6

Configuring High Availability

Objectives

After completing this lesson, you should be able to:

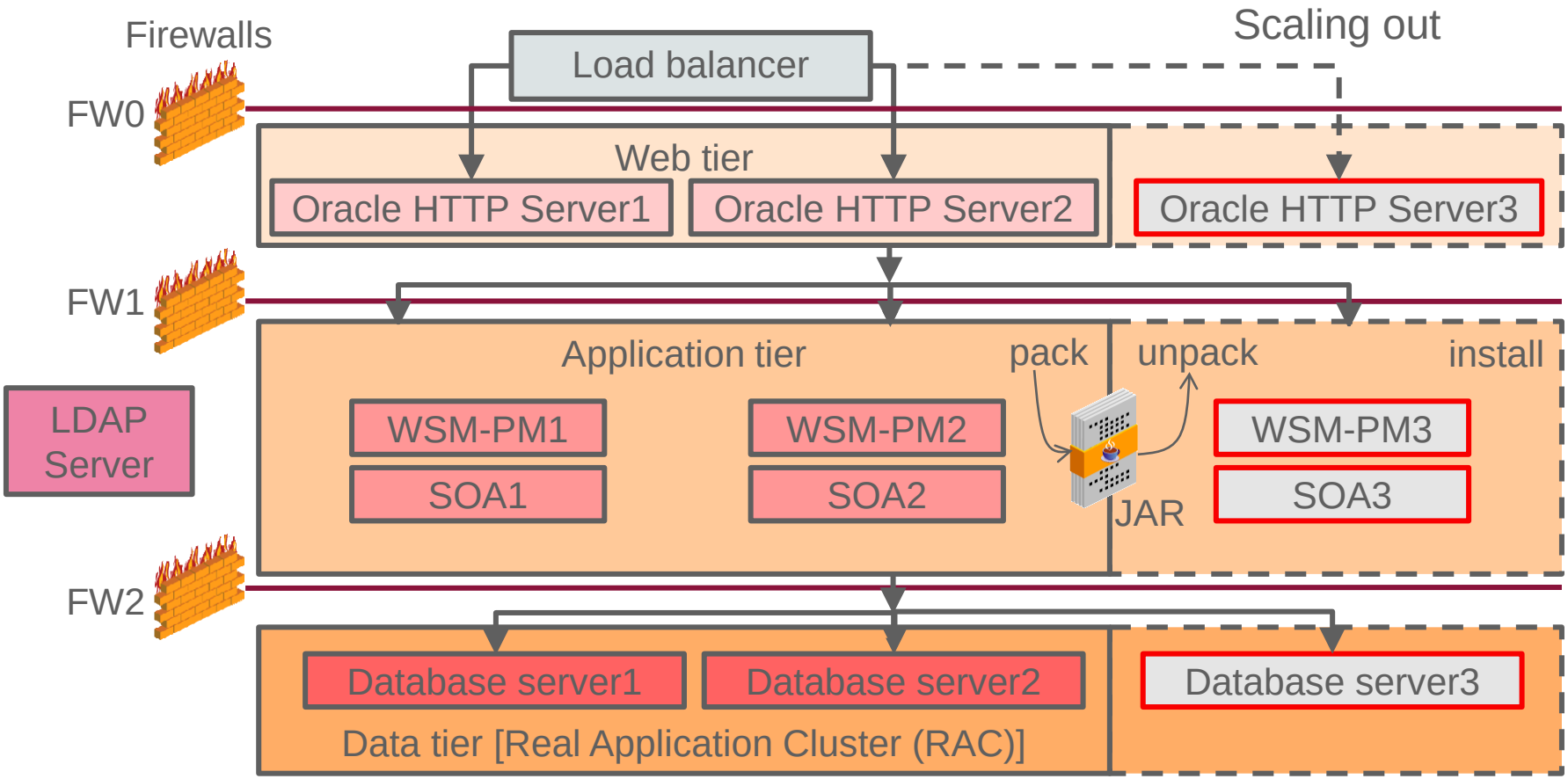
- Review the high availability architecture
  - Installing the data tier
  - Centralizing LDAP servers
  - Installing and configuring the web tier
  - Configuring a load balancer
  - Installing and configuring the middleware components
- Scale out an Enterprise Deployment topology
- Configure high availability for the Administration Server
- Configure a JCA Adapter and resources for applications

