



**Hewlett Packard**  
Enterprise

# Codar

Software version: 1.70

## Continuous Integration, Deployment and Testing by Codar using ALM and Jenkins

Document release date: July 2016

Software release date: July 2016

# Contents

<b>What is Codar?</b>	<b>3</b>
<b>Why is Codar required?</b>	<b>3</b>
<b>HPE Application Lifecycle Management</b>	<b>3</b>
<b>End-to-end flow</b>	<b>3</b>
<b>Continuous integration and deployment using Codar and Jenkins (build tool)</b>	<b>4</b>
Software required for the integration	4
Configuring Jenkins	5
Uploading the Codar plugin on the Jenkins server	5
Installing the Codar plugin	5
Enabling the Codar plugin	5
Configuring the Pet Clinic sample application project	6
Configuring the Codar plugin for the Pet Clinic sample application	7
Adding another project	10
<b>Continuous testing of the application using Codar and ALM through Jenkins</b>	<b>11</b>
Install and configure the Jenkins – ALM plugin	11
Enable the Jenkins plugin for ALM	11
Enabling the Artifact Deployer Plugin	11
Configure the Jenkins plugin for ALM	11
Create the test environment in ALM	15
Site Administration	16
ALM Lab Service	17
ALM Lab Management	18
ALM Desktop Client	19
<b>End-to-end flow</b>	<b>22</b>
<b>Send documentation feedback</b>	<b>24</b>
<b>Legal notices</b>	<b>24</b>

# What is Codar?

Codar is a continuous delivery solution that provides deployment and release management of complex multi-tier applications across the application lifecycle. It automates the deployment of applications by embracing existing content from Chef, HPE SA, and so on and representing this content as components. These components can be used in a graphical topology designer to create an application model.

One of the important features of Codar is that the model is used to trigger deployments automatically with Jenkins, trigger test cases that are on the deployed instances in ALM (Application Lifecycle Management), and update the results in ALM.

This document provides information about integrating Codar with ALM.

## Why is Codar required?

Software engineering builds are subject to continuous deployment and testing on the principles of frequent code commits, build automation, faster and frequent builds, automated application deployment, and test automation. On top of continuous integration, software development teams also continuously deliver qualified software applications to their test and production teams. One of the challenges that most software development teams face in the process of continuous integration and continuous delivery is the ability to automate the deployment of applications in a simple and consistent manner and run tests on the deployed instance. Codar is built to solve this problem.

Using Codar, users can deploy the application and run tests automatically by integrating with ALM. This white paper describes how Codar can be integrated with ALM. Codar is integrated with ALM through Jenkins. Jenkins acts as orchestrator between Codar and ALM.

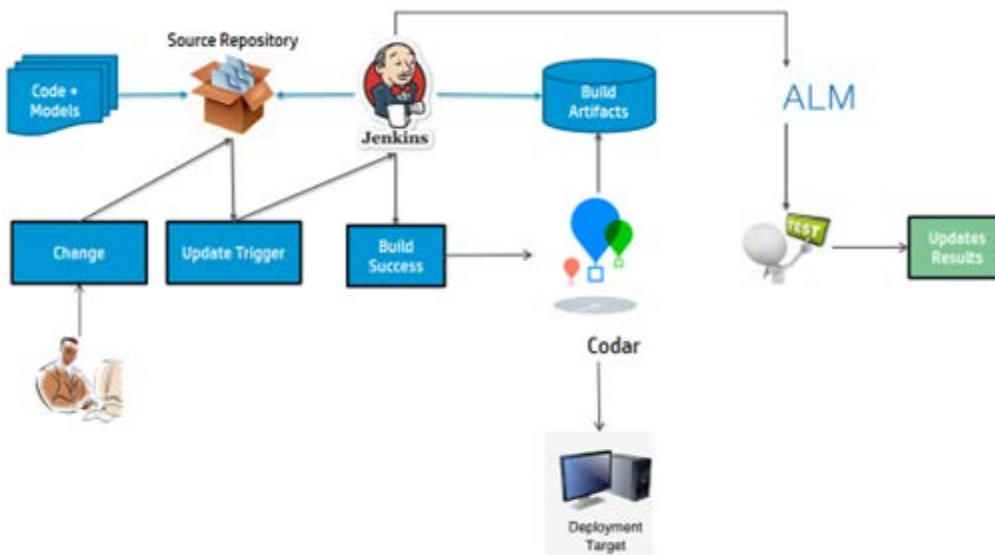
## HPE Application Lifecycle Management

ALM is a set of software products designed to accelerate the delivery of secure, reliable, and modern applications. It is a combination of a common platform, several key applications, and a dashboard targeted at managing the core lifecycle of applications.

## End-to-end flow

Figure 1 shows the end-to-end flow of how Codar and ALM are integrated through Jenkins. In this use case, Jenkins is the orchestrator.

*Figure 1: End to end workflow of the Codar-ALM integration*



© Copyright 2013 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice.



The developer makes a code change and checks it into source control. The source control system sends an update trigger and Jenkins triggers a build. After the build is successful, Jenkins invokes the Codar plugin which in turn invokes the Codar API and triggers a deployment.

After the deployment is complete, Jenkins creates a JSON file that contains the details of the deployed instance. It then invokes the ALM plugin. The ALM plugin logs on to ALM with the details provided during configuration. It invokes the test case and updates the result in ALM.

## Continuous integration and deployment using Codar and Jenkins (build tool)

### Software required for the integration

Jenkins must be integrated with Codar for the ALM integration to work. To integrate Jenkins with Codar, you must install the following software:

1. Install Jenkins from [jenkins-ci.org/](http://jenkins-ci.org/)
2. Install the JDK version 1.7x on the Jenkins server.
3. Install Collabnet Subversion Edge from [collab.net/support/documentation](http://collab.net/support/documentation)
4. Install TortoiseSVN from [tortoisesvn.net](http://tortoisesvn.net). Install the latest version and use the default settings. After the installation, you can see new options when you right-click a file or folder in Windows Explorer.
5. Download and install Maven from [maven.apache.org](http://maven.apache.org)

# Configuring Jenkins

After installing the software listed at Software required for the integration, configure Jenkins as follows (the following steps are for Jenkins version 1.583):

1. Ensure that JDK and Maven are installed.
2. Open Jenkins and click the **Manage Jenkins** option in the Jenkins dashboard.
3. Click **Configure System**.
4. In the **JDK** section, click **JDK installations** and then **Add JDK**.
5. Enter the name and path of the JAVA\_HOME environment variable.
6. Deselect the **Install automatically** check box.
7. In the Maven section, click **Maven installations** and then **Add Maven**.
8. Enter the name and path of the MAVEN\_HOME environment variable.
9. Deselect the **Install automatically** check box.
10. Enter the value of the MAVEN\_OPTS environment variable.
11. Click **Save**.

