



VMware vRealize Code Stream 2.1 Release Notes

Updated on 02 September 2016

vRealize Code Stream 2.1 | 23 August 2016 | Build 4270058

Check regularly for additions and updates to these release notes.

What's in the Release Notes

The release notes cover the following topics:

- [What's New](#)
- [License Key](#)
- [Documentation](#)
- [Support Information](#)
- [Do not enable Code Stream on an appliance that runs in High Availability mode](#)
- [Resolved Issues](#)
- [Known Issues](#)

What's New

vRealize Code Stream 2.1 offers several new out-of the-box integrations with PaaS (Cloud Foundry), Issue tracking (Atlassian JIRA, Bugzilla), Build & CI (Atlassian Bamboo) and Collaboration (VMware Socialcast) systems in the context of release automation. A brand new Dashboard provides Dev, QE, Release and Operations teams with configurable views to track the specific build and release information they are interested in. Furthermore, the pipeline modeling experience has been significantly enhanced with new capabilities such as nested pipelines, an overall faster and more intuitive user interface, and simpler invocation of scripts and vRealize Orchestrator workflows.

New out-of-the box Integrations

- **Cloud Foundry**

Cloud Foundry is an open source cloud platform as a service (PaaS) that enables deploying and running applications at scale. The Cloud Foundry integration works with its commercial implementations including Pivotal Cloud Foundry. vRealize Code Stream 2.1 supports Cloud Foundry version 255 (Pivotal Cloud Foundry version 1.6) and later. The integration provides the following capabilities:

- Manage Cloud Foundry server endpoints.
- Deploy application using either manifest or ad-hoc configuration.
- Start or stop a deployed application.
- Delete a deployed application.
- Bind or unbind services to an application.
- Scale up or down a deployed application.

- **Bamboo**

Atlassian Bamboo is a continuous integration and deployment tool. The Bamboo integration works across commercial and free versions of the Bamboo server. vRealize Code Stream 2.1 supports Bamboo server version 5.9 and later. The integration provides the following capabilities:

- Manage Bamboo server endpoints.
- Trigger build plans on Bamboo server.

- **JIRA**

Atlassian JIRA is an issue tracking tool. vRealize Code Stream 2.1 supports JIRA server version 6.3 and later. The JIRA integration provides the following capabilities:

- Manage JIRA server endpoints.
- Create an issue for any issue type including the user-defined types.

- Update an issue with comments, mark as done, and reassign.

- **Bugzilla**

Bugzilla is an open-source defect tracking system. vRealize Code Stream 2.1 supports Bugzilla version 5.0.1 and later. The Bugzilla integration supports the following capabilities:

- Create a defect with user-defined values for mandatory and optional parameters.
- Update a defect with comments and defect state, and reassign.

- **Socialcast**

Socialcast is a VMware enterprise collaboration platform. The Socialcast integration provides the capability of posting pipeline execution updates such as execution start, and failures to a user configured socialcast group.

Brand New Dashboard

vRealize Code Stream 2.1 introduces a brand new dashboard that gives a bird's eye view of specific pipeline executions and is customized on a user basis and shareable with other users within the same tenant.

The dashboard provides the following capabilities:

- Create custom dashboards for specific pipeline executions along with pertinent pipeline parameters.
- Monitor status of all task executions across the stages and concurrent executions of a pipeline in a compact format.
- Monitor status of specific task execution within the pipeline execution.
- Add columns representing important pipeline parameters such as Build-id, Commit-Id, and Triggered-by to the dashboard.
- Publish dashboards to other users in the tenant.

Other Enhancements

The other modeling enhancements are:

- **Nested Pipelines:** Model pipeline templates to trigger other pipelines during execution. Nested pipelines enable modeling more complex and re-useable pipeline logic such as roll-backs, and deployment setup.
- **Inline Scripts:** In addition to executing scripts on remote hosts, inline scripts (bash/PowerShell) can be specified as part of script task configuration. Inline scripts can use and dynamically consume pipeline parameters at runtime.
- **Endpoint Management:** Endpoint configuration for various integrations to external systems is now available within the **Code Stream** tab.
- **External vRealize Orchestrator:** In addition to the embedded instance of vRealize Orchestrator, pipeline templates can include tasks that invoke workflows on remote instances of vRealize Orchestrator.