Agenda

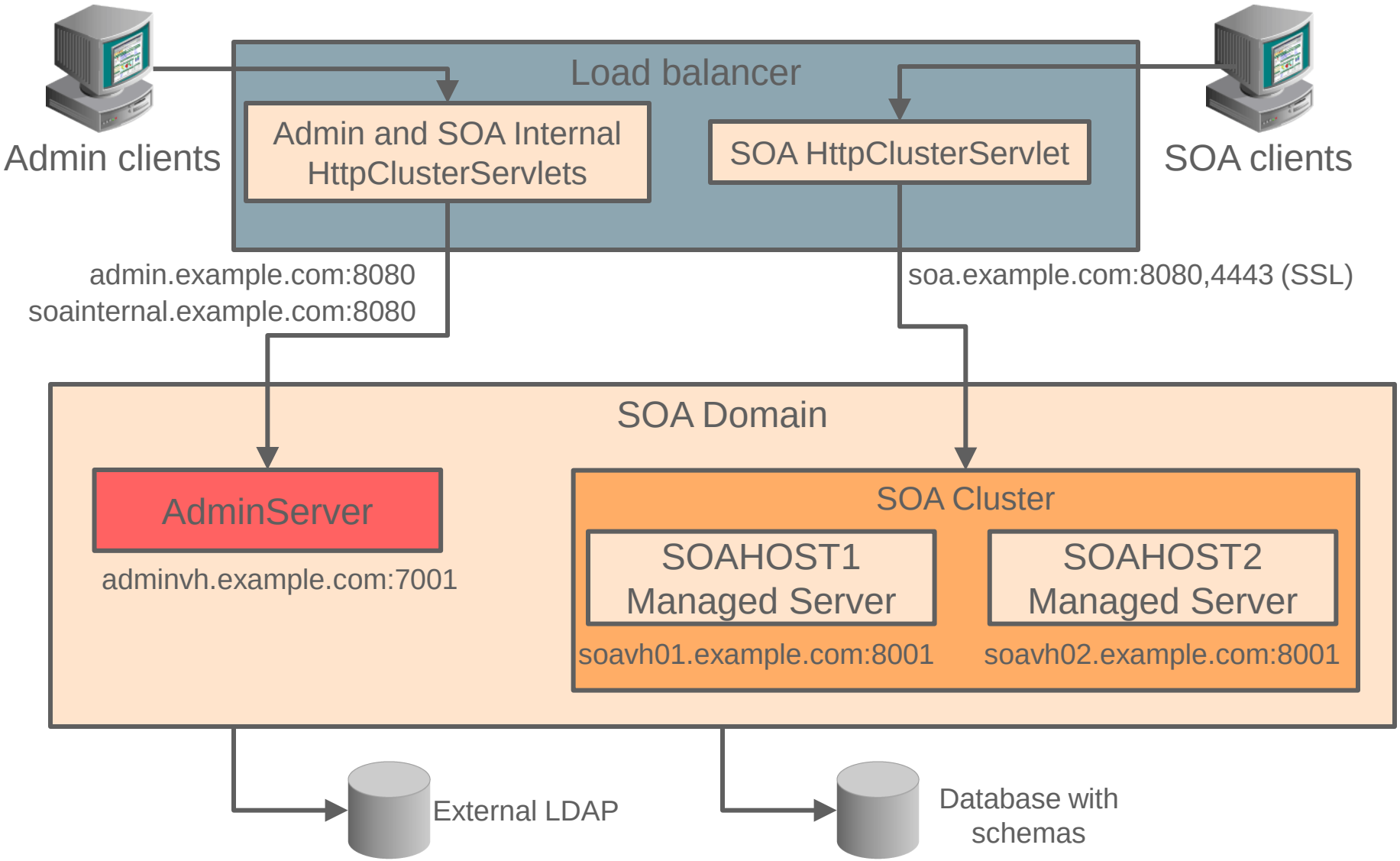
- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Reviewing a High Availability Architecture

The Oracle Enterprise Deployment (EDG) topology is a high availability reference model and guide.



Course High Availability Architecture: Review



Agenda

- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Roadmap for Scaling Out Your Topology

1. Create a virtual host name and IP address for the new host.
2. Create a new Managed Server in the cluster to listen with the new virtual host name and ports.
3. Create and target a new JMS Server for the new Managed Server instance.
4. Pack the updated domain configuration to create the Managed Server template.
5. Install the product binaries on the new host.
6. Unpack the Managed Server template on the new host.
7. Start and test the new Managed Server instance.
8. Consider updating the cluster messaging mode from unicast to multicast (depending on the number of servers in cluster).

Agenda

- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Planning Considerations for High Availability