## Uploading the Codar plugin on the Jenkins server

You must now upload and enable the Codar plugin on the server in which Jenkins is installed.

## Installing the Codar plugin

1. Log on to the Jenkins dashboard using the <http://<host>:<port>/> URL. Use the host and port information appropriate for your Jenkins environment.
2. Click **Manage Jenkins** on the Jenkins dashboard.
3. Click **Manage Plugins**.
4. Select the **Advanced** tab.
5. In the **Upload Plugin** section, browse to the path of the Codar plugin file at HPE\Codar\CSAKit-4.2\Content Archives\topology\Jenkins plugin\HP\_Helion\_Codar.hpi
6. Click **Upload**.

## Enabling the Codar plugin

1. Click **Manage Jenkins** on the Jenkins dashboard.
2. Click **Configure System**.
3. Scroll down to the **HP Helion Codar Plugin/ HP Codar Plugin** section and select the **Enable** check box.
4. Click **Save**.

**Note:** For security reasons, you must configure the Codar plugin with SSL and HTTPS enabled and with TLS Protocol Version 1.2 only.

*Figure 2: Enabling the Codar plugin*

Helion Codar Plugin

Enable☒

Check to enable Helion Codar plugin

SaveApply

# Configuring the Pet Clinic sample application project

This section describes how to configure the Codar-ALM integration by means of a sample application called Pet Clinic.

1. Download the Pet Clinic source code from [GitHub](#).
2. Check in the source code for the Pet Clinic project into the SVN server.
3. Create a new Pet Clinic project in the Jenkins dashboard:
  - a. Click **New Item** -> **Maven project**.
  - b. Enter Pet Clinic in the **Item name** text box.
  - c. Click **OK**.The Pet Clinic link is displayed in the Jenkins dashboard.
4. Click the Pet Clinic link on the Jenkins dashboard, and then click the **Configure** link in the page that opens.
5. Configure SVN for the Pet Clinic project by choosing the **Subversion Modules** option in the **Source Code Management** section area and adding the SVN Pet Clinic source code URL in the **Repository URL** field.
6. After saving, update the SVN credentials as shown in Figure 3.

Figure 3: Source code management

Source Code Management

☐ CVS☐ CVS Projectset☐ None☒ Subversion

Modules

Repository URLhttp://vm101.underworld.com:8090/svn/PetClinic/trunk/PetClinic

Local module directory (optional).

Repository depthinfinity

Ignore externals

Check-out Strategy

Use 'svn update' as much as possible

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to re

Repository browser

(Auto)

7. Configure Jenkins to automatically trigger a build if some code is checked in by selecting the **Poll SCM** check box and adding **\*/\*5 \* \* \* \*** in the **Schedule** field. This indicates the poll every five minutes if a code commit happens.

Figure 4: Triggering a build

**Build Triggers**

- ☒ Build whenever a SNAPSHOT dependency is built
- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ Poll SCM

Schedule

`*/* * * * *`

⚠ Spread load evenly by using `H/* * * * *` rather than `*/* * * * *`

Ignore post-commit hooks ☐

8. Scroll down to the **Post-build Actions** section and click **Add post-build action**, select **Archive the artifacts**, and then enter `*/*.war, target/classes/*/*.sh` in the **Files to archive** text box.

**Post-build Actions**

**Archive the artifacts**

Files to archive `*/*.war, target/classes/*/*.sh`

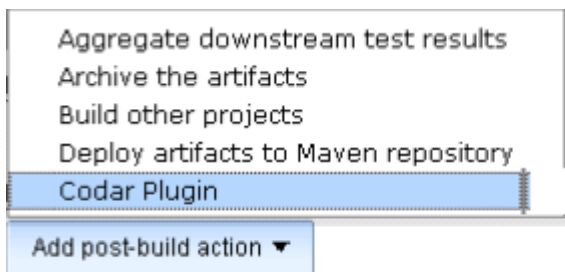
Advanced...

Delete

## Configuring the Codar plugin for the Pet Clinic sample application

1. Click the Pet Clinic link on the Jenkins dashboard, and then click the **Configure** link on the page that opens.
2. Click **Add post-build action** and select **Codar Plugin**.

Figure 5: Configuring the Codar plugin



3. Enter the Codar plugin properties as follows:

Figure 6: Codar properties

**HP Codar Plugin**

CodarUrl\*  ?

The url for access the Codar

Username\*  ?

The username of Codar

Password\*  ?

The password of Codar

SSLCertificatePath\*  ?

Password credential used to access the REST APIs

CertificatePassword\*  ?

Certificate keystore password

☒ Continuous Promote ?

☒ Enable Http Authentication

Application Design Location  ?

The relative location of the design

Environment\*  ?

The selected environment to provision from

Package Name

Package Name

Package Description  ?

Package Description

Application Design Type\*  ?

Select a Design Type

List Of Designs\*  ?

List of Designs of specified type.

Versions\*  ?

List of Versions of specified Design.

Package Properties\*

Component Id/Name	Property Name	Property Value	
Server	Domain Name	abcd	Delete

Add Package Property

Specify the build specific properties of the components in the design that will be dynamically generated based on the Jenkins build

Extended Properties File  ?

Save Apply

- CodarUrl – URL used to log on to Codar.
- Username – Name of the user that has administrative privileges in Codar.  
Do not use the default administrator user because it may cause a security issue. After installing Codar and configuring LDAP, a user is added to the Application Architect role. Use the credentials for that user here.
- Password – Password of the Codar user.
- SSLCertificatePath – Enter SSL certificate path for Codar and pick up the certificate from the Codar setup. If open JRE is used during the Codar installation, then it is in the computer in which Codar is installed (in the Program Files\Hewlett-Packard\Codar\openjre\lib\cacerts path). Details about the JRE used during installation is located in the **csa.properties** file.
- CertificatePassword – Enter the SSL certificate keystore password for Codar.
- Continuous Promote – If this option is selected, packages are created in the first stage and release gate actions are executed. If all actions are successful, only then is the package promoted to the next stage and so on till it reaches the last stage. In the last stage, the package is executed after all actions are executed. If this option is not selected, then the package is created and deployed. Release gate actions are not executed and not promoted.
- Enable Http Authentication – Select to enable the Jenkins user name and password. This is required for Codar to pull deployment artifacts from Jenkins.
- HttpUsername - User name for accessing artifacts from the HTTP location. For example, if the artifacts are at a location in Jenkins, enter the user name of the Jenkins server.
- HttpPassword – Password for accessing artifacts from the HTTP location. For example, if the artifacts are at a location in Jenkins, enter the password of the Jenkins server.
- Application Design Location – this can be configured in 3 ways:



- Relative path and file name of the application design JSON file from the source repository URL. The relative path must be separated by a slash. This design will be deployed by Codar. For example, designs\PetClinicApp.json. For example, if the source repository is <https://myrepo.mydomain.com/mypetclinicapp>, the application JSON can exist in a directory named designs and the JSON file can be created with the any name as required. The JSON file is a part of the repository in which the application source code is located.

