Secure an Oracle SOA Suite Instance

Oracle SOA Suite on Marketplace security includes configuring SSL and importing external web service certificates.

About Security in Oracle SOA Suite on Marketplace

Learn how to secure applications deployed to your Oracle SOA Suite on Marketplace instance through the capabilities of Oracle Cloud and Oracle WebLogic Server.

An Oracle SOA Suite on Marketplace instance includes an Oracle WebLogic Server domain, which is comprised of an Administration Server and one or more Managed Servers. A domain also defines a security realm that controls authentication, authorization, role mapping, credential mapping and security auditing across all of the servers in the domain. Java applications deployed to this WebLogic Server domain can be associated with security roles and policies that protect the applications against unauthorized access. WebLogic Server supports various security providers that assign an identity to the requesting user. By default, users, groups, roles and policies are all maintained in WebLogic Server's embedded LDAP server.

To provide the highest level of network security, Oracle SOA Suite on Marketplace implements an "access by exception" architecture. You must explicitly grant network access to your service instance for administrators, application users or other cloud services. By default, a service instance is accessible only through secure protocols like HTTPS and SSH, and only using specific ports. You're also able to customize the default network security configuration to support different access rules and security policies.

Set Up Oracle SOA Suite to Use CA-Verified SSL Certificates (without load balancer)

You can replace the identity and trust of Oracle SOA Suite with custom identity and custom trust and register the Oracle SOA Suite server with digital certificates procured from public certificate authorities like digicert or any other third party authority.

As a prerequisite, register Oracle SOA Suite domain with the public DNS for CA verification. For the documentation purposes, the public IP of the Oracle SOA Suite domain is registered with mydomain.com and takes the CA signed certificates from mydomain.

The Enterprise Manager (EM) console needs to be accessible using the public domain name.

Register a Domain Name for Oracle SOA Suite

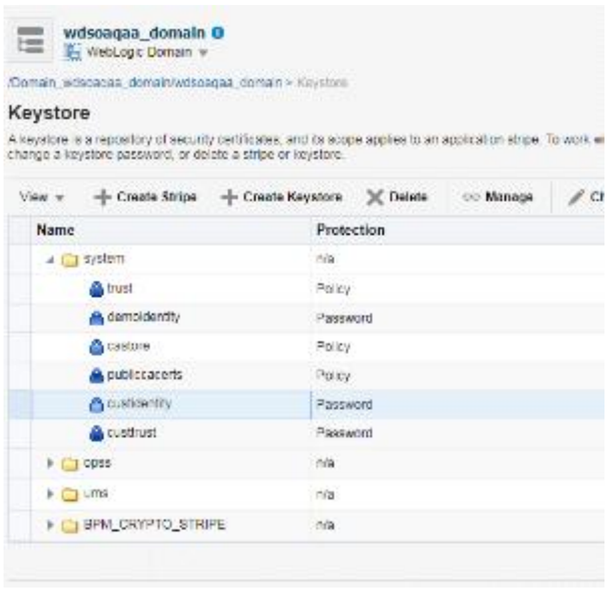
To register a domain name for Oracle SOA Suite:

1. Register a domain for Oracle SOA Suite server with a public DNS server. You can register your domain with any public DNS Server of your choice, mapping it to the public IP of Oracle SOA Suite. For example register soacs.oraclecloud.co.in domain in *mydomain.com*, mapping to the public IP of Oracle SOA Suite server.
2. Test access to the Enterprise Manager (EM) Console and the WebLogic Server Console through the domain name registered.

Create Custom Identity and Custom Trust Keystores and Generate a CSR

To create custom identity and custom trust keystores and generate a Certificate Signing Request (CSR):

1. Log in to the Enterprise Manager (EM) Console and access the Keystores page by opening WebLogic domain > **Security** > **Keystore**.
2. Under the system stripe, click **Create Keystore**.
3. Provide the following details for custom identity:
    a. **Keystore Name**: custIdentity
    b. **Protection**: select the **Password** option
    c. **Keystore Password**: enter the password
    d. **Confirm Password**: reenter the password
4. Click **Create Keystore** to create another new keystore.
5. Provide the following details for custom trust:
    a. **Keystore Name**: custTrust
    b. **Protection**: select the **Password** option
    c. **Keystore Password**: enter the password
    d. **Confirm Password**: reenter the password



Keystore

A keystore is a repository of security certificates, and its scope applies to an application stripe. To work w
change a keystore password, or delete a stripe or keystore.

