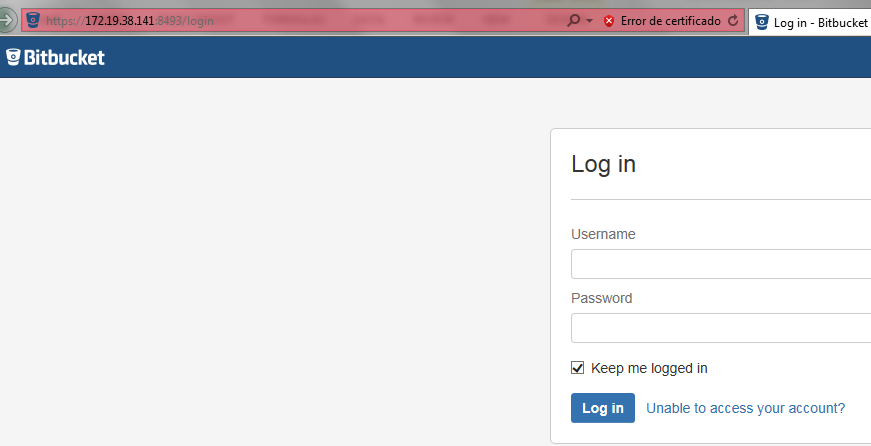
* Ahora indicaremos el lugar del certificado en el archivo catalina.sh



* Lo siguiente será indicar en el archivo bitbucket.properties



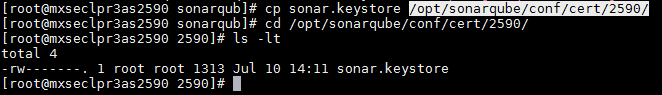
* Después iniciaremos el servicio y comprobaremos el funcionamiento de la herramienta.



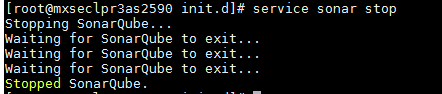
## SONARQUBE

A continuación mostraremos la integración de la herramienta con su base de datos.

* Para esto tomaremos el archivo **.crt** de la base de datos y lo depositaremos en el servidor. Después crearemos el archivo **.der** con el comando **openssl x509 -in server.crt -out server.crt.der** **-outform der** y con el comando **keytool -v -importkeystore -srckeystore server.crt.der -srcstoretype JKS -destkeystore jira.keystore -deststoretype JKS** crearemos el archivo **.keystore** de la misma manera se integraran todas las herramientas con las que esté integrada la herramienta.



* Detenemos el servicio



* Ahora especificaremos el certificado en el archivo **sonar.properties**

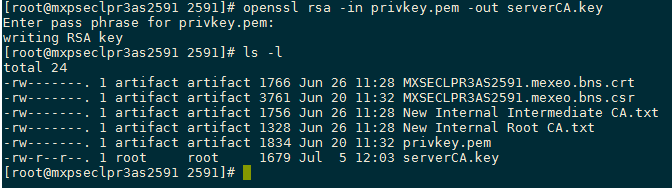




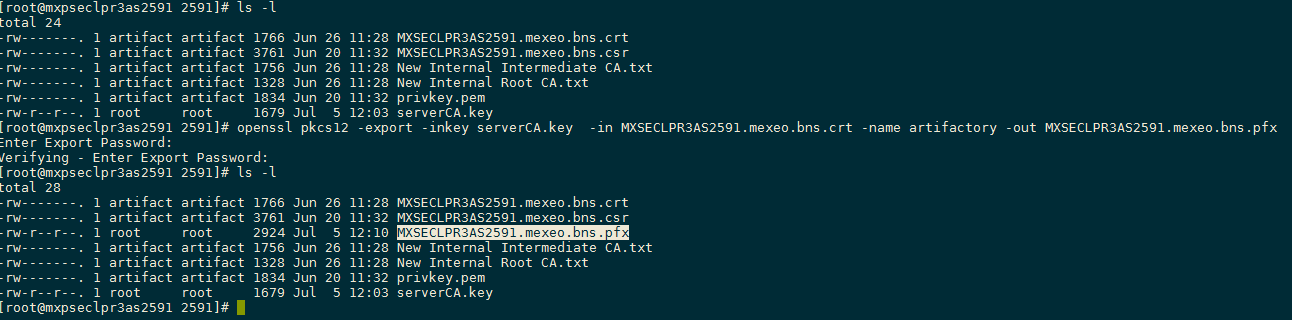
## ARTIFACTORY

A continuación realizaremos la instalación del certificado.