- URL of design location can be specified. For example, you can specify the location of design file **http://<hostname>/<designfile.json>**

- You can create an environment variable and pass it as a variable.

k. Application Design Type – From the list, select either Topology Designs or Sequenced Designs. If you select 'Topology Designs', then the List of Designs drop-down lists all topology designs for selection. If you select 'Sequenced Design', then the List of Designs drop-down lists all sequenced designs.

l. List of Designs – Select the design from the list. The list shows topology designs or sequenced designs based on the application design type you selected.

m. Versions – Select the version number of the design. The list shows all versions of the design selected above.

Meta Data

Preset Values

Values

Add

☐ Needs percentage based access to slave's resources

☐ Can run only single instance per slave

☒ This build is parameterized

String Parameter

Name: designfilelocation

Default Value: https://codarhost/content/petclinic.json

Description:

[Escaped HTML] [Preview](#)

Delete

Add Parameter

n. Environment – Environment in which the provider to be used for deployment is contained. Environments are created in the **Resource provider** tile under Codar. For details about environments, see the Codar Administration Guide.

o. Package Name – Name of the package

p. Package Description - Description of the package. The Jenkins URL is appended to the package description.

q. Package Properties – You can specify the value of the property in the following ways:

- Artifact name like petclinic.war that resolves to `http://<jenkins_host>/job/Petclinic/<buildno>/artifact/target/petclinic.war`
- Property value can be parameterized in one of the following ways:
- Property value – The property values allows user to specify parameterized values which allows users to dynamically reference the current build artifacts that built by the Jenkins Job
  - Within the property values users will be able to specify the Jenkins Parameters in the following notation `${Jenkins_param}` where Jenkins\_param is the parameter created in Jenkins. This allows the possibility of injecting the Jenkins environmental variables as part of the package property. For e.g Users can create a Jenkins String Parameter called BUILD\_NUM and provide the value of this property to the environmental variable provided by jenkins \$BUILD\_NUMBER .Users can then specify this Jenkins Parameter `${BUILD_NUM}` in the property value and this value will be assigned to the package property value. When a package is created in Codar, it will substitute the current Build number in Jenkins as the value of the property.
  - Users can also directly specify the Jenkins artifact name as depicted below. The Petclinic.war will then automatically be resolved to the http url of the Jenkins artifact, and substituted in the package property value.

For example, the package property can be:

PetCline App	artifacturl	petclinic.war
--------------	-------------	---------------

The ID can also be used instead of the name:

PetClinic Application VERSION 1 GROUPID com.hp.csa.type0001

artifacturl

petclinic.war

Use only the package ID instead of the package name.

r. Extended Properties File –Enter the name of the properties file. This is an optional field. Specify this file only if you want to specify a different CI process than what is provided by default. The properties file can specify a different OO flow containing the necessary CI logic. You can also specify a different flow D by creating a properties file with the UUID as the key and UUID of the OO flow as the value. For example, uuid=asdaasdasdsdasdad99f. You can also specify the required properties to this flow as key value pairs in the properties file.

4. Node ID – Enter the component ID for which the component properties have to be extracted. The component name can be obtained from the topology design or from the JSON file. Multiple components names or IDs can be provided by clicking the **Add** button. For example, to retrieve the IP address and host name of the vCenter, the component name can be vCenter Server and the component ID can be: vCenterServerType\_\_VERSION\_\_04.20.0000\_\_GROUPID\_\_com.hp.csa.type0002

The following is an example of the extended properties file of the Pet Clinic application:

*## Properties accessed by the ARA API to invoke OO flows.*

*##This properties file contains the oo flow id (uuid) as well as the relevant parameters to be passed to the oo flow.*

*#uuid of the oo flow. This flow contains the necessary logic for the Continuous integration process.*

*#uuid=377898bc-d92e-4e6a-b542-718539fdbcb9a*

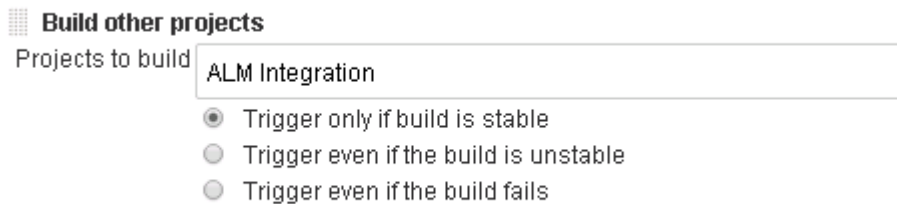
*#user can add more parameters in key value format*

## Adding another project

Add a step to build a downstream project (ALM Integration) as shown in Figure 7:

1. Open the Pet Clinic job in Jenkins and click **Configure**.
2. Scroll down to the **Post-build Actions** section click the **Add post-build actions** drop-down list.
3. Select **Build other projects**.
4. Enter the project name (in this case, ALM Integration) in the **Projects to build** field.  
This project triggers the ALM Integration project after the Pet Clinic project is complete.

Figure 7: Building another project



**Build other projects**

Projects to build

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

# Continuous testing of the application using Codar and ALM through Jenkins

## Install and configure the Jenkins – ALM plugin

Install the Jenkins plugin for ALM from the following location:

<https://wiki.jenkins-ci.org/display/JENKINS/HP+Application+Automation+Tools>

## Enable the Jenkins plugin for ALM

Go to **Jenkins** -> **Manage Jenkins** -> **Manage Plugins** and verify that the following entry appears:

Figure 8: Verifying the Jenkins – ALM plugin

<a href="#">HP Application Automation Tools</a>		3.0.7	3.0.7-SNAPSHOT (private-01/14/2015 17:52-MYD- VM04700\$)
<small>This plugin allows Jenkins to trigger HP tests such as: Test sets on Application Lifecycle Management, tests saved on the file system from Unified Functional Testing, QuickTest Professional, or Service Test, and LoadRunner scenarios.</small>			

Go to **Jenkins** -> **Manage Jenkins** -> **Configure System** and provide the details of the ALM server.

Figure 9: Providing details of the ALM server

**Application Lifecycle Management**

ALM server

ALM server name	<input type="text" value="My ALM"/>	?
ALM server URL	<input type="text" value="http://10.1.8.176:8080/qcbin"/>	?

