



Deploying SOA Composites
and Oracle Service Bus
Applications from Oracle
JDeveloper to Oracle SOA
Cloud Service

Deploying SOA Composites and Oracle Service Bus Applications from Oracle JDeveloper to Oracle SOA Cloud Service

This white paper documents the steps to deploy SOA/Oracle Service Bus composites/applications from JDeveloper (that runs outside the Oracle network) to the provisioned SOA Cloud Service instance.

This deployment scenario has been successfully tested against both a single-node cluster environment and a two-node cluster environment.

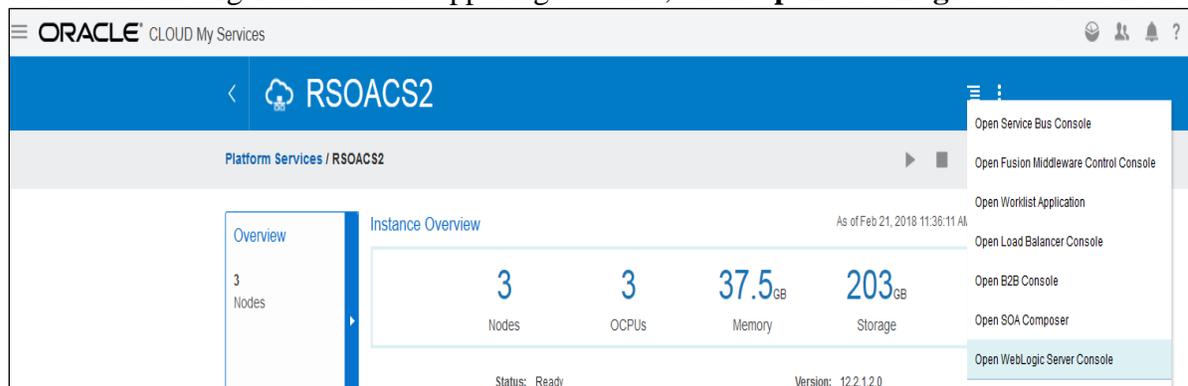
Note: These instructions work from outside the Oracle network, but not from within an Oracle network.

[Deploy to SOA Cloud Service from JDeveloper- Post-Provisioning Steps](#)

After provisioning the SOA Cloud Service instance, there are several post-provisioning steps that are required for deployment to be successful.

The post-provisioning steps are as follows:

- 1) Find the public IP address of the administration server. For this, go to the instance details page by clicking the SOA Cloud Service instance that was provisioned earlier.
- 2) From the hamburger menu in the upper right corner, click **Open WebLogic Server Console**.



- 3) After logging into the Oracle WebLogic Server Console, on the Summary of Servers page, check the host name associated with the administration server. For this example, the administration server is running on host **rsoacs2-wls-1** (marked in red).

Summary of Servers

Configuration Control

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.
This page summarizes each server that has been configured in the current WebLogic Server domain.

Customize this table

Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

New Clone Delete

Name	Type	Cluster	Machine	State	Health	Listen Port	Listen Address	SSL Enabled	SSL Listen Port	SSL Listen Port Enabled
RSOACS2_adminserver(admin)	Configured		RSOACS2_machine_1	RUNNING	OK	9071	rsoacs2-wls-1	true	9072	true

- 4) From the SOA Cloud Service instance details page, you get the public IP address associated with host **rsoacs2-wls-1**. This is the public IP address of the administration server.

Weblogic

State: Ready

Resources

Host Name:	rsoacs2-wls-1	CPUs:	1
Public IP:	192.0.2.254	Memory:	15 GB
Instance:	Runs RSOACS2__server_1	Storage:	77 GB

- 5) Configure the managed server's reachability.
Verify each of the managed server's listener addresses from Oracle WebLogic Service Console.

Note: If the listener address is a host name or a private IP address, JDeveloper cannot access the managed servers during deployment. In this case you must perform the following step.

- a. Map the host name to the same administration server public IP address in the `hosts` file as shown below. For Windows, the `hosts` file is typically located at `C:\Windows\System32\Drivers\etc\hosts`.

