# HP Codar

Software Version: 1.50

# Concepts Guide

Document Release Date: July 2015
Software Release Date: July 2015

# Legal Notices

## Warranty

## Restricted Rights Legend

## Copyright Notice

## Trademark Notices

# 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: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

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

# Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

**http://h20229.www2.hp.com/passport-registration.html**

To find more information about access levels, go to:

**http://h20230.www2.hp.com/new_access_levels.jsp**

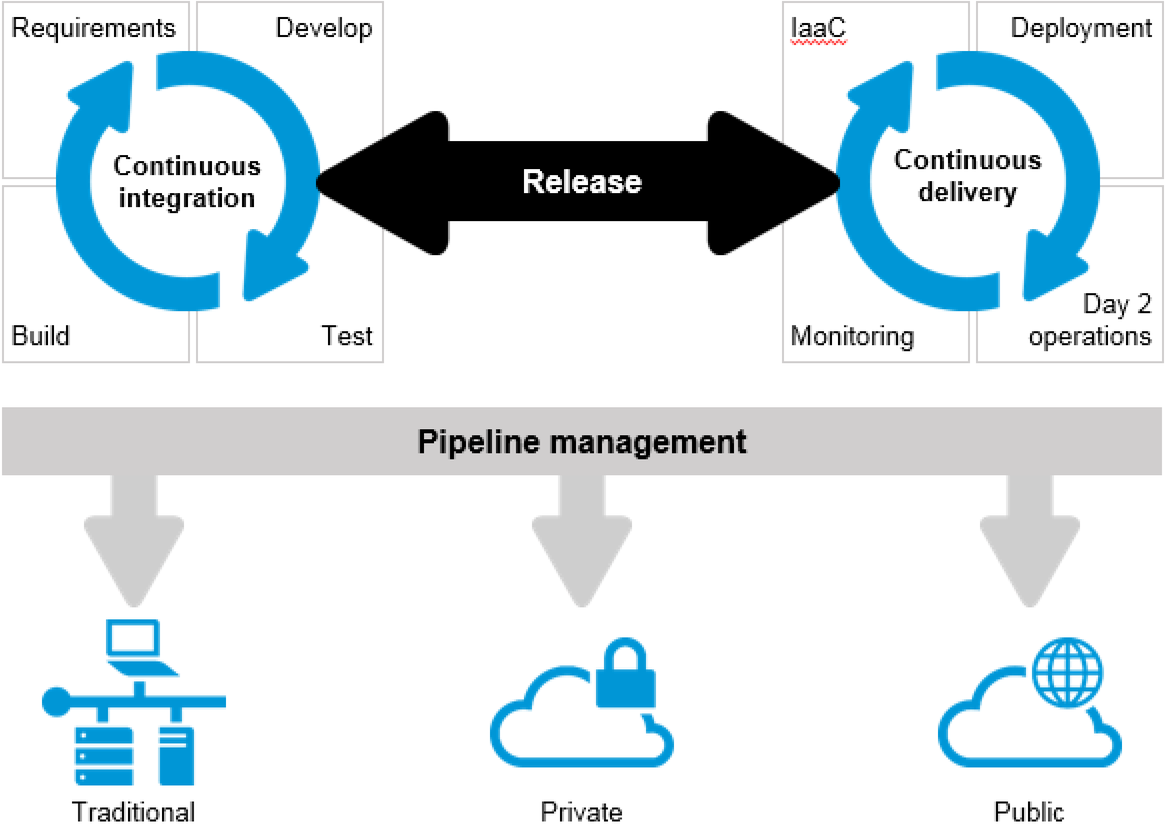
**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

# Contents

# HP Codar

Organizations are facing new challenges when extending continuous integration into continuous delivery. Challenges include consistently deploying applications through development to production environments while considering the differences in those environments.

DevOps provides a framework to bridge the gaps between the development (Dev) and operations (Ops) environments by using a set of principles, methods, and practices around collaboration, automation, and governance. The goal is to extend continuous build or assembly integration to repeatable and consistent application deployment across heterogeneous environments. The following diagram illustrates the continuous integration and continuous delivery cycle in a DevOps environment.



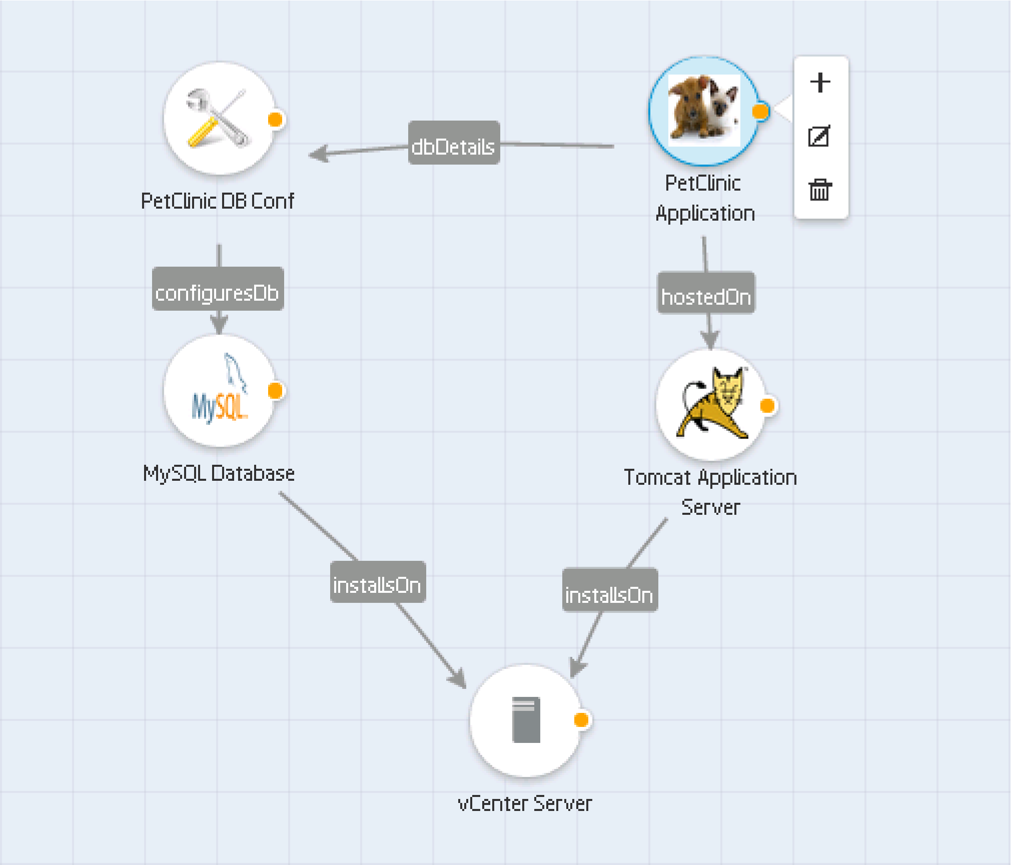
# HP Codar overview

HP Codar facilitates continuous delivery in which every change to the system is releasable and that every code change can be deployed in production. It enables automation of continuous delivery where every code change triggers a build, which is deployed, automated unit tests are executed, and the application is automatically deployed to an environment based on policies that are defined in a runbook automation flow. Continuous delivery aims to deliver frequently and get fast feedback from users.

## Declarative based modeling

Automating the deployment of applications using declarative based modeling allows the user to declare the end state of the application deployment (the application components and the dependencies between them) while the process to get to that state is triggered in the background. This allows the user to focus on what is deployed rather than how it gets deployed, which results in a shorter time to automate the deployment of multi-tier applications and greater simplicity in managing them over time.

HP Codar supports declarative-based model development that involves creation, integration, and maintenance of complex designs through a user interface. A model consists of a topology design and its properties. HP Codar provides flexibility for the user to modify the properties during the time of realization (similar to late binding).

