

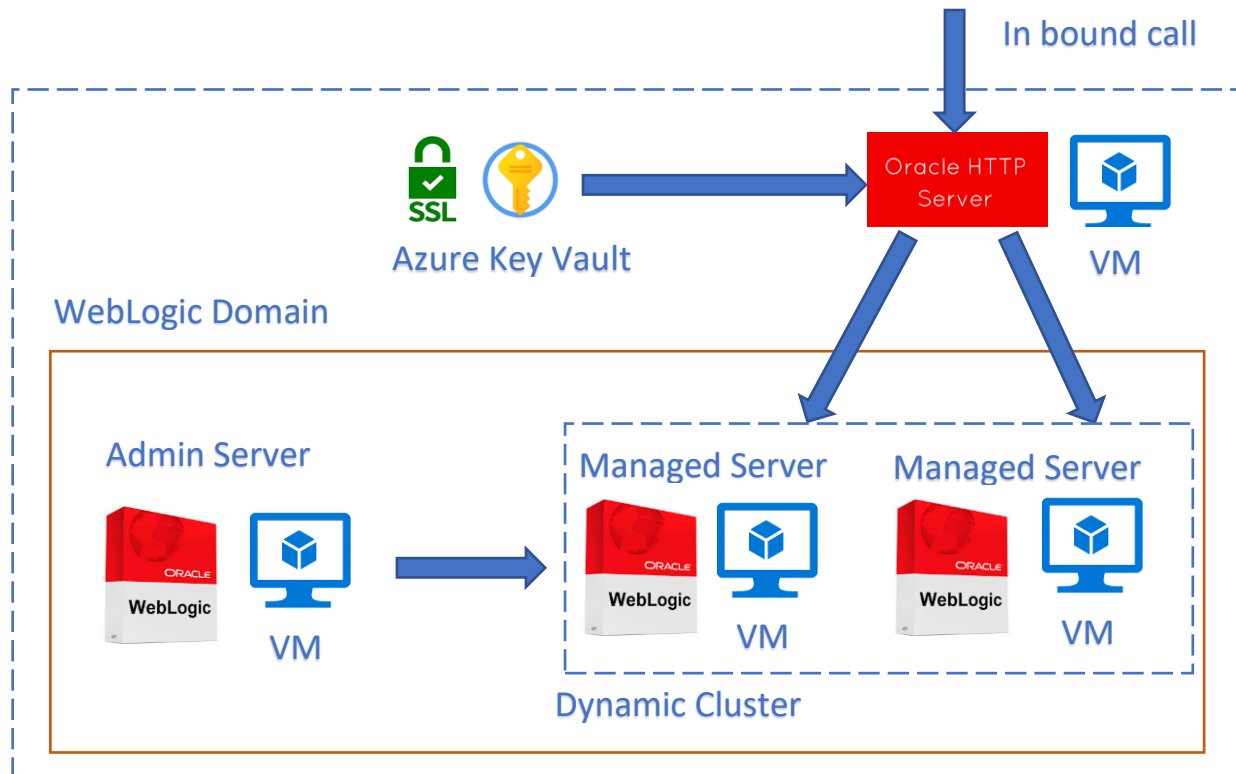
Oracle HTTP Server as load balancer for WebLogic Server dynamic cluster

Table of Contents

- Introduction..... 2
- Prerequisites..... 3
- Deploy WebLogic Server dynamic cluster with Oracle HTTP Server 3
 - Oracle HTTP Server details at Oracle HTTP Server Load Balancer deployment 3
 - Choose How would you like to provide required configuration 5
 - Option one: Upload existing KeyStores..... 5
 - Option two: Use KeyStores stored in Azure Key Vault 6
- Validate successful deployment of WebLogic Server and Oracle HTTP Server..... 9
- References 10
 - Create TLS/SSL certificate 10

Introduction

This tutorial walks you through the process of deploying WebLogic Server dynamic cluster with Oracle HTTP Server as load balancer. It covers the specific steps for uploading TLS/SSL certificates, creating Azure Key Vault, storing a TLS/SSL certificate and store the password in the Azure Key Vault. It describes how Azure Key Vault can be used for configuring TLS/SSL termination for Oracle HTTP Server.



Load balancing is an essential part of migrating your Oracle WebLogic Server dynamic cluster to Azure. The easiest solution is to use the [Oracle HTTP Server](#) as load balancer.

In this tutorial, you learn how to:

- ✓ Choose how to provide the TLS/SSL certificate to the Oracle HTTP Server
- ✓ Deploy WebLogic Server with Oracle HTTP Server
- ✓ Validate successful deployment of WebLogic Server and Oracle HTTP Server

Prerequisites

- [OpenSSL](#) on a computer running a UNIX-like command-line environment.