Note: Perform this task on the host on which JDeveloper is running.

```
192.0.2.254 rsoacs2-wls-1
```

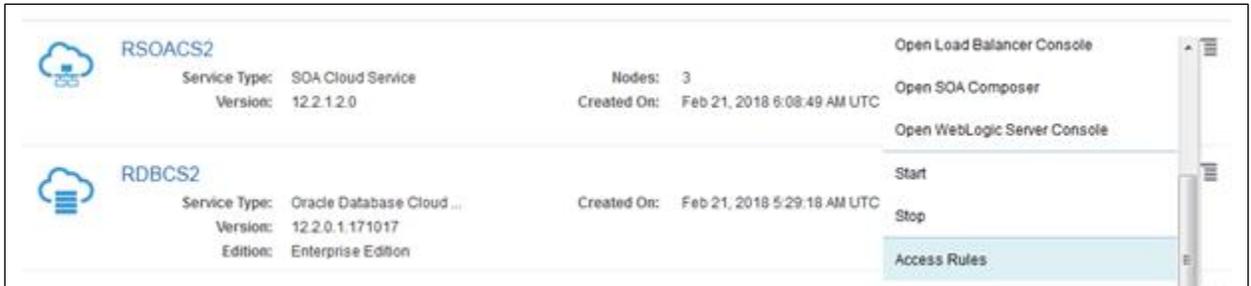
```
192.0.2.254 rsoacs2-wls-2
```

An alternative is to modify the managed server's listener address. However, this is not recommended because of unintended consequences such as disaster recovery breakage.

- 6) Add the access rule in the SOA Cloud Service Console to permit traffic from JDeveloper (that exits outside of the Oracle network) to the SSL listener port of the managed server.

- a. Click the **Manage this service** menu and choose the **Access Rules** option to launch the Access Rules page.

NOTE: Opening the managed server port is not as secure as the default, where the port is not exposed. If you do not want to open this port, this solution does not work for you.

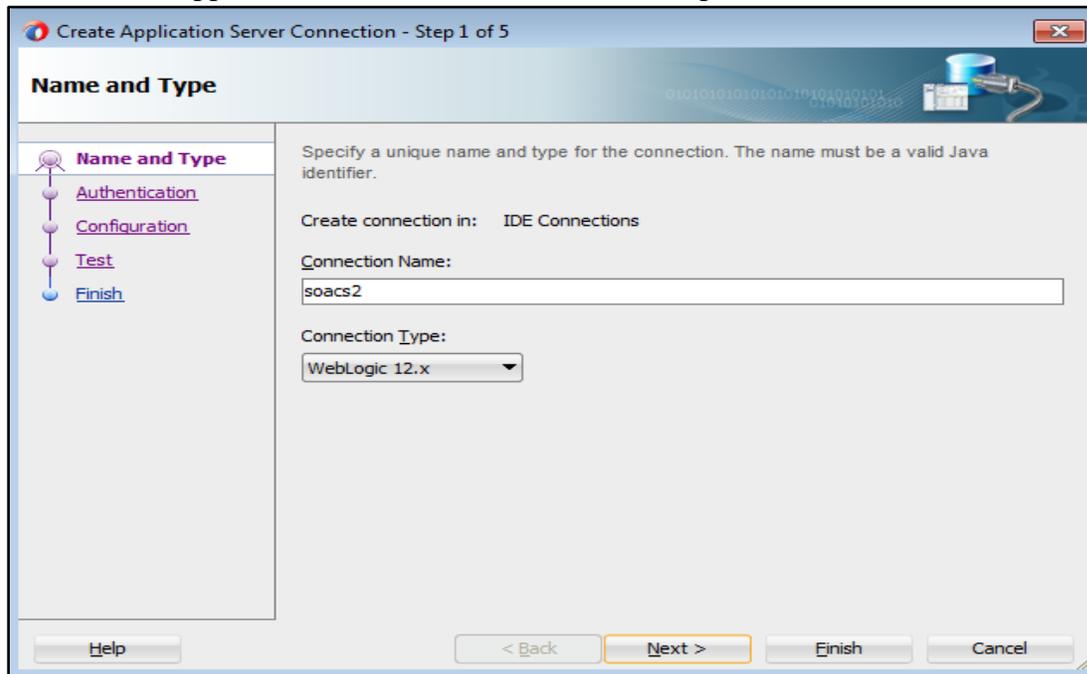


- b. Enter the details as shown in the screen shot. Add the SSL listener port of the managed server in the **Destination Port(s)** field.