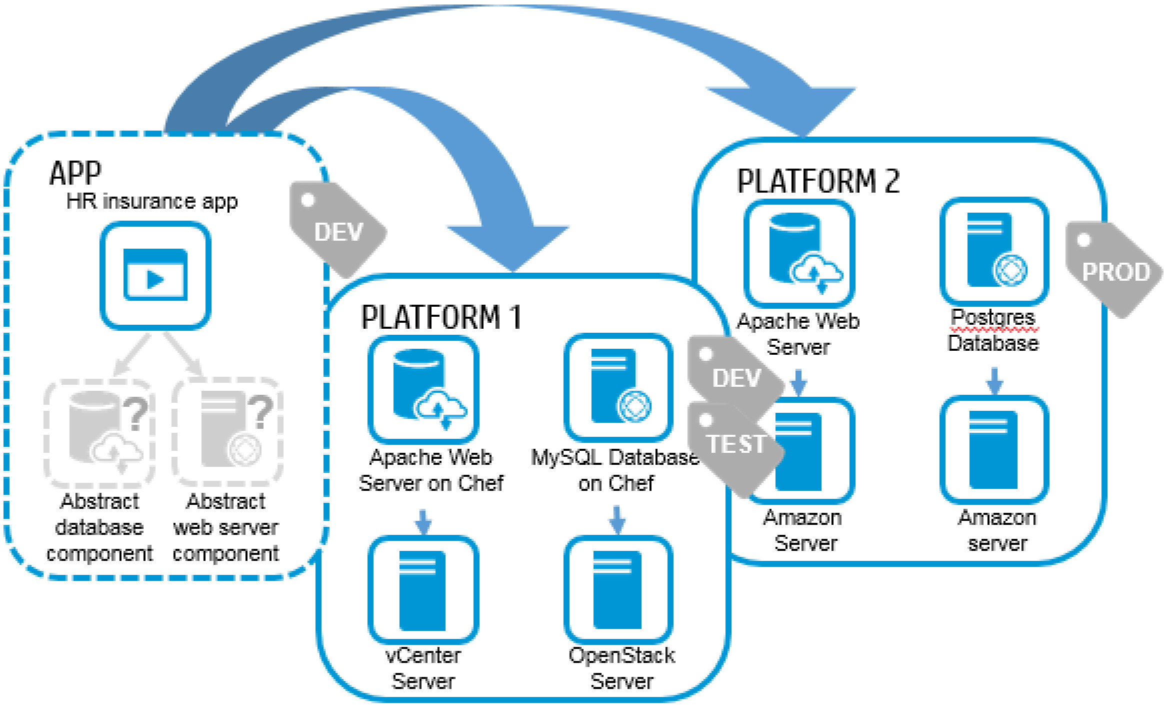


# Topology composition

Topology composition is used to compose the application design with the infrastructure design at run time. During the application deployment, the infrastructure need varies for each deployment, the topology composition helps in defining these variable infrastructure needs in the application design and allows to compose with different infrastructure design in deploy time.

The capabilities and characteristics are used to describe the components. The application design will define the requirements using the capability components and characteristics in the design. The application design cannot be provisioned on its own and requires the selection of a compatible infrastructure design. The infrastructure design components are matched for their capability and characteristics to check the compatibility and the matching designs are chosen as compatible infrastructure design during the deployment.

The following illustration shows the topology composition for an HR insurance app. The app requires a database component and web server component, which are defined in the application design APP. This is fulfilled by PLATFORM1 as it has the Apache Web Server which has the web server capabilities and its characteristics and MySQL database, which has the database capability and its characteristics. Similarly the PLATFORM2 also matches the APP requirements.

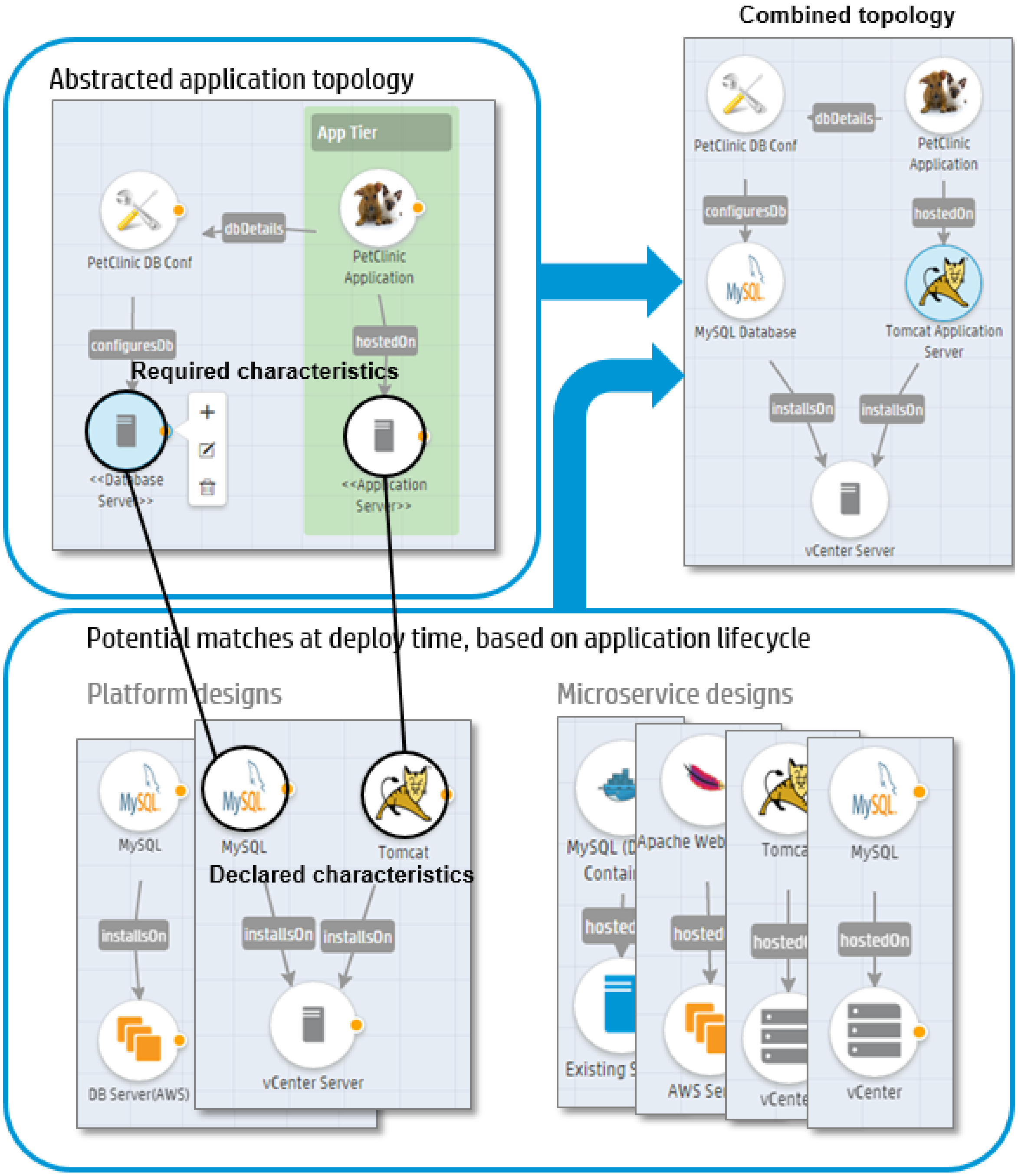


# Microservices

A partial application design can be deployed using multiple infrastructure designs that provide the platform or infrastructure services rather than single service. A partial application can use components from multiple infrastructure. You can choose either a single service that matches all capabilities, or you can choose components from different designs.