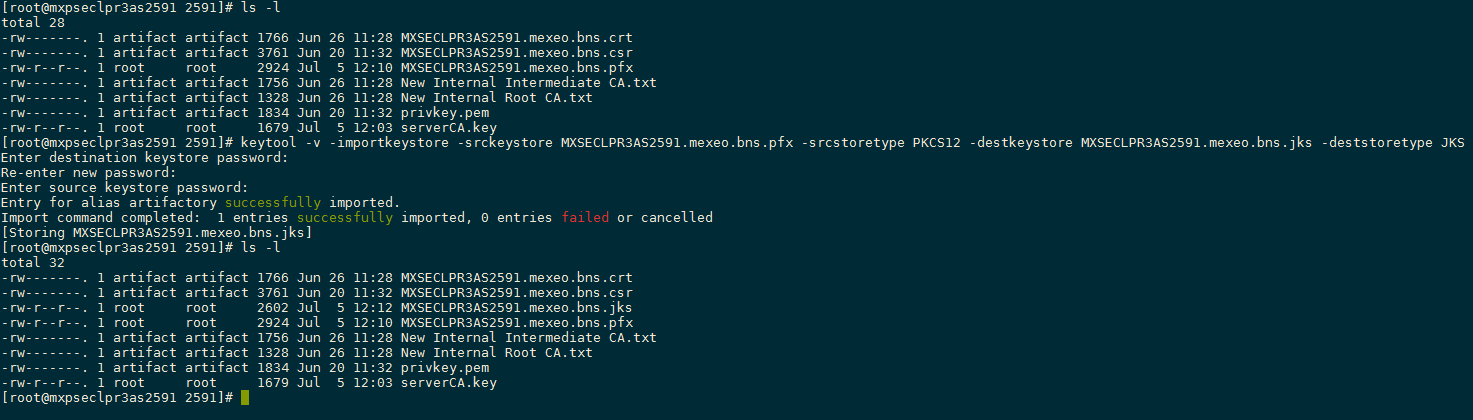
* Lo primero es crear el archivo **.key** desde el **.pem.**



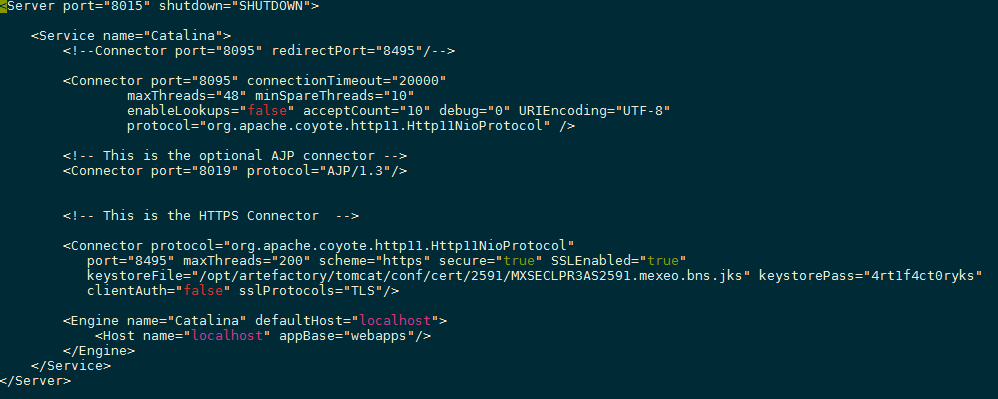
* Después creamos el archivo **.pfx** con el archivo **.key** y el archivo **.crt.**



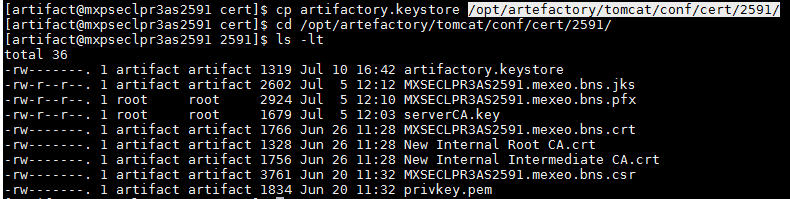
* El siguiente paso es crear el archivo **.jks** desde el archivo **.pfx**.



