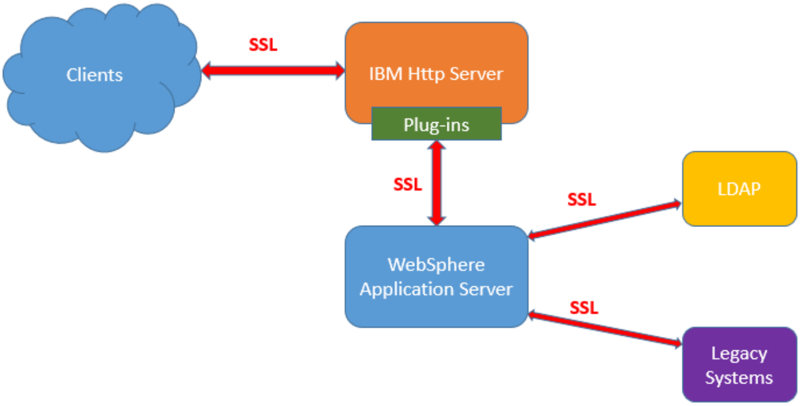
CHAPTER 18: SSL

Theory

WebSphere Application Server communicates with other components of a middleware environment and in order to prevent intrusions, WebSphere uses Java Secure Socket Extension (JSSE) which is an SSL implementation. SSL capabilities such as handshaking are delivered by JSSE using X.509 standard public key infrastructure (PKI).

PKI contains hardware, software, people and policies to handle whole lifecycle of digital certificates including activities like create, manage, store and renew. The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository and revokes them if needed. Certificate Authority (CA) digitally sign and publish the public key bound to a given user. This is done using the CA's own private key, so that trust in the user key relies on one's trust in the validity of the CA's key.

SSL is a protocol which is designed to provide secure communication over networks. It uses certificates to authenticate the service provider by exchanging keys between server and client. WebSphere Application Server store these certificates in password protected files, in keystores. WebSphere allows you to work with certificates using administrative console where you can create and manage keystores.



WebSphere Application Server can use SSL in the cell between the nodes. The certificate required for this secure communication is created during profile creation. This certificate contains a signer certificate and a personal certificate created by embedded CA. You can also use your own certificate for this purpose.

Another important aspect of secure communication is between WebSphere Application Server and web server. For this purpose, plug-in uses only one keyring file. WebSphere allows you to create your personal keys and add them to the keyring files to be delivered to the web server.

It is also possible to secure the communication between external systems that WebSphere Application Server transmits sensitive information. Following systems worth to be considered to use SSL secured communication:

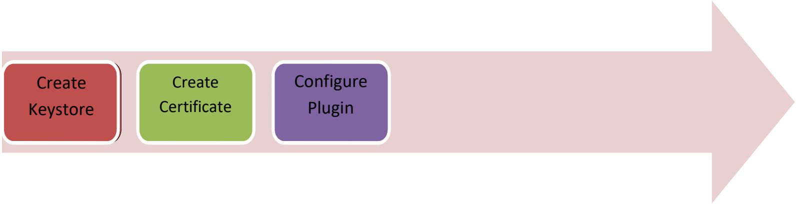
- Database connection
- LDAP connection
- Message channels (i.e. WebSphere MQ connection)
- Web services connections

AIM

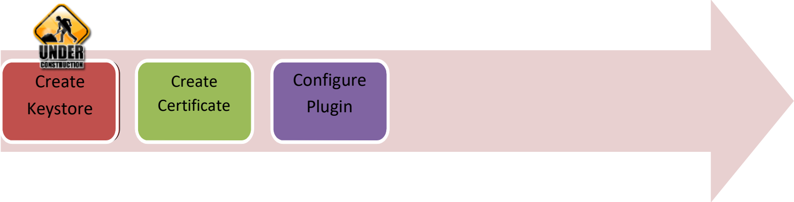
In this lab exercise, you will perform common SSL operations. When you finish this lab, you will be able to create a keystore and a personal certificate and configure it to use in communication between WebSphere Application Server and the web server. In order to achieve this goal, you need to complete following tasks:

- Create a keystore
- Create a self-signed certificate
- Configure plug-in to use new certificate

Lab Exercise 18: SSL



1. **Create a keystore**
2. **Create a self-signed certificate**
3. **Configure plug-in to use new certificate**



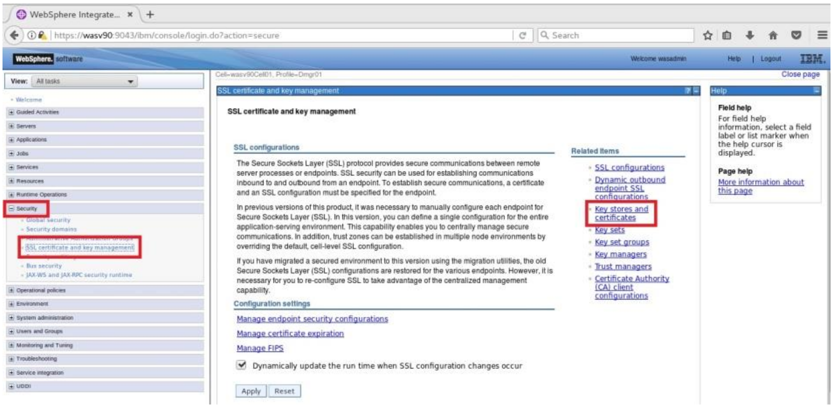
Create Keystore

Create Certificate

Configure Plugin

Task 1: Create a keystore

Step 1: Navigate to “Security>SSL certificate and key management” and click “Key stores and certificates”.



Cell=wasv90Cell01, Profile=Dmgr01

SSL certificate and key management

SSL certificate and key management

SSL configurations

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

Configuration settings

Manage endpoint security configurations

Manage certificate expiration

Manage FIPS

☑ Dynamically update the run time when SSL configuration changes occur

Apply Reset

Related Items

- SSL configurations
- Dynamic outbound endpoint SSL configurations
- Key stores and certificates
- Key sets
- Key set groups
- Key managers
- Trust managers
- Certificate Authority (CA) client configurations

