

VMware vRealize Code Stream Reference Architecture

12 APRIL 2018

vRealize Code Stream 2.4



vmware®

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

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

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

docfeedback@vmware.com

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

Copyright © 2018 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

	vRealize Code Stream Reference Architecture	4
1	vRealize Code Stream Services	5
2	Integration With External Services	6
	Integration with the vRealize Automation Service	6
	Integrations with Third-Party Services	7
3	Deploying vRealize Code Stream	8
	Deployment Recommendations	8
4	Execution Scenarios	14

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

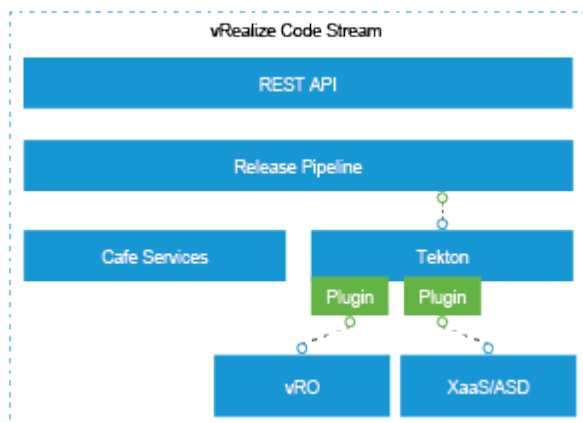
This information is intended for anyone who wants to configure and manage vRealize Code Stream. The information is written for Linux users and administrators who are experienced and familiar with virtual machine technology and data center operations.

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

vRealize Code Stream includes services that run on a single host.



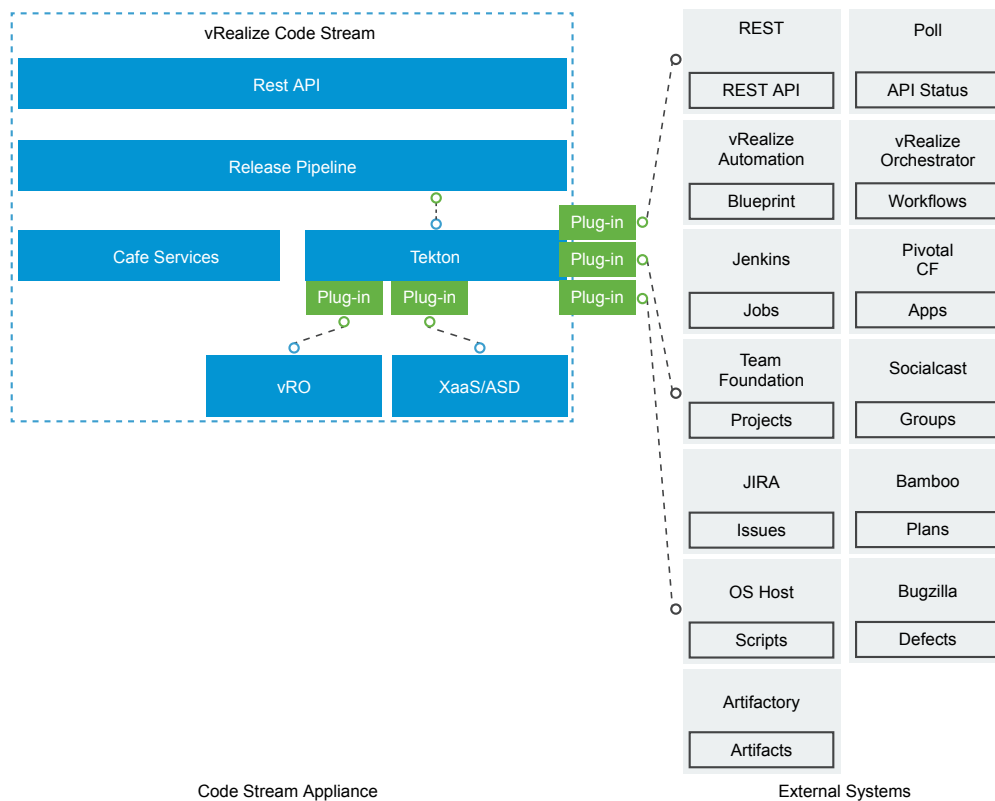
vRealize Code Stream is comprised of the following services:

Service	Description
Release Management	A vRealize Code Stream service that provides the capability to model and run pipelines.
vRealize Code Stream plug-in	A vRealize Code Stream service that provides a plug-in framework to support integrations with other services. Each integration is implemented as a custom plug-in. This plug-in provides the capability to orchestrate execution of the pipeline tasks.
vRealize Orchestrator	vRealize Code Stream uses the embedded vRealize Orchestrator to run vRealize Orchestrator workflows. You can configure a pipeline task or gating rule to run a workflow.
XaaS (Anything as Service)	XaaS and ASD use vRealize Orchestrator. vRealize Code Stream invokes the vRealize Orchestrator workflows used in 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 the Café plug-in or vRealize Automation Appliance. The platform services natively use the functionality of many common embedded services such as Identity Services, UI Shell services, Notification and Approval (Work Items), and Endpoint Administration.

Integration With External Services

2

The vRealize Code Stream plug-in framework is based on Xenon, which supports integrations with other services. Each integration is implemented as a plug-in, which uses an endpoint definition to connect to external systems for interaction.



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

This chapter includes the following topics:

- [Integration with the vRealize Automation Service](#)
- [Integrations with Third-Party Services](#)

Integration with the vRealize Automation Service

vRealize Automation is considered to be an external system, regardless of its hosted location. The hosted location 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.

Depending on the version of vRealize Automation, the integration with vRealize Code Stream enables pipeline tasks to provision machines in specific ways.

vRealize Code Stream 2.0 and later includes the following versions of vRealize Automation plug-ins:

vRealize Automation Integration	vRealize Code Stream Version	Provisioning Feature	Connection to vRealize Automation
7.x plug-in	2.x	Provision a machine using Composite Blueprints	through Endpoint
6.2.x plug-in	2.x	Provision a single virtual machine	through Endpoint

Integrations with Third-Party Services

vRealize Code Stream offers integrations with Jenkins, Team Foundation Server, Cloud Foundry, Socialcast, JIRA, Bugzilla, and Bamboo. You can also use an external Artifactory repository.

vRealize Code Stream offers integrations with the following third-party services as of the releases shown:

Third-Party Service	vRealize Code Stream Version	Feature
Applications that use a REST API	2.3	REST Plug-in to invoke a REST API
jFrog Artifactory	2.2	External Artifactory repository
Jenkins	2.x	Run Jenkins Jobs
Microsoft Team Foundation Server	2.x	Run Team Foundation Server Projects
Generic OS Host	2.x	Run scripts (Bash or PowerShell)
Cloud Foundry	2.1	Manage applications in the 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	Run Bamboo Plans

Deploying vRealize Code Stream

3

You can deploy vRealize Code Stream in a production environment or a lab environment.

Deployment Recommendations

The following deployment recommendations apply to vRealize Code Stream 2.0 and later.

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

vRealize Code Stream Appliance

The vRealize Code Stream server is packaged with the 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 and production environments.