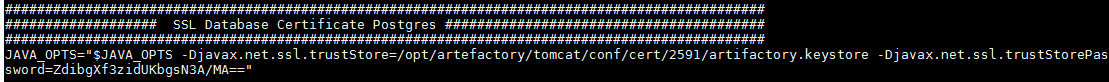
* Ahora modificaremos el archivo **server.xml** e indicaremos la ubicación del certificado y el puerto.



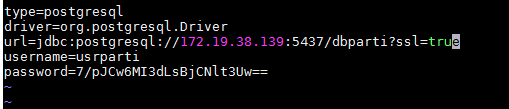
* El siguiente paso será la integración de la herramienta con su base de datos para esto tomaremos el archivo **.crt** de la base de datos y lo depositaremos en el servidor. Despues crearemos el archivo **.der** con el comando **openssl x509 -in server.crt -out server.crt.der** -outform dery con el comando **keytool -v -importkeystore -srckeystore server.crt.der -srcstoretype JKS -destkeystore jira.keystore -deststoretype JKS** crearemos el archivo **.keystore** de la misma manera se integraran todas las herramientas con las que esté integrada la herramienta.



* Ahora indicaremos el lugar del certificado en el archivo **catalina.sh**



* Lo siguiente será indicar en el archivo **db.properties**



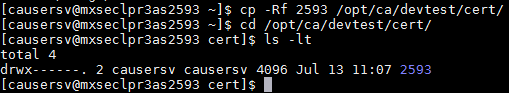
* Después iniciaremos el servicio y comprobaremos el funcionamiento de la herramienta.



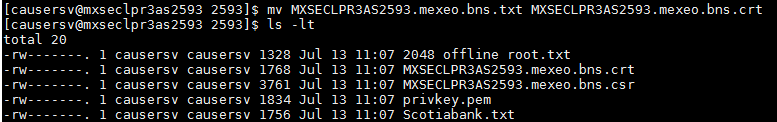
## CA SERVICE VIRTUALIZATION

A continuación realizaremos la instalación del certificado.

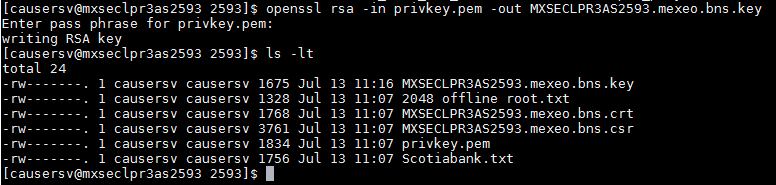
* El primer paso es copiar la carpeta que contiene el certificado.



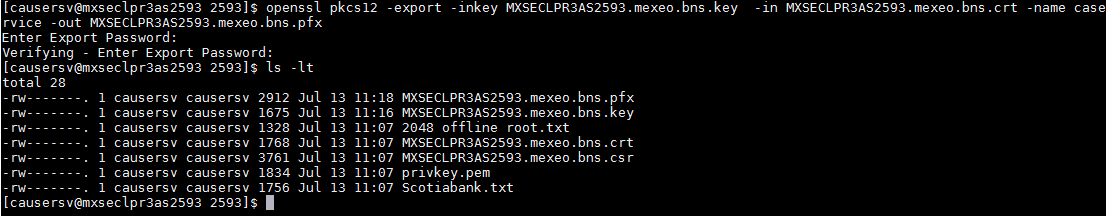
* Cambiamos la extensión **.txt** a **.crt** del certificado.



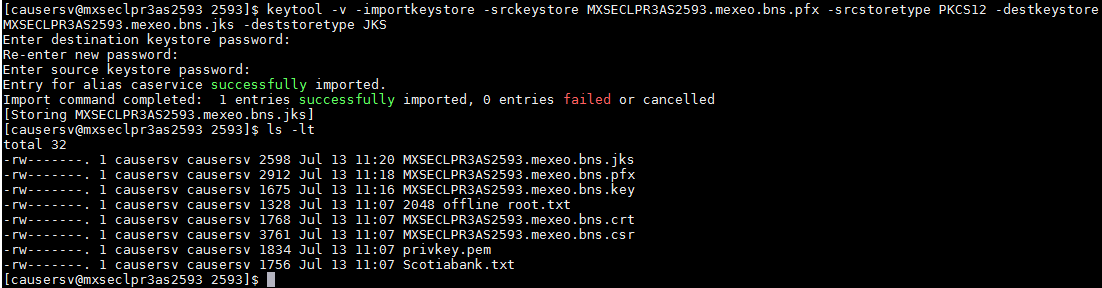
* Lo siguiente es crear el archivo **.key** desde el **.pem.**