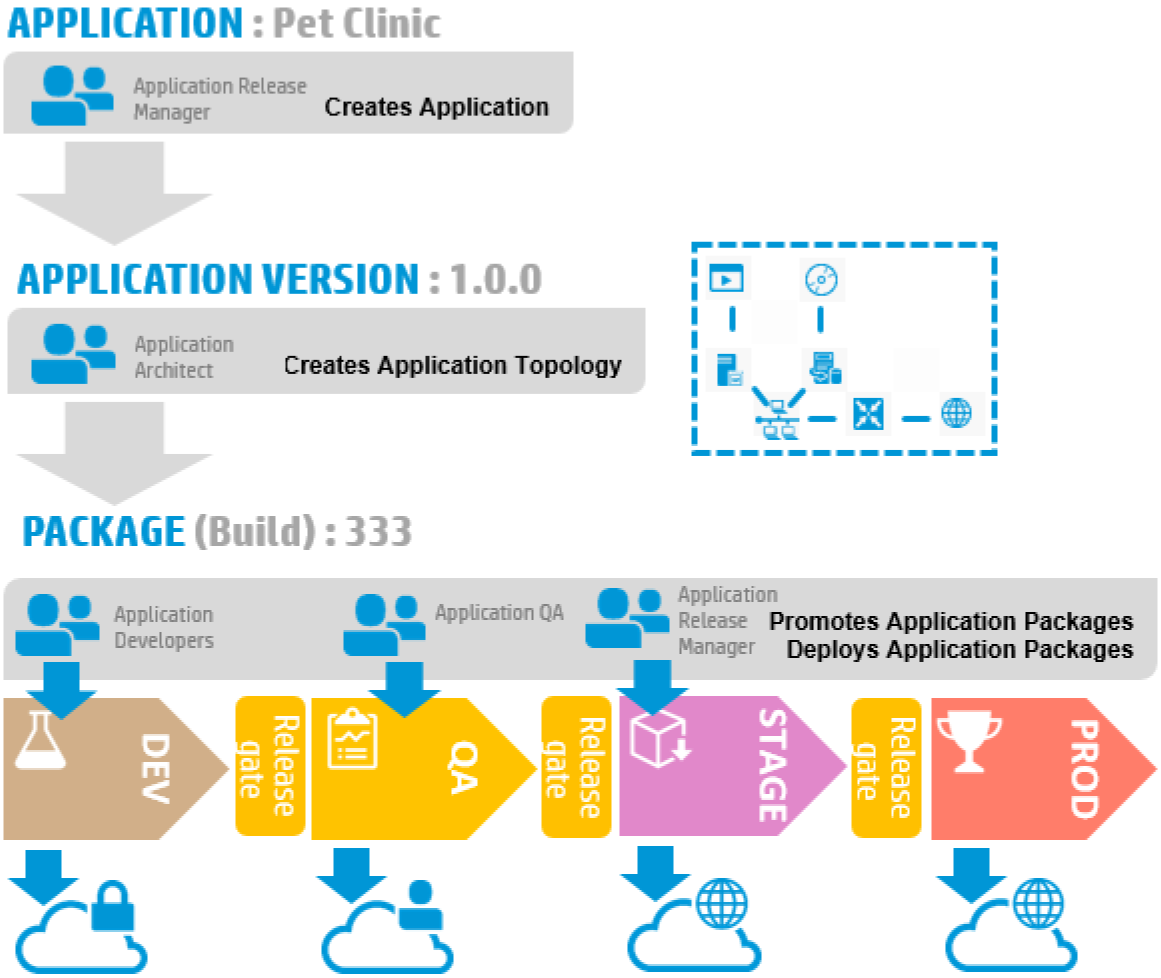
For example, if the application requires database and application services, it should be possible to select a design which has database and application in a single design or you can choose a database from one service design and an application from another service design.

A combined topology is created at run time based on microservice selection. The microservice can be associated with lifecycle stages.



# Application pipeline management

Automating the deployment of applications is a complicated and lengthy process and requires significant investment. Applications are deployed differently in development and in production, causing many errors. Application pipeline management allows you to deploy applications across different environments using the same topology model as described earlier.You can choose different microservices in different stages; however, the application design remains the same. This means that the same design will be deployed and tested across different lifecycle stages.

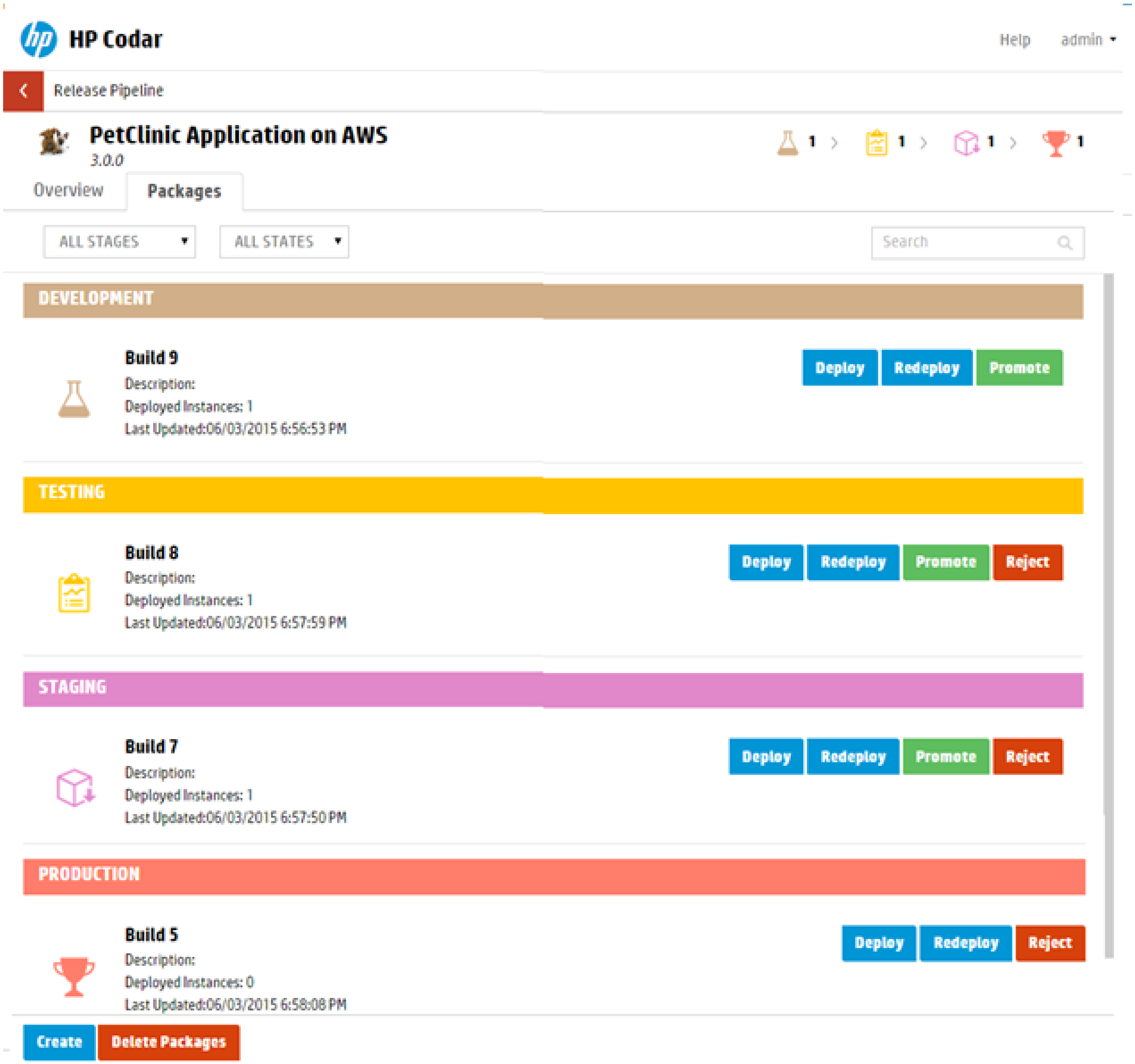


## Package operations

HP Codar is a centralized structure for implementing a DevOps environment. Different roles can deploy, redeploy, promote, or reject the packages. Packages are promoted from one stage to another in a consistent and repeatable manner. This ensures visibility to the developers when their applications are pushed into production.

When packages are created and deployed, new virtual machines will be created and packages will be deployed. You can execute tests on a deployed instance, and the package can be either promoted or rejected

HP Codar facilitates application pipeline management capabilities, as show in the following screenshot.