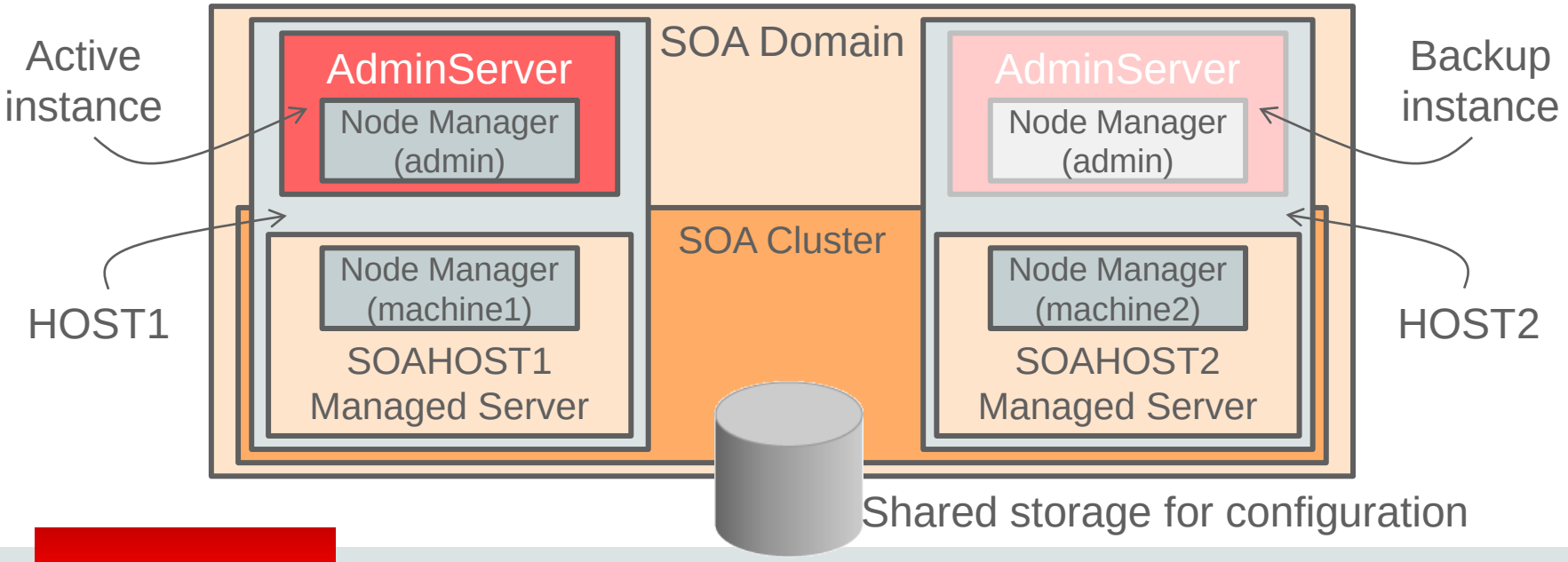
Part of configuring a highly available topology includes:

- Recognizing the importance of virtual host names and the related virtual IP addresses
- Starting Node Manager instances (per domain or per host)
- Testing manual failover of the Administration Server in an environment that is configured for high availability (only one per domain)
- Configuring for Whole Server Migration (underlying WebLogic Server functionality)

Administration Server High Availability Topology

The Administration Server:

- Can be active on a single host at any one time
- Must be configured on a virtual host to support failover to another host in the domain
- Hosts that are considered for running the AdminServer instance require access to the shared folders of the domain configuration



Failing Over the Administration Server

To fail over the Administration Server:

- Check that the Node Manager and Administration Server (on for example, adminvh.example.com) are inaccessible
- Perform the following steps on the backup host:
  - Enable the virtual IP address assigned to the virtual host name.
    - `$ ifconfig eth0:2 192.0.2.20 netmask 255.255.255.0`
  - Update the IP tables on the host.
    - `$ arping -q -U -c 3 -I eth0 192.0.2.20`

Run as super user.

  - Verify that the virtual host name is reachable.
    - `$ ping –c 1 adminvh.example.com`
  - Start Node Manager, and then the Administration Server on the host that services the adminvh.example.com virtual host name.
- Repeat the steps when the original host becomes available after shutting down and disabling items on the backup host.

Additional Post-Configuration Tasks

- Update the JCA Adapter configuration properties (mostly determined by application requirements):
  - File Adapter to make it highly available
  - Database Adapter
  - JMS Adapter
  - Others (as required)
- Create the JCA Adapter resources.