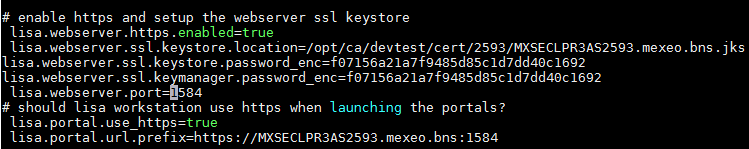
* Después creamos el archivo **.pfx** con el archivo **.key** y el archivo **.crt.**



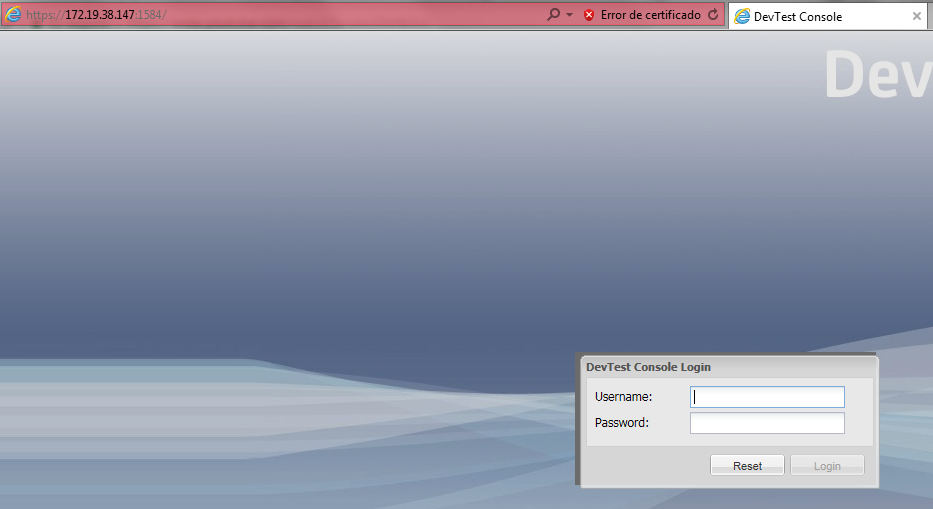
* El siguiente paso es crear el archivo **.jks** desde el archivo **.pfx**.



* Ahora modificaremos el archivo **lisa.properties** e indicaremos la ubicación del certificado y el puerto.



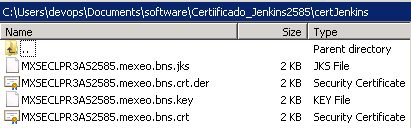
* Después iniciaremos el servicio y comprobaremos el funcionamiento de la herramienta.



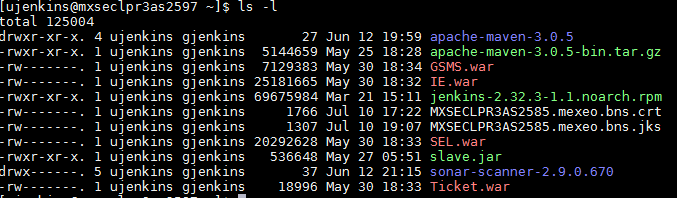
## CONEXIÓN SERVIDOR ESCLAVO 1 (LINUX) CON EL SERVIDOR JENKINS MAESTRO

Para realizar la conexión con el servidor debemos de realizar los siguientes pasos:

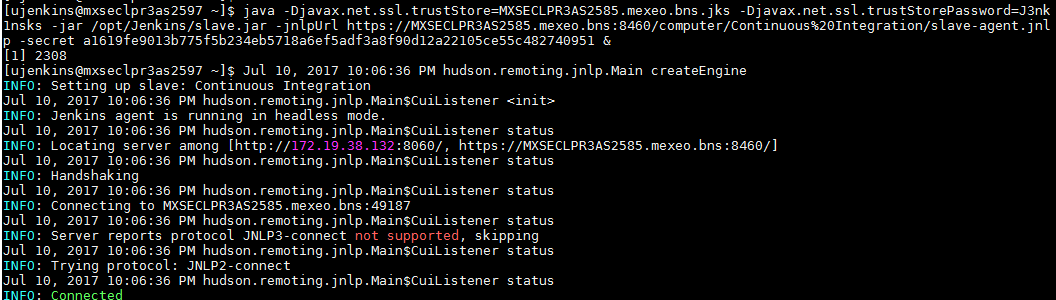
Nos conectamos al servidor Jenkins Master **MXSECLPR3AS2585**, tomamos los certificados de este servidor para agregarlos al servidor **MXSECLPR3AS2597** como se visualiza en la imagen:



Ahora los agregamos al servidor **MXSECLPR3AS2597** como se muestra en la imagen:



Una vez realizados los pasos anteriores, ahora ejecutaremos el siguiente comando para conectar el servidor con el servidor maestro:



Con la respuesta **CONNECT** que se muestra en la ejecución del comando garantiza que se conectó el servidor **MXSECLPR3AS2597** con el servidor maestro **MXSECLPR3AS2585**.