PetClinic Application on AWS
3.0.0

DEVELOPMENT

Build 9
Description:
Deployed Instances: 1
Last Updated:06/03/2015 6:56:53 PM

TESTING

Build 8
Description:
Deployed Instances: 1
Last Updated:06/03/2015 6:57:59 PM

STAGING

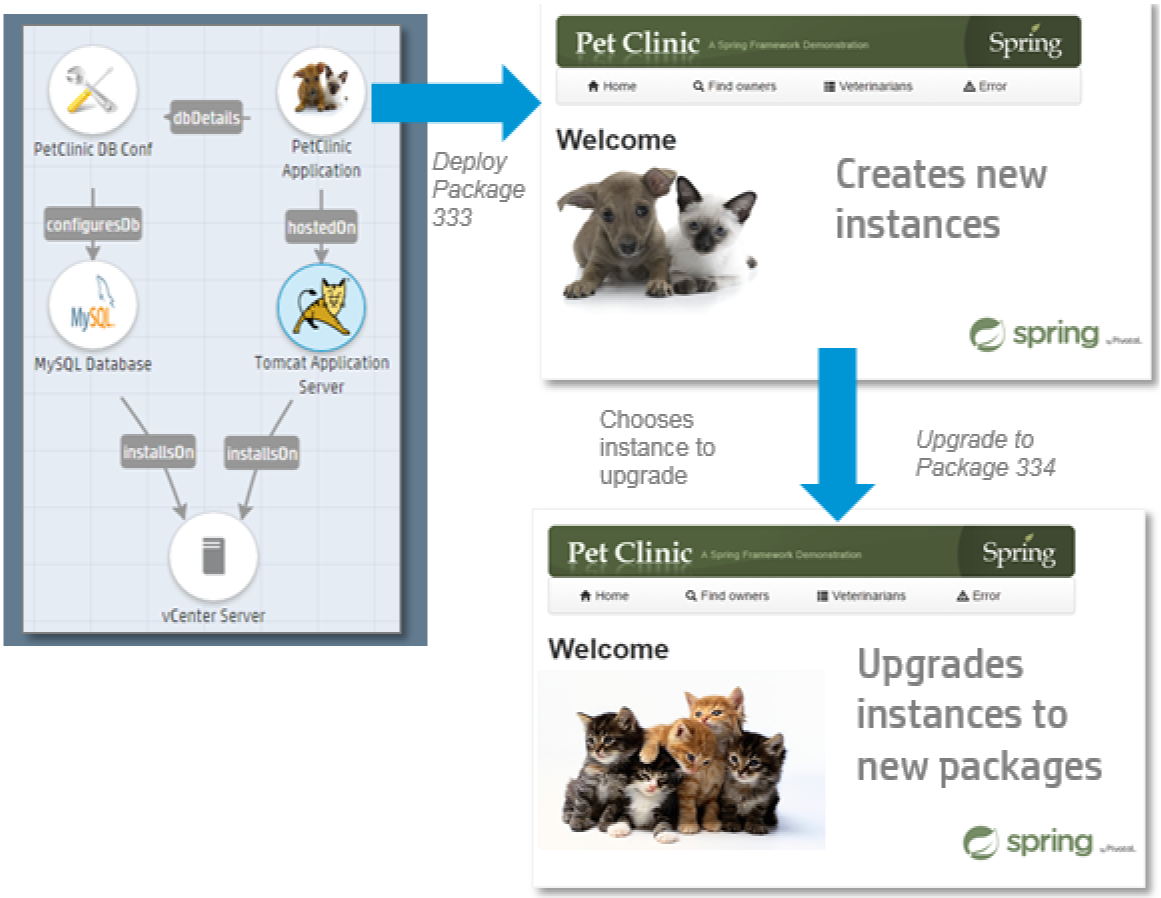
Build 7
Description:
Deployed Instances: 1
Last Updated:06/03/2015 6:57:50 PM

PRODUCTION

Build 5
Description:
Deployed Instances: 0
Last Updated:06/03/2015 6:58:08 PM

# Deploy and redeploy

A package can be redeployed on an instance that has an older package. You can view instance details and pick an existing insistence. Redeploy can also be used to upgrade or patch a component. Because redeploy invokes a modify action for all components, all components in a design can be upgraded to new versions.

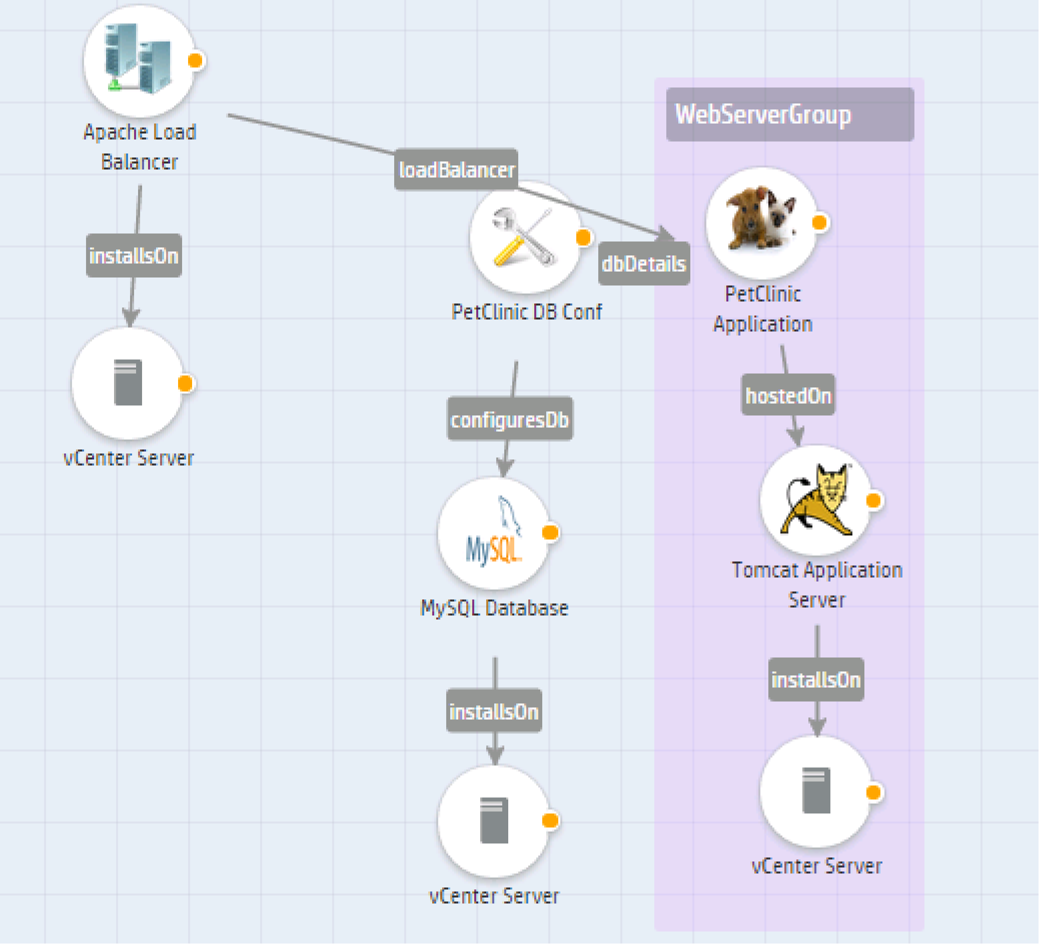


## Scale out

During Topology design creation you can create a scaling group. A scaling group represents a scalable stack. There can be multiple scalable groups in an application design.

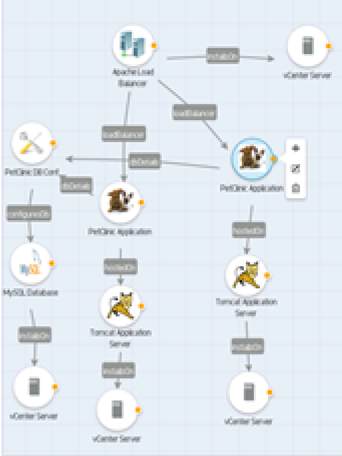
You can scale out after the deployment is complete. When you scale out, the full stack is replicated.

For example, the image below shows the web tier as a logical group named webServerGroup.

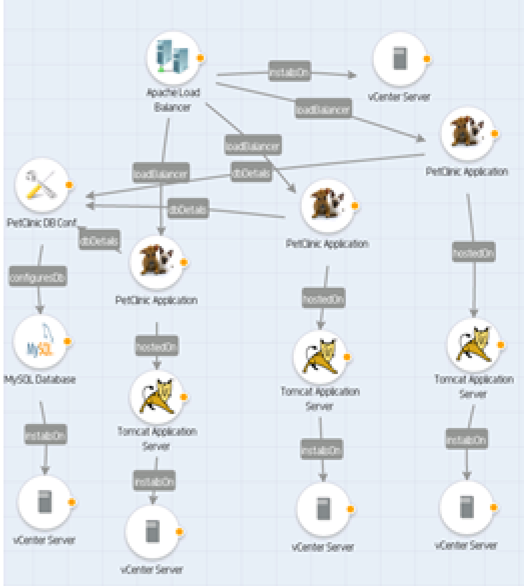


This group was scaled in the Development stage to one group and in Testing it is scaled to two groups in the image below.