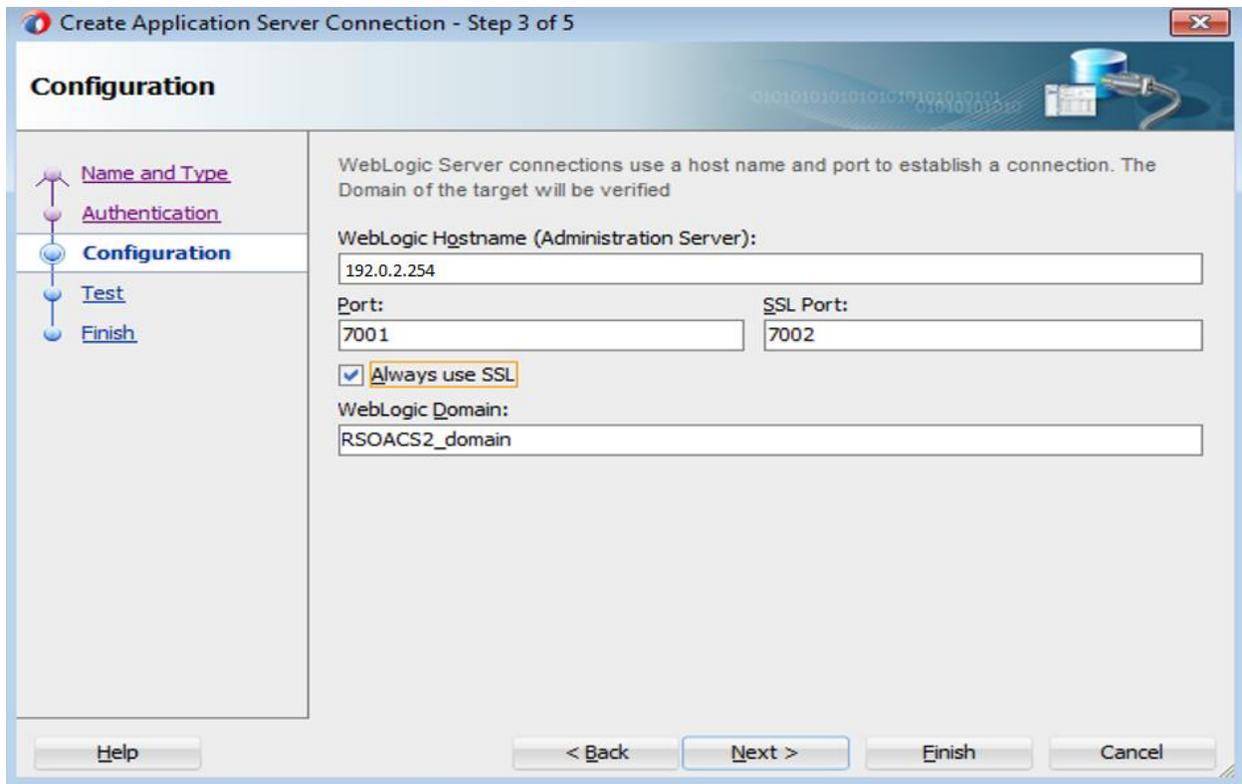
The 'Create Access Rule' dialog box is shown with the following fields and values: 'Rule Name' is 'permitJdevTraffic', 'Source' is 'PUBLIC-INTERNET', 'Destination' is 'WLS_MS', 'Destination Port(s)' is '9074', and 'Protocol' is 'TCP'. There is a 'Description' field which is currently empty. At the bottom left, a green warning icon is followed by the text 'This operation may take some time.' At the bottom right, there are 'Create' and 'Cancel' buttons. A small 'Enter' tooltip is visible near the Destination field.