ALM servers list

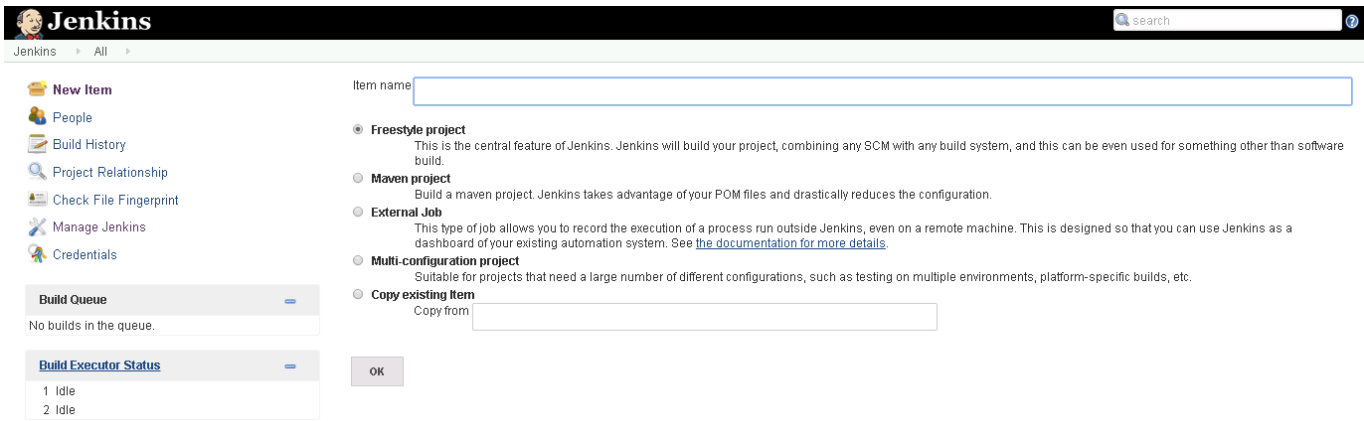
## Enabling the Artifact Deployer Plugin

Install (if not available) the artifact plugin from **Manage Jenkins** -> **Manage Plugins** -> **Available** tab -> **Artifact Deployer Plugin**.

## Configure the Jenkins plugin for ALM

1. Create a new freestyle project and name it *ALM Integration*.

Figure 10: Creating a project



2. Click the **Configure** link in the ALM Integration page.
3. Click the **This build is parameterized** check box.
4. Click **Add Parameter** three times and enter the details as listed in Parameter details. All parameters must be string parameters. These parameters are environment variables that are used when configuring the build steps.

Table 1: Parameter details

Parameter name	Default value	Description
AUT_Environment_Configuration_ID	Blank	Blank
Codar_Output	CodarOutput.json	Blank
Port	8080	Blank

Figure 11: Parameter details

Jenkins

ALM Integration configuration

Back to Dashboard  
Status  
Changes  
Workspace  
Build with Parameters  
Delete Project  
Configure

Build History trend

Project name: ALM Integration

Description:

☐ Discard Old Builds  
☒ This build is parameterized

**String Parameter**

Name: AUT\_Environment\_Configuration\_ID  
Default Value:  
Description:

**String Parameter**

Name: Codar\_Output  
Default Value: CodarOutput.json  
Description:

**String Parameter**

Name: port  
Default Value: 8080  
Description:

RSS for all RSS for failures

- Click **Add build step**.
- In the **Copy artifacts from another project** section, enter Pet Clinic in the **Project name** field. This copies the output of the file that is generated to the ALM Integration job's workspace. The copied file contains information such as host name, user name and so on of the node.

Figure 12: Configuring the values of the ALM Integration project

Build

**Copy artifacts from another project**

Project name: Petclinic

Which build: Latest successful build  
☒ Stable build only

Artifacts to copy: \$Codar\_Output

Artifacts not to copy:

Target directory:

Parameter filters:

☐ Flatten directories ☐ Optional ☒ Fingerprint Artifacts

Delete

- Click **Add build step**.
- Select **Execute AUT Environment Preparation using HP ALM Lab Management**.
- Enter the following values:

Figure 13: Specifying values for the Execute AUT Environment Preparation using ALM Lab Management plugin

[Delete](#)

Execute AUT Environment preparation using HP ALM Lab Management

Use this build step to assign values to AUT Environment Configuration in ALM.

ALM server	<input type="text" value="almserver"/>	
User name	<input type="text" value="willalex"/>	
Password	<input type="password" value="*****"/>	
Domain	<input type="text" value="HP"/>	
Project	<input type="text" value="Petclinic"/>	
AUT Environment ID	<input type="text" value="1001"/>	<a href="#">?</a>
AUT Environment Configuration	<input checked="" type="checkbox"/> Create a new configuration named: <input type="text" value="TriggeredByJenkins\$BUILD"/>	
	<input type="checkbox"/> Use an existing configuration with ID: <input type="text" value=""/>	
Path to JSON file	<input type="text" value="\${WORKSPACE}/\${Codar_Output}"/>	
Assign AUT Environment Configuration ID to:	<input type="text" value="AUT_Environment_Configuration_ID"/>	
AUT Environment Parameters		
	Parameter type: <input type="text" value="From JSON"/>	<a href="#">?</a>
	Parameter name: <input type="text" value="Parameters/App Server/IpAddress"/>	<a href="#">?</a>
	Parameter value: <input type="text" value="\$Nodes[?(@.id==VcenterServerType__VERSION__04.20.0000__GROUPID__com.hp.csa.type0002)].properties.[?(@.id==IpAddress)].values[0]"/>	<a href="#">?</a>
	<input checked="" type="checkbox"/> Get only the first value in case an array will be returned	
	<a href="#">Delete</a>	
	Parameter type: <input type="text" value="Environment"/>	<a href="#">?</a>
	Parameter name: <input type="text" value="Parameters/App Server/port"/>	<a href="#">?</a>
	Parameter value: <input type="text" value="port"/>	<a href="#">?</a>
	<a href="#">Delete</a>	

Table 2: Values of the ALM Lab Management plugin

Field	Description
ALM server	Name of the ALM plugin configuration
User name	Name of the ALM user who has permission to trigger the test suite
Password	Password of the ALM user who has permission to trigger the test suite
Domain	ALM domain name
Project	Name of the project that is associated with the domain and that which contains the test suite
AUT Environment ID	ALM environment ID
AUT Environment Configuration	Select this check box and enter a unique name with the build ID. You can see the value of the build ID from the \$BUILD_NUMBER environment variable. For example, TriggeredByJenkins \$BUILD_NUMBER
Path to JSON file	Path in which the JSON file is copied during the Copy artifacts from another project step.
Assign AUT Assignment Configuration ID to	Parameter that is passed to the test script. This parameter is created in the AUT environment. Enter Parameters/App Server/IpAddress
AUT Environment Parameters	Parameter type: From JSON Parameter name: Parameters/App Server/IpAddress Parameter value: \$.Nodes[?(@.id==<VcenterServerType>__<VERSION>__04.20.0000__GROUPID__com.hp.csa.type0002)].properties.[?(@.id==IpAddress)].values[0]  You can obtain these values from the JSON file that refers to the node component which contains the application server details.