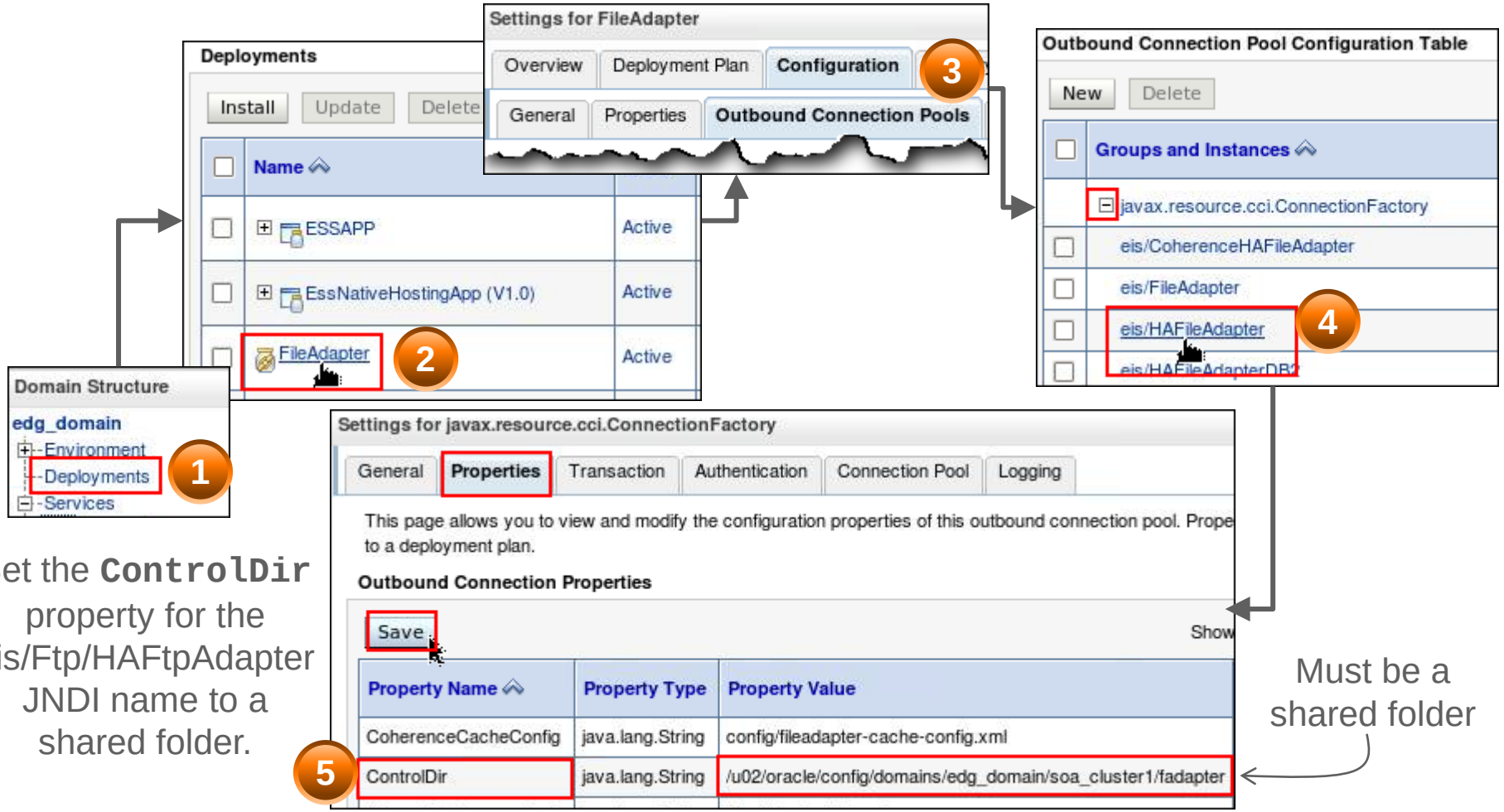
Note: JCA Adapter configuration is done in Oracle WebLogic Administration Server Console. In addition, JDBC data sources can also be created in Oracle Enterprise Manager Fusion Middleware Control.

Agenda

- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Making the File Adapter Highly Available

To make the Oracle File Adapter highly available:



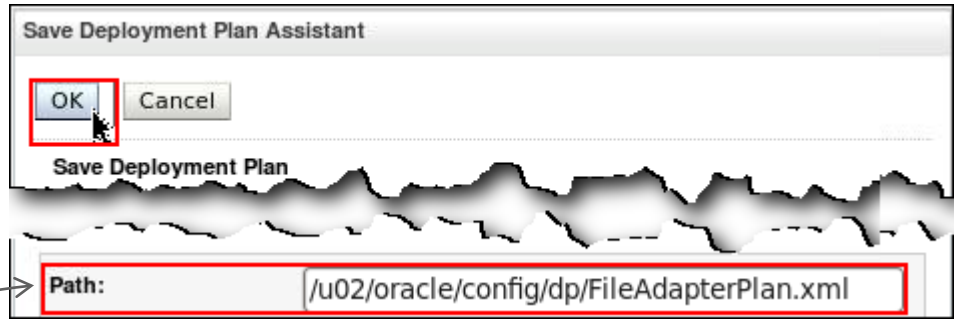
Set the **`ControlDir`** property for the eis/Ftp/HAFtpAdapter JNDI name to a shared folder.

Must be a shared folder

Modifying the File Adapter Deployment Descriptor

To save the updated File Adapter configuration:

- Create a folder for deployment plans (if not already present) on a shared disk that is accessible to all Managed Server instances in the cluster
- Save the Adapter deployment descriptor in the deployment plan folder



The path should be on a shared disk, below the `config` folder for the domain configuration.

High Availability for the Oracle Database Adapter

- High availability for the Oracle Database Adapter is supported (by default) by using an Oracle Database feature called skip locking, which is a distributed polling technique.
- Earlier versions of Oracle SOA Suite Database Adapters may be using Logical Delete polling because it performed better than a physical delete.
  - In a clustered environment with multiple nodes polling for the same data, a single record might get processed multiple times.
  - To avoid the problem of using Logical Delete polling, in the Database Adapter properties file (in `db.jca`), remove or clear `MarkReservedValue` (on the Logical Delete page of the Database Adapter wizard) to automatically enable skip locking.