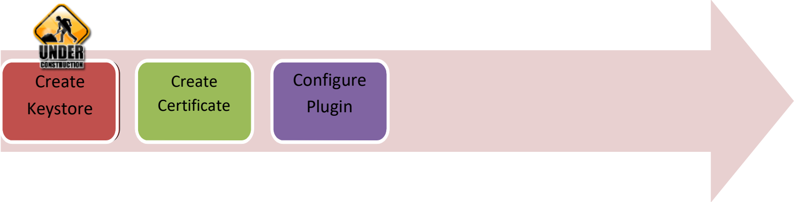
Help

Field help

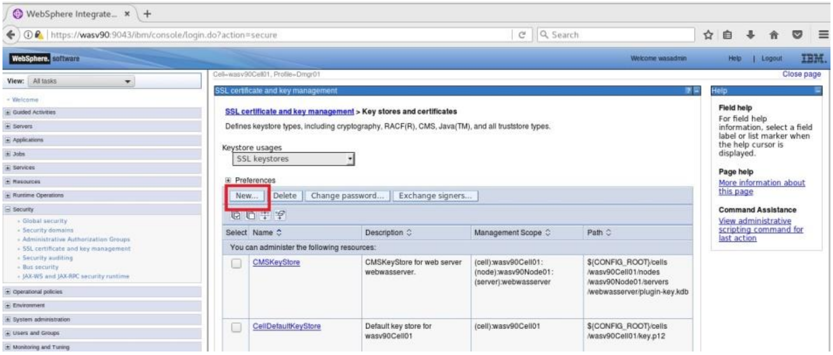
For field help information, select a field label or list marker when the help cursor is displayed.

Page help

More information about this page



Step 2: Click “New”.



Cell=wasv90Cell01, Profile=Dmgr01

SSL certificate and key management

SSL certificate and key management > Key stores and certificates

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

Keystore usages: SSL keystores

⊞ Preferences

Select	Name	Description	Management Scope	Path
You can administer the following resources:				
□	CMSKeyStore	CMSKeyStore for web server webwasserver.	(cell):wasv90Cell01:(node):wasv90Node01:(server):webwasserver	${CONFIG_ROOT}/cells/wasv90Cell01/nodes/wasv90Node01/servers/webwasserver/plugin-key.kdb
□	CellDefaultKeyStore	Default key store for wasv90Cell01	(cell):wasv90Cell01	${CONFIG_ROOT}/cells/wasv90Cell01/key.p12

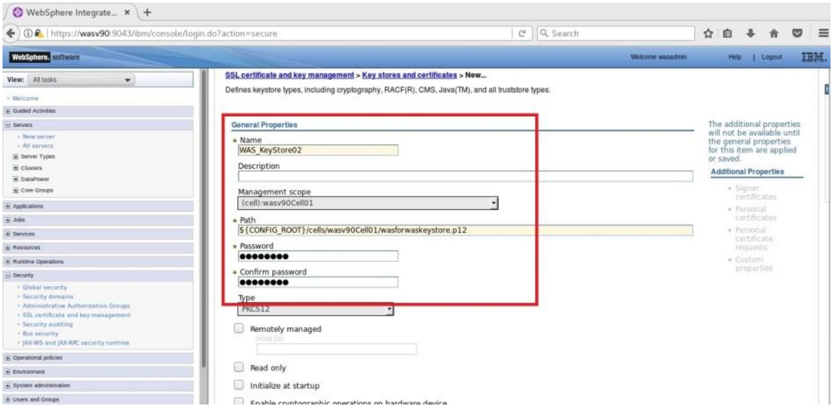
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

Step 3: Enter the properties of the new keystore similar to following picture.



SSL certificate and key management > Key stores and certificates > New...

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

General Properties

* Name: WAS_KeyStore02

Description:

Management scope: (cell):wasv90Cell01

* Path: ${CONFIG_ROOT}/cells/wasv90Cell01/wasforwaskeystore.p12

* Password: ●●●●●●●●

* Confirm password: ●●●●●●●●

Type: PKCS12

□ Remotely managed

Host list

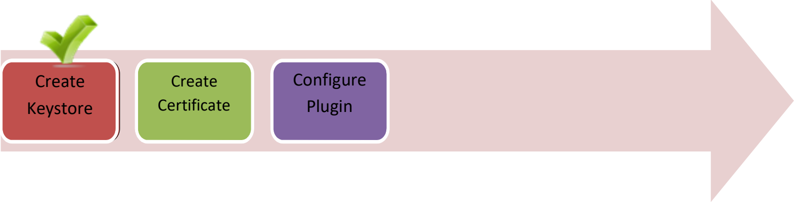
□ Read only

□ Initialize at startup

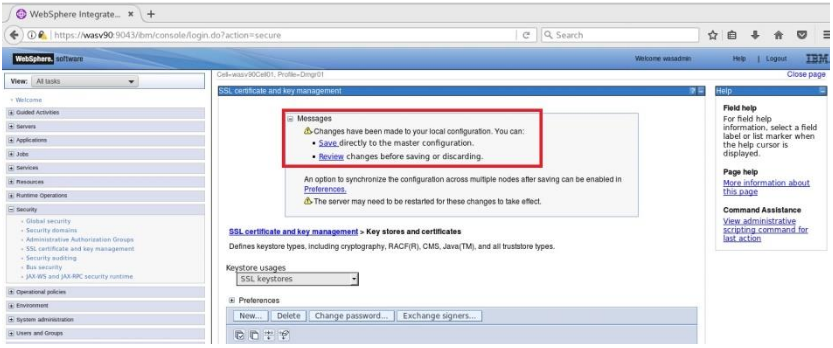
□ Enable cryptographic operations on hardware device

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties
- Signer certificates
- Personal certificates
- Personal certificate requests
- Custom properties



Step 4: Click “Save” to write changes to the master configuration file.



Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

⚠ The server may need to be restarted for these changes to take effect.

SSL certificate and key management > **Key stores and certificates**

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

Keystore usages: SSL keystores

Preferences

Field help

For field help information, select a field label or list marker when the help cursor is displayed.

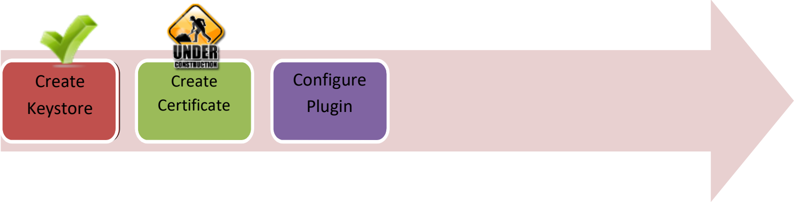
Page help

More information about this page

Command Assistance

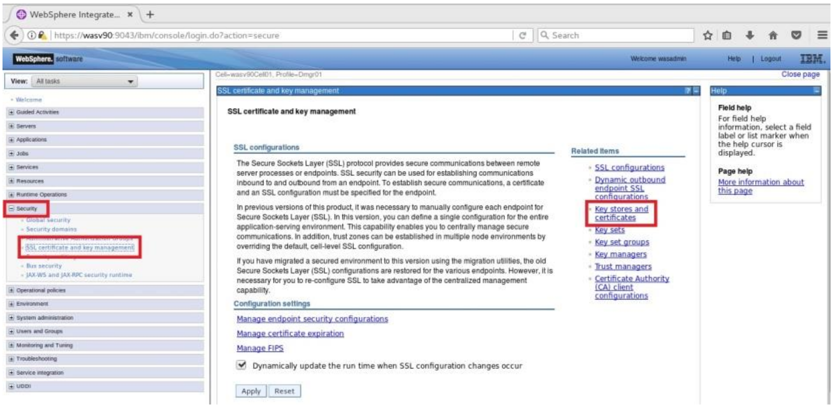
View administrative scripting command for last action

Task 1 is complete!