Create an Application Server Connection in JDeveloper

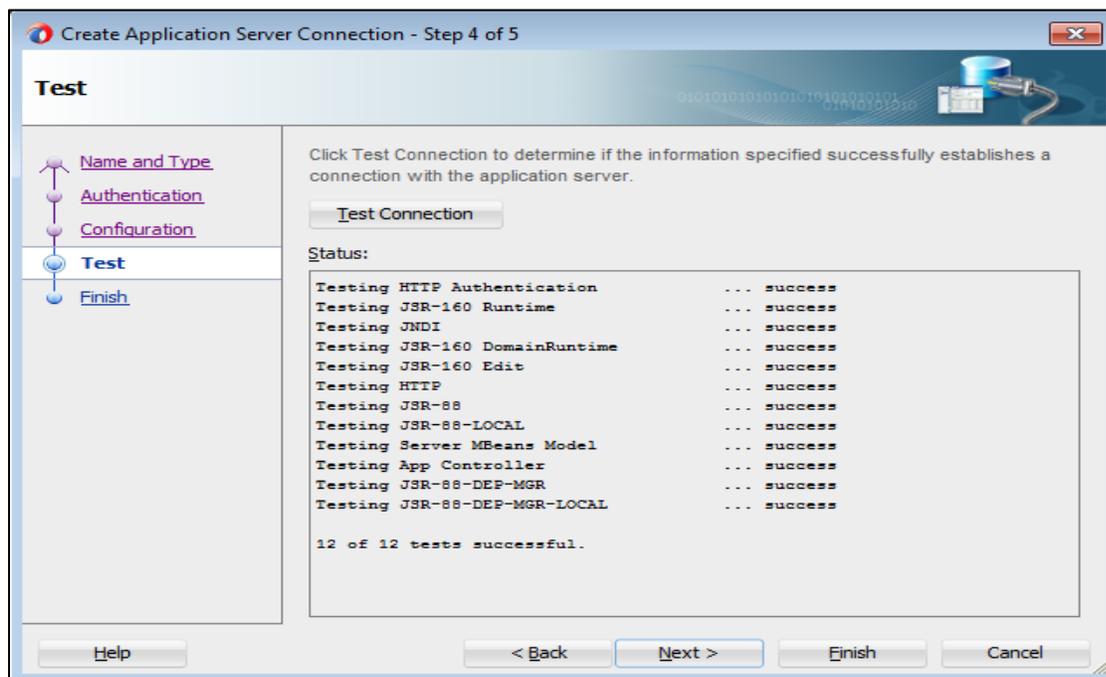
- 1) Create a new application server connection in JDeveloper.



- 2) In the Configuration page, provide the administration server public IP address that was obtained earlier in the **WebLogic Hostname** field.
- 3) Select **Always use SSL** and provide port 7002 as the SSL port.

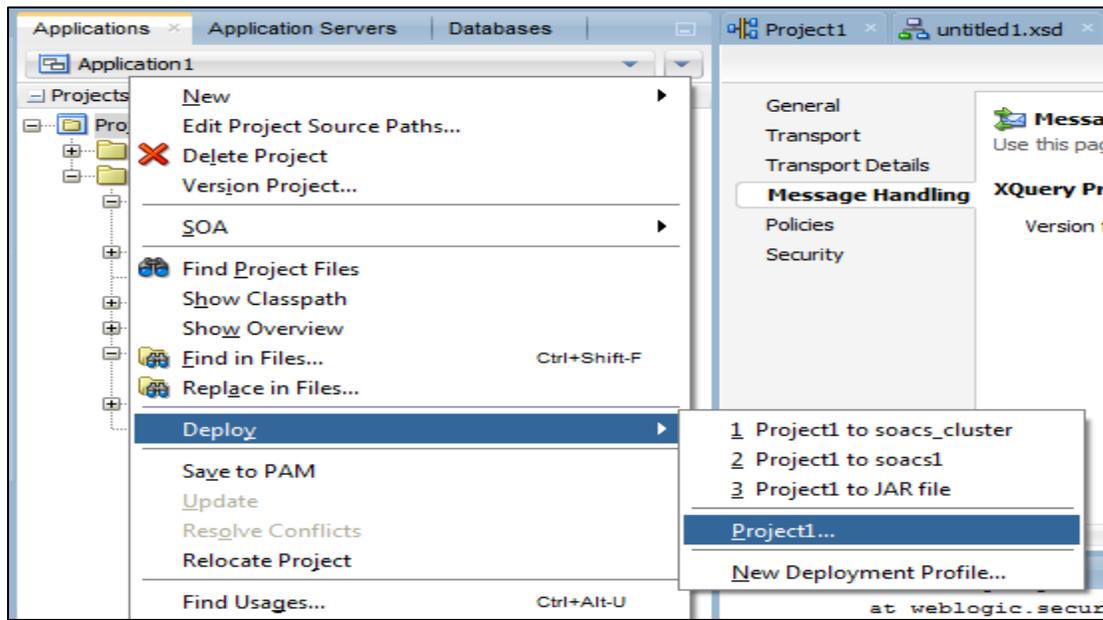


- 4) Test the connection after accepting the certificates. The dialog to accept the certificate is only displayed the first time you connect to this server. The test connection should now pass without any errors.