Development stage



Testing stage



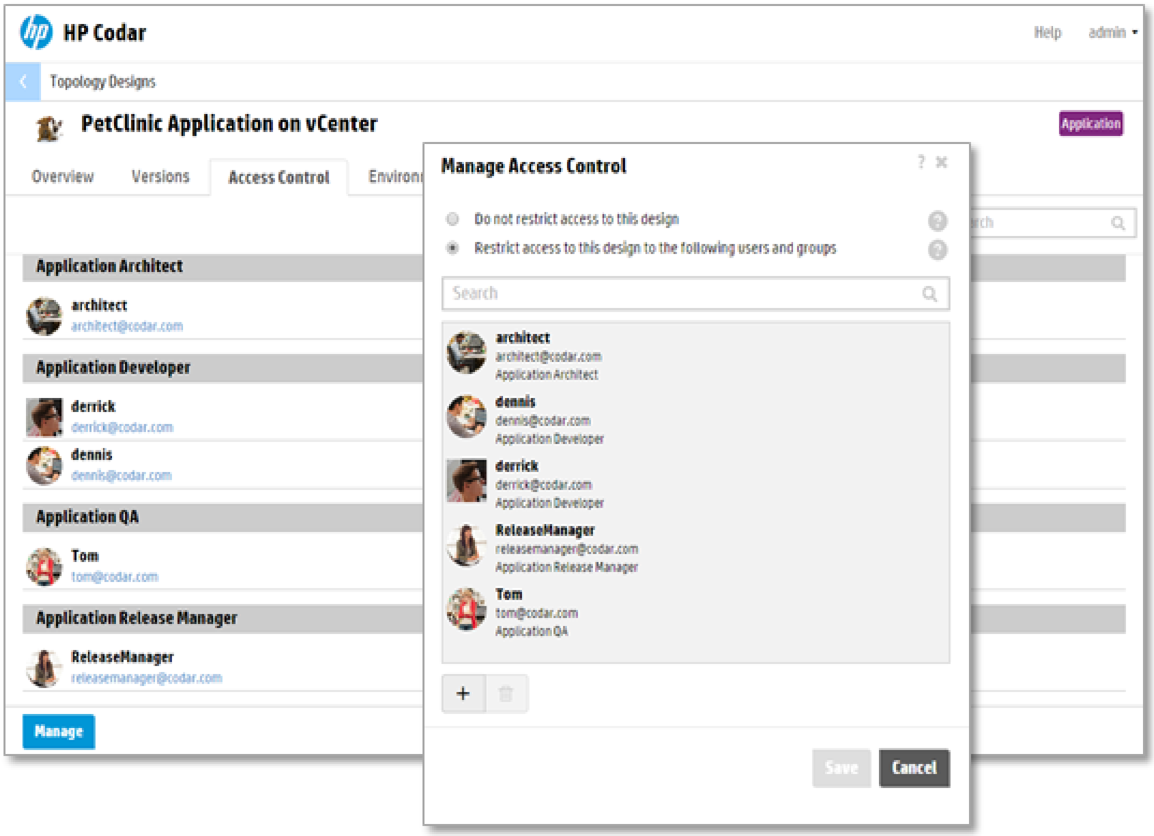
# User access

User access can be configured for topology designs. Both users and LDAP groups can be added to the designs. An application architect can create a design and either make the design public or restrict it to certain users.

The application architect can confugre users and groups:

- Users for each role are defined at the application level for a granular level of control.
- Groups should represent application teams to automatically assign roles for the application.

The following image shows a design configured for various users including an application architect, developers, QA, and release manager.



PetClinic Application on vCenter

Application Architect

architect
architect@codar.com

Application Developer

derrick
derrick@codar.com

dennis
dennis@codar.com

Application QA

Tom
tom@codar.com

Application Release Manager

ReleaseManager
releasemanager@codar.com

Manage Access Control

○ Do not restrict access to this design

● Restrict access to this design to the following users and groups

Search

architect
architect@codar.com
Application Architect

dennis
dennis@codar.com
Application Developer

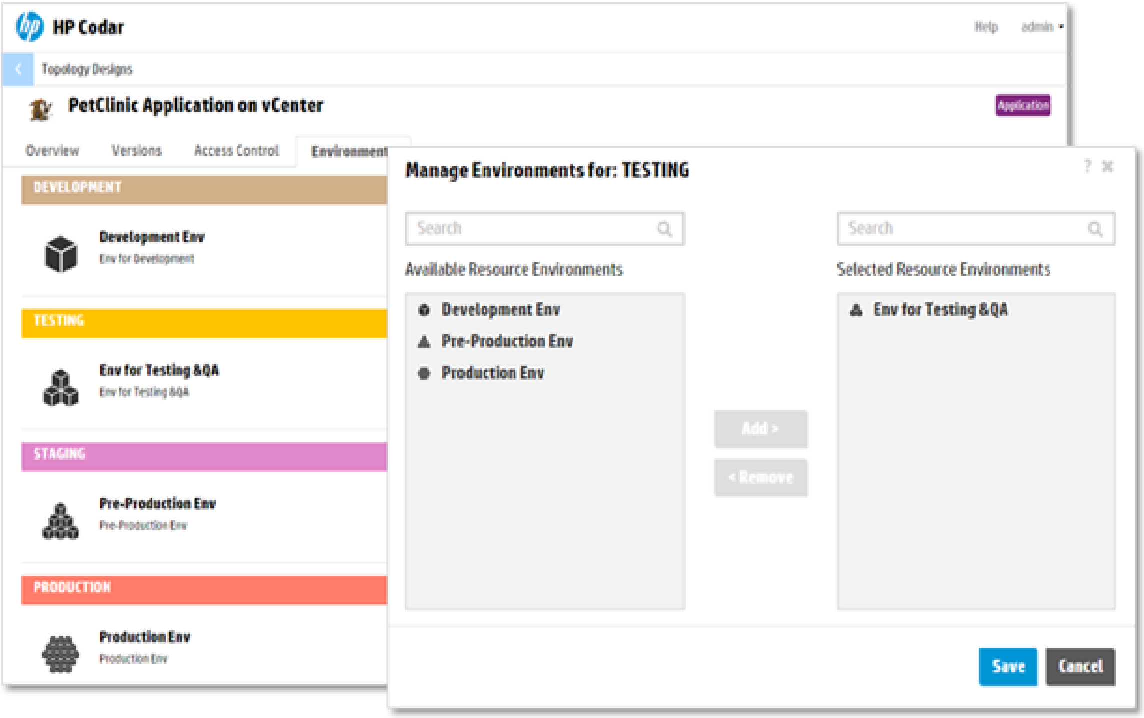
derrick
derrick@codar.com
Application Developer

ReleaseManager
releasemanager@codar.com
Application Release Manager

Tom
tom@codar.com
Application QA

# Environments

You can select different environments for each lifecycle stage at the application level. For example, you can configure vCenter for deployment but a public cloud environment for staging.

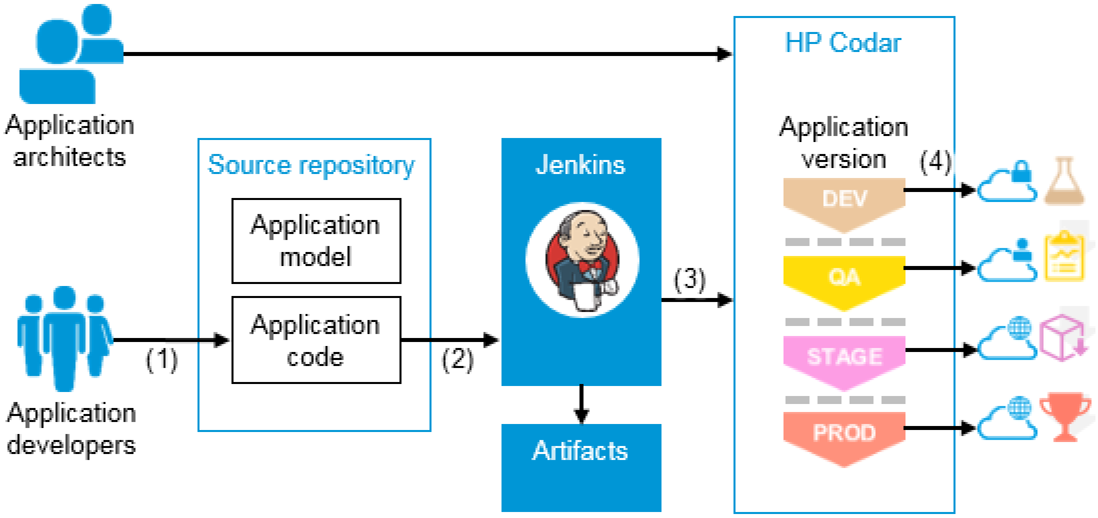


# External integrations

HP Codar is open and extensible, and can be integrated with different build systems such as Jenkins, Hudson, etc. A comprehensive set of REST APIs can be used with other external tools to achieve continuous integration, deployment, and delivery. The HP Codar architecture also provides options for you to hook into customized flows for DevTest and DevOps.