Task 2: Create a self-signed certificate

Step 1: Navigate to “Security>SSL certificate and key management” and click “Key stores and certificates”.



SSL certificate and key management

SSL configurations

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

Configuration settings

Manage endpoint security configurations

Manage certificate expiration

Manage FIPS

☑ Dynamically update the run time when SSL configuration changes occur

Apply Reset

Related Items

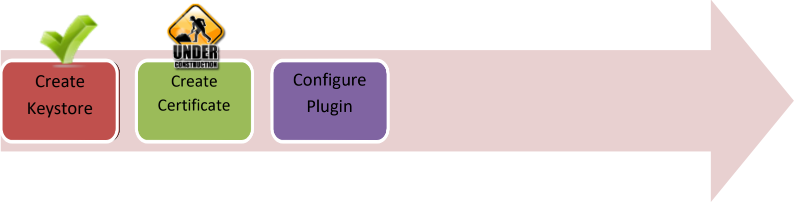
- SSL configurations
- Dynamic outbound endpoint SSL configurations
- Key stores and certificates
- Key sets
- Key set groups
- Key managers
- Trust managers
- Certificate Authority (CA) client configurations

Field help

For field help information, select a field label or list marker when the help cursor is displayed.

Page help

More information about this page

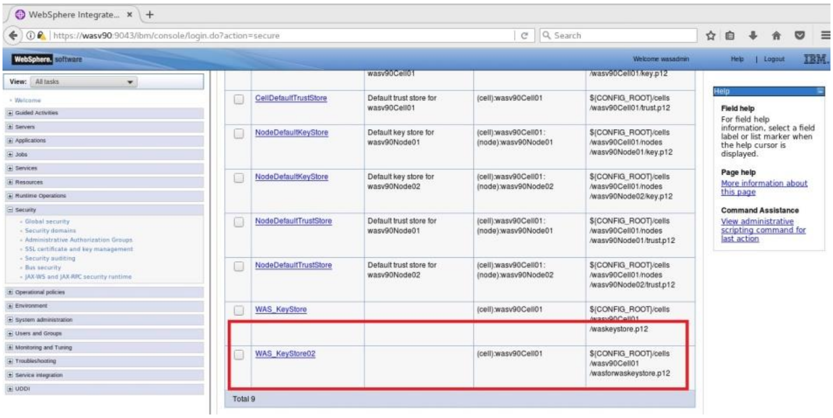


Create Keystore

Create Certificate

Configure Plugin

Step 2: Click on “WAS_Keystore02”.

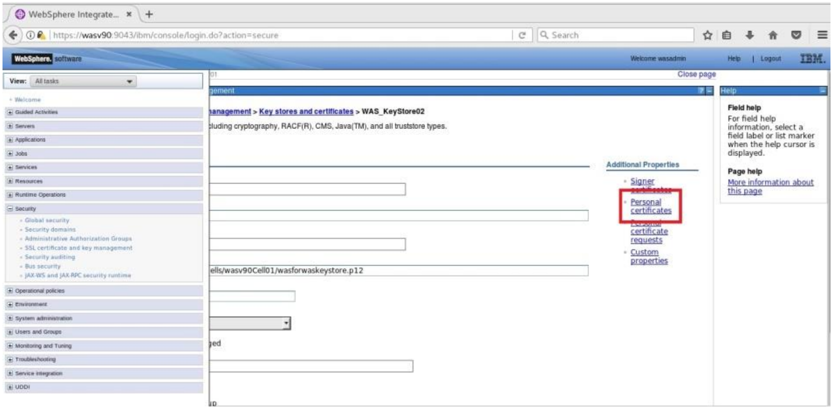


		wasv90Cell01		/wasv90Cell01/key.p12
□	CellDefaultTrustStore	Default trust store for wasv90Cell01	(cell):wasv90Cell01	${CONFIG_ROOT}/cells /wasv90Cell01/trust.p12
□	NodeDefaultKeyStore	Default key store for wasv90Node01	(cell):wasv90Cell01:(node):wasv90Node01	${CONFIG_ROOT}/cells /wasv90Cell01/nodes /wasv90Node01/key.p12
□	NodeDefaultKeyStore	Default key store for wasv90Node02	(cell):wasv90Cell01:(node):wasv90Node02	${CONFIG_ROOT}/cells /wasv90Cell01/nodes /wasv90Node02/key.p12
□	NodeDefaultTrustStore	Default trust store for wasv90Node01	(cell):wasv90Cell01:(node):wasv90Node01	${CONFIG_ROOT}/cells /wasv90Cell01/nodes /wasv90Node01/trust.p12
□	NodeDefaultTrustStore	Default trust store for wasv90Node02	(cell):wasv90Cell01:(node):wasv90Node02	${CONFIG_ROOT}/cells /wasv90Cell01/nodes /wasv90Node02/trust.p12
□	WAS_KeyStore		(cell):wasv90Cell01	${CONFIG_ROOT}/cells /wasv90Cell01 /waskeystore.p12
□	WAS_KeyStore02		(cell):wasv90Cell01	${CONFIG_ROOT}/cells /wasv90Cell01 /wasforwaskeystore.p12

Total 9

Help
Field help
For field help information, select a field label or list marker when the help cursor is displayed.
Page help
More information about this page
Command Assistance
View administrative scripting command for last action

Step 3: Click “Personal certificates”.