- Click **Add build step**.
- Select the **Execute HP Tests using HP ALM Lab Management** plugin and enter the values. For information about these values, see the Jenkins ALM plugin documentation.

Figure 14: Values of the Execute Tests using ALM Lab Management plugin

Execute HP tests using HP ALM Lab Management

Use this build step to run ALM server-side functional test sets and Build Verification Suites.

Don't forget to enable the **Publish HP tests result** option in the **Post-build Actions** section so that the tests results are published.

ALM server

My ALM ▾

User name

tester

Password

\*\*\*\*\*

Domain

DEFAULT

Project

PetClinic

Run type

Test Set ▾

Test Set/Build Verification Suite ID

1

Description

Timeslot Duration (in minutes, minimum = 30)

30

Environment Configuration ID

\$(AUT\_Environment\_Configuration\_ID) ⓘ

☐ Use CDA for provisioning and deployment

Delete

Add build step ▾

Post-build Actions

Add post-build action ▾

## Create the test environment in ALM

The Jenkins ALM integration is supported only in ALM version 12.20. See the ALM Installation Guide for information about installing ALM 12.20.

After installing ALM, the configurations required for the Jenkins-ALM integration are shown in Figure 16.

Figure 15: Configurations required for the integration



After installing ALM, open the <http://localhost:8080/qcbin/> link using Internet Explorer.

# Site Administration

Use Site Administration to create projects and users. The users should be associated with the project. The following screens show how the Pet Clinic project is created.

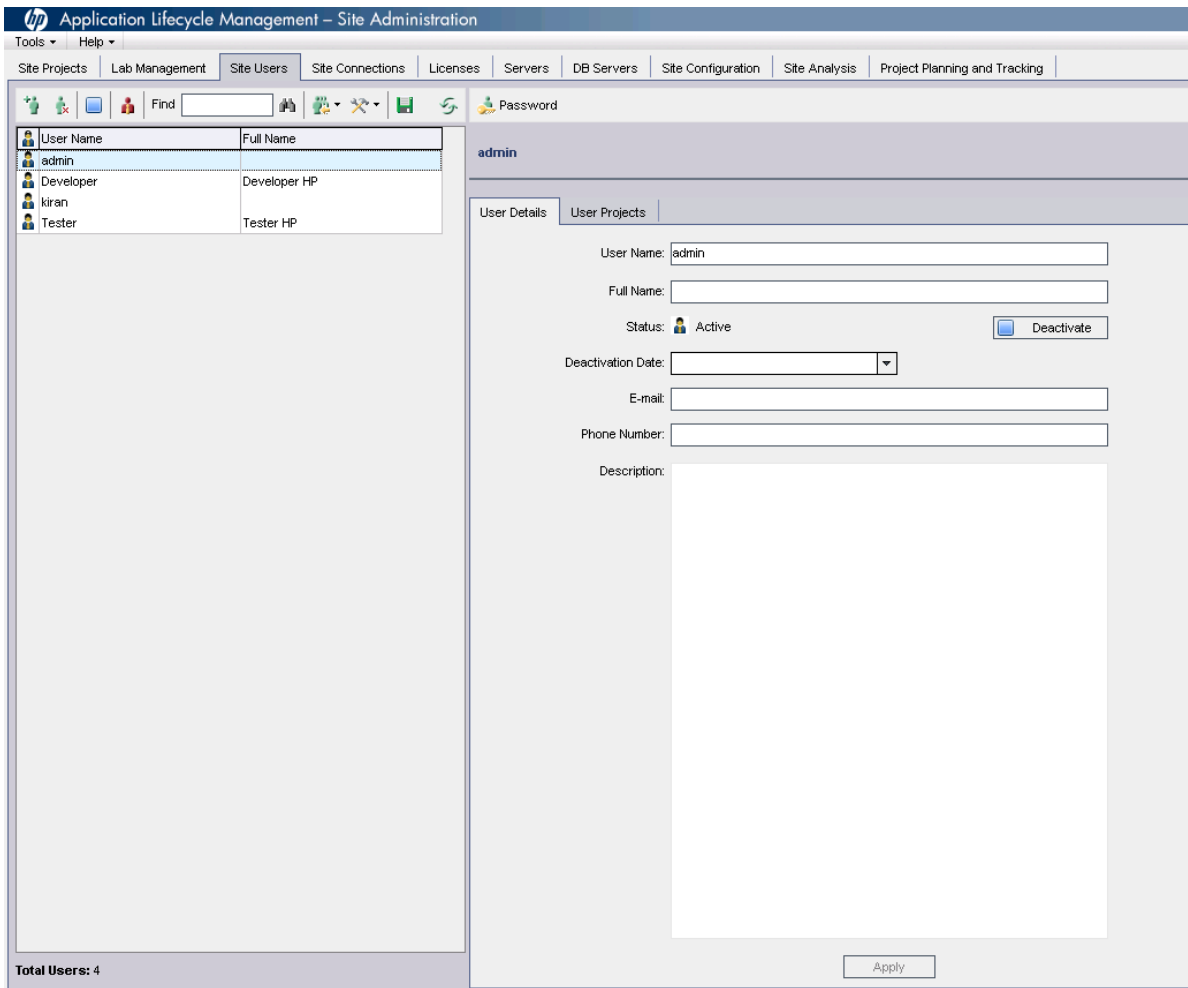
Figure 16: Entering the project details

The screenshot displays the HP Application Lifecycle Management - Site Administration web interface. The top navigation bar includes tabs for Site Projects, Lab Management, Site Users, Site Connections, Licenses, Servers, DB Servers, Site Configuration, Site Analysis, and Project Planning and Tracking. The left sidebar shows a tree view with 'DEFAULT' and 'Projects' folders, with 'PetClinic' selected under 'Projects'. The main content area is titled 'PetClinic' and contains several sections: 'Project Database' with fields for Database Type (MS-SQL), Database Name (default\_petclinic\_db), Database Server (localhost), Created From Project (Empty Database), Created From Domain (Templates), Maintenance State (Idle), Unicode supported (N), and Connection String (jdbc:mercury:sqlserver://localhost:1433); 'Project Directory' with a text field (C:\ProgramData\HP\ALM\repository\qc\Default\PetClinic) and a Search Language dropdown; 'Repository Cleanup' with a Promote Repository Cleanup button; 'Project Planning and Tracking' with a checked Automatic Calculations State and a Run Now button; and 'Miscellaneous' with a Send Mail Automatically checkbox, a Send E-mail Now button, and fields for User Quota (Unlimited Connections) and Description (Created on 2/18/2015 4:17:03 PM).

After entering the project details, users have to be created and associated with the Pet Clinic project. In this example, a user called Tester has been created and Tester has been associated with the Pet Clinic project. In this case, Tester is also a project administrator (to avoid any permission issues).

Figure 17: Creating a user





For more information about using Site Administration, see the ALM Administration Guide.