# Jenkins integration

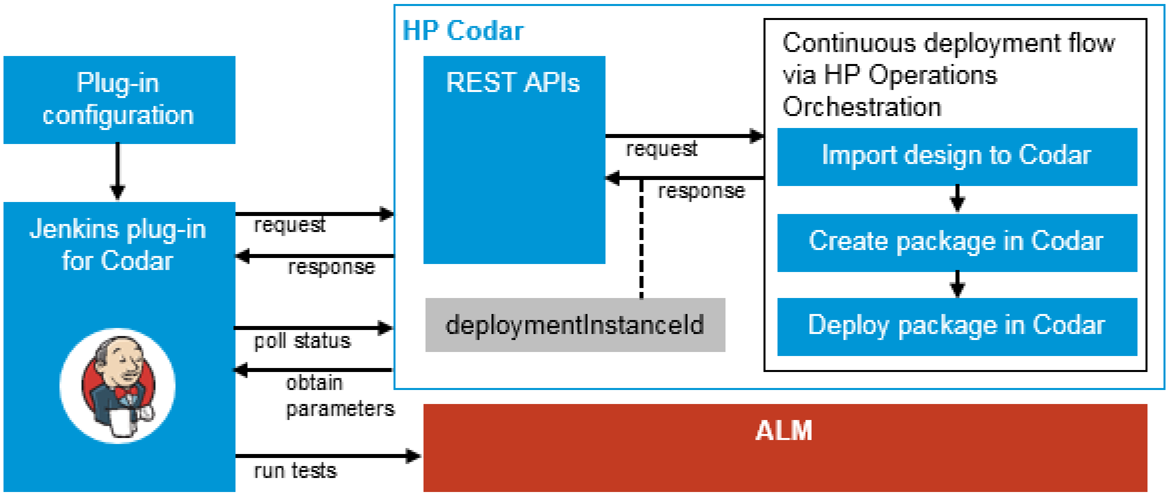
HP Codar includes a Jenkins plug-in for continuous deployment. The following illustration shows how it works.



1. Developers check in changes.
2. Continuous integration triggers build.
3. Jenkins plug-in creates and deploys package.
4. Application is deployed to different environments depending on lifecycle stage.

# ALM integration

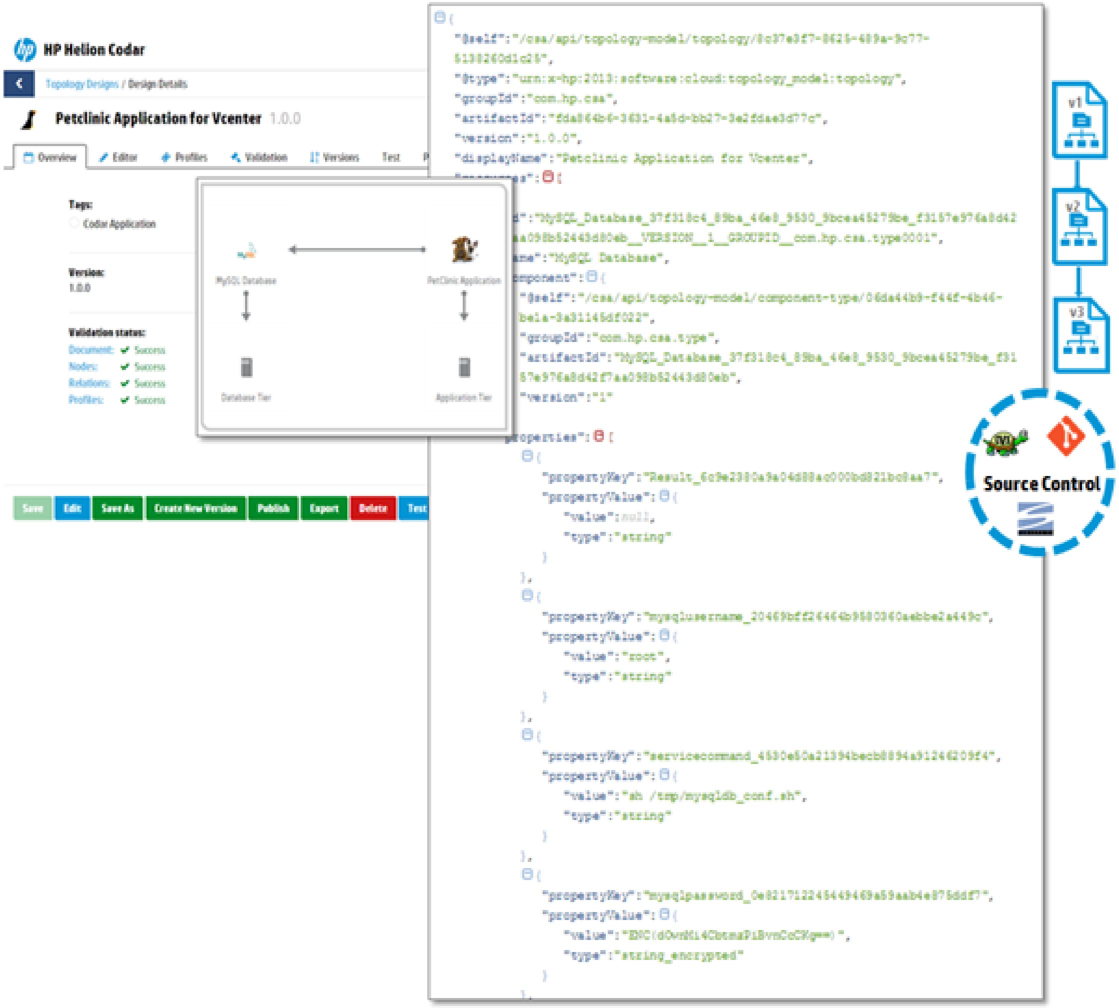
You can integrate Application Lifecycle Management (ALM) with HP Codar to run tests after successful deployment. The following illustration shows how Jenkins acts as an orchestrator.



# Infrastructure as code (IaaC)

Managing infrastructure as code (IaaC) allows IT teams to leverage the best practices for developing code, such as code reviews and unit testing for how infrastructure and applications get provisioned.

HP Codar can manage infrastructure as code. Topology designs that can contain server configurations, networks, volumes, relationships, and application-specific details like the application version and package information can be exported in JSON format and managed with the application in the source control system. Developers can make changes to the model using a text editor and use it for automation. The modified model can also be imported back into HP Codar.



Tags:
Codar Application

Version:
1.0.0

Validation status:
Document: ✓ Success
Nodes: ✓ Success
Relations: ✓ Success
Profiles: ✓ Success