While there are could be other tools available for certificate management, this tutorial uses Open SSL for PKCS12 format certificate. You can find OpenSSL bundled with many GNU/Linux distributions.

- An active Azure subscription
 - If you don't have an Azure subscription, [create a free account](#) .
- JDK

While there are could be other tools available for certificate management, this tutorial uses keytool for JKS format certificate creation.

Deploy WebLogic Server dynamic cluster with Oracle HTTP Server

This section will show you how to provision a WebLogic Server dynamic cluster with Oracle HTTP Server as load balancer for the dynamic cluster nodes. Oracle HTTP Server will use the provided TLS/SSL certificate for TLS/SSL termination. For advanced details on TLS/SSL termination with Oracle HTTP Server, see [About Terminating SSL at Oracle HTTP Server](#).

To create the WebLogic Server dynamic cluster and Oracle HTTP Server, use the following steps.

First, begin the process of deploying a WebLogic Server configured with dynamic cluster as described in the [Deploy Oracle WebLogic Server N-Node Dynamic Cluster](#) documentation, but come back to this page when you reach Oracle HTTP Server Load Balancer.

Oracle HTTP Server details at Oracle HTTP Server Load Balancer deployment

Use the following steps to provide Oracle HTTP Server details

1. Select **Oracle HTTP Server Load Balancer** section during deployment.
2. Next to **Connect to Oracle HTTP Server?** select **Yes**.
3. Select **Oracle HTTP Server image** from drop down.
Refer [Azure Marketplace Offers](#) for more details.
4. Enter **Oracle HTTP Server Domain name**.
Oracle HTTP Server domain will be configured with provided domain name.
5. Enter **Oracle HTTP Server Component name**.
Oracle HTTP Server component name will be configured with provided component name.
Oracle HTTP Server component name holds all customized values like ports, WebLogic cluster address and TLS/SSL configuration.

6. Enter **Oracle HTTP Server NodeManager username**.
Oracle HTTP Server domain will be configured with provided NodeManager username.
7. Enter the password and confirm password for **Oracle HTTP Server NodeManager Password**.
Oracle HTTP Server domain will be configured with provided NodeManager password.
8. Enter the port for **Oracle HTTP Server HTTP Port**
Deployed WebLogic application can be accessed through Oracle HTTP Server using provided HTTP port.
9. Enter the port for **Oracle HTTP Server HTTPS Port**
Deployed WebLogic application can be accessed through Oracle HTTP Server using provided HTTPS port.
10. Enter the password and confirm password for **Oracle Vault Password**.

Connect to Oracle HTTP Server? Yes
 No

Oracle HTTP Server integration requires a TLS/SSL certificate to enable TLS/SSL termination. End-to-end TLS/SSL encryption is not supported by the template. The template will look for the certificate and its password in the Azure Key Vault specified here.
[Learn more](#)

Oracle HTTP Server image ⓘ	<input type="text" value="OHS 12.2.1.4.0 and JDK8 on Oracle Linux 7.6"/>
Oracle HTTP Server Domain name * ⓘ	<input type="text" value="ohsStandaloneDomain"/>
Oracle HTTP Server Component name * ⓘ	<input type="text" value="ohs_component"/>
Oracle HTTP Server NodeManager username * ⓘ	<input type="text" value="weblogic"/> ✓
Oracle HTTP Server NodeManager Password * ⓘ	<input type="password" value="●●●●●●●●"/> ✓
Confirm password * ⓘ	<input type="password" value="●●●●●●●●"/> ✓
Oracle HTTP Server HTTP port * ⓘ	<input type="text" value="7777"/>
Oracle HTTP Server HTTPS port * ⓘ	<input type="text" value="4444"/>
Oracle Vault Password * ⓘ	<input type="password" value="●●●●●●●●"/> ✓
Confirm password * ⓘ	<input type="password" value="●●●●●●●●"/> ✓
How would you like to provide required configuration ⓘ	<input checked="" type="radio"/> Upload existing KeyStores <input type="radio"/> Use KeyStores stored in Azure Key Vault

Choose How would you like to provide required configuration

There are two options to provide the TLS/SSL certificate for the Oracle HTTP Server. You can choose one of them, during deployment, based on your requirement.

The below section explains each of the two options in detail.

Option one: Upload existing KeyStores

This option is suitable when certificates are available, and user doesn't want to configure Azure Key Vault manually.

