

Getting Started with vRealize Automation Code Stream

February 2022

vRealize Automation 8.7

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

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

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

Copyright © 2022 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

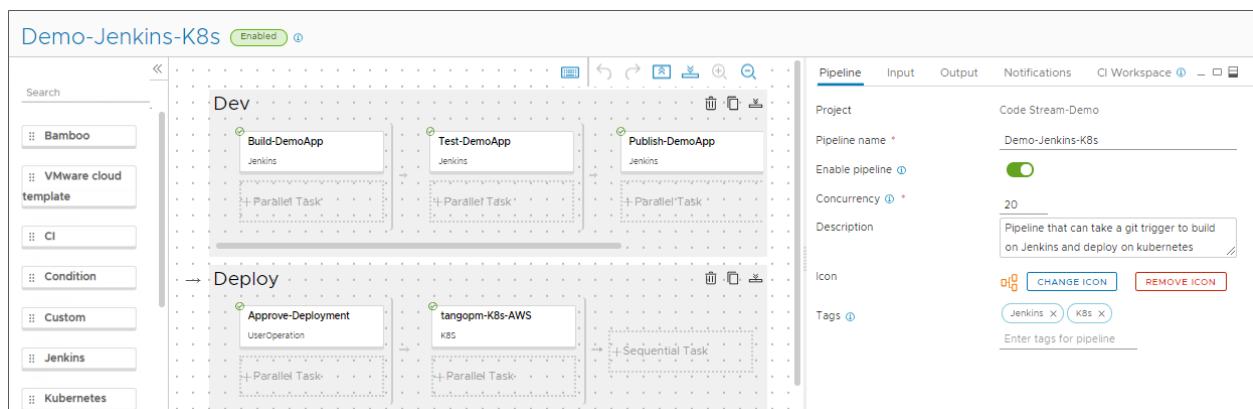
- 1** What Is Code Stream 4
- 2** What Does Code Stream Do 5
- 3** Before You Begin 8
- 4** How Do I Set It Up 10
- 5** What Else Can I Do With It 14

What Is Code Stream

1

vRealize Automation Code Stream™ is continuous integration and continuous delivery (CI/CD) software that delivers software rapidly and reliably, with little overhead. Code Stream supports deploying monolithic legacy applications, and uses Docker containers and Kubernetes containers running on multiple clouds.

With Code Stream, you create pipelines that automate your entire DevOps lifecycle while using existing development tools such as Git and Jenkins.



Code Stream simplifies the ability to build, test, and deploy your applications, and increases your productivity as you release source code from the development repository, through testing, to production. Code Stream supports custom and commercial applications, and objects such as VMware Cloud Templates.

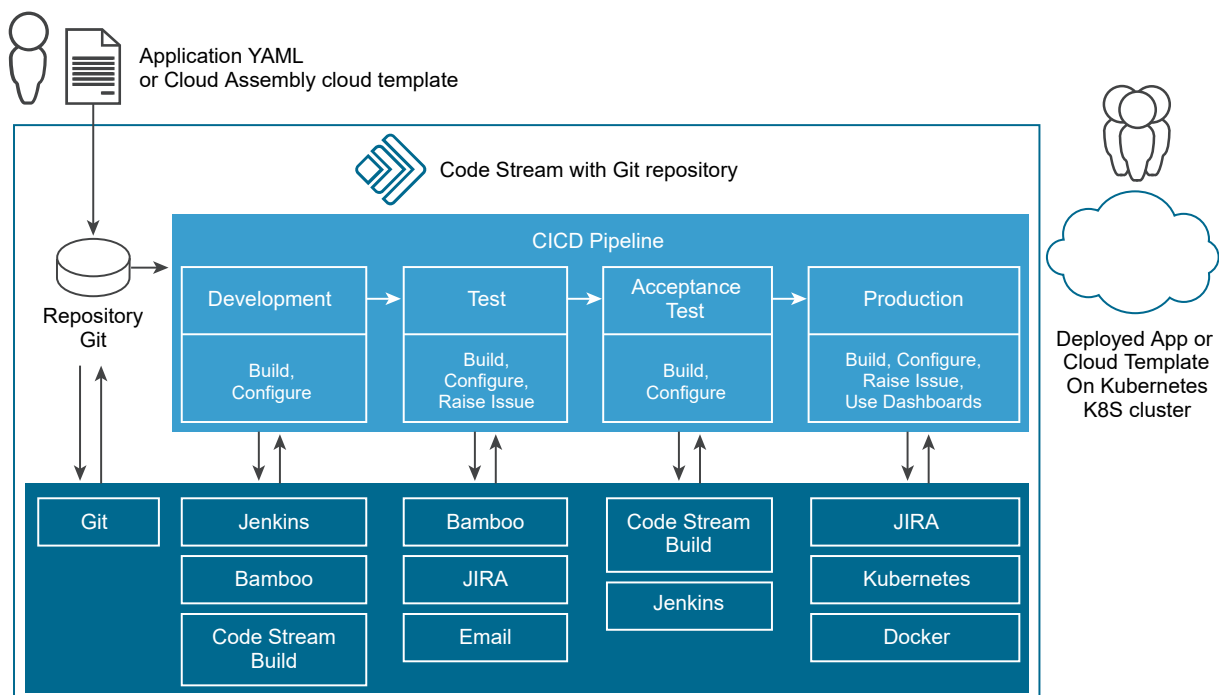
Starting with vRealize Automation 8.2, Blueprints are called VMware Cloud Templates.