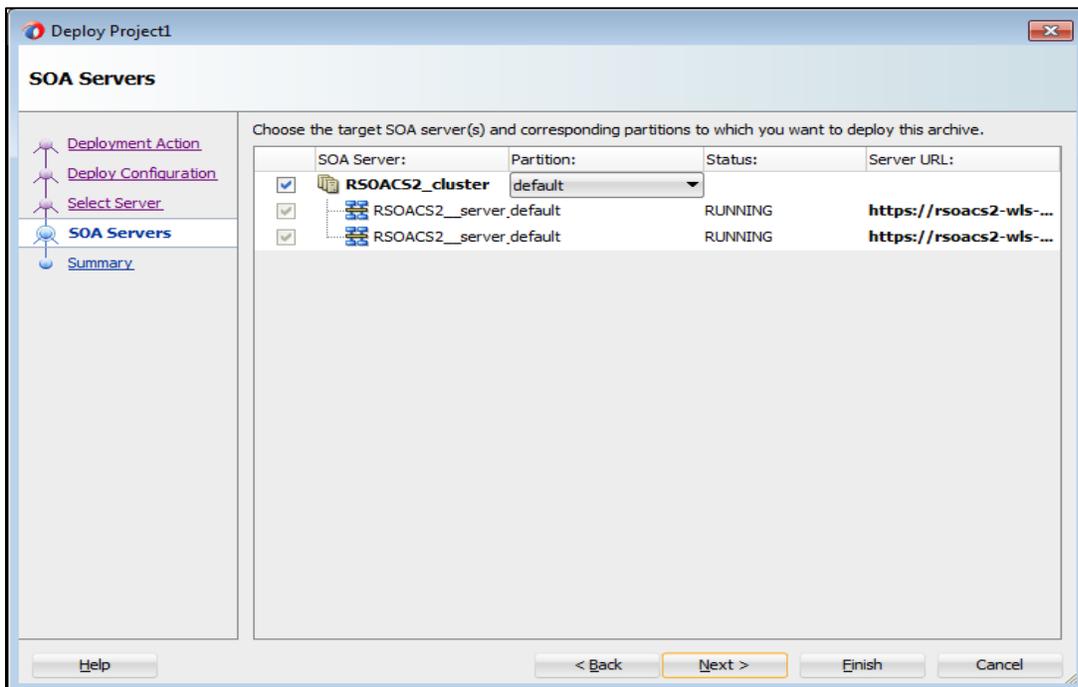
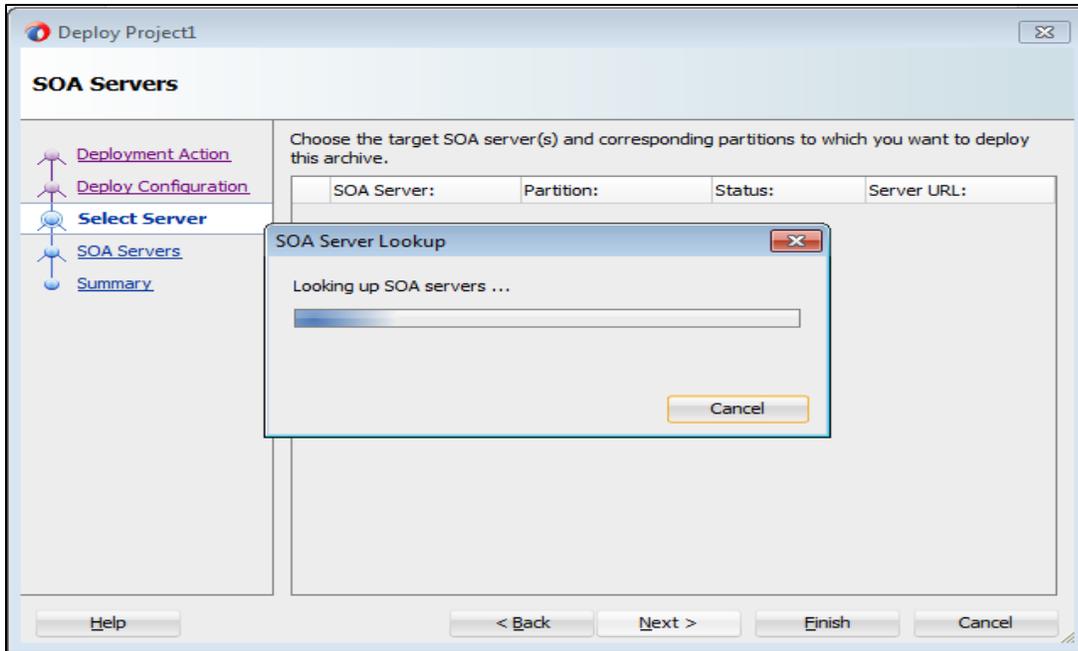
[Deployment of SOA Composites to SOA Cloud Service from JDeveloper.](#)

- 1) In JDeveloper, right-click the SOA project you want to deploy and select the **Deploy** menu. This invokes the deployment wizard.

