

VMware vRealize Code Stream Reference Architecture

vRealize Code Stream 2.1

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<https://docs.vmware.com/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2017 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

	vRealize Code Stream Reference Architecture	5
1	vRealize Code Stream Services	7
2	Integration with External Services	9
	Integration with vRealize Automation Service	10
	Integration with Third-Party Services	10
3	Deployment	11
	Deployment Recommendations	11
4	Execution Scenarios	17
	Index	19

vRealize Code Stream Reference Architecture

The vRealize Code Stream *Reference Architecture* guide describes the structure and configuration of typical vRealize Code Stream deployments.

Intended Audience

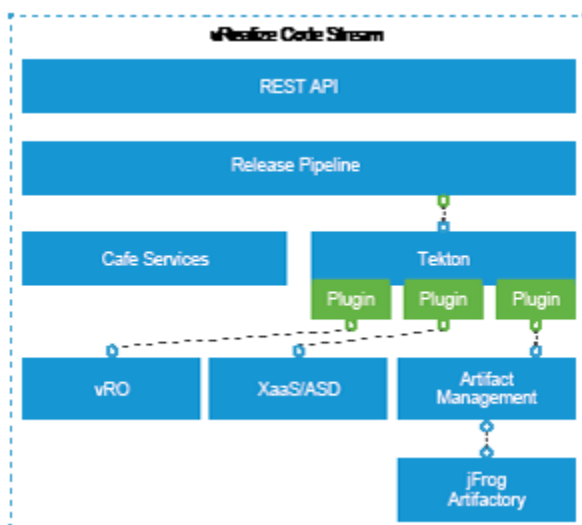
This information is intended for anyone who wants to configure and manage vRealize Code Stream. The information is written for experienced Linux system users and administrators who are familiar with virtual machine technology and data center operations.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to <http://www.vmware.com/support/pubs>.

vRealize Code Stream Services

The vRealize Code Stream system includes services that run on the same host.



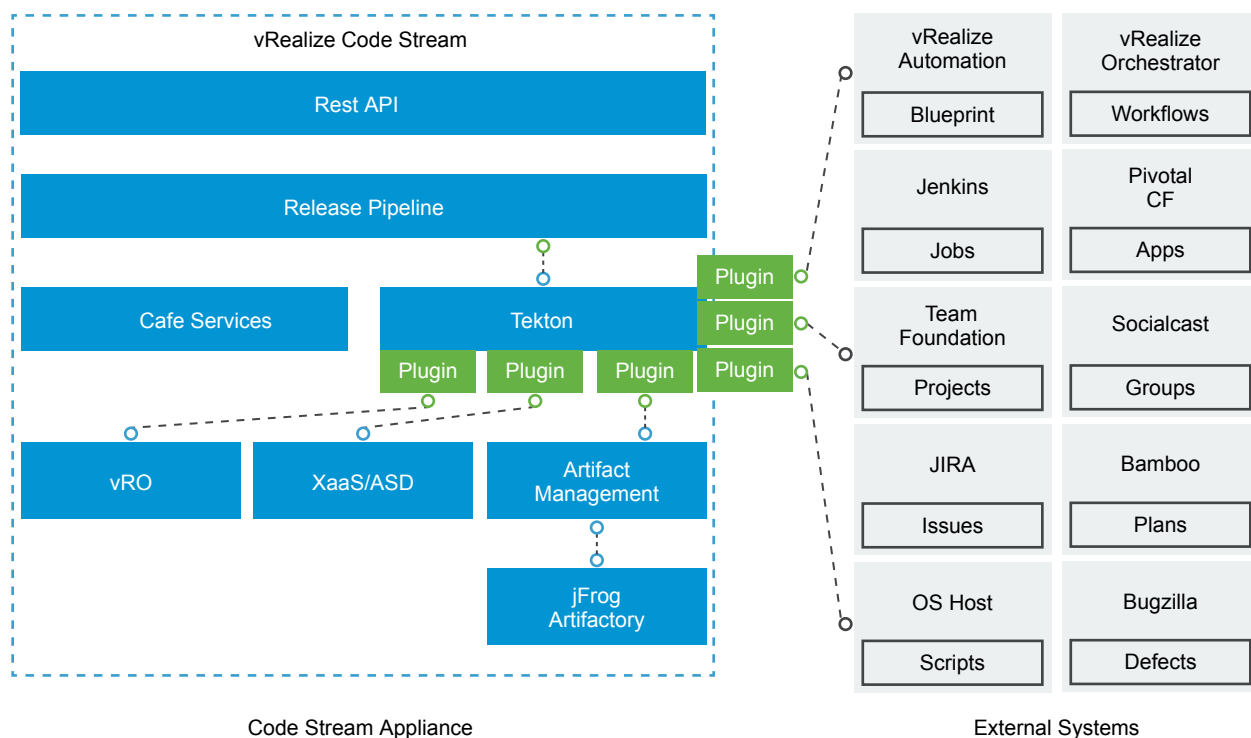
The vRealize Code Stream services comprise of:

Service	Description
Release Management	A vRealize Code Stream Service that provides the capability to model and execute pipelines.
Artifact Management	A vRealize Code Stream Service that provides the capability to locate and download artifacts for deployment. This service uses jFrog Artifactory Service as artifact repository.
vRealize Code Stream plug-in	A vRealize Code Stream Service that provides a plug-in framework to support integrations with other services. Each of the integrations is implemented as a custom plugin. It also provides the capability to orchestrate execution of pipeline tasks.
jFrog Artifactory	An Artifact Repository Service NOTE vRealize Code Stream does not support direct integration with a remote instance of jFrog Artifactory Service. However, the embedded Artifactory instance may be configured to connect with remote Artifactory instances.
vRealize Orchestrator	vRealize Code Stream uses embedded vRealize Orchestrator to execute vRealize Orchestrator Workflows. A pipeline task or gating rule can be configured to run a workflow. A Pipeline Task or Gating Rule can be configured to execute a workflow.

Service	Description
XaaS (Anything as Service)	XaaS and ASD use vRealize Orchestrator. vRealize Code Stream invokes the vRealize Orchestrator workflows used within the release pipeline. Note vRealize Orchestrator does not support integration with a remote instance of XaaS.
Café Platform Services	vRealize Code Stream Services are built using Café plug-in or vRealize Automation Appliance. The platform services natively leverage functionality of many of the common embedded services such as Identity Services, UI Shell services, Notification and Approval (Work Items), and Endpoint Administration.

Integration with External Services

vRealize Code Stream plug-in framework is based on Xenon that supports integrations with other services. Each integration is implemented as a plug-in and a few plug-ins use endpoint definition to connect and interact with the external system.



For more information on Xenon, see <https://vmware.github.io/xenon/>.

This chapter includes the following topics:

- “Integration with vRealize Automation Service,” on page 10
- “Integration with Third-Party Services,” on page 10

Integration with vRealize Automation Service

vRealize Automation is considered as an external system irrespective of the hosted location, which can be local or remote with reference to the vRealize Code Stream Appliance.

The plug-ins connect to a vRealize Automation instance using an endpoint configuration. For more information on how to configure endpoints, see the *Installation and Configuration* guide.

Integration with vRealize Automation enables pipeline tasks to provision machines using Converged Blueprints (7.x) and Single Machine Blueprints (6.2.x).

The following versions of vRealize Automation plug-ins are shipped with vRealize Code Stream 2.0 and above:

Integration	vRealize Code Stream Version	Feature	Connection to vRealize Automation
vRealize Automation 7.x plug-in	2.x	Provision a blueprint using Converged Blueprint	via Endpoint
vRealize Automation 6.2.x plug-in	2.x	Provision a single virtual machine	via Endpoint

Integration with Third-Party Services

vRealize Code Stream offers integrations with Jenkins, Team Foundation Server, Cloud Foundry, Socialcast, JIRA, Bugzilla, and Bamboo.

vRealize Code Stream offers integrations with the following third-party services:

Third-Party Service	vRealize Code Stream Version	Feature
Jenkins	2.x	Execute Jenkins Jobs
Microsoft Team Foundation Server	2.x	Execute Team Foundation Server Projects
Generic OS Host	2.x	Run scripts (Bash or Powershell)
Cloud Foundry	2.1	Manage applications in Pivotal Cloud Foundry server
Socialcast	2.1	Publish messages and pipeline execution events
JIRA	2.1	Issue tracking using JIRA
Bugzilla	2.1	Defect tracking using Bugzilla
Bamboo	2.1	Execute Bamboo Plans

Deployment

vRealize Code Stream can be deployed in a production environment or a lab environment.