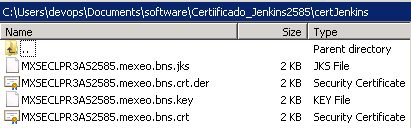
## XL RELEASE

## XL DEPLOY

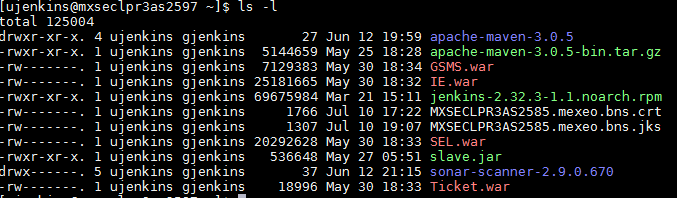
## CONEXIÓN SERVIDOR ESCLAVO 1 (LINUX) CON EL SERVIDOR JENKINS MAESTRO

Para realizar la conexión con el servidor debemos de realizar los siguientes pasos:

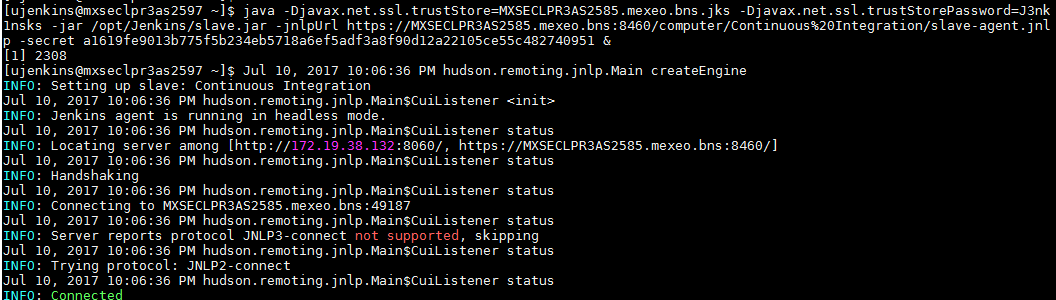
Nos conectamos al servidor Jenkins Master **MXSECLPR3AS2585**, tomamos los certificados de este servidor para agregarlos al servidor **MXSECLPR3AS2597** como se visualiza en la imagen:



Ahora los agregamos al servidor **MXSECLPR3AS2597** como se muestra en la imagen:



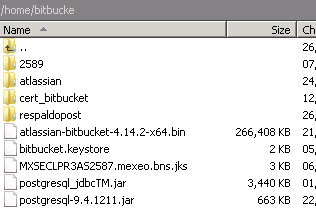
Una vez realizados los pasos anteriores, ahora ejecutaremos el siguiente comando para conectar el servidor con el servidor maestro:



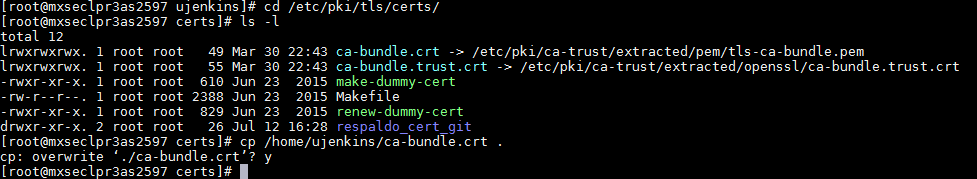
Con la respuesta **CONNECT** que se muestra en la ejecución del comando garantiza que se conectó el servidor **MXSECLPR3AS2597** con el servidor maestro **MXSECLPR3AS2585**.