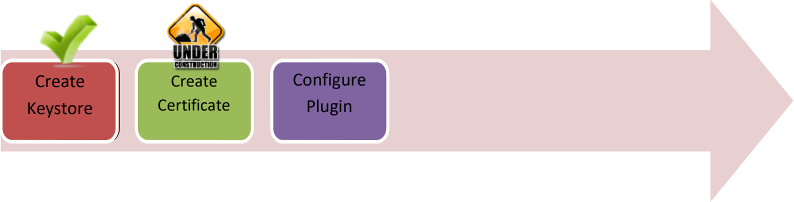


WebSphere Integrate...
https://wasv90:9043/ibm/console/login.do?action=secure
Search
WebSphere. software
Welcome wasadmin
Help | Logout
View: All tasks
Close page
Additional Properties
Signer certificates
Personal certificates
Personal certificate requests
Custom properties
Help
Field help
For field help information, select a field label or list marker when the help cursor is displayed.
Page help
More information about this page

anagement > Key stores and certificates > WAS_KeyStore02

cluding cryptography, RACF(R), CMS, Java(TM), and all truststore types.

ells/wasv90Cell01/wasforwaskeystore.p12

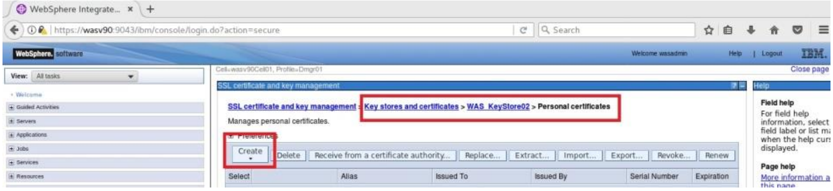


Create Keystore

Create Certificate

Configure Plugin

Step 4: Click “Create” to define a new personal certificate.



Cell=wasv90Cell01, Profile=Dmgr01

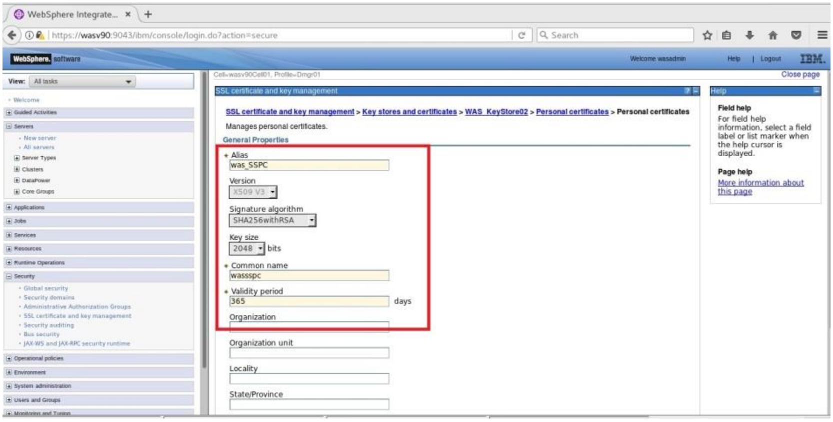
SSL certificate and key management > Key stores and certificates > WAS_KeyStore02 > Personal certificates

Manages personal certificates.

Field help
For field help information, select field label or list ma when the help curs displayed.

Page help
More information a this page

Step 5: Enter the values for the certificate as follows and click “OK”.



Cell=wasv90Cell01, Profile=Dmgr01

SSL certificate and key management > Key stores and certificates > WAS_KeyStore02 > Personal certificates > Personal certificates

Manages personal certificates.

General Properties

* Alias
was_SSPC

Version
X509 V3

Signature algorithm
SHA256withRSA

Key size
2048 bits

* Common name
wasspc

* Validity period
365 days

Organization

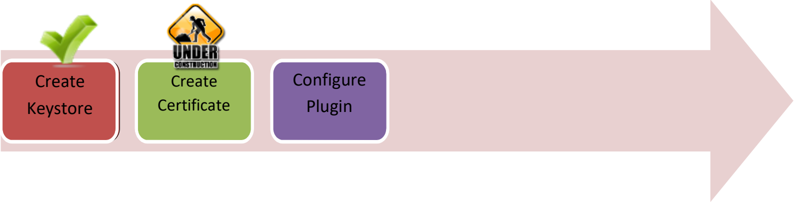
Organization unit

Locality

State/Province

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

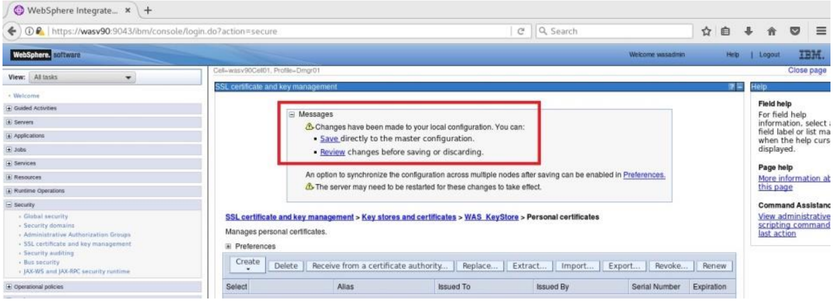


Create Keystore

Create Certificate

Configure Plugin

Step 6: Click “Save” to write changes.



Cell=wasv90Cell01, Profile=Dmgr01

SSL certificate and key management

⊟ Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

⚠ The server may need to be restarted for these changes to take effect.

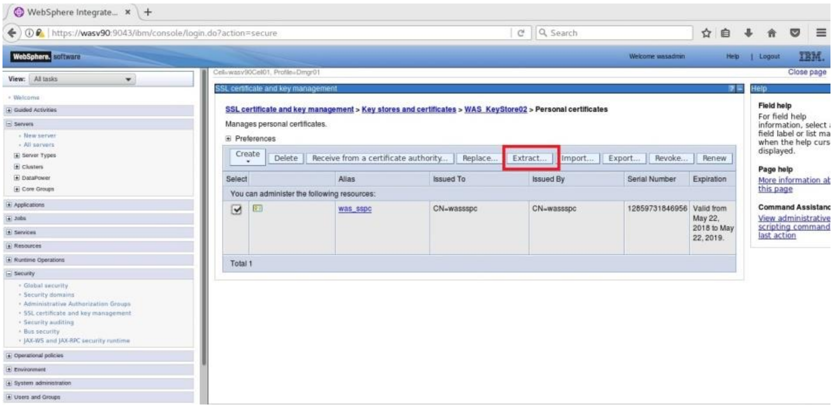
SSL certificate and key management > Key stores and certificates > WAS_KeyStore > Personal certificates

Manages personal certificates.

⊞ Preferences

Select		Alias	Issued To	Issued By	Serial Number	Expiration

Step 7: Select the “Alias” name and click “Extract”.



Cell=wasv90Cell01, Profile=Dmgr01

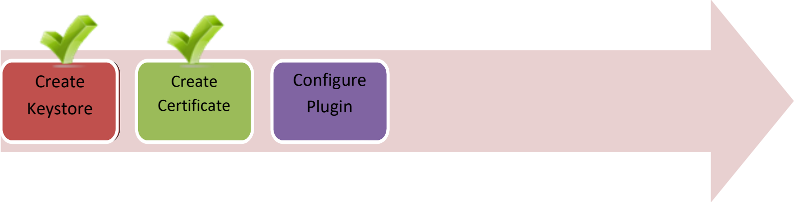
SSL certificate and key management

SSL certificate and key management > Key stores and certificates > WAS_KeyStore02 > Personal certificates

Manages personal certificates.