Deployment Recommendations

The deployment recommendations in this guide are applicable for vRealize Code Stream 2.0 and above.

All appliances must be in the same time zone with their clocks synchronized.

vRealize Code Stream Appliance

vRealize Code Stream server is packaged with vRealize Automation appliance.

The vRealize Automation appliance is a preconfigured virtual appliance that contains the vRealize Code Stream server. vRealize Automation is delivered as an open virtualization format (OVF) template. The system administrator deploys the virtual appliance to an existing virtualized infrastructure.

The vRealize Code Stream services are installed automatically on any deployment of vRealize Automation appliance.

Deployment Models

You can deploy vRealize Code Stream in both lab or production environments.

- In a lab environment, vRealize Code Stream and vRealize Automation services are deployed on the same appliance. For more information, see [“Deploying vRealize Code Stream for Evaluation,”](#) on page 12.
- In a production environment, vRealize Code Stream and vRealize Automation services are deployed on two separate appliances. For more information, see [“Deploying vRealize Code Stream for Production,”](#) on page 13.

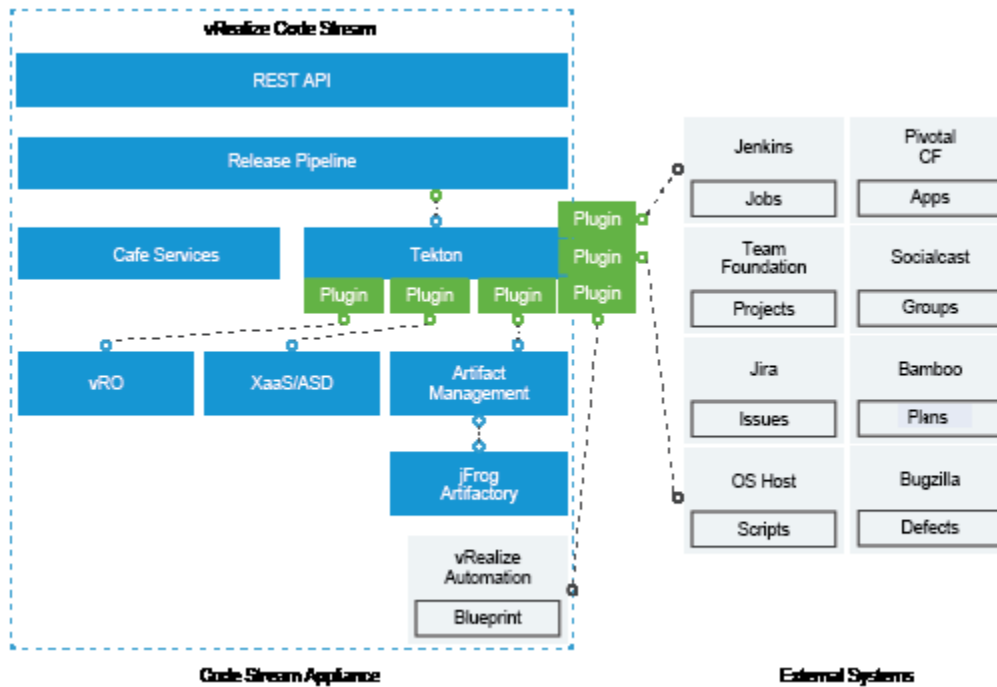
NOTE

- vRealize Code Stream and vRealize Automation must be enabled on two separate appliances for production environments. While you can enable vRealize Code Stream on the same virtual appliance as vRealize Automation in lab or evaluation environments, it is not a supported configuration for production systems, particularly when vRealize Automation is configured in High Availability (HA) mode.
-

Deploying vRealize Code Stream for Evaluation

You can host vRealize Code Stream and vRealize Automation services on the same appliance by deploying vRealize Code Stream for evaluation.

When you deploy vRealize Code Stream for evaluation, you can configure up to 50 pipeline models, execute up to 10 concurrent pipelines, and provision up to 2 concurrent virtual machines.



Requirements

The components configured on the same appliance are:

- vRealize Automation Services

NOTE Setting up a vRealize Business Appliance is optional.

For more information, see the *Reference Architecture for vRealize Automation* guide.

- vRealize Code Stream Services

vRealize Code Stream Services are installed automatically on any deployment of vRealize Automation. The services are enabled only after you apply the vRealize Code Stream license.

For more information on how to apply vRealize Code Stream license, see the *Installation and Configuration* guide.