To upload the TLS/SSL certificate, use the following steps:


1. Either [Create TLS/SSL certificate](#) or get certificate from the issuer in the format of PKCS12 or JKS.
2. Select **Upload existing KeyStores** for **How would you like to provide required configuration**.
3. Under **TLS/SSL Configuration Settings**
 - Next to **TLS/SSL certificate Data file (. jks,.p12)**, browse the certificate and upload.
 - Enter the password for the certificate in the **Password** and **Confirm password** boxes.
 - Under **Type of the Certificate format (JKS,PKCS12)**, select PKCS12 or JKS depending on what certificate format is uploaded.
4. Choose whether to deny public traffic directly to the managed server nodes. Selecting **Yes** will make that managed server ports are not accessible to the internet. Deployed WebLogic application will only be accessible through Oracle HTTP Server configured ports.
5. Select **Review+Create**.
6. Once you see **Validation passed**, select **Create**.


How would you like to provide required configuration ⓘ


Upload existing KeyStores


Use KeyStores stored in Azure Key Vault

TLS/SSL Configuration Settings

TLS/SSL certificate Data file(jks,.p12) * ⓘ 

Password * ⓘ 

Confirm password * ⓘ 

Type of the certificate format(JKS,PKCS12) * ⓘ 

Deny public traffic for managed server? * ⓘ

No

Yes

[Review + create](#) [< Previous](#) [Next : DNS Configuration >](#)

This will start the process of creating the WebLogic Server domain with dynamic cluster and its front-end Oracle HTTP Server. When the deployment completes, select **Go to resource group**.

Option two: Use KeyStores stored in Azure Key Vault

This option is suitable for production or non-production workloads, depending on the TLS/SSL certificate provided. This option requires you to store the certificate and its password in the Azure Key Vault before continuing. If you have an existing Key Vault you want to use, skip to the section [Create TLS/SSL certificate](#). Otherwise, continue to the next section.

Create an Azure Key Vault

This section shows how to use the Azure portal to create an Azure Key Vault.

1. From the Azure portal menu, or from the Home page, select Create a resource.
2. In the Search box, enter **Key Vault**.
3. From the results list, choose **Key Vault**.
4. On the Key Vault section, choose **Create**.
5. On the Create key vault section provide the following information:
 - **Subscription:** Choose a subscription
 - Under **Resource group**, choose **Create new** and enter a resource group name. Take note of the resource group name. You'll need it later when deploying WebLogic Server.
 - **Key Vault Name:** A unique name is required. Take note of the key vault name. You'll need it later when deploying WebLogic Server.

Note

You may use the same name for both **Resource Group** and **Key vault name**.

- In the **Location** pull-down menu, choose a location.
 - Leave the other options to their defaults.
6. Select **Next: Access Policy**.
 7. Under **Enable Access to**, select **Azure Resource Manager for template deployment**.
 8. Select **Review + Create**.
 9. Select **Create**

Key vault creation is lightweight, typically completing in less than two minutes. When deployment completes, select **Go to resource** and continue to the next section.

Store the KeyStores in the Key Vault

This section shows how to store the keystores and its password in the Key Vault created in the preceding sections.

To store the certificate, follow these steps:

1. From the Azure portal, put the cursor in the search bar at the top of the page and type the name of the Key Vault you created earlier in the tutorial.
2. Your Key Vault should appear under the **Resources** heading. Select it.
3. In the **Settings** section, select **Secrets**.
4. Select **Generate/Import**.
5. Under **Upload options**, leave the default value.
6. Under **Name**, enter myCertSecretData, or whatever name you like.
7. Under **Value**, enter the content of the *mycert.txt* file.
8. Leave the remaining values at their defaults and select **Create**.

To store the password for the certificate, follow these steps:

1. You'll be returned to the **Secrets** page. Select **Generate/Import**.
2. Under **Upload options**, leave the default value.
3. Under **Name**, enter myCertSecretPassword, or whatever name you like.
4. Under **Value**, enter the note down password for the certificate.
5. Leave the remaining values at their defaults and select **Create**.
6. You'll be returned to the **Secrets** page.

Key Vault details at Use KeyStores stored in Azure Key Vault

Now that you have a Key Vault with signed SSL certificate and its password stored as secrets, return to the **Oracle HTTP Server Load Balancer** section to enter Key Vault details for the deployment.

1. Select **Use KeyStores stored in Azure Key Vault** for **How would you like to provide required configuration**.
2. Under **Certificate Type**, select PKCS12 or JKS depending on what certificate format is created as per [Create TLS/SSL certificate](#) .
3. Under **Resource group name in current subscription containing the KeyVault**, enter the name of the resource group containing the Key Vault you created earlier.
4. Under **Name of the Azure KeyVault containing secrets for the Certificate for TLS/SSL Termination**, enter the name of the Key Vault.
5. Under **The name of the secret in the specified KeyVault whose value is the TLS/SSL Certificate Data**, enter myCertSecretData, or whatever name you entered previously.
6. Under **The name of the secret in the specified KeyVault whose value is the password for the TLS/SSL Certificate**, enter myCertSecretPassword, or whatever name you entered previously.