License Key

You must upgrade your license key to use the latest version of the vRealize Code Stream. For more information, see [Licensing Help Center](#).

Documentation

vRealize Code Stream 2.1 includes the following product documentation.

- [Installation and Configuration](#)
- [Upgrading vRealize Code Stream](#)
- [Using vRealize Code Stream](#)
- [Reference Architecture](#)

Support Information

- Applying a vRealize Code Stream license from the vRealize Automation CLI is supported. Run the command, `/usr/sbin/vcac-vami license-update --key CodeStreamLicenseKey` in the vRealize Automation CLI.
- vRealize Code Stream supports the Internet Explorer 10.0 or later, Firefox, and Chrome Web browsers.

Do not enable Code Stream on an appliance that runs in High Availability mode

Although vRealize Code Stream can be enabled on the same virtual appliance as vRealize Automation in lab or evaluation environments, it is not a recommended nor supported configuration for production systems, particularly when vRealize Automation is configured in High Availability (HA) mode. The current version of vRealize Code Stream does not support HA configuration, and if enabled on a vRealize Automation system in HA mode, can leave the overall system in an unpredictable state. For more information, see [KB 2145084](#).

Resolved Issues

The resolved issues in the vRealize Code Stream 2.1 release:

- The Jenkins task experiences an issue when binding its input properties to JSON output properties from other tasks like vRealize Automation provisioning task. The job parameter, which is a string input property does not receive a valid value when bound to JSON output property.
- When you attempt to create user or groups with names containing upper case characters, the **Approval** button gets disabled during the pipeline lifecycle.
- When you resolve an artifact using the Artifactory REST API, the API returns the HTTP path to the object instead of the HTTPS path.

Known Issues

The known issues are grouped as follows.

- [Upgrade](#)
- [Artifact Management](#)
- [Powershell Task](#)

Upgrade

- **vRealize Code Stream roles are not reassigned to users after you migrate identity store from vRealize Automation 6.2.x and later to 7.1**

Workaround: Reassign the roles manually for Release Manager and Release Engineer after you upgrade vRealize Automation 6.2.x and later to 7.1.

Artifact Management

- **Attempts to configure an Artifactory server fail with an error message**

When you configure the Artifactory server using an IP address, an error message Unable to register the Artifactory server because the server is unresponsive. Contact your system administrator is displayed.

Workaround: Configure the Artifactory server using a host name instead of the IP address.

Powershell Task

- **PowerShell task fails when concurrent operations for a user exceed the recommended value**

A few pipelines fail when you execute concurrent pipelines exceeding the maximum number. This issue occurs when the pipeline consists of powershell task. An error The WS-Management service cannot process the request. The maximum number of concurrent operations for this user has been exceeded. Close existing operations for this user, or raise the quota for this user is displayed.

Workaround: Perform the following steps to resolve the issue:

1. Modify winrm service configuration to set a higher value for MaxShellsPerUser. The recommended value is 500 for 50 concurrent pipelines with 5 powershell tasks for each pipeline.
2. Run the command `winrm set winrm/config/winrs '@{MaxShellsPerUser="500"}'`.
3. Verify the configuration by running the command `winrm get winrm/config`.
4. Restart the winrm service.

- **Powershell task might fail to respond due to memory constraint on winrm service**

When you execute concurrent pipelines that contain one or more powershell tasks, some pipelines might fail during execution.

Workaround: Perform the following steps to resolve this issue:

1. Modify winrm service configuration to set a higher value for MaxMemoryPerShellMB. The recommended value is 2048.
2. Run the command `winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="2048"}'`
3. Verify the configuration by running the command `winrm get winrm/config`.
4. Restart the winrm service.