⊞ Preferences

Select		Alias	Issued To	Issued By	Serial Number	Expiration
You can administer the following resources:						
☑		was_sspc	CN=wassspc	CN=wassspc	12859731846956	Valid from May 22, 2018 to May 22, 2019.
Total 1						

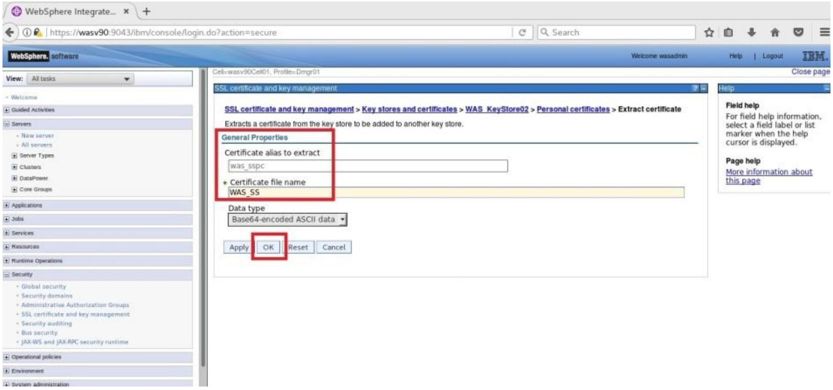


Create Keystore

Create Certificate

Configure Plugin

Step 8: Enter a name for “Certificate file name” and click “OK”.



SSL certificate and key management > Key stores and certificates > WAS_KeyStore02 > Personal certificates > Extract certificate

Extracts a certificate from the key store to be added to another key store.

General Properties

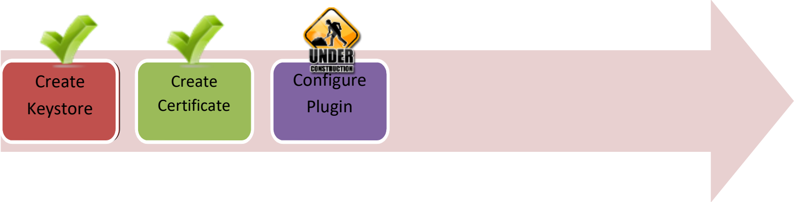
Certificate alias to extract: was_sspc

* Certificate file name: WAS_SS

Data type: Base64-encoded ASCII data

Apply | OK | Reset | Cancel

Task 2 is complete!



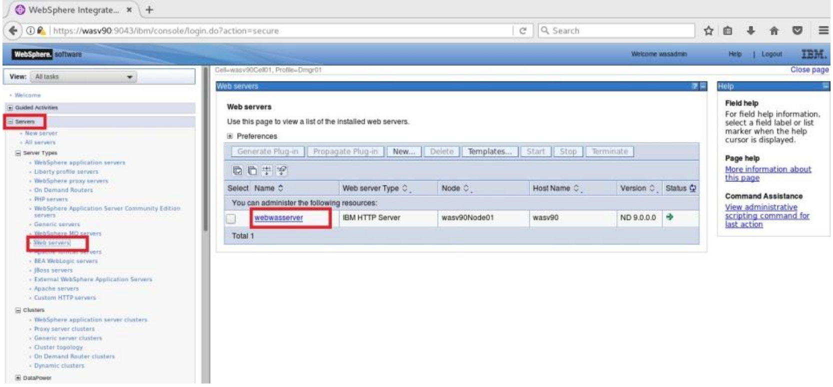
Create Keystore

Create Certificate

Configure Plugin

Task 3: Configure plug-in to use new certificate

Step 1: Navigate to “Servers>Server Types>Web servers” and click on the web server name.

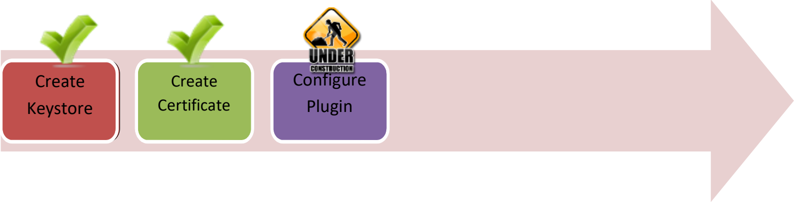


Web servers

Use this page to view a list of the installed web servers.

For field help information, select a field label or list marker when the help cursor is displayed.

Select	Name	Web server Type	Node	Host Name	Version	Status
You can administer the following resources:						
□	webwasserver	IBM HTTP Server	wasv90Node01	wasv90	ND 9.0.0.0	➔
Total 1						

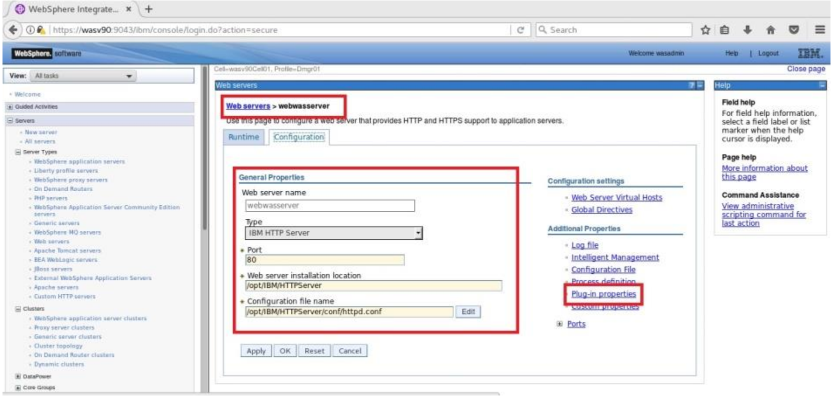


Create Keystore

Create Certificate

Configure Plugin

Step 2: Click on “Plug-in properties”.



https://wasv90:9043/ibm/console/login.do?action=secure

Cell=wasv90Cell01, Profile=Dmgr01

Web servers > webwasserver

Use this page to configure a web server that provides HTTP and HTTPS support to application servers.