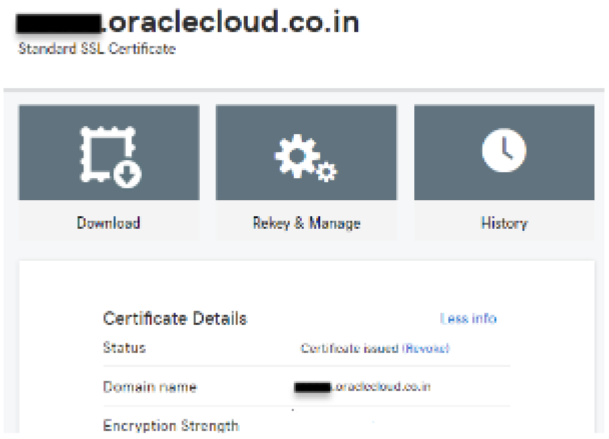
Name	Protection
system	n/a
trust	Policy
demoidentity	Password
castore	Policy
publiccacerts	Policy
custidentity	Password
custtrust	Password
opss	n/a
ums	n/a
BPM_CRYPTO_STRIPE	n/a

6. Click **Manage** on the custIdentity keystore name, click **Generate Keypair** to create a new key pair, and provide the following details:
    a. **Alias Name**: custIdentity
    b. **Common Name**: common name; for example, soacs.mydomain.com (domain name registered with public DNS)
    c. Organizational Unit: name of the organizational unit
    d. Organization: organization name
    e. Enter City, State, and Country names
    f. Key Type: RSA
    g. Key Size: 2048
7. Click **OK** to generate the key pair.
8. Select the newly created key pair and click **Generate CSR**.
9. Export the created CSR, share it with Certificate Authority, such as digicert CA, and get **root**, **intermediate**, and **signed** certificates.
   The certificate is generated for the domain name you specified in the **Common Name** field.
10. Download the certificates shared in the zip file from CA.
   It is not mandatory to create identity and trust keystore under the system stripe that comes with Oracle SOA Cloud Service provisioning by default. You can create a new custom stripe and create identity and trust keystores under it.

Share the CSR with CA to Get CA-Signed Certificates

To share the CSR with CA to get CA-signed certificates:

1. Select the new key pair you created under the custIdentity and click **Generate CSR**.
2. Export the created CSR and share it with the Certificate Authority and get root, intermediate, and signed certificates. The certificate is generated for the domain name you specified in the **Common Name** field.


[redacted].oraclecloud.co.in

Standard SSL Certificate

Certificate Details — Less info

Status: Certificate issued (Revoke)

Domain name: [redacted].oraclecloud.co.in

Encryption Strength

3. Download the certificates shared in the zip file from the CA.
   The zip file contains either of the following:
   - the three certificates individually - root, intermediate, and signed certificates
   - two root and intermediate certificates in one chain and the signed certificate separately
4. Double-click the certificate chain for the root and intermediate certificates. You can see the full chain when you click on certification path.
5. Extract the root and intermediate certificates individually by going to the certification path, select the certificate to be extracted (root or intermediate), and click **View Certificate**.
6. In the View Certificates popup, select the **Details** tab and click **Copy to File**.
7. In the Certificate Export wizard, click **Next**, select **Base 64 encoded X.509 (CER)**, then click **Next**. Export the certificate.
8. Name the exported certificate as root and intermediate certificates respectively.

Import CA Certificates

Certificate Authority (CA) certificates must be imported in the following order: first the signed server certificate, then the intermediate certificate, and then the root certificate.

To import CA certificates:

1. Use WLST commands to import the certificate chain into the identity keystore (custIdentity):
   a. Combine the three certificates into a single text file called chain.pem in the following order: signed server certificate, followed by intermediate certificate, followed by root certificate:

```
-----BEGIN CERTIFICATE-----
<signed server certificate>
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
<intermediate certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<root certificate>
-----END CERTIFICATE-----
```

b. As the opc user, use an FTP client such as WinSCP to copy chain.pem to the /tmp directory of the Administration Server VM.
c. Enter the following command to change the file ownership to the oracle:oracle user/group:

```
sudo chown oracle:oracle /tmp/chain.pem
```

d. Use the ssh command to connect to the Administration Server VM:

```
ssh -i private_key opc@AdminServerVM_IP_address
```

e. Change to the oracle user:

```
sudo su - oracle
```

f. Start WLST and access the Oracle Platform Security Services (OPSS) key store service:

```
/u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
connect('username','password','t3s://SOACS_hostname:7002')
svc = getOpssService(name='KeyStoreService')
```

Note:

If connecting to port 7002 does not work, try port 9071 or 9074 with the *SOACS_hostname*, or alternatively the internal hostname (as reported by the uname -n command at the Linux prompt).