## ALM Lab Service

You can install ALM Lab Service either on the same computer in which ALM 12.20 is installed or on a different computer. Ensure that ALM Lab Service is running. For information about installing and configuring ALM Lab Service, see the ALM Guide.

**Note** – If ALM Lab Service is installed on a different computer, then VAPI-XP must be installed. To do this, open Internet Explorer on the computer in which ALM Lab Service is installed and type `http://<alm server hostname>:<port>/qcbn/addins.html`

Figure 18: Installing VAPI-XP



# Application Lifecycle Management - Tools

## HP ALM Connectivity

Enables you to integrate HP ALM with other tools.

## HP ALM Lab Service

Enables you to remotely trigger functional tests and maintenance tasks on a testing host using HP ALM. Install and configure the HP ALM Lab Service agent on functional testing hosts (such as VAPI and United Functional Testing) that need to connect to Lab Management.

## HP ALM Client Registration

Deploys and registers ALM components on a client machine.

Click on "HP ALM Client Registration"

## Shared Deployment for Virtual Environments

Deploys ALM components on a shared location of a client machine.

## Webgate Customization

Customizes the WebGate client component.

## More HP ALM Add-ins

To download the VAPI-XI libraries, register the client with ALM as shown in Figure 20.

Figure 19: Downloading libraries



# Application Lifecycle Management - Tools

## HP ALM Client Registration

To work with other HP testing tools as well as third-party and custom tools, HP ALM must be registered on client machines. HP ALM Client Registration enables you to deploy and register the following ALM components on a client machine:

- HP ALM Client components
- HP ALM Site Administration Client components

For a list of the tools for which you must register ALM on a client machine, see the "Registering ALM" topic in the *HP Application Lifecycle Management Installation Guide*.

### Instructions:

1. Log on to the client machine as a local user or a domain user with administrator privileges.
2. Make sure you have the file system and registry permissions listed below.
3. Make sure you close all instances of ALM Quality Center and any integration tools.
4. Open the browser as an administrator (for example, right-click the Internet Explorer icon and select **Run as Administrator**).
5. Start ALM and re-access this Tools page for HP ALM Client Registration.
6. Click **Register HP ALM** below for ALM Client components.
7. Click **Register HP ALM Site Administration** below for ALM Site Administration Client components.
8. Close and re-open the browser.

### Notes:

- After components are registered on the client machine by a user with administrator privileges, users without administrator privileges can start ALM client components.

### Required Permissions:

You must have the following file system permissions:

- Full read and write permissions on the **HP\ALM-Client** deployment folder. This is located at:
  - Windows 8, 7, 2008R2: %ALLUSERSPROFILE%
- Full read and write permissions to the **Temp** (%TEMP% or %TMP%) directory. The installer program writes installation and log files to this directory. This is generally located at:
  - Windows 8, 7, 2008R2: C:\Users\<username>\AppData\Local\Temp

You must have full read and write permissions on the following registry keys:

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER\Software
- HKEY\_LOCAL\_MACHINE\SOFTWARE

Versions supported: HP Application Lifecycle Management 12.20.

Click on "Register"



## ALM Lab Management

1. Open ALM Lab Management from the computer in which ALM 12.20 is installed. Login as an administrator.

2. Go to **Testing Host** → **New Testing host**.

Figure 20: Testing hosts

Testing Host Details

Host ID: 1003 Name/IP: adash-alm

Details

- Purpose: VAPI:XP
- Installation: Windows Host
- Belongs To Pools: General:PetclinicPool
- Status: Operational
- Location: Default
- Priority: 5
- Host Attributes:
- Registration Status: Registered
- Registration Auto Approve: Y
- Exclude From Automatch: N
- Source: Local
- Reserved for User:

Description

3. Ensure that the purpose is VAPI-XP.
4. After adding the new host, it must appear as shown in Figure 22.

Figure 21: New host

HP Application Lifecycle Management - Lab Management - Internal

Domain: DEFAULT, Project: LAB\_PROJECT | User: admin

Testing Hosts Edit View Favorites

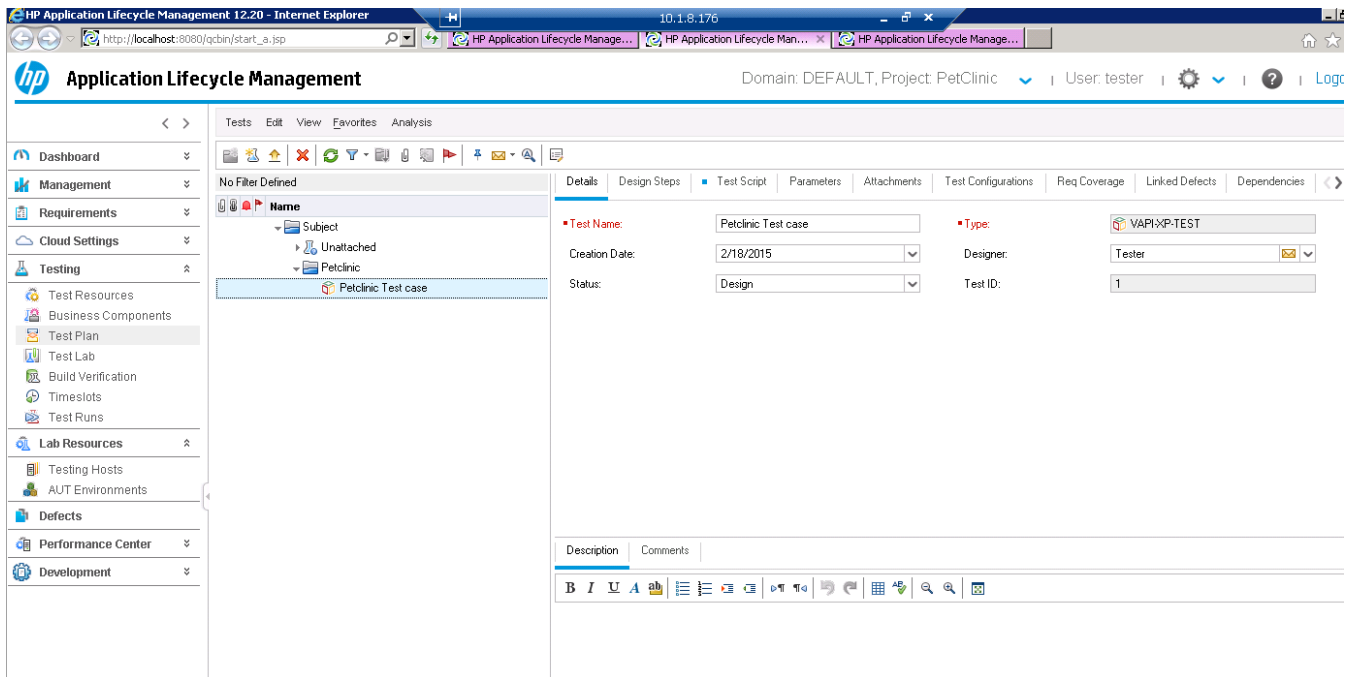
No Filter Defined

Source	Name/IP	Status	Registration...	Belongs To...	Purpose	Location	Host Attributes	Priority	Host State	Host Version	Last Poll T
Local	adash-alm	Operational	Registered	General:Petclinic...	VAPI:XP	Default		5	Idle	12.20.192.0	3/17/2015 1

## ALM Desktop Client

1. Open ALM Desktop Client from the computer in which ALM 12.20 is installed.
2. Login as Tester (domain is default and the project is Pet Clinic for this example)
3. Go to **Test Plan** → **Create a New Test Case**.