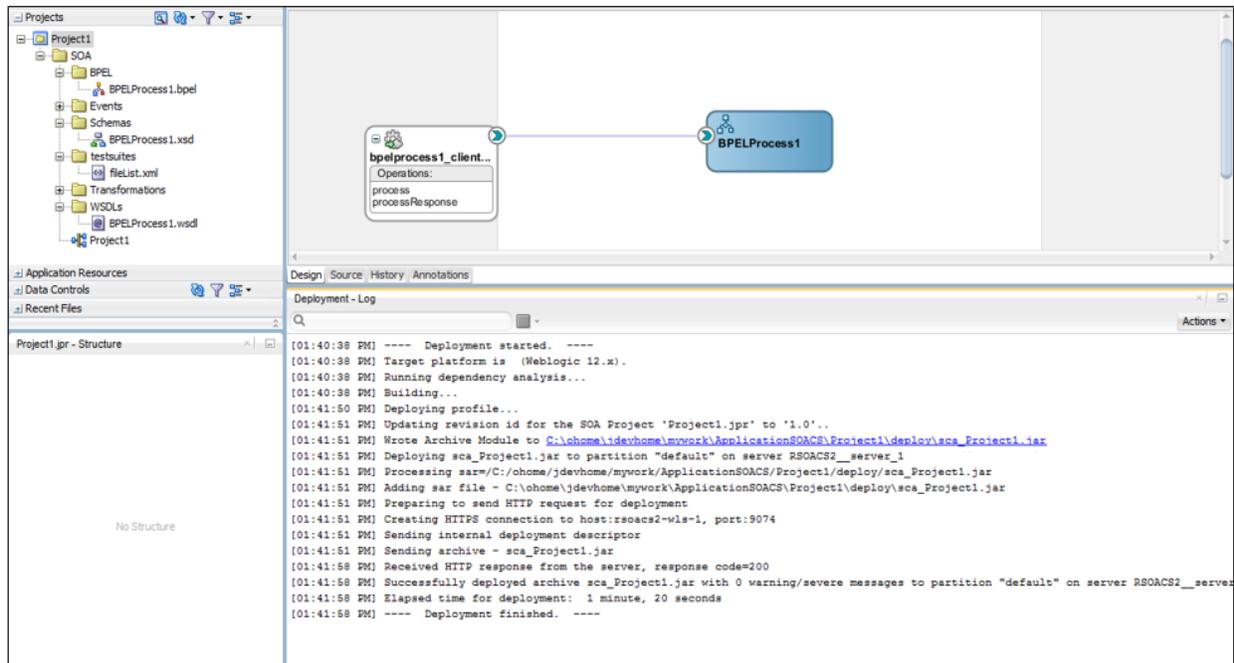


- 2) In the deployment wizard, select the application server connection that was created earlier.

If the server has been configured correctly, the next step looks up the SOA servers and shows the various managed servers to which to deploy the composite.



- 3) Complete the deployment as shown below. The JDeveloper console logs indicate that the composite was deployed successfully.



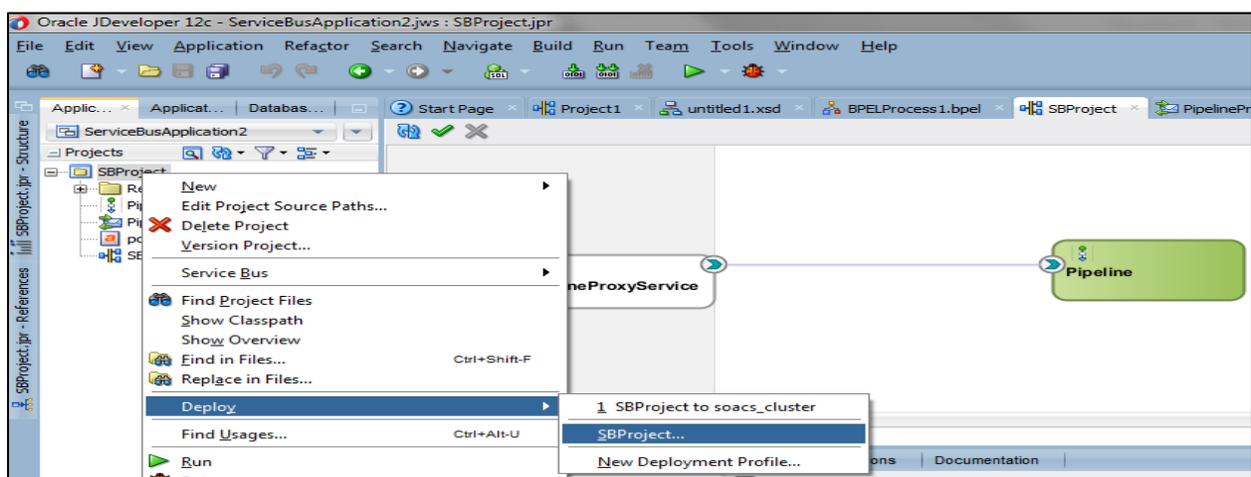
[Deployment of Oracle Service Bus Applications to SOA Cloud Service from JDeveloper](#)