General Properties

Web server name: webwasserver

Type: IBM HTTP Server

* Port: 80

* Web server installation location: /opt/IBM/HTTPServer

* Configuration file name: /opt/IBM/HTTPServer/conf/httpd.conf

Field help

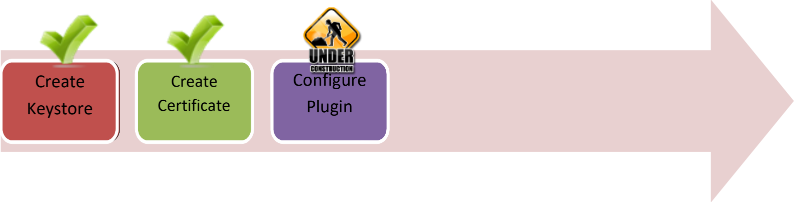
For field help information, select a field label or list marker when the help cursor is displayed.

Page help

More information about this page

Command Assistance

View administrative scripting command for last action

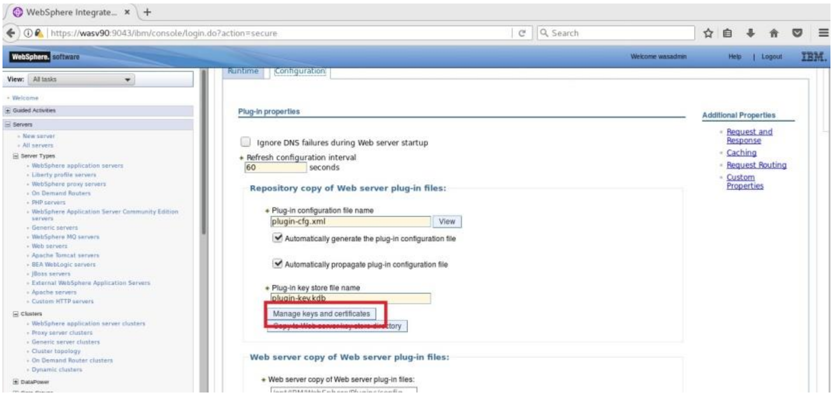


Create Keystore

Create Certificate

Configure Plugin

Step 3: In the configuration tab, click on “Manage keys and certificates”.



Plug-in properties

☐ Ignore DNS failures during Web server startup

* Refresh configuration interval
60 seconds

Repository copy of Web server plug-in files:

* Plug-in configuration file name
plugin-cfg.xml

☑ Automatically generate the plug-in configuration file

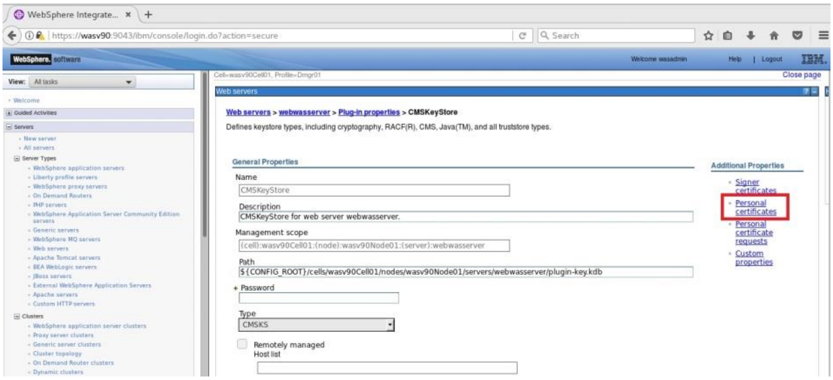
☑ Automatically propagate plug-in configuration file

* Plug-in key store file name
plugin-key.kdb

Web server copy of Web server plug-in files:

* Web server copy of Web server plug-in files:

Step 4: Click on “Personal certificates”.



Cell=wasv90Cell01, Profile=Dmgr01

Web servers > webwasserver > Plug-in properties > CMSKeyStore

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

General Properties

Name
CMSKeyStore

Description
CMSKeyStore for web server webwasserver.

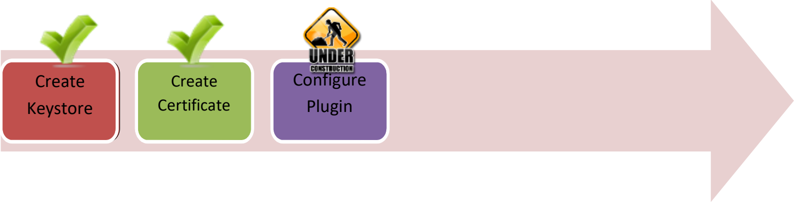
Management scope
(cell):wasv90Cell01:(node):wasv90Node01:(server):webwasserver

Path
${CONFIG_ROOT}/cells/wasv90Cell01/nodes/wasv90Node01/servers/webwasserver/plugin-key.kdb

* Password

Type
CMSKS

☐ Remotely managed
Host list

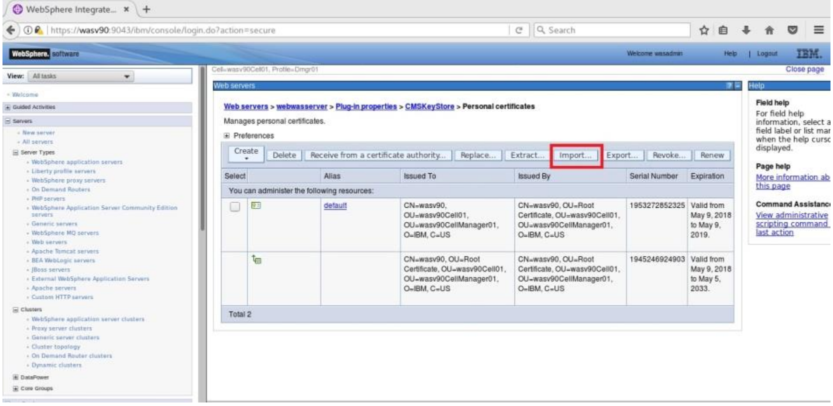


Create Keystore

Create Certificate

Configure Plugin

Step 5: Click “Import”.



Cell=wasv90Cell01, Profile=Dmgr01 Close page

Web servers

Web servers > webwasserver > Plug-in properties > CMSKeyStore > **Personal certificates**

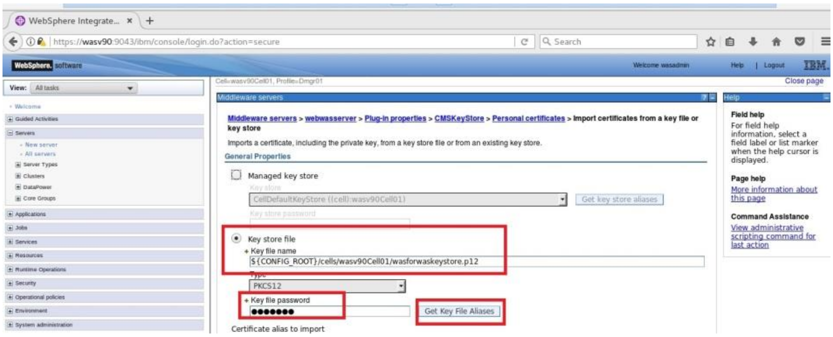
Manages personal certificates.