g. Use the WLST importKeyStoreCertificate command to import chain.pem:

```
svc.importKeyStoreCertificate(appStripe='stripe', name='keystore', password='password',
alias='alias', keypassword='keypassword', type='entrytype',filepath='absolute_file_path')
```

For example:

```
svc.importKeyStoreCertificate(appStripe='system', name='custIdentity', password=welcome1,
alias='custIdentity', keypassword='welcome1, type='CertificateChain',
filepath='/tmp/chain.pem')
```

h. Exit WLST:

```
exit()
```

2. Use Oracle Enterprise Manager to import the certificate chain into the trust keystore (custTrust):
   a. Log in to the Enterprise Manager Console and access the Keystores page by opening WebLogic domain > **Security** > **Keystore**.
   b. Select the trust keystore (custTrust) and click **Manage**.
   c. Click **Import Certificate** and import the certificates in this order:

      i. the signed server certificate as a trusted certificate (alias `mySignedCert`)
      ii. the intermediate certificate from CA as a trusted certificate (alias `myInterCA`)
      iii. the root certificate from CA as a trusted certificate (alias `myRootCA`)

3. Set up `cacerts`:
   a. Use the `ssh` command to connect to the Administration Server VM:
      ```
      ssh -i private_key opc@AdminServerVM_IP_address
      ```
   b. Open `/u01/jdk/jre/lib/security`.
   c. Import the root and intermediate certificates into `cacerts` using the following commands:
      ```
      keytool -import -keystore cacerts -storepass keystorepassword -file rootCA.crt
      keytool -import -keystore cacerts -storepass keystorepassword -file interCA.crt
      ```
   d. Take a backup of the `cacerts` file for future use (for example, in case of JDK upgrade).
      Whenever there is an upgrade in the JDK, the backup copy needs to be copied back after upgrade as `cacerts`. Since all the upgrades are handled automatically, this is a critical step and all the upgrades need to be tracked.

Synchronize the Local Keystore with the Security Store

Synchronize keystores to synchronize information between the domain home and the Oracle Platform Security Services (OPSS) store in the database.

To synchronize keystores:

1. Use the `ssh` command to connect to the Administration Server VM:
   ```
   ssh -i private_key opc@AdminServerVM_IP_address
   ```
2. Change to the `oracle` user:
   ```
   sudo su - oracle
   ```
3. Start WLST and access the Oracle Platform Security Services (OPSS) key store service:
   ```
   /u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
   connect('username','password','t3s://hostname:7002')
   svc = getOpssService(name='KeyStoreService')
   ```

   Note:

   If connecting to port `7002` does not work, try port `9071` or `9074` with the *hostname*, or alternatively the internal hostname (as reported by the `uname -n` command at the Linux prompt).

4. Enter the following commands to synchronize the custom identity and custom trust keystores:

   Note:

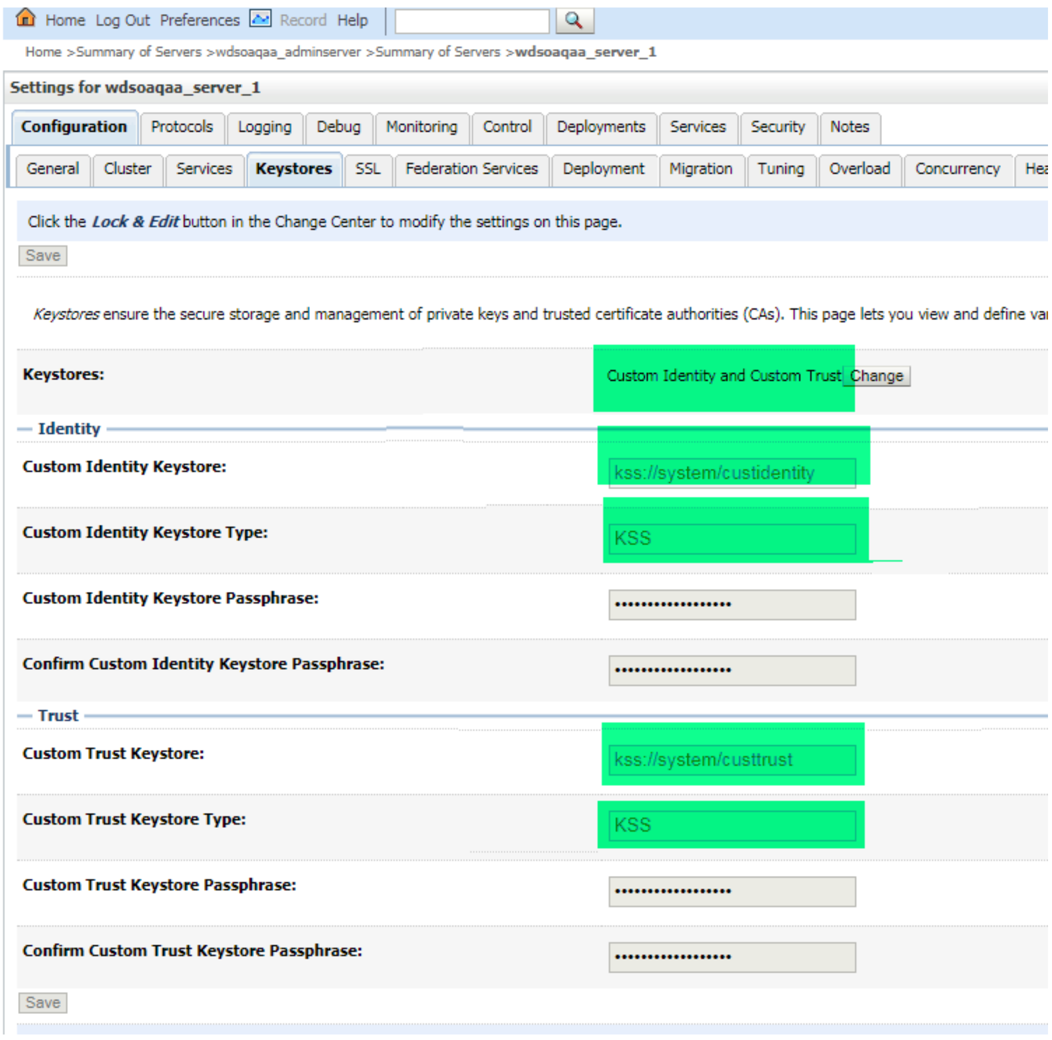
   This step is necessary only if you are using the `system` stripe. You do not need to synchronize the keystores if you are using a custom stripe:

```
svc. listKeyStoreAliases (appStripe="system”, name="custIdentity”, password=”*****”, type="*")
syncKeyStores(appStripe='system',keystoreFormat='KSS')
svc. listKeyStoreAliases (appStripe="system”, name="myKSSTrust”, password='****’, type="*")
syncKeyStores(appStripe='system',keystoreFormat='KSS')
```

Update WebLogic Keystores with Custom Identity and Trust

To update the WebLogic keystores with custom identity and custom trust:

1. Log in to the WebLogic Server Administration Console.
2. Navigate to **Servers** > **Admin Server** > **Configurations** > **Keystores** tab.
3. Change the **Keystores** to **Custom Identity** and **Custom Trust** and **Save**.
4. Provide the values for **Custom Identity**:
   - **Custom Identity Keystore:** kss://system/custidentity
   - **Custom Identity KeyStore Type:** KSS
   - **Custom Identity PassPhrase:** enter the password given while creating the custIdentity keystore
   - **Confirm Custom Identity PassPhrase:** reenter the password
5. Provide the values for **Custom Trust**:
   - **Custom Trust Keystore:** kss://system/custTrust
   - **Custom Trust KeyStore Type:** KSS
   - **Custom Trust PassPhrase:** enter the password given while creating the custIdentity keystore
   - **Confirm Custom Trust PassPhrase:** reenter the password
6. Click **Save** and then activate changes.


Home Log Out Preferences Record Help

Home >Summary of Servers >wdsoaqaa_adminserver >Summary of Servers >**wdsoaqaa_server_1**

Settings for wdsoaqaa_server_1

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes

General | Cluster | Services | Keystores | SSL | Federation Services | Deployment | Migration | Tuning | Overload | Concurrency | Hea

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Save

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define va

Keystores:	Custom Identity and Custom Trust Change
Identity	
Custom Identity Keystore:	kss://system/custidentity
Custom Identity Keystore Type:	KSS
Custom Identity Keystore Passphrase:	••••••••••••••••••
Confirm Custom Identity Keystore Passphrase:	••••••••••••••••••
Trust	
Custom Trust Keystore:	kss://system/custtrust
Custom Trust Keystore Type:	KSS
Custom Trust Keystore Passphrase:	••••••••••••••••••
Confirm Custom Trust Keystore Passphrase:	••••••••••••••••••

Save

7. On the **SSL** tab, provide the following details:
   - **Private Key Alias:** custIdentity (this is the alias given while creating keypair in the custIdentity keystore)
   - **Private Key PassPhrase:** enter the password given while creating the key pair under the custIdentity keystore.
   - **Confirm Private Key PassPhrase:** reenter the password.
8. In the **Advanced** section, change **Hostname Verification** to **None**. Click **Save** and activate changes.
   The Managed Server steps do not require a restart. Therefore, after activating the changes, you can check if the SSL URLs that open on Managed Server ports show the updated certificates.