## CONEXIÓN SERVIDOR ESCLAVO 1 (GIT) CON EL SERVIDOR BITBUCKET

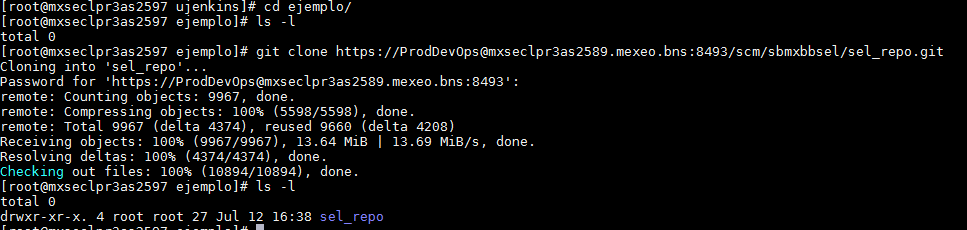
Nos conectamos al servidor **MXSECLPR3AS2589** y copiamos los certificados en la carpeta **cert\_bitbucket** en el home del usuario:



Una vez realizado lo anterior, nos conectamos al servidor **MXSECLPR3AS2597** y agregamos los certificados al archivo **ca-blunde.crt**, como se muestra en la imagen:



Para realizar la prueba nos conectamos al repositorio de SEL, para verificar la conexión con los certificados.

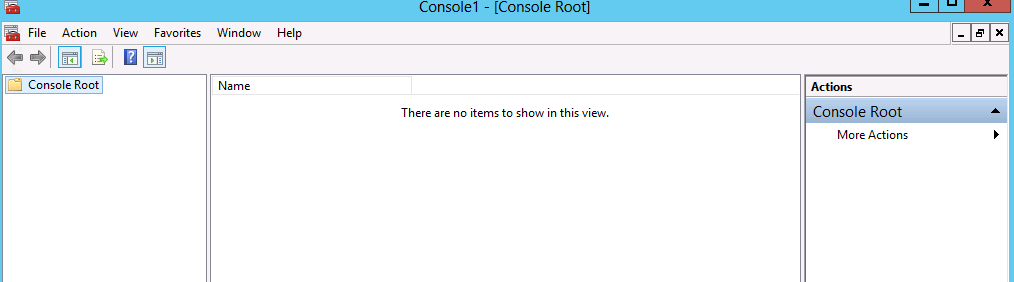


La conexión es correcta ya que se pudo descargar sin problemas el repositorio.

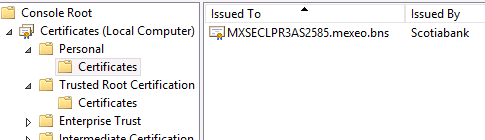
Nota: Para **Artifactory** con **Maven** está pendiente ya que los certificados tienen detalles en su configuración, la conexión entre para el **sonar-scanner** está pendiente ya que el servidor de **SonarQube** no tiene salida a internet.

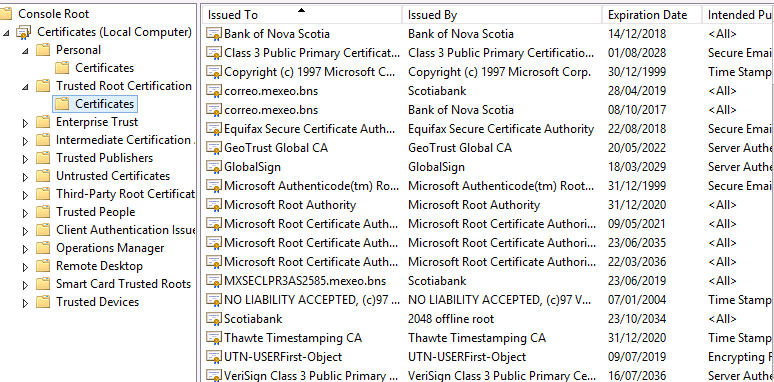
## CONEXIÓN ESCLAVO 2 Y 3 (WINDOWS SERVER) CON EL SERVIDOR JENKINS MAESTRO

En el procedimiento anterior ya teníamos los certificados del servidor Jenkins maestro **MXSECLPR3AS2585**, ahora los configuraremos los certificados en el servidor 3 **MXSECIPR3AS2584,** entraremos a la siguiente consola:

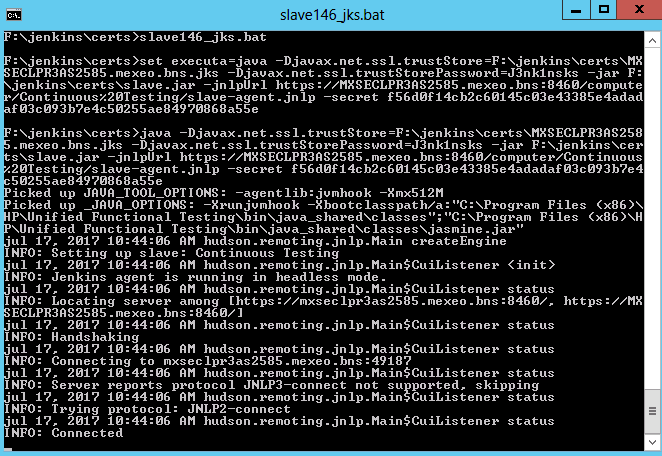


Agregaremos el certificado a las secciones de **Certificates** como certificado **Personal** y **Trusted Root Certification**.