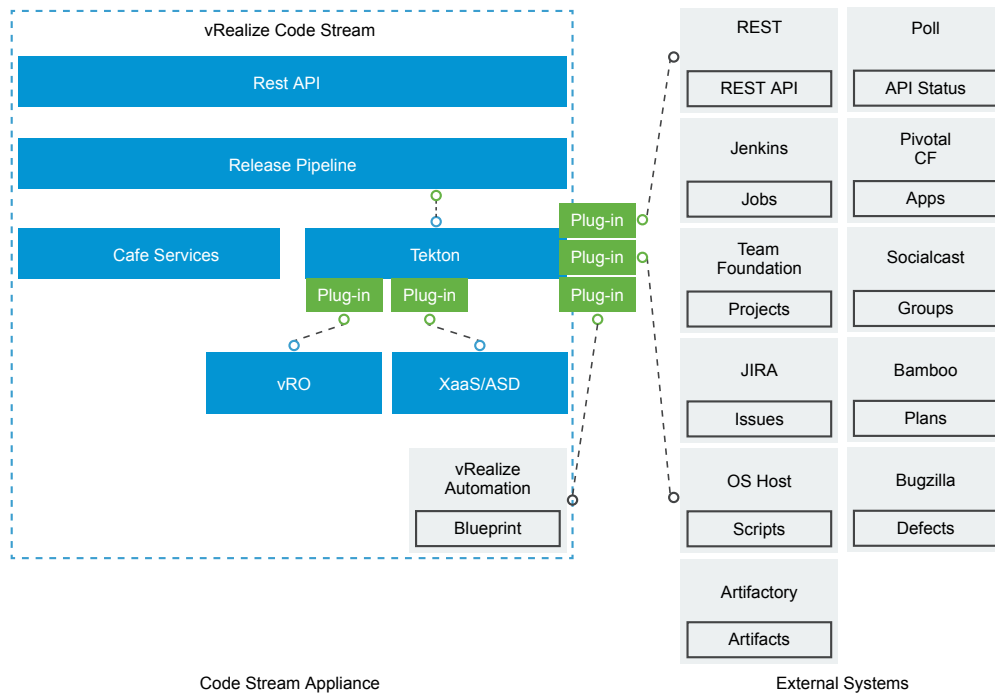
- In a lab environment, vRealize Code Stream and vRealize Automation services are deployed on the same appliance. For more information, see [Deploying vRealize Code Stream for Evaluation](#).
- In a production environment, vRealize Code Stream and vRealize Automation services are deployed on two separate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

Note In production environments, vRealize Code Stream and vRealize Automation must be enabled on two separate appliances. Although in lab or evaluation environments, you can enable vRealize Code Stream on the same virtual appliance as vRealize Automation, it is not a supported configuration for production environments, 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 a maximum of 50 pipeline models, run a maximum of 10 concurrent pipelines, and provision a maximum of 2 concurrent virtual machines.



Requirements

The vRealize Automation services and the vRealize Code Stream services are configured on the single appliance.

- 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

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

For more information on how to apply the 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 the vRealize Automation instance through 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 about editing the 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 Required

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 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

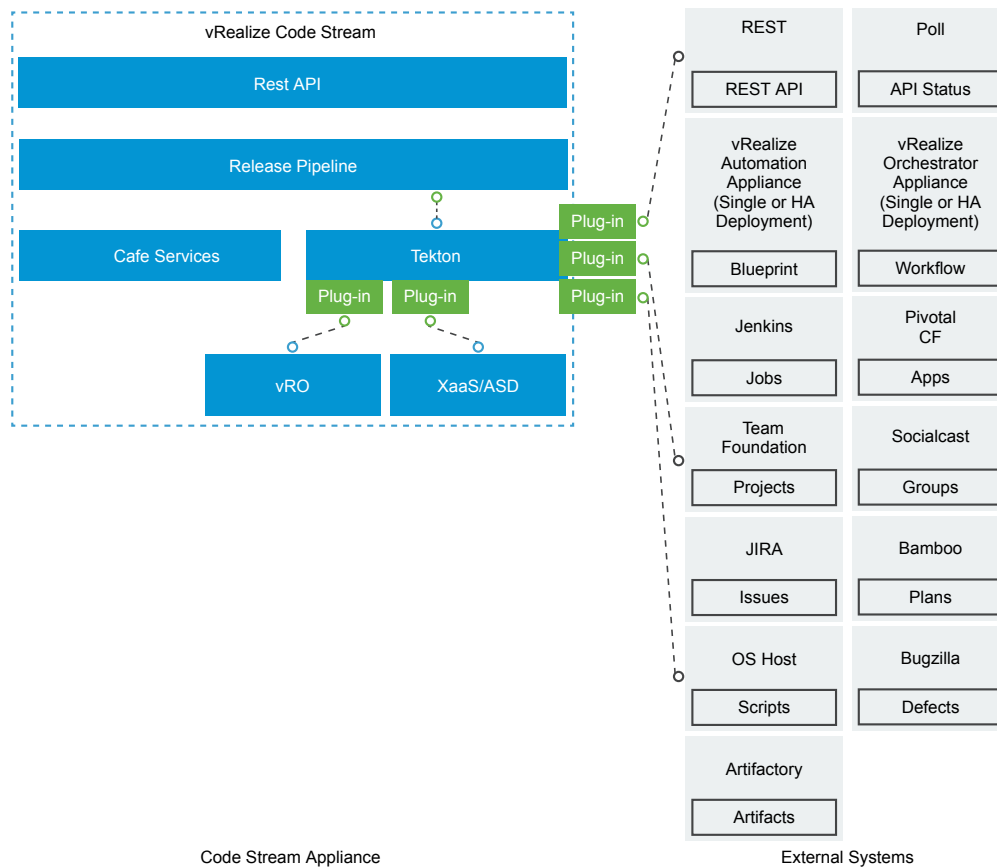
In a production environment, you must deploy vRealize Code Stream on a dedicated appliance.