9. Repeat steps 1 to 7 for the Administration Server. Administration Server changes require a restart.
10. Stop the Administration Server, Managed Server, and Node Manager.
    Before restart, make sure that the Node Manager changes are done.

Update the Node Manager and boot.properties File

To update the Node Manager and boot.properties file:

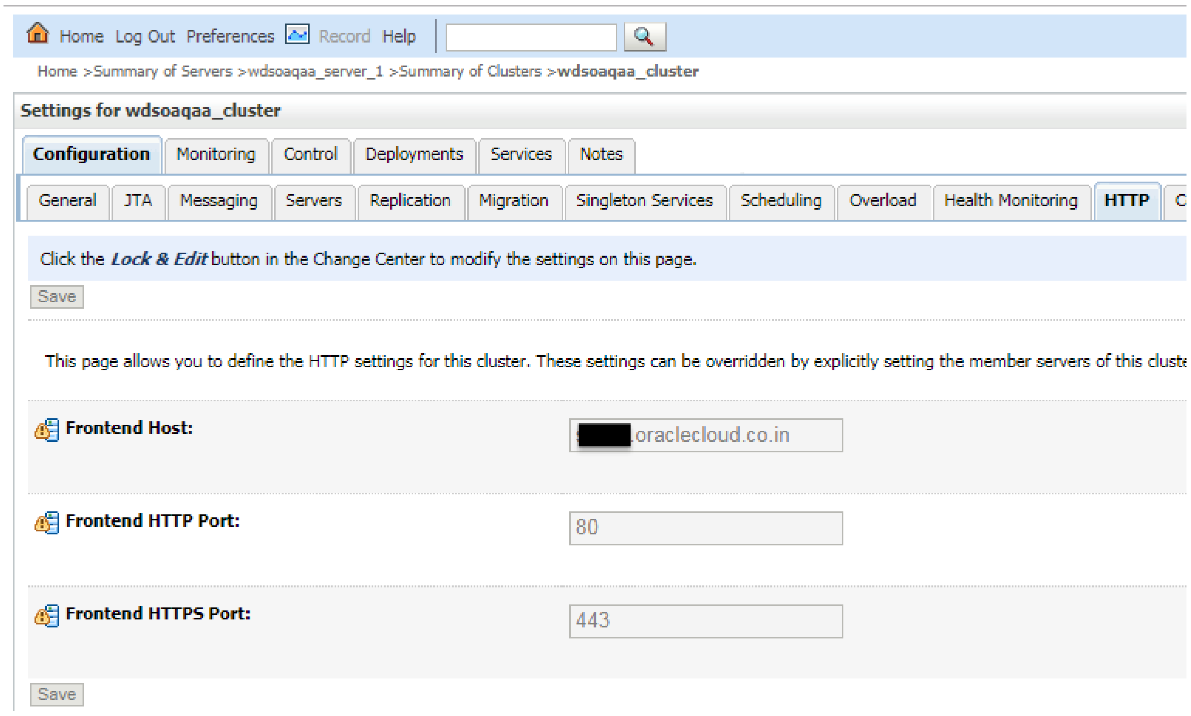
1. Access the Node Manager:
   ```
   cd /u01/data/domains/DomainName/nodemanager
   ```
2. Edit nodemanager.properties and add the following properties:
   ```
   # added for custom identity and custom trust
   KeyStores=CustomIdentityAndCustomTrust
   CustomIdentityAlias=custIdentity
   CustomIdentityKeyStoreFileName=kss://system/custIdentity
   CustomIdentityKeyStorePassPhrase=*********
   CustomIdentityKeyStoreType=KSS
   CustomIdentityPrivateKeyPassPhrase=*********
   CustomTrustKeyStoreFileName=kss://system/custTrust
   ```
3. Edit startNodeManager.sh under /u01/data/domains/YourDomain/bin/ to add the following properties during startup in JAVA_OPTIONS:
   ```
   JAVA_OPTIONS="${JAVA_OPTIONS}
   -Dweblogic.nodemanager.sslHostNameVerificationEnabled=false -
   Djava.security.egd=file:/dev/./urandom"
   ```

The JAVA_OPTIONS for a 12.2.1.4 environment is as follows:

```
JAVA_OPTIONS="${JAVA_OPTIONS}
-Doracle.security.jps.config=/u01/data/domains/TPLSOADE_domain/config/fmwconfig/jps-config-jse.xml
-Dcommon.components.home=/u01/app/oracle/middleware/oracle_common -Dopss.version=12.2.1.2
-Dweblogic.nodemanager.sslHostNameVerificationEnabled=false -Djava.security.egd=file:/dev/./urandom"
```