7. Choose whether to deny public traffic directly to the managed server nodes. Selecting **Yes** will make that managed server ports are not accessible to the internet. Deployed WebLogic application will only be accessible through Oracle HTTP Server configured ports.
8. Select **Review+Create**.
9. Once you see **Validation passed**, select **Create**.

How would you like to provide required configuration ⓘ Upload existing KeyStores Use KeyStores stored in Azure Key Vault

TLS/SSL Configuration Settings

Certificate Type ⓘ ✓

Resource group name in current subscription containing the Key Vault * ⓘ ✓

Name of the Azure Key Vault containing secrets for the certificate for TLS/SSL Termination * ⓘ ✓

The name of the secret in the specified Key Vault whose value is the TLS/SSL certificate Data * ⓘ ✓

The name of the secret in the specified Key Vault whose value is the password for the TLS/SSL certificate * ⓘ ✓

Deny public traffic for managed server? * ⓘ No Yes

[Review + create](#) [< Previous](#) [Next : DNS Configuration >](#)

This will start the process of creating the WebLogic Server domain with dynamic cluster and its front-end Oracle HTTP Server. When the deployment completes, select **Go to resource group**.

Note

Details on certificate data and password will be validated during Oracle HTTP Server deployment. In order to deploy Oracle HTTP Server (OHS) successfully, make sure JKS/PKCS12 certificate is created properly with same password at all places.

Validate successful deployment of WebLogic Server and Oracle HTTP Server

This section shows a technique to quickly validate the successful deployment of the WebLogic Server dynamic cluster and Oracle HTTP Server.

1. Upon successful deployment, following values will be seen.

```
"outputs": {
  "adminConsole": {
    "type": "String",
    "value": "http://wls0-157be2b5ff-wlsd.eastus.cloudapp.azure.com:7001/console"
  },
  "adminHostName": {
    "type": "String",
    "value": "wls0-157be2b5ff-wlsd.eastus.cloudapp.azure.com"
  },
  "adminSecuredConsole": {
    "type": "String",
    "value": "https://wls0-157be2b5ff-wlsd.eastus.cloudapp.azure.com:7002/console"
  },
  "ohsAccessURL": {
    "type": "String",
    "value": "http://wls-0657f9dc82-ohsstandalonedomain.eastus.cloudapp.azure.com:7777"
  },
  "ohsSecureAccessURL": {
    "type": "String",
    "value": "https://wls-0657f9dc82-ohsstandalonedomain.eastus.cloudapp.azure.com:4444"
  },
}
```

At the end you should see

```
"provisioningState": "Succeeded",
```

```
"template": null,
```

References

Create TLS/SSL certificate

This section shows how to create a self-signed SSL certificate in a format suitable for use by Oracle HTTP Server deployed with WebLogic on Azure. The example provided below is one of the ways to create self-signed certificates for JKS and PKCS12 format.

Note

Self-signed certificates should only be used for testing purpose and it is not recommended for production purpose.

You need to choose one of the formats JKS or PKCS12 format and note down which format is selected. You'll need it later when deploying WebLogic Server with Oracle HTTP Server.

2. JKS format certificate

- Create JKS format file using following `$JAVA_HOME/bin/keytool` command

```
keytool -genkey -keyalg RSA -alias selfsigned -keystore mycert.jks -storepass password -  
validity 360 -keysize 2048 -keypass password -storetype jks
```

Provide all information prompted and store in a file, *mycert.jks*.

- Base64 encode *mycert.jks* file
This step is required only if you have opted [Option two: Use KeyStores stored in Azure Key Vault](#).

```
base64 mycert.jks -w 0 > mycert.txt
```

3. PKCS12 format certificate

- Create an RSA PRIVATE KEY

```
openssl genrsa 2048 > private.pem
```

- Create a corresponding public key

```
openssl req -x509 -new -key private.pem -out public.pem
```

Provide all information prompted.

- Export the certificate as a “. p12” file

```
openssl pkcs12 -export -in public.pem -inkey private.pem -out mycert.p12
```

Note down the Export Password provided.

- Base64 encode *mycert.pfx* file
This step is required only if you have opted [Option two: Use KeyStores stored in Azure Key Vault](#).

```
base64 mycert.pfx -w 0 > mycert.txt
```