Figure 22: Creating a test case



4. Ensure that the type is VAPI-XP-TEST.

5. Add the following script on the **Test Script** tab.

```
' Petclinic Test [VBScrip]
' Created by Application Lifecycle Management
' 2/9/2015 3:25:28 AM
' =====

' -----
' Main Test Function
' Debug - Boolean. Equals to false if running in [Test Mode] : reporting to Application Lifecycle Management
' CurrentTestSet - [OTA COM Library].TestSet.
' CurrentTSTest - [OTA COM Library].TSTest.
' CurrentRun - [OTA COM Library].Run.
' -----

Sub Test_Main(Debug, CurrentTestSet, CurrentTSTest, CurrentRun)

' *** VBScrip Limitation ! ***
' "On Error Resume Next" statement suppresses run-time script errors.
' To handle run-time error in a right way, you need to put "If Err.Number <> 0 Then"
' after each line of code that can cause such a run-time error.
On Error Resume Next

' clear output window
TDOutput.Clear
```

```

' TODO: put your code here
  Dim ipAdd : ipAdd = "10.1.9.136"

'url_base= "10.1.9.136"
url_base=CurrentRun.getRuntimeParameterByName("ipAddress")
url= "http://" & url_base & ":8080/petclinic"

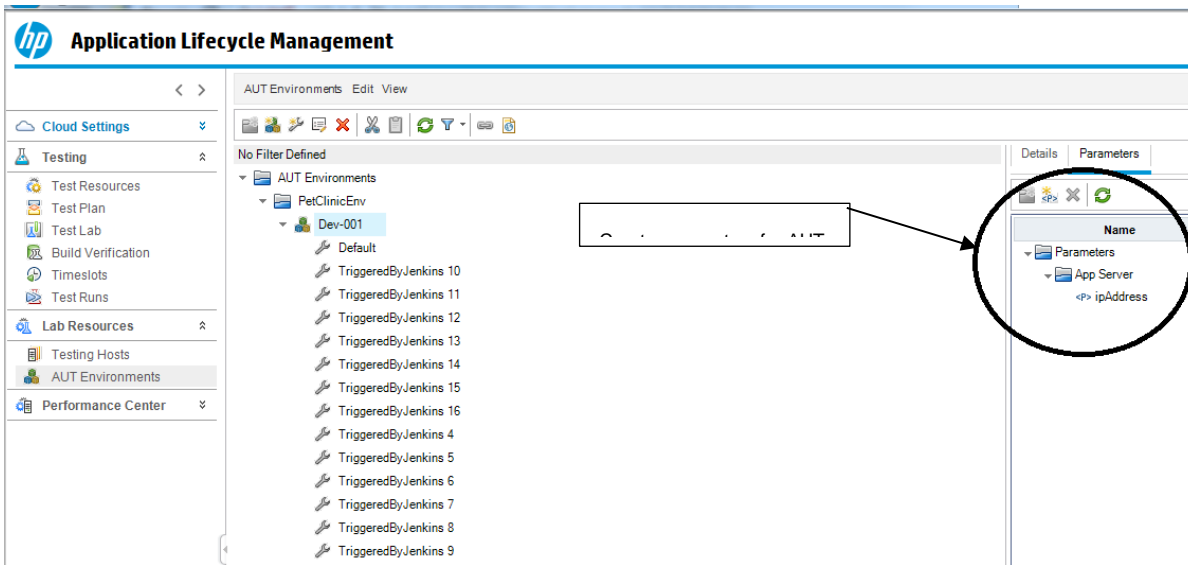
set ie = CreateObject("InternetExplorer.Application")
'   ie.Navigate "http://10.1.9.136:8080/petclinic"
      ie.Navigate url
      ie.Visible = true

If Not Debug Then
End If
' handle run-time errors
If Err.Number <> 0 Then
  TDOOutput.Print "Run-time error [" & Err.Number & "] : " & Err.Description
  ' update execution status in "Test" mode
  If Not Debug Then
    CurrentRun.Status = "Failed"
    CurrentTSTest.Status = "Failed"
  End If
End If
End Sub

```

6. Ensure that Testing Hosts is updated with the information added in Lab Management.
7. Verify that the Pet Clinic test case is successful by manually running the test case in the test lab.
8. Go to **Lab Resources** -> **AUT Environments** and add a new AUT parameter called App Server as shown in Figure 24.

*Figure 23: Added the App Server parameter*



## End-to-end flow

Now Jenkins is set up for the Pet Clinic project and the ALM\_Integration project. The ALM\_Integration project is configured as a dependent project for Pet Clinic. The following steps provide the end-to-end flow:

1. Jenkins pulls in Pet Clinic changes from SVN.
2. The Pet Clinic build is triggered.
3. After the build is complete, Jenkins archives the artifacts.
4. The Codar plugin that is configured as a post build action is triggered as show in Figure 25.

Figure 24: Triggering the post build action

```
Archiving artifacts
Jenkins URL: http://10.1.6.54:8088/
Codar Deploy URL: https://10.1.6.54:8444/csa/api/codar/codar-oointegration/invokeFlow
Codar Deploy Status URL: https://10.1.6.54:8444/csa/api/codar/app-deploy/
No Properties file.
JSON Input to Codar :: {
  "designurl": "http://10.1.6.54:8088/job/Petclinic/29/artifact/PetclinicApp.json", "environment": "De
9fa310fd_VERSION_1__GROUPID__com.hp.csa.type0001=[{artifacturl:http://10.1.6.54:8088/job/Petcli
[{vmNamePrefix:ChangedDBName}],VcenterServerType__VERSION__04.20.0000__GROUPID__com.hp.csa.type0
Output from the Codar REST CALL ....
Received Output from the Codar REST CALL ...

{

{
  "serviceInstanceId" : "8a8186b64b11370b014bda5aa02d3612"
```

5. The Codar plugin calls an API in Codar and triggers the deployment.
6. It keeps polling till the deployment is complete as shown in Figure 26.