4. Use the ssh command to connect to the VM as the opc user:
   ```
   ssh -i private_key opc@VM_IP_address
   ```
5. Change to the oracle user:
   ```
   sudo su - oracle
   ```
6. Access the boot.properties file:
   ```
   cd /u01/data/domains/YourDomain/servers/YourManagedServer/security
   ```
7. Take a backup of boot.properties.
8. Open boot.properties and comment the line #TrustKeyStore=DemoTrust (if present) and save.
9. Update the managedServer boot properties by accessing the nodemanager:
   ```
   cd /u01/data/domains/YourDomain/servers/YourManagedServer/data/nodemanager
   ```
10. Take a backup of boot.properties.
11. Edit the boot.properties file, comment the line #TrustKeyStore=DemoTrust
12. Add the following lines at the end of boot.properties in the Managed Server:
    ```
    CustomTrustKeyStoreFileName=kss://system/custTrust
    TrustKeyStore=CustomTrust
    CustomTrustKeyStorePassPhrase=****
    CustomTrustKeyStoreType=KSS
    ```
13. Save boot.properties.

14. To make changes in SetDomainEnv.sh, remove the following property: Djavax.net.ssl.trustStore=%WL_HOME%\server\lib\DemoTrust.jks
15. To update the **Frontend Host Port**, update the host and port in the WebLogic Server Console to reflect the domain name and ports:
    a. In the WebLogic Server Console, navigate to **Environments** > **Clusters** > **Cluster Name** > **HTTP** tab.
    b. Update the **Frontend Host** as the domain name.
    c. Update the **Frontend HTTP Port**, default is port 80.
    d. Update the **Frontend HTTPS Port**, default is port 443.


Home Log Out Preferences Record Help
Home >Summary of Servers >wdsoaqaa_server_1 >Summary of Clusters >wdsoaqaa_cluster
Configuration Monitoring Control Deployments Services Notes
General JTA Messaging Servers Replication Migration Singleton Services Scheduling Overload Health Monitoring HTTP C
Save

Settings for wdsoaqaa_cluster

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

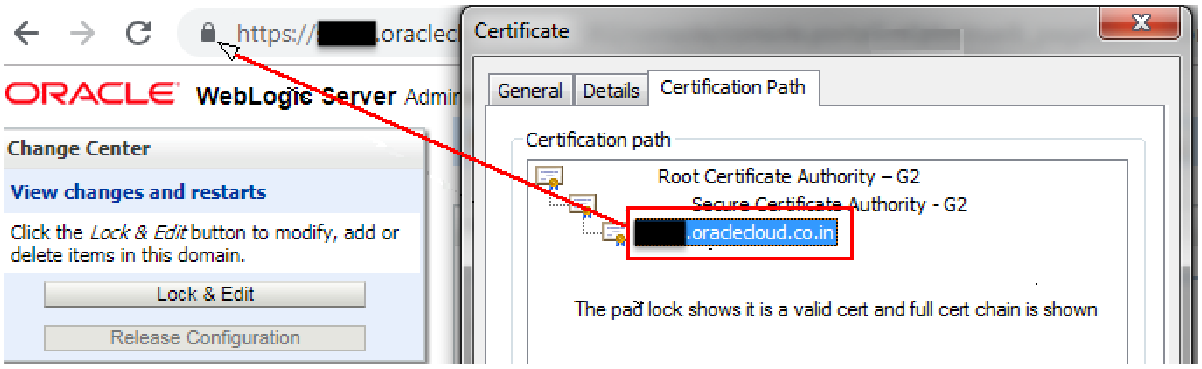
This page allows you to define the HTTP settings for this cluster. These settings can be overridden by explicitly setting the member servers of this cluste

Frontend Host:	[redacted].oraclecloud.co.in
Frontend HTTP Port:	80
Frontend HTTPS Port:	443

16. Start the Node Manager, Administration Server, and Managed Server, in this order.

Verify the Environment

When you restart the environment, the Aministration Server and Managed Server user interface shows the certificates as trusted:

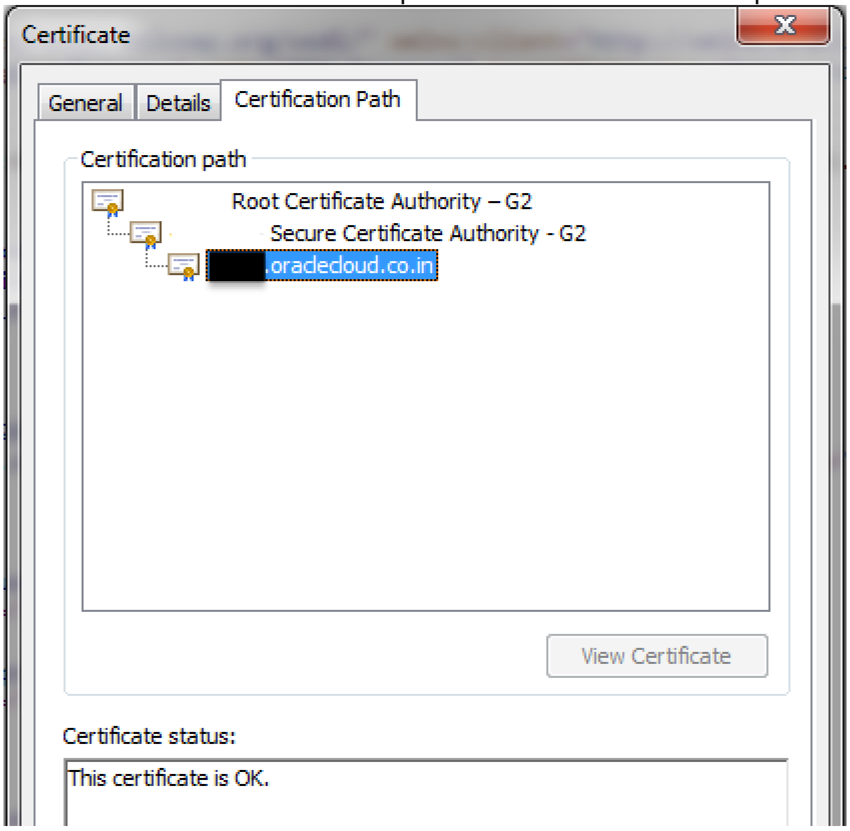

https://[redacted].oraclec
ORACLE WebLogic Server Admin
Change Center
View changes and restarts
Click the *Lock & Edit* button to modify, add or delete items in this domain.
Lock & Edit
Release Configuration
Certificate
General
Details
Certification Path
Certification path
Root Certificate Authority – G2
Secure Certificate Authority - G2
[redacted].oraclecloud.co.in

The pad lock shows it is a valid cert and full cert chain is shown

To verify the environment:

1. Deploy a HelloWorld composite and verify that the client endpoint URL can be opened on https host and port.

   The valid certificate chain is present on the client endpoint URL:


Certificate
General
Details
Certification Path
Certification path
Root Certificate Authority – G2
Secure Certificate Authority - G2
[redacted].oraclecloud.co.in
View Certificate

Certificate status:

This certificate is OK.

2. To invoke the client end point from any other composite, import all the certificates (signed server, intermediate, and root) present in the WSDL into the truststore of the server from where the parent composite is deployed.

Set Two-Way SSL Authentication

Two-way SSL authentication creates a truststore and a keystore on both the client and the server. It is not mandatory to set the two-way authentication.

To set the two-way authentication:

1. Log in to the WebLogic Server Administration Console.
2. On the Managed Server, select the **SSL** tab and click **Advanced**.
3. Select **Lock and Edit**.
4. For **Two Way Client Cert Behavior**, select **Client Certs Requested and Enforced** from the drop-down list.
5. Click **Save** and activate the changes.
   This change in the property does not require a WebLogic Server restart.

Import Certificates of External Web Services with HTTPS in Oracle SOA Suite

To import the certificate chain, which prevents a SSLHandshakeExceptions error from occurring while invoking an HTTPS service, complete the following steps:

- Export the Certificate Chain of the HTTPS WSDL Called in Oracle SOA Suite
- Import the Certificate Chain of the HTTPS WSDL Called in the Oracle SOA Suite Trust Store
- Import the Certificate Chain of the HTTPS WSDL Called in the Java Trust Store
- Restart the Administration and Managed Servers
- Troubleshoot Issues

Export the Certificate Chain of the HTTPS WSDL Called in Oracle SOA Suite

To export the certificate chain of the HTTPS WSDL:

1. Open the HTTPS URL that is called from the Oracle SOA/Oracle Service Bus composite in the Firefox browser.
2. Click the **padlock** icon to the left of the URL.
3. Under **Secure Connection**, select **More Information**.
4. Go to the **Security** tab and click **View Certificates**.
5. In Certificate Viewer dialog, click the **Details** tab and select each certificate.
6. Click **Export**.
   Once the certificates are exported, you can use secure copy (SCP) to copy them onto the virtual machines where the Oracle SOA/Oracle Service Bus servers are running.

Import the Certificate Chain of the HTTPS WSDL Called in the Oracle SOA Suite Trust Store

Note:

In a multinode cluster, the certificate chain must be imported to the keystores on all nodes of the cluster.