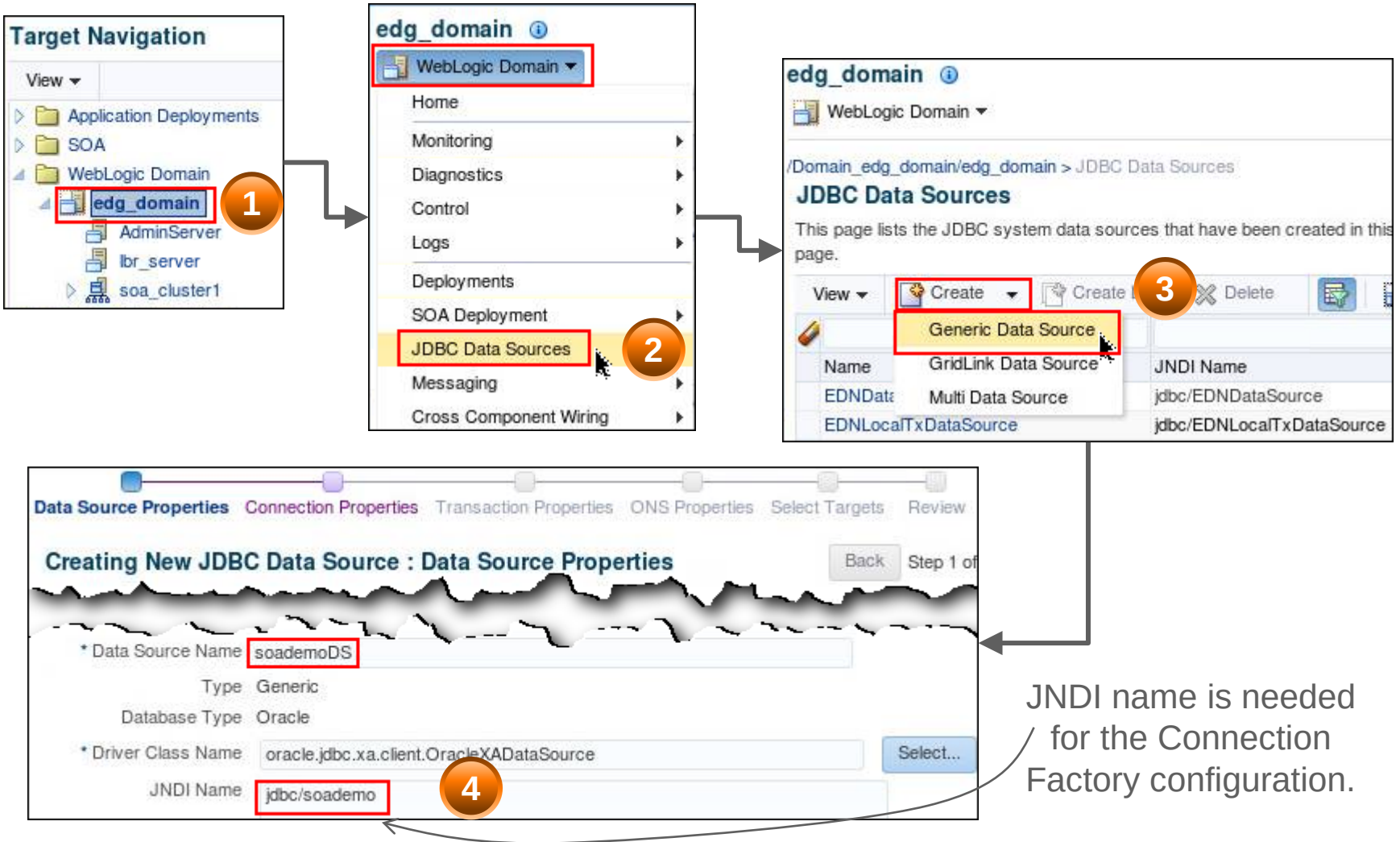
Preparing Resources for the Database Adapter

The resources required by a composite application that uses the Database Adapter are:

- JDBC data sources, which can be created in either of the two administration consoles
- A connection factory, which is:
  - Created by updating the Database Adapter in the WebLogic Administration Console
  - Associated with a target JDBC data source