⊞ Preferences

Create | Delete | Receive from a certificate authority... | Replace... | Extract... | Import... | Export... | Revoke... | Renew

Select		Alias	Issued To	Issued By	Serial Number	Expiration
You can administer the following resources:						
□		default	CN=wasv90, OU=wasv90Cell01, OU=wasv90CellManager01, O=IBM, C=US	CN=wasv90, OU=Root Certificate, OU=wasv90Cell01, OU=wasv90CellManager01, O=IBM, C=US	1953272852325	Valid from May 9, 2018 to May 9, 2019.
			CN=wasv90, OU=Root Certificate, OU=wasv90Cell01, OU=wasv90CellManager01, O=IBM, C=US	CN=wasv90, OU=Root Certificate, OU=wasv90Cell01, OU=wasv90CellManager01, O=IBM, C=US	1945246924903	Valid from May 9, 2018 to May 5, 2033.
Total 2						

Step 6: Enter the “Key store file” path and name, “${CONFIG_ROOT}/cells/wasv90Cell01/wasforwaskeystore.p12” and the password. Click on “Get Key File Aliases”.



Cell=wasv90Cell01, Profile=Dmgr01 Close page

Middleware servers

Middleware servers > webwasserver > Plug-in properties > CMSKeyStore > Personal certificates > **Import certificates from a key file or key store**

Imports a certificate, including the private key, from a key store file or from an existing key store.

General Properties

□ Managed key store

Key store

CellDefaultKeyStore ((cell):wasv90Cell01) | Get key store aliases

Key store password

● Key store file

* Key file name

${CONFIG_ROOT}/cells/wasv90Cell01/wasforwaskeystore.p12

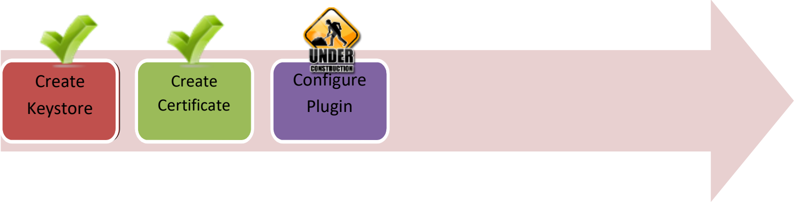
Type

PKCS12

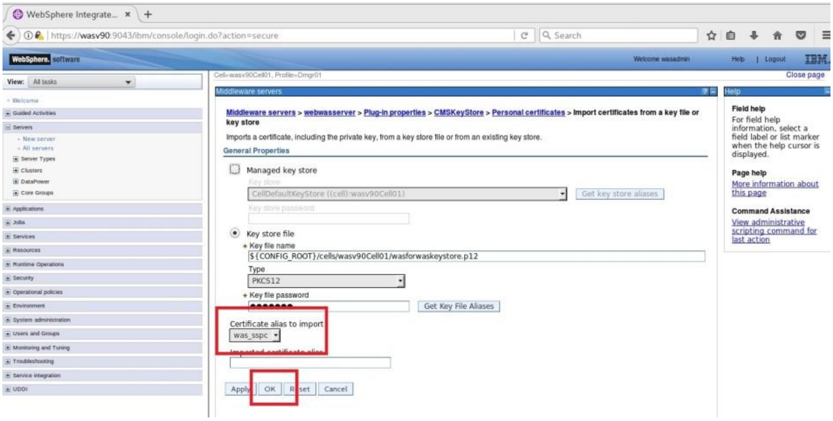
* Key file password

●●●●●●● | Get Key File Aliases

Certificate alias to import



Step 7: Select the alias created on the second task, and click “OK”.



Cell=wasv90Cell01, Profile=Dmgr01

Middleware servers

Middleware servers > webwasserver > Plug-in properties > CMSKeyStore > Personal certificates > **Import certificates from a key file or key store**

Imports a certificate, including the private key, from a key store file or from an existing key store.

General Properties

□ Managed key store

Key store: CellDefaultKeyStore ((cell):wasv90Cell01) [Get key store aliases]

Key store password

● Key store file

* Key file name: ${CONFIG_ROOT}/cells/wasv90Cell01/wasforwaskeystore.p12

Type: PKCS12

* Key file password: ●●●●●●● [Get Key File Aliases]

Certificate alias to import: was_sspc

Imported certificate alias

[Apply] [OK] [Reset] [Cancel]

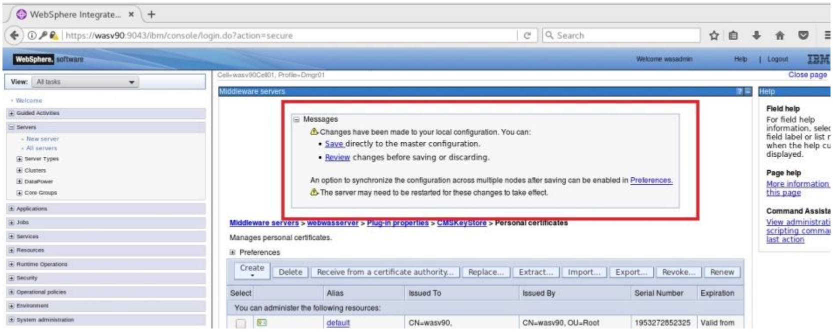
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

Step 8: Click “Save” to write changes to the master configuration file.



Cell=wasv90Cell01, Profile=Dmgr01

Middleware servers

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

⚠ The server may need to be restarted for these changes to take effect.

Middleware servers > webwasserver > Plug-in properties > CMSKeyStore > **Personal certificates**

Manages personal certificates.