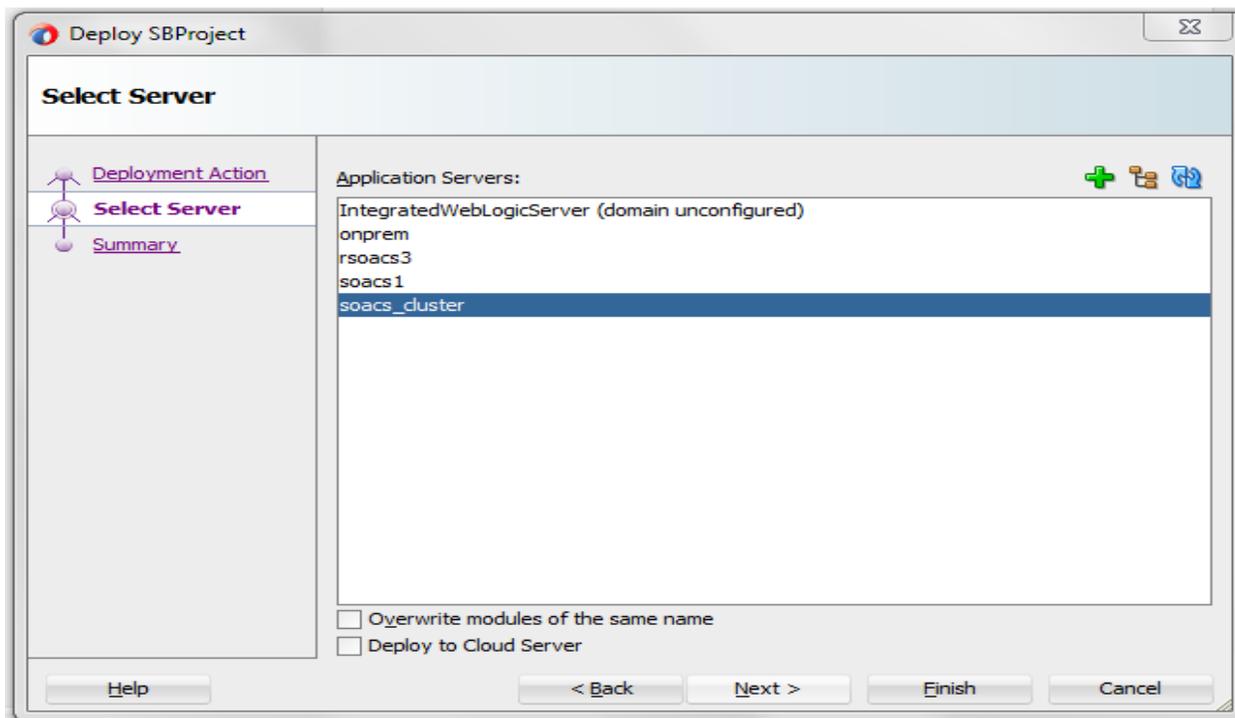
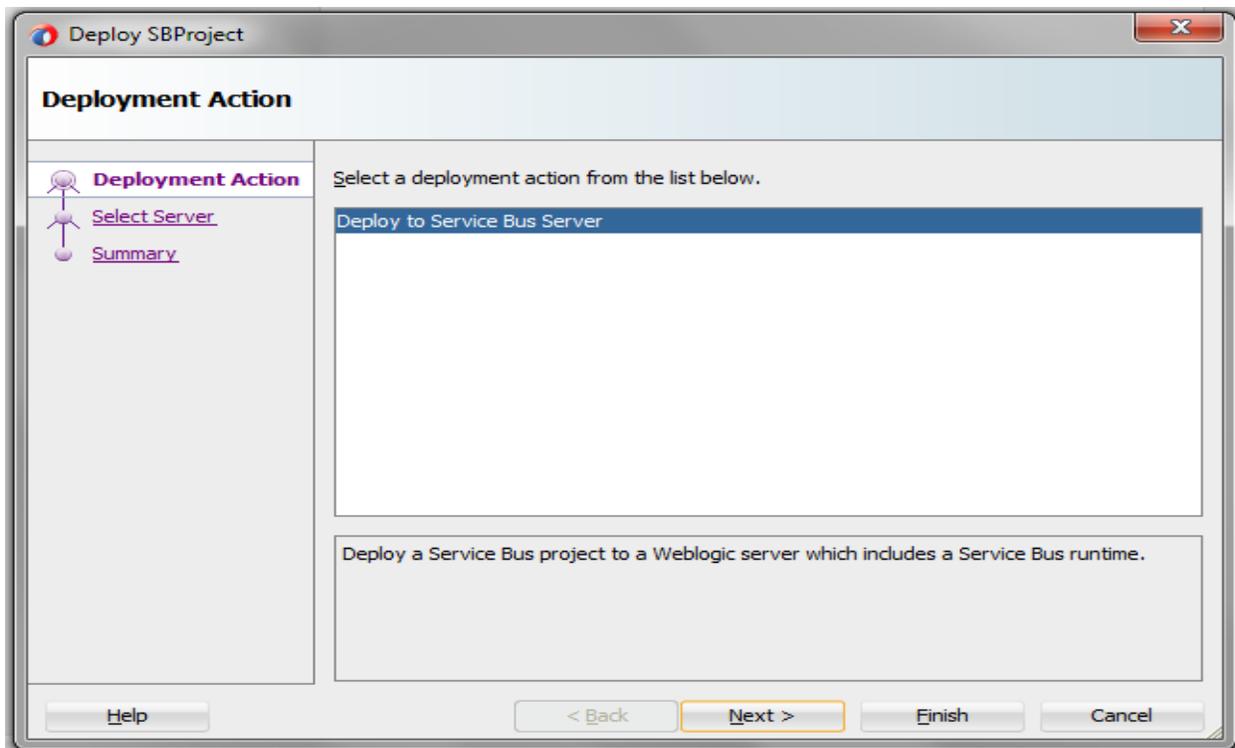
Follow the same steps for deployment of Oracle Service Bus applications.