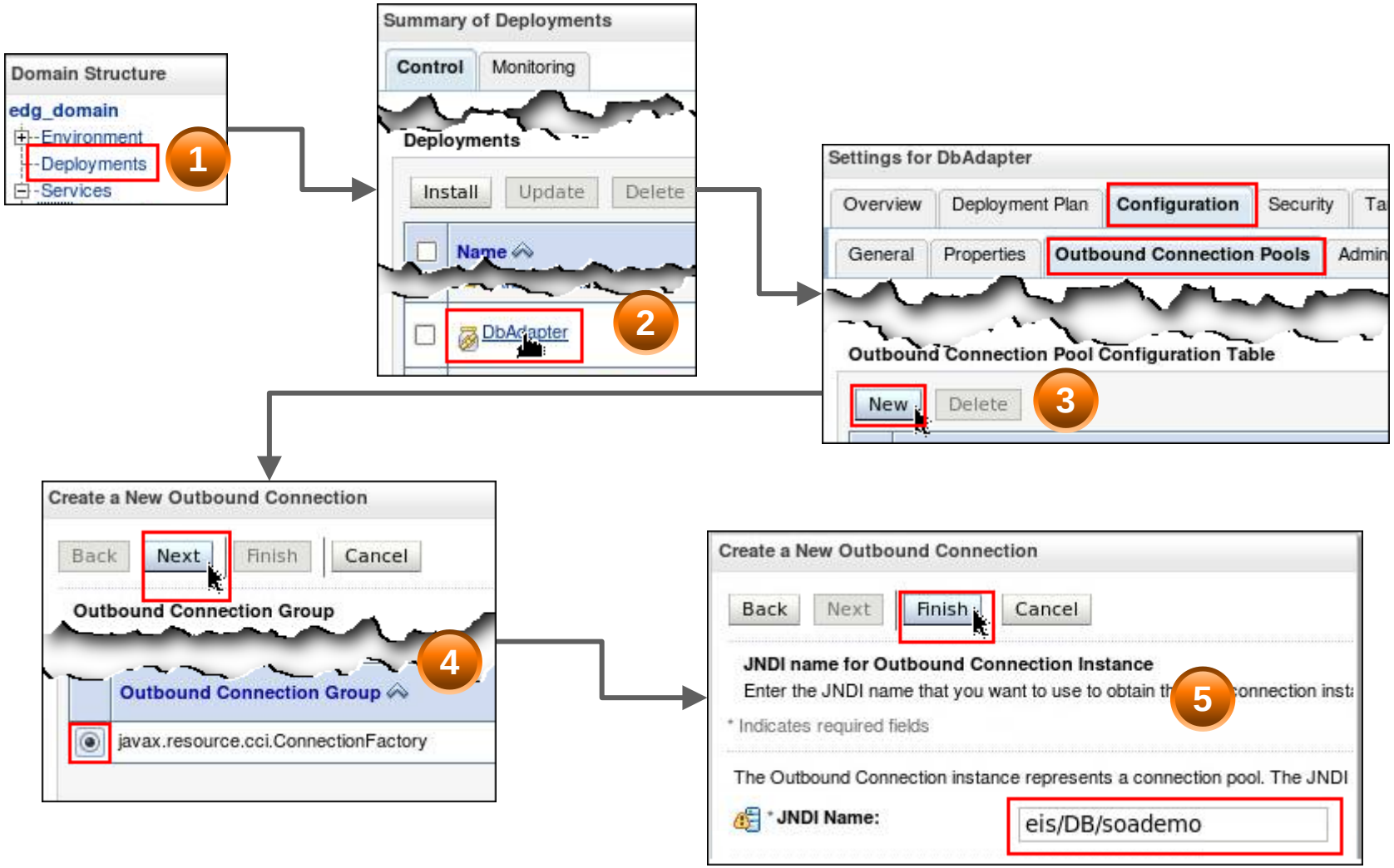
Note: There is a one-to-one association between a connection factory and a JDBC data source.

Creating a JDBC Data Source



JNDI name is needed for the Connection Factory configuration.

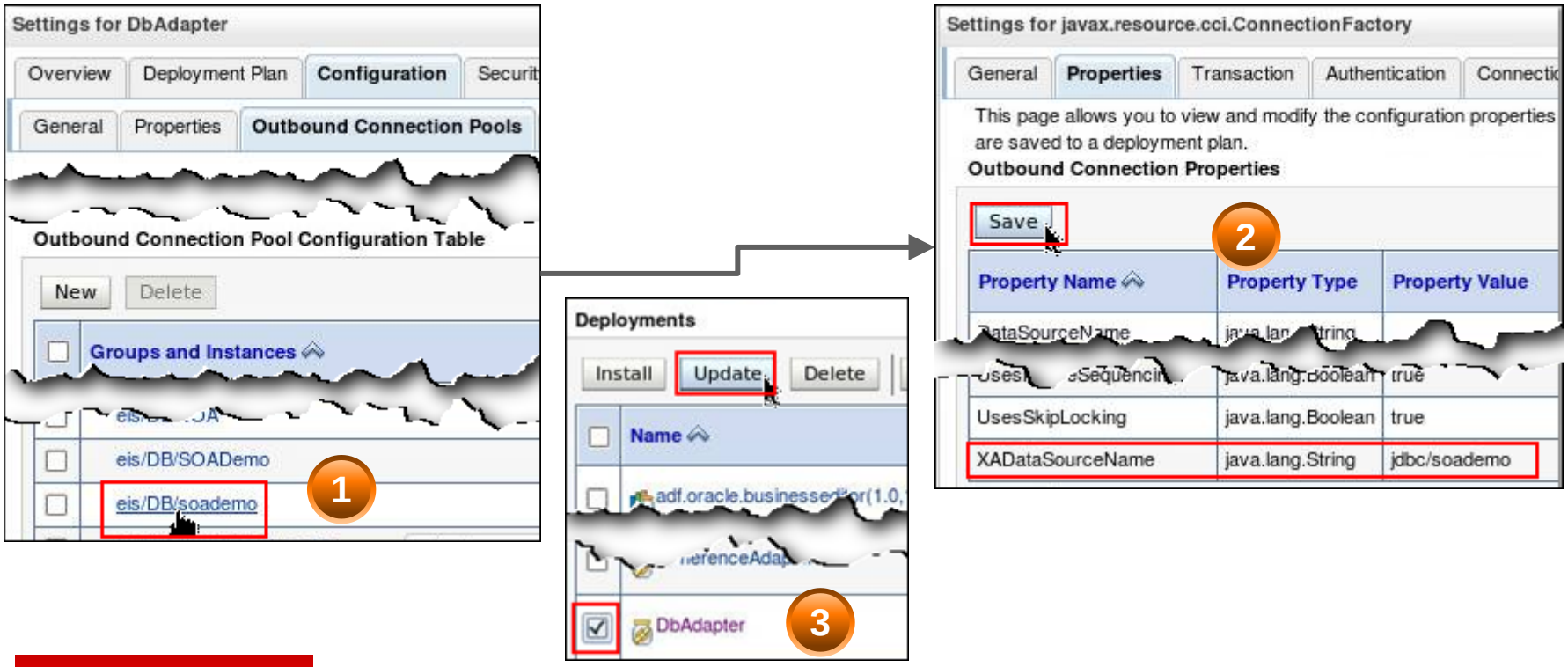
Creating a Database Adapter Connection Factory



Note: This configuration is saved to a deployment plan in a shared folder.