Although both vRealize Automation and vRealize Code Stream are deployed on the same appliance, the plug-ins connect to vRealize Automation instance using an endpoint configuration. For more information on how to define an endpoint for vRealize Automation plug-ins, see the *Installation and Configuration* guide.

Hardware Specification

The deployment must be configured with the minimal and required hardware specification. For more information on editing OVF settings for a virtual machine, see the *vSphere Virtual Machine Administration* guide.

Server Role	Component	Hardware Specifications
vRealize Code Stream Appliance	vRealize Code Stream Services	CPU: 4 vCPU RAM: 18 GB Disk: 108 GB Network: 1 GB/s

Ports

Users require access to certain ports. All ports listed are default ports.

Server Role	Port
vRealize Code Stream Appliance	443 8444 Port 8444 is required for the Virtual Machine Remote Console.

Administrators require access to certain additional ports, in addition to the ports that users require.

Server Role	Port
vRealize Code Stream Appliance	443 5480 8443 Port 8443 is required for advanced identity management configuration.

The system must support the appropriate inter-application communications.

Server Role	Inbound Ports	Services/System Outbound Ports
vRealize Code Stream Appliance	HTTPS: 443 Identity Management Configuration: 8443 Virtual Machine Remote Console Proxy: 8444 SSH: 22 Virtual Appliance Management Console: 5480	LDAP: 389 LDAPS: 636 VMware ESXi: 902 vRealize Code Stream Appliance: 443 The vRealize Appliance requires access to ESXi host Port 902 to proxy console data to the user.

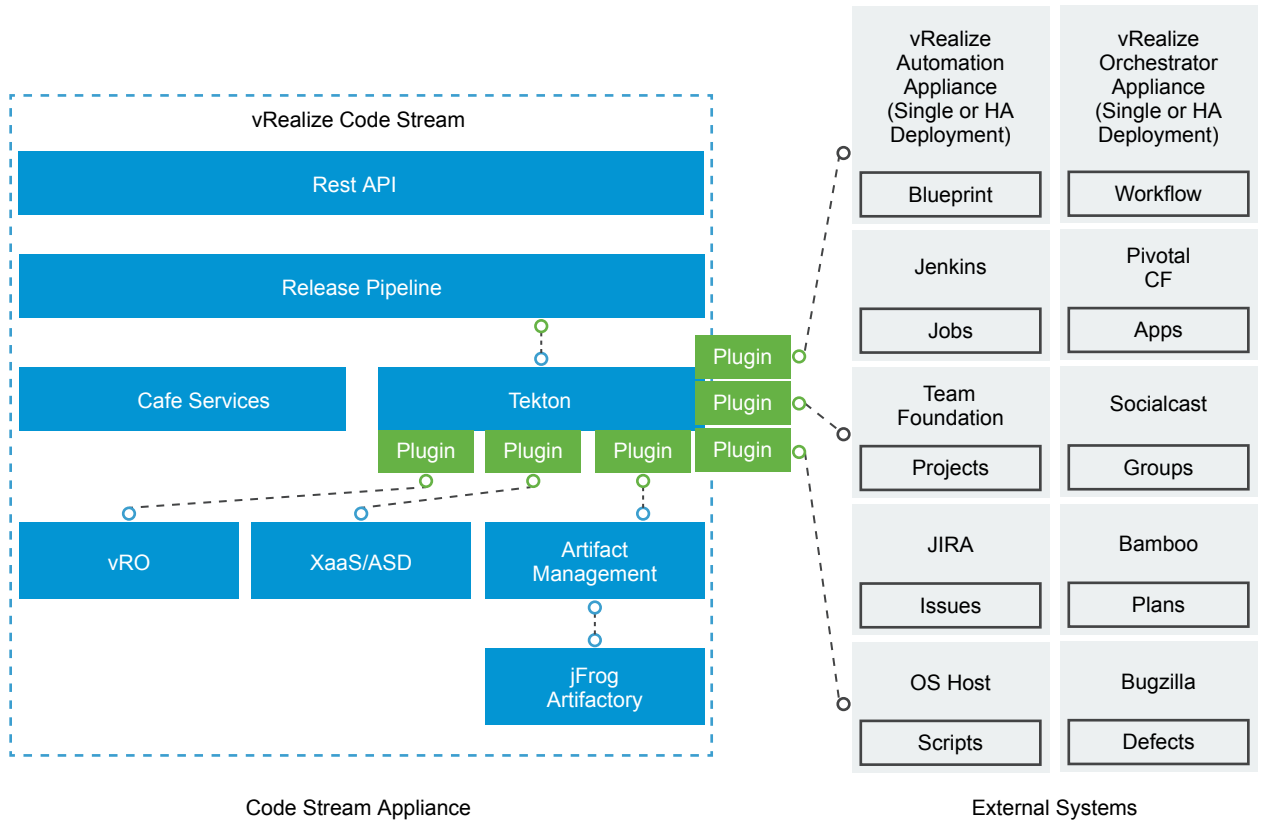
Deploying vRealize Code Stream for Production