To import the certificate chain of the HTTPS WSDL called in the Oracle SOA Suite trust store:

1. Check the setDomainEnv.sh file to see if you have a DemoTrust.jks entry in EXTRA_JAVA_PROPERTIES present under DOMAIN_HOME.
2. If a DemoTrust.jks entry exists, use the keytool command to import the certificates in the JKS-based trust store:
   ```
   keytool -import -alias rootcrt1 -keystore
   ```

```
/u01/app/oracle/middleware/wlserver/server/lib/DemoTrust.jks -file RootcertFile.crt -
storepass DemoTrustKeyStorePassPhrase

keytool -import -alias intercrt2 -keystore
/u01/app/oracle/middleware/wlserver/server/lib/DemoTrust.jks -file InterMedCertFile.crt -
storepass DemoTrustKeyStorePassPhrase

keytool -import -alias cert3 -keystore
/u01/app/oracle/middleware/wlserver/server/lib/DemoTrust.jks -file cert3file.crt -storepass
DemoTrustKeyStorePassPhrase
```

3. If a DemoTrust.jks entry does not exist, use Oracle Enterprise Manager Fusion Middleware Control to import certificates in the KSS-based trust store:
   a. Go to the **Keystore** > **Weblogic Domain** drop down list, and select **Security** > **Keystore**.
   b. In the navigation tree, click **trust**.
   c. Click the **Manage** button.
   d. Click the **Import** button.
   e. In the Import Certificate dialog, select **Trusted Certificate** from the **Certificate Type** list.
   f. Provide the root certificate you previously exported from the WSDL URL.
   g. Repeat the same steps for other certificates in the WSDL URL chain.

   Synchronizing the keystores copies the certificates from the central repository to the local domain file. Perform the following commands:

   h. Start WLST:

   ```
   /u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
   ```

   i. Enter the administrator password and public IP address (the IP address used to access Oracle Enterprise Manager Fusion Middleware Control/Oracle WebLogic Server Console).

   ```
   connect('username', 'password', 'admin-server-host:admin-server-port')
   ```
   For example:

   ```
   connect('weblogic', 'welcome', 't3s://public IP:7002')
   ```

   j. Run the following commands:

   ```
   svc = getOpssService(name='KeyStoreService')
   syncKeyStores(appStripe='system', keystoreFormat='KSS')
   ```

Import the Certificate Chain of the HTTPS WSDL Called in the Java Trust Store

Note:

In a multinode cluster, the certificate chain must be imported into the cacerts location on all nodes of the cluster.

To import the certificate chain of the HTTPS WSDL called in the Java trust store:

- Add the certificate chain into the cacerts location. Sample keytool commands for importing certificates into the cacerts location are as follows:

  ```
  keytool -import -alias rootcrt1 -keystore /u01/jdk/jre/lib/security/cacerts -storepass changeit -file RootcertFile.crt

  keytool -import -alias intercrt2 -keystore /u01/jdk/jre/lib/security/cacerts -storepass changeit -file InterMedCertFile.crt

  keytool -import -alias cert3 -keystore /u01/jdk/jre/lib/security/cacerts -storepass changeit -file cert3file.crt
  ```

Restart the Administration and Managed Servers

Restart the Administration Server and Managed Servers once the certificates are imported. This is required for both JKS- and KSS-based certificates.

Troubleshoot Issues

Issue:

The following error occurs when invoking external Web Services:

```
Caused By: javax.xml.ws.WebServiceException: Could not determine wsdl ports.
WSDLException: faultCode=PARSER_ERROR:  Failed to read wsdl file at:
https://abc.xxx.com/...Service?WSDL%22, caused by:
java.security.NoSuchAlgorithmException: Error constructing implementation
```

Workaround:

1. Back up $DOMAIN_HOME/bin/setDomainEnv.sh.
2. Edit $DOMAIN_HOME/bin/setDomainEnv.sh and remove the following entries:

   ```
   -Djavax.net.ssl.trustStore=kss://system/xxx
   -Djavax.net.ssl.trustStoreType=kss
   ```

   Before:

   ```
   EXTRA_JAVA_PROPERTIES="-Djavax.net.ssl.trustStore=kss://system/xxx
   -Djavax.net.ssl.trustStoreType=kss ${EXTRA_JAVA_PROPERTIES}
   -Dsoa.archives.dir=${SOA_ORACLE_HOME}/soa
   ...
   ```

After:

```
EXTRA_JAVA_PROPERTIES=" ${EXTRA_JAVA_PROPERTIES}
-Dsoa.archives.dir=${SOA_ORACLE_HOME}/soa
...
```