⊞ Preferences

[Create] [Delete] [Receive from a certificate authority...] [Replace...] [Extract...] [Import...] [Export...] [Revoke...] [Renew]

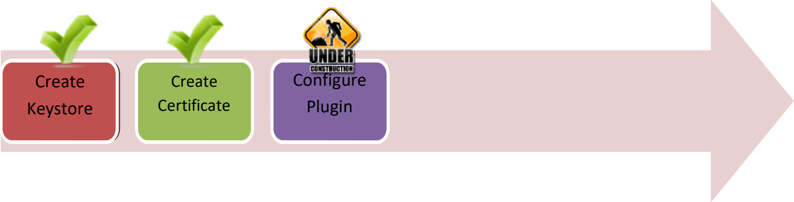
Select		Alias	Issued To	Issued By	Serial Number	Expiration
You can administer the following resources:						
□		default	CN=wasv90,	CN=wasv90, OU=Root	1953272852325	Valid from

Help

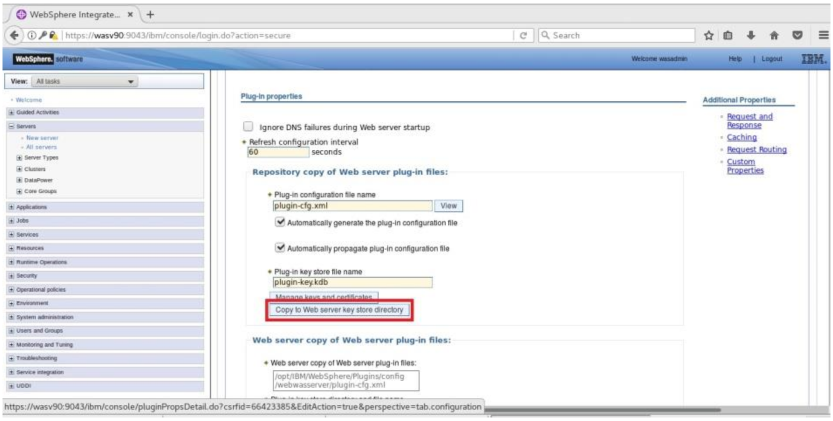
Field help
For field help information, selec field label or list r when the help cu displayed.

Page help
More information this page

Command Assista
View administrati scripting comma last action



Step 9: On the “Configuration” tab, click “Copy to Web server key store directory”.



Plug-in properties

☐ Ignore DNS failures during Web server startup

* Refresh configuration interval
60 seconds

Repository copy of Web server plug-in files:

* Plug-in configuration file name
plugin-cfg.xml

☑ Automatically generate the plug-in configuration file

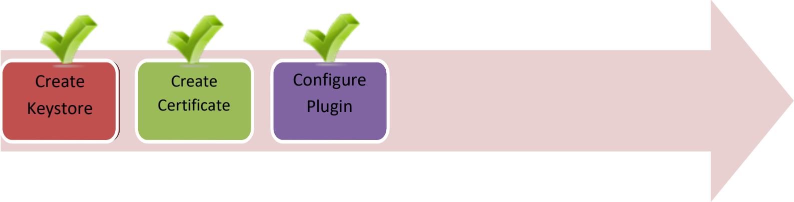
☑ Automatically propagate plug-in configuration file

* Plug-in key store file name
plugin-key.kdb

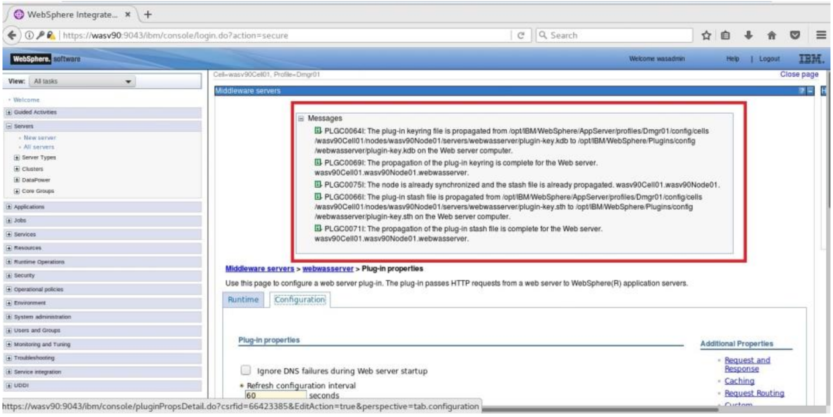
Web server copy of Web server plug-in files:

* Web server copy of Web server plug-in files:
/opt/IBM/WebSphere/Plugins/config/webwasserver/plugin-cfg.xml

https://wasv90:9043/ibm/console/pluginPropsDetail.do?csrfid=66423385&EditAction=true&perspective=tab.configuration



Step 10: Keyring file should be propagated to the web server.



Messages

- PLGC0064I: The plug-in keyring file is propagated from /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells /wasv90Cell01/nodes/wasv90Node01/servers/webwasserver/plugin-key.kdb to /opt/IBM/WebSphere/Plugins/config /webwasserver/plugin-key.kdb on the Web server computer.
- PLGC0069I: The propagation of the plug-in keyring is complete for the Web server. wasv90Cell01.wasv90Node01.webwasserver.
- PLGC0075I: The node is already synchronized and the stash file is already propagated. wasv90Cell01.wasv90Node01.
- PLGC0066I: The plug-in stash file is propagated from /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells /wasv90Cell01/nodes/wasv90Node01/servers/webwasserver/plugin-key.sth to /opt/IBM/WebSphere/Plugins/config /webwasserver/plugin-key.sth on the Web server computer.
- PLGC0071I: The propagation of the plug-in stash file is complete for the Web server. wasv90Cell01.wasv90Node01.webwasserver.

Middleware servers > webwasserver > Plug-in properties

Use this page to configure a web server plug-in. The plug-in passes HTTP requests from a web server to WebSphere(R) application servers.

Runtime | Configuration

Plug-in properties

□ Ignore DNS failures during Web server startup

* Refresh configuration interval
60 seconds

Additional Properties
- Request and Response
- Caching
- Request Routing
- Custom

https://wasv90:9043/ibm/console/pluginPropsDetail.do?csrfid=66423385&EditAction=true&perspective=tab.configuration

Task 3 is complete!

SUMMARY

WebSphere uses Java Secure Socket Extension (JSSE) which is an SSL implementation. SSL capabilities such as handshaking are delivered by JSSE using X.509 standard public key infrastructure (PKI). SSL uses certificates to authenticate the service provider by exchanging keys between server and client. WebSphere Application Server store these certificates in password protected files, in keystores. WebSphere Application Server can use SSL in the cell between the nodes, or to secure the communication between external systems that WebSphere Application Server transmits sensitive information such as database connection, LDAP connection, messaging connection and web services connection.

REFERENCES

- http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp?topic=/com.ibm.swg.im.iis.productization.iisinfsv.install.doc/tasks/configurethewebsphereapplicationserverforssl.html
- http://www.ibm.com/developerworks/websphere/techjournal/1210_lansche/1210_lansche.html
- http://www-01.ibm.com/support/knowledgecenter/linuxonibm/liaag/l0wascry00_2013.htm?cp=linuxonibm%2F0-4-3-2-0
- http://www-01.ibm.com/support/docview.wss?uid=swg21179559

INDEX