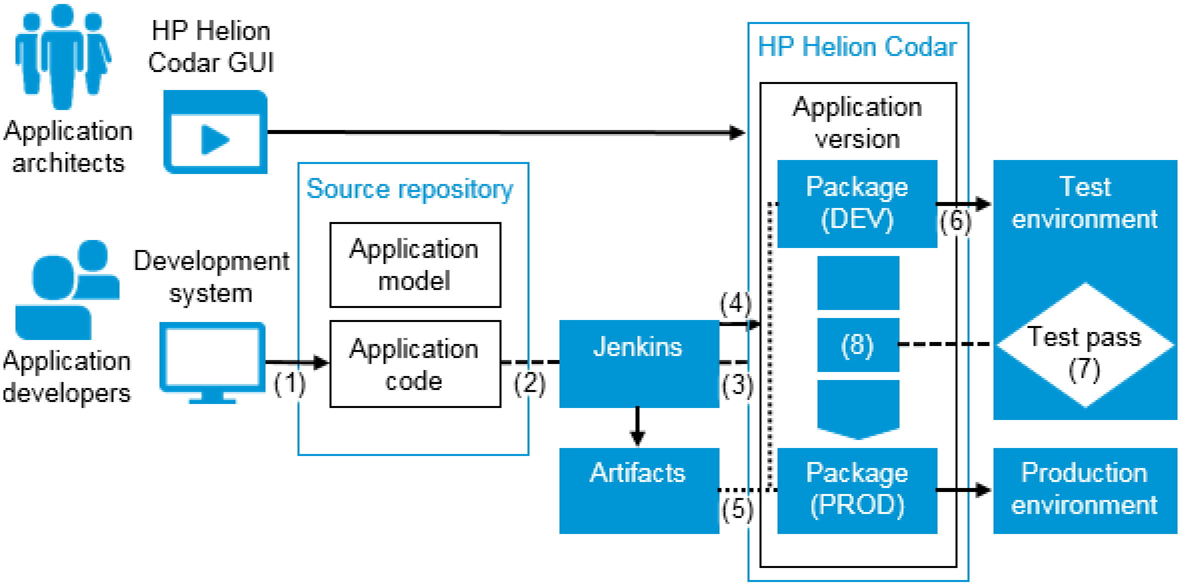
```
{
   "@self":"/csa/api/topology-model/topology/8c37e3f7-8625-489a-9c77-
   5138260d1c25",
   "@type":"urn:x-hp:2013:software:cloud:topology_model:topology",
   "groupId":"com.hp.csa",
   "artifactId":"fda864b6-3631-4a5d-bb27-3e2fdae3d77c",
   "version":"1.0.0",
   "displayName":"Petclinic Application for Vcenter",
   "resources":[
          d":"MySQL_Database_37f318c4_89ba_46e8_9530_9bcea45279be_f3157e976a8d42
          a098b52443d80eb__VERSION__1__GROUPID__com.hp.csa.type0001",
          me":"MySQL Database",
          mponent":{
            "@self":"/csa/api/topology-model/component-type/06da44b9-f44f-4b46-
            be1a-3a31145df022",
            "groupId":"com.hp.csa.type",
            "artifactId":"MySQL_Database_37f318c4_89ba_46e8_9530_9bcea45279be_f31
            57e976a8d42f7aa098b52443d80eb",
            "version":"1"

          roperties":[
            {
               "propertyKey":"Result_6c9e2380a9a04d88ac000bd821bc8aa7",
               "propertyValue":{
                  "value":null,
                  "type":"string"
               }
            },
            {
               "propertyKey":"mysqlusername_20469bff26464b9580360aebbe2a449c",
               "propertyValue":{
                  "value":"root",
                  "type":"string"
               }
            },
            {
               "propertyKey":"servicecommand_4530e50a21394becb8894a91246209f4",
               "propertyValue":{
                  "value":"sh /tmp/mysqldb_conf.sh",
                  "type":"string"
               }
            },
            {
               "propertyKey":"mysqlpassword_0e8217122454494 69a59aab4e875ddf7",
               "propertyValue":{
                  "value":"ENC(dOvnMi4CbtmaPiBvnCcCKg==)",
                  "type":"string_encrypted"
               }
            },
```

# Use case: Continuous integration, deployment, and delivery

The goal is for an application to be enabled for continuous integration (CI) and continuous deployment. An application developer codes the application and an application architect models the application in the HP Codar interface and then exports the application model as code (IaaC). When the application developer checks in the code, a Jenkins build is triggered and the application is deployed using the application model on a specific environment. After the application is deployed, the continuous deployment process is extended to continuous delivery whereas application-specific tests can be automatically run on the deployed instance, with the application possibly being deployed to a different environment dependent on the outcome of the tests.

The following section describes how HP Codar achieves this scenario.



## Application modeling

Application architects model applications graphically by including the necessary components of the design in the designer interface and connecting them via relationships. HP Codar contains a palette of standard components, and components can be imported (embraced) from various deployment engines such as HP Operations Orchestration and Chef. Such designs, called application models, are representations of the methods in which applications are to be deployed. An application model can be exported in JSON format and managed in an external source repository, achieving infrastructure as code (IaaC).

# Continuous integration and deployment

In continuous integration, the code for the sample application and the model for the deployment of the application (in JSON format) is available in a source repository.

When an application developer makes a code change to the application and checks it into the source repository (1), Jenkins triggers a build (2).

HP Codar provides a Jenkins plug-in which has details such as the IP address, user name, and password for HP Codar. It establishes a connection and invokes an API as part of a post-build step (3). The API then invokes a workflow that executes various actions for achieving continuous deployment and continuous delivery.

# Importing an application design

If the application model has not already been imported into HP Codar or if it has changed, the continuous deployment workflow imports it, in JSON format (IaaC), into HP Codar (4) as a new version of the application design. This allows changes that have been made by application developers and architects to be taken into consideration at the time of deployment.

It is important to note that if the application model has already been imported or if there is no change in the application design, then this import operation does not take place and the application version within HP Codar remains the same. You can view the application model in the Topology tile in the designer.

# Managing packages

Packages represent a snapshot of an application design and allow properties to be parameterized within the design. We can also say that the package represents a particular build of an application.

A package is the smallest unit that can be deployed for an application. It represents both the implementation artifacts (the manner in which an application should be deployed) and deployment artifacts(the location of libraries like war, ear, etc., that should be deployed).

Packages are associated with a lifecycle stage. A package can belong to Development, Testing, Staging, or Production stages.

Packages are associated with pipeline management. They can be managed across lifecycle stages, such as promotion or rejection in a given stage. For example, a user with the QA role can reject a package.

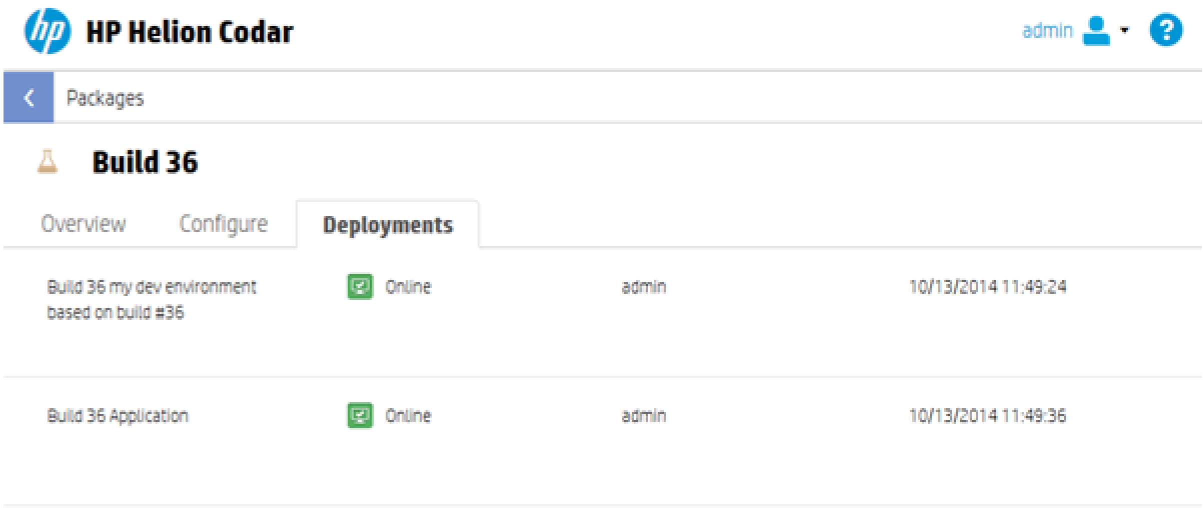
## Tasks

- **Create a package from a specific application version.** An application version can consist of multiple packages.
- **Deploy or redeploy a package.** In this case the corresponding state of an application design along with the properties of the design specified in the package will be fulfilled.
- **Delete a package.** Go to the Release Pipeline tab, hold Ctrl to select multiple packages, and click **Delete**.

  You cannot delete a package that has an instance associated with it.

## Deploying on an environment

After the package is created, the continuous deployment workflow fulfills the application design based on the environment (6). You can view deployments for the package on the Deployments tab:



| | | | |
|---|---|---|---|
| Build 36 my dev environment based on build #36 | Online | admin | 10/13/2014 11:49:24 |
| Build 36 Application | Online | admin | 10/13/2014 11:49:36 |

A runbook automation engine creates an execution plan based on the design that fulfills the infrastructure layer, platform layer, and application layer. You can monitor the status of the deployment of a particular package and view a graphical representation of the deployed application, which includes component-level properties and actions.

## Publishing a design

Publishing a design makes it available as an offering to service consumers. You must have an HP CSA license installed before you can publish a design.