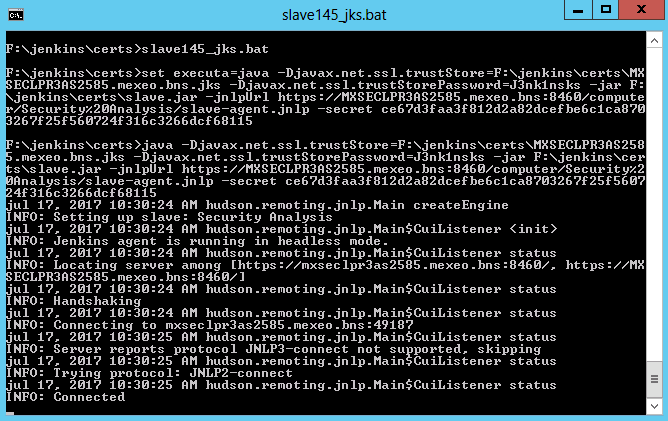




Después de haber agregado los certificados al servidor ejecutamos el siguiente archivo en la consola de Windows para que se conecte al servidor master Jenkins **MXSECLPR3AS2585**.

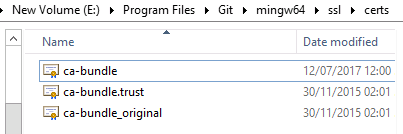


Nota: Este procedimiento se realiza de la misma forma para el servidor 3 **MXSECIPR3AS2583.**

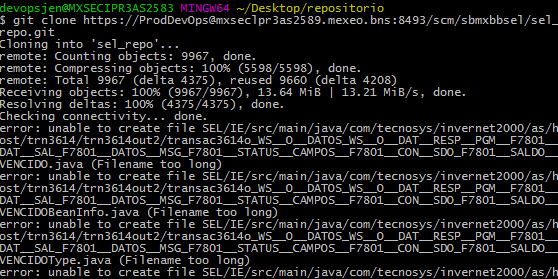


## CONEXIÓN JENKINS ESCLAVO 2 (GIT) CON EL SERVIDOR BITBUCKET

Una vez realizada la conexión de **git** con el esclavo 1, tomaremos el archivo actualizado **ca-bundle** y lo agregaremos a la ruta como se muestra en la imagen:



Con el paso anterior, realizaremos la conexión con **git** abriendo una terminal, conectándonos al repositorio de **bitbucket**.



La conexión con **bitbucket** es correcta con los certificados.

Nota: La conexión de **Maven** con **Artifactory** esta pendiente por los certificados, **sonar-scanner** está pendiente por el servidor de **SonarQube** por la salida a internet para su conexión.

## PERMISOS POR USUARIO

Todos los usuarios fueron configurados con los mismos permisos. Ensegida se muestra la tabla de usuarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SERVIDOR** | **APLICACIÓN** | **CUENTA** | | **GRUPO** |
| SERVIDOR 1 | **Atlassian Crowd** | | usracrowd | usracrowd |
| SERVIDOR 2 | **Atlassian JIRA** | | jira | jira |
| SERVIDOR 3 | **Atlassian Confluence** | | confluence | confluence |
| SERVIDOR 4 | **Atlassian BitBucket** | | atlbitbucket | atlbitbucket |
| SERVIDOR 5 | **SonarQube** | | sonar | sonar |
| SERVIDOR 6 | **Artifactory** | | artifactory | artifactory |
| SERVIDOR 7 | **Jenkins** | | jenkins | jenkins |
| SERVIDOR 8 | **Jenkins** | | jenkins | jenkins |
| SERVIDOR 9 | **Jenkins** | | devops | Administrator |
| SERVIDOR 10 | **Jenkins** | | devops | Administrator |
| SERVIDOR 11 | **XL Deploy** | | xluser | xluser |
| SERVIDOR 12 | **XL Release** | | xluser | xluser |
| SERVIDOR 13 | **Puppet Enterprise** | | pu-puppetu | pu-puppetu |
| SERVIDOR 14 | **CA Service Virtualization** | | causr | causr |
| SERVIDOR 15 | **PostgreSQL** | | postgres | postgres |
| SERVIDOR 16 | **PostgreSQL** | | postgres | postgres |
| SERVIDOR 17 | **PostgreSQL** | | postgres | postgres |
| SERVIDOR 18 | **PostgreSQL** | | postgres | postgres |
| SERVIDOR 19 | **PostgreSQL** | | postgres | postgres |
| SERVIDOR 20 | **Microsot SQL Server** | | devops | Administrator |