Do not apply the vRealize Code Stream license on a production vRealize Automation appliance.

When you deploy vRealize Code Stream for production, you can configure a maximum of 1000 pipeline models and run a maximum of 50 concurrent pipelines.

With 6 GB for processing, 8 vCPU, and 32 GB of RAM at the system level, vRealize Code Stream can support a maximum of 250 to 300 concurrent pipeline executions.

Note You must host the vRealize Automation services on a different appliance. For more information about the deployment of vRealize Automation, see the *Reference Architecture for vRealize Automation* 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.

To use vRealize Automation services for provisioning blueprints, you must configure the vRealize Automation appliance.

For more information about 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 about editing the 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 Required

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 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.

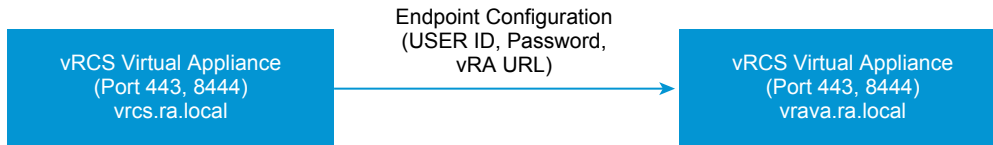
The system must support the appropriate inter-application communications.

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.

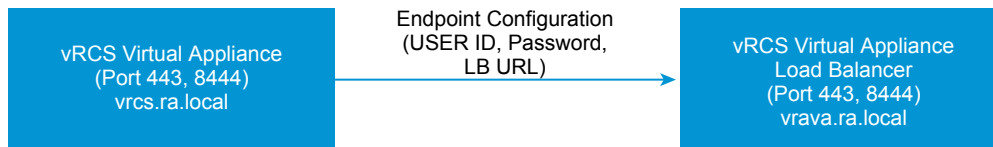
Integrating vRealize Code Stream with vRealize Automation

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 that is configured for vRealize Automation must be used as an external endpoint in vRealize Code Stream. All the VM provisioning requests in the vRealize Code Stream provisioning task are routed through the 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.
-

4

Execution Scenarios

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

The tests conducted are for different pipelines that ran 100 times, with five stages in each pipeline and 5 tasks in each stage.

Table 4-1. Execution Scenario for 30 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
Number of Pipeline Templates: 100 Pipelines Run: 5000	4 vCPU/18 GB	Defaults (out-of-the box)

Table 4-2. For 100 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
Number of Pipeline Templates: 200 Pipelines Run: 10000	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

Table 4-3. For 125 Concurrent Executions

Pipeline Details	Host vCPU/Memory	Parameters
Number of Pipeline Templates: 200 Pipelines Run: 10000	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