If the SOA Cloud Service instance has been correctly configured as previously described, the deployment is also successful for Oracle Service Bus applications.

Because Oracle Service Bus applications are deployed to the administration server (unlike SOA, where composites are deployed to the managed servers), you do not see the SOA server lookup step while deploying the Oracle Service Bus project.

The following image shows the steps involved in deploying an Oracle Service Bus application.





The screenshot displays the JDeveloper IDE interface. On the left, the Project Explorer shows a project named 'SBProject' with a folder 'Resources' containing 'Pipeline.pipeline', 'PipelineProxyService.proxy', and 'pom.xml'. The main workspace shows a diagram with two components: 'PipelineProxyService' and 'Pipeline', connected by a line. Below the workspace, the 'Design' tab is active, and the 'Deployment - Log' window shows the following output:

```
[12:00:59 PM] ---- Deployment started. ----
[12:00:59 PM] Target platform is Standard Java EE.
[12:01:18 PM] Deployed: SBProject/Pipeline (type: Pipeline)
[12:01:18 PM] Deployed: SBProject/PipelineProxyService (type: ProxyService)
[12:01:18 PM] Deployed: SBProject/Resources/RestService (type: WADL)
[12:01:18 PM] Elapsed time for deployment: 19 seconds
[12:01:18 PM] ---- Deployment finished. ----
```

[Other Important Points](#)

1. Deployment to SOA Cloud Service always occurs on the SSL port, and not on the non-SSL port.
2. As you can see in the Create Application Server Connection step, JDeveloper uses SSL port 7002 to connect and deploy to the administration server in the SOA Cloud Service environment. However, the mapping of port 7002 to 9072 (SSL port on the administration server) happens internally.
3. If deployment failures occur even after following the above steps, retry deployment after clearing the JDeveloper cache. Remember that clearing the JDeveloper cache removes all previously-created connections.
4. The deployment has been tested with the following versions:
SOA Cloud Service – provisioned with SOA version 12.2.1.2.0
JDeveloper – version 12.2.1.2.0