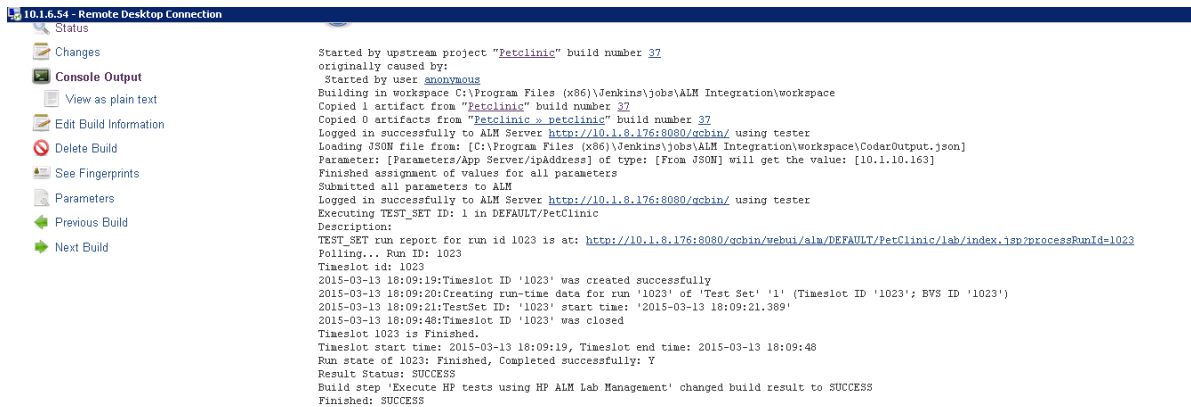
Figure 25: Deployment complete

```
Deployment Status :DEPLOYING
getserviceInstance from the Codar REST CALL .... "status" : "ACTIVE",
Deployment Status :ACTIVE
Final Deployment Status :ACTIVE
Writing Codar output on this folder:C:\Program Files (x86)\Jenkins\jobs\Petclinic\builds\2015-03-02_17-27-38\archive
Codar output File Location:C:\Program Files (x86)\Jenkins\jobs\Petclinic\builds\2015-03-02_17-27-38\archive
Warning: you have no plugins providing access control for builds, so falling back to legacy behavior of permitting any downstream builds to be triggered
Triggering a new build of ALM\_Integration
Finished: SUCCESS
```

7. After the deployment is complete, the status of the final deployment becomes active.

8. The Codar plugin creates a JSON file that contains the IP address, user name, password, and other details of the deployed instance. The JSON file is created based on the node ID input given during the configuration of the plugin.
9. The JSON file is stored in the workspace as CodarOutput.json. This is the integration point between Codar and ALM.
10. Because ALM\_Integration is configured as a subsequent project, the ALM\_Integration build job is triggered after the Pet Clinic job is successful.
11. Open the output console of ALM\_Integration to see the status of that job. It must be successful as shown in Figure 27.

Figure 26: Successful job status



12. The ALM\_Integration job copies the JSON file from workspace. It then reads the JSON file and looks for parameters such as IP address.
13. The ALM\_Integration job logs on to ALM with the credentials provided during configuration. It triggers the test case and executes the test script.
14. The test script is executed in the computer that is configured in Lab Management. VAPI tests are executed. In this case, it opens the Pet Clinic link [http://localhost:8080/qcbin/start\\_a.jsp](http://localhost:8080/qcbin/start_a.jsp) in a browser and verifies that the application is deployed successfully.
15. The test results are updated back into Test Run as shown in Figure 28.

Figure 27: Updated test results

Run ID	Test Set	State	Start Time	End Time	Completed...	Tester
1023	Petclinic Test	Finished	3/13/2015 6:09:2...	3/13/2015 6:09:4...	Y	tester
1022	Petclinic Test	Finished	3/11/2015 4:36:5...	3/11/2015 4:37:0...	Y	tester
1021	Petclinic Test	Finished	3/11/2015 2:53:5...	3/11/2015 2:54:1...	Y	tester
1020	Petclinic Test	Finished	3/6/2015 10:49:3...	3/6/2015 10:49:4...	Y	tester
1019	Petclinic Test	Finished	3/6/2015 10:44:2...	3/6/2015 10:44:3...	Y	tester
1018	Petclinic Test	Finished	3/3/2015 3:32:59...	3/3/2015 3:33:16...	Y	tester
1017	Petclinic Test	Finished	3/3/2015 2:15:40...	3/3/2015 2:15:56...	N	tester
1016	Petclinic Test	Finished	3/3/2015 2:13:16...	3/3/2015 2:13:33...	N	tester
1015	Petclinic Test	Finished	3/3/2015 11:55:2...	3/3/2015 11:55:4...	N	tester
1014	Petclinic Test	Finished	3/3/2015 11:41:0...	3/3/2015 11:41:2...	N	tester
1013	Petclinic Test	Finished	3/3/2015 11:28:1...	3/3/2015 11:28:3...	N	tester
1012	Petclinic Test	Finished	3/3/2015 11:23:4...	3/3/2015 11:23:5...	N	tester
1011	Petclinic Test	Finished	3/2/2015 5:46:34...	3/2/2015 5:46:54...	N	tester
1010	Petclinic Test	Finished	3/2/2015 2:56:27...	3/2/2015 2:57:22...	N	tester
1009	Petclinic Test	Finished	3/2/2015 2:34:12...	3/2/2015 2:35:21...	N	tester
1008	Petclinic Test	Finished	2/24/2015 5:58:0...	2/24/2015 5:58:1...	N	tester
1007	Petclinic Test	Finished	2/24/2015 5:43:5...	2/24/2015 5:44:1...	N	tester
1006	Petclinic Test	Finished	2/24/2015 5:33:1...	2/24/2015 5:33:2...	N	tester
1005	Petclinic Test	Finished	2/24/2015 5:30:4...	2/24/2015 5:31:0...	N	tester
1004	Petclinic Test	Finished	2/24/2015 4:27:3...	2/24/2015 4:27:4...	N	tester
1003	Petclinic Test	Finished	2/24/2015 3:38:4...	2/24/2015 3:38:4...	N	tester
1002	Petclinic Test	Finished	2/24/2015 3:23:2...	2/24/2015 3:23:2...	N	tester
1001	Petclinic Test	Finished	2/24/2015 2:58:3...	2/24/2015 2:58:5...	N	tester

1.

# Send documentation feedback

If you have comments about this document, you can send them to [clouddocs@hpe.com](mailto:clouddocs@hpe.com).

## Legal notices

### Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

### Restricted rights legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright notice

© Copyright 2016 Hewlett Packard Enterprise Development LP

### Trademark notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

## Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to the following URL and sign-in or register:

<https://softwaresupport.hpe.com>.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your Hewlett Packard Enterprise sales representative for details.

## Support

Visit the Hewlett Packard Enterprise Software Support Online web site at <https://softwaresupport.hpe.com>.