In a production environment, you must deploy vRealize Code Stream on a dedicated appliance.

vRealize Code Stream license should not be applied on a production vRealize Automation appliance.

When you deploy vRealize Code Stream for production, you can configure up to 1000 pipeline models and execute up to 50 concurrent pipelines.

NOTE You must host vRealize Automation Services on a different appliance. For more information on deployment of vRealize Automation, see *Reference Architecture for vRealize Automation 7.0* guide.



Requirements

Deploying vRealize Code Stream requires the following dedicated appliances:

- **vRealize Code Stream Appliance**
A standalone appliance dedicated for vRealize Code Stream services.
- **vRealize Automation Appliance**
The vRealize Automation deployment can be a single appliance or a load balanced cluster.
The vRealize Automation appliance should be configured if you want to use vRealize Automation services for provisioning blueprints.
For more information on deployment topologies, see the *Reference Architecture for vRealize Automation* guide.

Hardware Specification

The deployment must be configured with the minimal and required hardware specification. For more information on editing OVF settings for a virtual machine, see the *vSphere Virtual Machine Administration* guide.

Server Role	Component	Required Hardware Specifications
vRealize Code Stream Appliance	vRealize Code Stream Services	CPU: 4 vCPU RAM: 18 GB Disk: 108 GB Network: 1 GB/s

Ports

Users require access to certain ports. All ports listed are default ports.

Server Role	Port
vRealize Code Stream Appliance	443 8444 Port 8444 is required for the Virtual Machine Remote Console.

Administrator requires access to certain ports, in addition to the ports that users require.

Server Role	Port
vRealize Code Stream Appliance	5480 8443 Port 8443 is required for advanced identity management configuration.

The system must support the appropriate inter-application communications.

Server Role	Inbound Ports	Services/System Outbound Ports
vRealize Code Stream Appliance	HTTPS: 443 Identity Management Configuration: 8443 Virtual Machine Remote Console Proxy: 8444 SSH: 22 Virtual Appliance Management Console: 5480	LDAP: 389 LDAPS: 636 VMware ESXi: 902 vRealize Code Stream Appliance: 443 The vRealize Appliance requires access to ESXi host Port 902 to proxy console data to the user.

Deploying vRealize Code Stream Management Pack

You can deploy vRealize Code Stream Management Pack when you have two vRealize Automation appliances in a production environment.

The vRealize Code Stream Management Pack requires two vRealize Automation appliances. vRealize Automation, vRealize Code Stream, vRealize Orchestrator, and Artifactory provided with vRealize Code Stream are hosted on a single appliance. The second appliance hosts the content storage and the persistent storage.

Requirements

To deploy vRealize Code Stream Management Pack for IT DevOps 2.1, you require:

- vRealize Automation 7.x appliance with 4 CPU and 20-Gb RAM, which is available with vRealize Code Stream 2.x on the same appliance.
- Content Server

For production systems, you can deploy the content server on a secondary vRealize Automation appliance that has vRealize Automation and vRealize Orchestrator services disabled. The content server exports and imports software-defined content in the form of binary files between external systems and the JFrog Artifactory artifact repository.

For non-production systems, you can deploy the content server on the same appliance as vRealize Automation. However, you must increase the memory to manage the extra load and avoid any performance impact. For optimum results, install a secondary server with an external content server host.

- vRealize Automation Advanced (or equivalent) license
- vRealize Code Stream license
- (optional) SMTP server

Note Increase the RAM size on the vRealize Automation appliance from 18 Gb to 32 Gb if content server and the vRealize Automation appliance are deployed on a single appliance.

Ports

The system must support the appropriate inter-application communications.