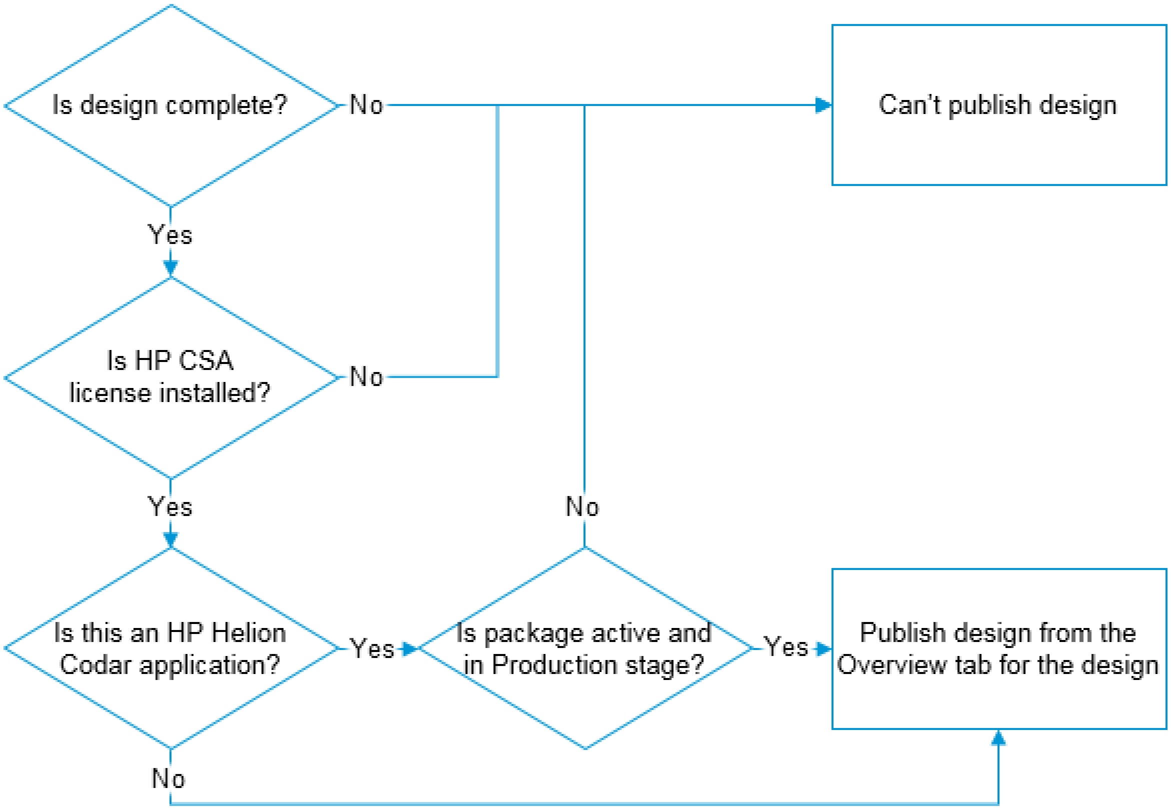
A complete design with an active package in the Production stage contains package-specific properties as part of the design and can be published.

A partial design with an active package in the Production stage contains package-specific properties as part of the design, but it cannot be published until a final composed design is created by deploying the production package.

Publishing a partial design is different depending on which licenses you have installed:

- An HP Codar application design that has been advanced to the Production stage is deployed on a production infrastructure, and then the composed production design is made visible on successful production deployment. The design can then be published to service consumers.
- A design that is not an HP Codar application design must be saved as a composed design from the Test tab. The design can then be published to service consumers.

The following figure illustrates when a design can be published based on the license used:


Is design complete?
No
Can't publish design
Yes
Is HP CSA license installed?
No
Yes
No
Is this an HP Helion Codar application?
Yes
Is package active and in Production stage?
Yes
Publish design from the Overview tab for the design
No

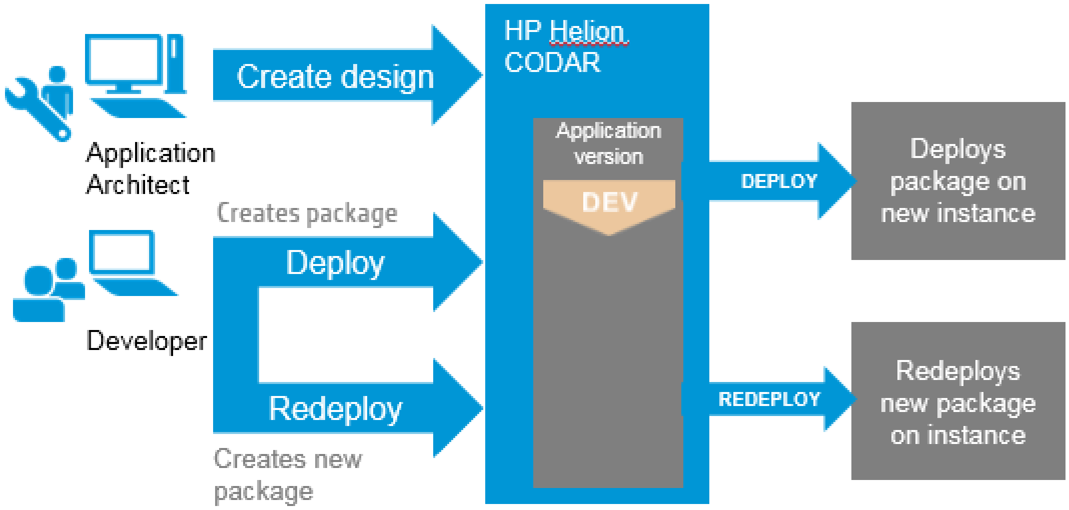
# Use case: Deploy and redeploy packages

The goal is to deploy a package and then use the same instance to redeploy a newer version o the package on an instance that has older versions.

An application architect models the application and marks it for pipeline management. The developer will then create a package and deploy it. A new instance is created when the package is deployed. The deployment is based on the topology design.

After deployment, the same instance can be used to redeploy a newer version of the package. The instance can be upgraded or patched to newer packages or builds.

The following diagram shows the process:



# Use case: Deployment and scale out

The goal is to create a scalable stack and scale out the stack on demand after deployment. An application architect models the application and marks the application for pipeline management. The architect identifies the components that should be scaled out in different life cycle stages. The scalable stack can contain a single component or group of components. During development, the stack can be scaled in by one, and then during testing the stack can be scaled to two, and so on.

# Lifecycle stages and actions

Packages have the following lifecycle stages and corresponding actions:

| Stage | Promote | Deploy, Redeploy | Edit | Delete | Reject |
|---|---|---|---|---|---|
| Development | Yes | Yes | Yes | Yes | No |
| Testing | Yes | Yes | No | Yes | Yes |
| Staging | Yes | Yes | No | Yes | Yes |
| Production | No | Yes | No | Yes | Yes |

> Note: You may not have access to some or all of this functionality, depending on your role.

You can find the lifecycle stages on the Packages tab, and the current stage. The lifecycle stage is also shown for each deployment on the Deployments tab for a package.

Use the following actions to deploy or move a package through the stages:

- **Promote**: Moves the package to the next lifecycle stage. The package state remains Active.
- **Deploy, Redeploy**: Deploys the package.
- **Edit**: Change the properties of a package.
- **Reject**: Stops the package from advancing to another stage. The package will remain in its current stage, its state will be set to Rejected, and the action buttons will no longer be available.
- **Delete**: Delete a package. The package will be removed permanently from the system. A package can only be deleted if all associated deployed instances are canceled and deleted.
- **Refresh**: Retrieves current package status.

# Package states

Packages have the following states:

- Active
- Rejected

If you reject a package, then it remains in its current stage, its state is set to Rejected, and no further actions can be applied; however, it can be delete and will be removed from the system.

When a package is promoted, it moves to the next stage and remains in the active state. Packages are always created in the Development stage. If the HP Codar Jenkins plug-in is configured, then after a successful build the Jenkins plug-in talks to HP Codar and creates a package.

# Grouping service designs by lifecycle stage

A partial design with an active package requires a selection of a service design for provisioning in the deploy package wizard. These service designs can be grouped for different lifecycle stages so that the package deployment in the lifecycle stage will list only those grouped service designs from the respective lifecycle stage.

To group the service designs for a lifecycle stage, create a tag with the name of the lifecycle stage in each topology design. For example, you could create a Development tag and associate it with all required designs in the Development lifecycle state.

Note: The test run wizard in the Test tab will list all designs and will not group by tag.

# Next steps

*HP Codar Installation Guide* and *HP Codar Configuration Guide* explain how to download, install, and configure the software, and *HP Codar API and CLI Reference* provides a brief introduction to the REST APIs and explains how to obtain detailed information for each API. It also explains the command-line interface. You can access online help from the application for task-oriented assistance.

# Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Concepts Guide (Codar 1.50)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to csadocs@hp.com.

We appreciate your feedback!

Java™
COMPATIBLE