Configuring the Connection Factory Data Source

1. Configure the XADataSourceName value with the desired JNDI name of the JDBC data source.
2. Save the changes in a Database Adapter deployment plan.
3. Redeploy (update) the Database Adapter.



High Availability for Oracle JMS Adapters

Configuring the JMS Adapter to communicate with multiple servers in a cluster involves:

- Creating a JMS resource, such as a queue or topic, for application destinations
- Creating a JMS connection factory resource
- Creating a JMS connection pool with the following configured:
  - ConnectionFactoryLocation with the connection factory JNDI name
  - FactoryProperties with a list of all available cluster servers

Example FactoryProperties value:

```
java.naming.factory.initial=weblogic.jndi.WLInitialContex
   tFactory;java.naming.provider.url=t3://soavh01.exampl
   e.com:8001,soavh02.example.com:8001;java.naming.secur
   ity.principal=weblogic;java.naming.security.credentia
   ls=mypassword
```

Agenda

- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Configuring SSL Communication with the Load Balancer

During or after extending the domain with Oracle SOA Suite, configure the Administration Server and Managed Servers to access the front-end SSL URL of the hardware load balancer to allow:

- SOA composite applications and web services to invoke callbacks and other communications
- Oracle Service Bus to perform invocations with endpoints exposed through the load balancer SSL virtual servers
- Oracle Business Process Management to retrieve role information through specific web services

Configuring SSL for Oracle SOA Suite Applications

1. Generating Self-Signed Certificates by Using the utils.CertGen Utility
2. Creating an Identity Keystore by Using the utils.ImportPrivateKey Utility
3. Creating a Trust Keystore by Using the Keytool Utility
4. Importing the Load Balancer's Certificate into the Trust Store
5. Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts
6. Configuring Node Manager to Use the Custom Keystores
7. Configuring WebLogic Servers to Use the Custom Keystores
8. Testing Composites By Using SSL Endpoints

Generating Self-Signed Certificates with `utils.CertGen`

- Syntax

```
$ java utils.CertGen pswd cert_file key_file [export|domestic] hostname
```

- Example:

```
$ source $WL_HOME/server/bin/setWLSEnv.sh
$ mkdir $ASERVER_HOME/certs
$ cd $ASERVER_HOME/certs
# Generate certificates for physical and virtual host names
$ java utils.CertGen password adminvh.example.com_cert \ #virtual
 adminvh.example.com_key domestic adminvh.example.com
$ java utils.CertGen password host01.example.com_cert \ # physical
    host01.example.com_key domestic host01.example.com
$ java utils.CertGen password soavh01.example.com_cert \ # virtual
    soavh01.example.com_key domestic soavh01.example.com
```

Note: The digital certificates and private keys that are generated by the `utils.CertGen` tool are for demonstration or testing, not for production. Production certificates must be obtained from a recognized Trusted Certificate Authority (CA).

Creating an Identity Keystore by Using the `utils.ImportPrivateKey` Utility

Used for setting a Node Manager property

- Syntax:

```
$ java utils.ImportPrivateKey keystore_file ks_pswd certificate_alias
    pk_pswd cert_file key_file [keystore_type]
```

- Repeat the command for all the hosts in the system.
- The Identity Store is created (if none exists) when you import a certificate and the corresponding key into the Identity Store by using the `utils.ImportPrivateKey` utility.

```
$ java utils.ImportPrivateKey appIdentityKeyStore.jks password
    appIdentity1 password
    ASERVER_HOME/certs/SOAHOST1.example.com_cert.pem
    ASERVER_HOME/certs/SOAHOST1.example.com_key.pem
$ java utils.ImportPrivateKey appIdentityKeyStore.jks password
    appIdentity2 password
    ASERVER_HOME/certs/soavh01.example.com_cert.pem
    ASERVER_HOME/certs/soavh01.example.com_key.pem
$ java utils.ImportPrivateKey appIdentityKeyStore.jks password
    appIdentity3 password
    ASERVER_HOME/certs/ADMINVHN.example.com_cert.pem
    ASERVER_HOME/certs/ADMINVHN.example.com_key.pem
```

Creating a Trust Keystore By Using the Keytool Utility

To create the Trust Keystore on each host:

- Copy the standard Java keystore to create the new trust keystore because it contains most of the root CA certificates
- Change the default password (`changeit`) for the standard Java keystore by using the `keytool` utility

Note: The CA certificate `CertGenCA.der` is:

- Used to sign all certificates generated by the `utils.CertGen` tool
- Located in the `WLS_HOME/server/lib` directory
- Required to be imported into appTrustKeyStore by using the `keytool` utility

Importing the Load Balancer's Certificate into the Trust Store

For the SSL handshake to behave properly, the load balancer certificate must be added to the WebLogic Server trust store. To add the load balancer certificate:

- With a web browser, access the URL for the website that is exposed through the load balancer for SOA Infrastructure
- Export the certificate to a file in a shared location for the domain, by using the browser's certificate management tools
- Import the load balancer's certificate into the trust store by using `keytool`. For example:

```
$keytool –import –file soa.example.com –v –keystore appTrustKeyStore.jks
```

Adding the Trust Store to the Server Start Scripts

To add the trust store to the WebLogic Server start scripts:

- Edit the `setUserOverrides.sh` script, which executes when the Administration Server and Managed Servers start
- Ensure that each server can access the updated trust store by using the shared directories configured for the enterprise deployment:

```
$ edit ASERVER_HOME/bin/setUserOverrides.sh

# Replace references to the existing DemoTrustStore entry
# with the following
# NOTE: that all the values for EXTRA_JAVA_PROPERTIES must be on one
# line in the file, followed by the export command on a new line

EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
 -Dsoa.archives.dir=${SOA_ORACLE_HOME}/soa ...
 -Djavax.net.ssl.trustStore=/u02/oracle/certs/appTrustKeyStore.jks..."
export EXTRA_JAVA_PROPERTIES
```

Configuring Node Manager to Use Custom Keystores

To configure the Node Manager to use custom keystores, append the following properties:

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=Identity KeyStore
CustomIdentityKeyStorePassPhrase=Identity KeyStore Passwd
CustomIdentityAlias=Identity Key Store Alias
CustomIdentityPrivateKeyPassPhrase=Private Key to create Certificate
```

to the `nodemanager.properties` files in:

- `ASERVER/nodemanager`
- `MSERVER/nodemanager` for all nodes

For example:

Custom identity alias created when importing the certificate with the utils.ImportPrivateKey utility.

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=
 /u01/oracle/config/domains/edg_domain/certs/appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=password
CustomIdentityAlias=appIdentity1
CustomIdentityPrivateKeyPassPhrase=password
```

Configuring WebLogic Server to Use Custom Keystores

To configure WebLogic Server to use custom keystores requires several steps to be performed by using the Oracle WebLogic Server Administration Console for:

- The Administration Server
- The Managed Servers (and other servers) that require SSL access to the front-end load balancer URLs

Deploying Applications to an Enterprise Deployment

Oracle SOA Suite applications are deployed as composites that consist of one or more components. Composite applications should be deployed to:

- A specific server (called a pinned application)
- The cluster through an internal host name, such as soainternal.example.com

Testing Composites By Using SSL Endpoints

With SSL enabled, composite endpoints can be verified on SSL from Oracle Enterprise Manager Fusion Middleware Control by using the following steps:

1. Log in to Fusion Control via the URL http://adminvh.example.com:7001/em.
2. Expand SOA and click soa-infra(soa_server1), for example.
3. Expand the partition to which the composite is deployed and select the composite.
4. On the composite page, click the Test tab.
5. In the WSDL or WADL address field, replace http://soa.example.com:[80]80 with https://soa.example.com:[4]443. Click Parse WSDL or WADL.
6. Verify that the Endpoint URL shown is SSL and click Test.
7. Check that the response is as expected for the web service.

Agenda

- Reviewing the high availability architecture
- Scaling out an enterprise deployment topology
- Configuring high availability for the Administration Server
- Configuring a JCA Adapter and resources for applications
- Configuring SSL
- Configuring Whole Server Migration: Overview

Configuring Whole Server Migration: Overview

The key tasks required to configure Whole Server Migration are:

1. Setting up a user and tablespace for the server migration leasing table
2. Creating a GridLink Data Source for leasing by using the administration console
3. Editing the Node Manager's properties file to enable Whole Server Migration
4. Setting environment and super user privileges for the `wlsifconfig.sh` script
5. Configuring server migration targets
6. Testing Whole Server Migration

Components for Whole Server Migration

Component	Whole Server Migration	Automatic Server Migration
Oracle Web Services Manager (OWSM)	NO	NO
Oracle SOA Suite	YES	NO
Oracle Business Process Management	YES	NO
Enterprise Scheduler Services	NO	NO
Oracle Business Activity Monitoring	NO	YES

Note: The course topology and architecture are not conducive for Whole Server Migration because OWSM, SOA Suite, and ESS are co-located in the same Managed Server.

Summary

In this lesson, you should have learned how to:

- Describe the high availability architecture
- Scale out an Enterprise Deployment topology
- Configure high availability for the Administration Server
- Configure a JCA Adapter and resources for applications

Practice 6: Overview

This practice covers the following tasks:

- 6-1: Testing Manual Failover of the Administration Server
- 6-2: Configuring the File Adapter for High Availability
- 6-3: Configuring the DBAdapter and Resources
- 6-4: Configuring the JMS Adapter and Resources