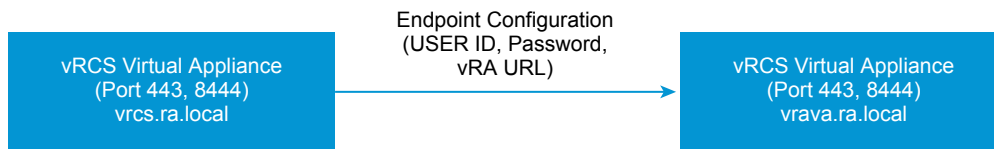
vRealize Orchestrator

Server Role	Inbound Ports	Outbound Ports
vRealize Automation and vRealize Code Stream appliance	HTTPS: 443	HTTPS: 443 and 8281 SSH: 22
Content Server (Second vRealize Automation)	SSH: 22	HTTPS: 443 and 8281 SSH: 22 The ports are applicable to out-of-box Package Types.
External test workflow server running vRealize Orchestrator	HTTPS: 443 and 8281	HTTPS: 443 and 8281 The ports are applicable to out-of-box Package Types. Additional ports might be required for custom test workflow.

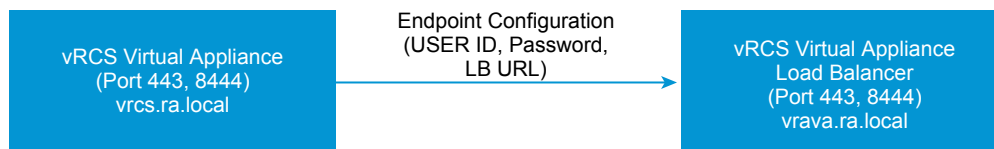
Integrating vRealize Code Stream with vRealize Automation

vRealize Code Stream uses vRealize Automation to provision virtual machines.

For vRealize Code Stream to communicate with vRealize Automation while provisioning a virtual machine, vRealize Code Stream requires the endpoint of vRealize Automation.



vRealize Automation appliance might require a full scale high availability deployment. The load balancer configured for vRealize Automation should be used as external endpoint in vRealize Code Stream. All the VM provisioning requests in the vRealize Code Stream provisioning task is routed through vRealize Automation load balancer.



NOTE

- Identity Federation support between the two appliances is not provided.
- vRealize Code Stream does not support connecting to vRealize Orchestrator, XaaS services on the vRealize Automation appliance. It uses the services embedded on its appliance.

Execution Scenarios

The supporting infrastructure based on tests conducted in a lab environment with different configurations is provided for reference. Based on your setup, you can modify the settings in the config files.

The tests conducted are for different pipelines executed 100 times with 5 stages in each pipeline and 5 tasks in each stage.

The execution scenarios are:

Table 4-1. For 30 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
No. of Pipeline Templates: 100	4 vCPU/18 GB	Defaults (out-of-the box)
Executed Pipelines: 5000		

Table 4-2. For 100 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
No. of Pipeline Templates: 200	6 vCPU/20 GB	<ul style="list-style-type: none"> ■ jvm memory for Tekton: 5 GB ■ jvm memory for vcac instance: 8 GB ■ jvm memory for vco instance: 2.5 GB ■ vcac Application db pool: 100 ■ connectionTimeout for vcac instance: 90 sec ■ connectionTimeout for vco instance: 40 sec ■ vco application thread pool size: 300 ■ postgres max_connections: 500
Executed Pipelines: 10000		

Table 4-3. For 125 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
No. of Pipeline Templates: 200	8 vCPU/25 GB	<ul style="list-style-type: none"> ■ jvm memory for Tekton: 6 GB ■ jvm memory for vcac instance: 8 GB ■ jvm memory for vco instance: 4 GB ■ vcac Application db pool: 200 ■ vcac Application thread pool: 600 ■ connectionTimeout for vcac instance: 40 sec ■ connectionTimeout for vco instance: 40 sec ■ vco application thread pool size: 300 ■ postgres max_connections: 500
Executed Pipelines: 10000		

Index

D

- deploy for evaluation **12**
- deployment recommendations **11**
- deployment topology **11**

E

- external service **9**

I

- integrating with vRealize Automation **16**
- integration **10**
- intended audience **5**

M

- management pack deployment **15**

P

- production environment **13**

S

- sizing **17**
- system architecture **7**

T

- third-party integrations **10**

V

- vRealize Code Stream Appliance **11**