Tabla 3 Usuarios

Enseguida se describen los permisos que tiene cada usuario.

|  |
| --- |
| **PERMISOS** |
| /bin/yum update \* , \ |
| /bin/yum search \* , \ |
| /bin/yum info \* , \ |
| /bin/yum install \* , \ |
| /bin/yum remove \* , \ |
| /bin/yum list \*, \ |
| /bin/rpm -qa \* , \ |
| /bin/rpm -ivh \* , \ |
| /bin/rpm -qRp \* , \ |
| /bin/rpm -ev \*, \ |
| /bin/ls, \ |
| /bin/mv, \ |
| /bin/vi, \ |
| /bin/cat, \ |
| /bin/more, \ |
| /bin/find |

Tabla 4 Permisos

## AUDITORIA

Se muestra a continuación la ubicación de los **logs** de cada herramienta relacionados a su funcionalidad general y al registro de las operaciones relacionadas a los procesos de auditoria.

|  |  |  |  |
| --- | --- | --- | --- |
| **COMPONENTE** | **APP USER** | **APP GROUP** |  |
| **Atlassian Crowd** | AccessLog | /opt/atlassian/application-data/crowd/apache-tomcat/logs/localhost\_access\_log\* | -rwxrwxr-x |
| ToolLog | /opt/atlassian/application-data/crowd/logs atlassian-crowd.log | -rwxrwxr-x |
| **Atlassian Confluence** | AccessLog | /opt/atlassian/application-data/confluence/logs/atlassian-confluence-access.log | -rwxr-xr-x |
| ToolLog | /opt/atlassian/application-data/confluence/logs/catalina.out | -rwxr-xr-x |
| **Atlassian JIRA** | AccessLog | /opt/atlassian/application-data/jira/logs/access\_log.\* | -rw-r----- |
| ToolLog | /opt/atlassian/application-data/jira/logs/ catalina.out | -rw-r----- |
| **Atlassian BitBucket** | AccessLog | /var/bitbucket/log/atlassian-bitbucket-access.log  /var/bitbucket/log/audit/atlassian-bitbucket-audit.log | -rw-------  -rw------- |
| ToolLog | var/bitbucket/log/atlassian-bitbucket.log | -rw------- |
| **Artifactory** | AccessLog | /opt/artefactory/logs/access.log | -rwxrwxr-x |
| ToolLog | /opt/artefactory/logs/artifactory.log | -rw-r--r-- |
| **Jenkins Master** | AccessLog | /var/log/jenkins/access\_log | -rw-r--r-- |
| ToolLog | /var/log/jenkins/jenkins.log | -rw-r--r-- |
| **XL Deploy** | Audit Log | /opt/xlsdeploy/xldeploy/log/audit.log | -rw-rw-rw- |
| WrapperLog | /opt/xlsdeploy/xldeploy/log/wrapper.log | -rw-rw-rw- |
| ToolLog | /opt/xlsdeploy/xldeploy/log/deployit.log | -rw-r--r-- |
| **XL Release** | Audit Log | /opt/xlrelease/log/access.log | -rw-rw-rw- |
| WrapperLog | /opt/xlrelease/log/wrapper.log | -rw-rw-rw- |
| ToolLog | /opt/xlrelease/log/xl-release.log | -rw-r--r-- |
| **SonarQube** | AccessLog | /opt/sonarqube/logs/access\*.log | -rw-r--r-- |
| ToolLog | /opt/sonarqube/logs/sonar.log | -rwxr-xr-x |

ÍNDICE DE TABLAS

[Tabla 1 Instancias Postgresql 4](#_Toc488940248)

[Tabla 2 Instancias Microsoft Sql 7](#_Toc488940249)

[Tabla 3 Usuarios 38](#_Toc488940250)

[Tabla 4 Permisos 39](#_Toc488940251)