What Does Code Stream Do

2

Code Stream models the tasks in your software release process, automates the development and test of developer code, and releases it to your production environment.

It integrates your release process with developer tools to accomplish specific tasks, and tracks all code artifacts and versions.



You create a pipeline that runs actions to build, deploy, test, and release your software. Code Stream runs your software through each stage of the pipeline until it is ready to be released to production.

You integrate your pipeline with one or more DevOps tools such as data sources, repositories, or notification systems, which provide data for the pipeline to run.

For example, you might need to use several endpoints in a pipeline that downloads and deploys a Kubernetes application from GitHub to a Kubernetes cluster.



Code Stream integrates with various endpoint types.

Table 2-1. Code Stream Integrates with DevOps Tools

Endpoint	What it does
Git	Pulls developer code from the repository and works with the Git trigger to trigger pipelines when developers check in code.
Kubernetes	Automates the steps to deploy, scale, and manage containerized applications.
Code Stream Build	Creates native builds for continuous integration instead of using third-party integrations.
Jenkins	Builds code artifacts.
Email	Sends notifications to users.
JIRA	Creates a ticket when a pipeline fails.
Bugzilla	Creates and track bugs.

Why You Use Code Stream

As a Code Stream administrator or developer, you use Code Stream to automate your entire DevOps release lifecycle, while you continue to use your existing development tools. Code Stream gives you:

- Easy automation
- Out-of-the-box plug-ins that work without open source

- Simple modeling experience and pipeline as code
- Straightforward integration with VMware Tanzu Kubernetes Grid Integrated Edition (formerly known as VMware Enterprise PKS)
- Reporting and insights
- End-to-end visibility with detailed dashboards
- Custom dashboards
- DevOps metrics and insights
- Governance
- Role-based access
- Secret and restricted variables, and approvals
- Projects

Code Stream simplifies the deployment of software applications as your source code runs through the development and test phases, and is released to production. It also increases your productivity by using the Git trigger. When a developer checks in code, Code Stream can trigger the pipeline and automate the build, test, and deployment of your application.

You can use Code Stream with other vRealize Automation components.

- Deploy a Cloud Assembly cloud template, and use the parameter values that the cloud template exposes.
- Publish your pipeline to Service Broker so that other members of your team can request and deploy it to their cloud regions.

For other ways to use Code Stream, see [Chapter 5 What Else Can I Do with Code Stream](#).

Before You Begin with Code Stream

3

As a Code Stream administrator or developer, you must sign up for a MyVMware account and log in before you can begin using Code Stream.

Log In

After you log in, you can create, run, and manage pipelines.

Table 3-1. Preparing to Use Code Stream

Step	Action	What you need to do
1	Create a MyVMware account.	Use your corporate email address to create your MyVMware account at https://my.vmware.com/ .
2	Log in to vRealize Automation.	Open HTTPS port 443 to outgoing traffic with access through the firewall to: <ul style="list-style-type: none">■ *.vmwareidentity.com■ gaz.csp-vidm-prod.com■ *.vmware.com
3	Open the Code Stream service.	

Getting Started for Code Stream Administrators

As an administrator, you have the Code Stream `Administrator` role, and can create endpoints so that you can ensure that working instances are available for developers. You can create, manage, and trigger pipelines, and more.

For example, your developers might need to connect their pipeline tasks to a data source, repository, or notification system. These components provide data for their pipelines to run.

You can also integrate Code Stream with other vRealize Automation components.

- Use Cloud Assembly to deploy VMware Cloud Templates.
- Use Service Broker to publish pipelines and trigger them.

Getting Started for Code Stream Developers

As a developer, you use Code Stream to build and run pipelines, and monitor pipeline activity on the dashboards. You'll need the `User` role.

After you run a pipeline, you'll want to know whether:

- Your code succeeded through all stages of your pipeline. Results appear in pipeline executions.
- Your pipeline failed and what caused the failure. Key errors appear in pipeline dashboards.

Using the Smart Pipeline Templates

To save time when you create a pipeline that natively builds, tests, and deploys your application, use the smart pipeline templates. Each smart pipeline template asks you several questions, and creates a pipeline based on how you answer the questions about:

- Your build goals, environments, and where your source code resides.
- Your deployment goals, and where you intend to deploy your application. For example, the smart pipeline template identifies your existing Kubernetes clusters. You can then select a cluster to use when you build and deploy your application.

After the smart pipeline template creates the pipeline, you can modify the pipeline further to make it even more specific to your needs.

How Do I Set Up Code Stream

4

As an administrator who sets up Code Stream, after you log in, you can add endpoints, create and run pipelines, and view the results.

Code Stream connects to endpoints to obtain data for your pipelines to run. In this use case, Code Stream connects to a GitLab repository so that your pipeline can download a Kubernetes file.

A getting started process is also available as a guided setup in the Code Stream user interface. Click the **Help** icon and click **Guided Setup**.

Prerequisites

- Verify that a GitLab repository or a GitHub repository exists on premises, and contains the code that your pipeline will use.

Procedure

- 1 Add a Git endpoint that connects Code Stream to your on-premises GitLab repository.
 - a Click **Endpoints**.
 - b Select the Git endpoint type, and enter a name and description.
 - c Enter the remaining information.
 - d To test the connection to the endpoint, click **Validate**, then save the endpoint.

- Click **Pipelines**, create a pipeline, and add a task that uses the Git endpoint. You can optionally add an email notification.

The screenshot displays the vRealize Automation Code Stream interface for a pipeline named 'MyKubeApp4' (status: Disabled). The interface is divided into a workspace on the left and a task configuration panel on the right.

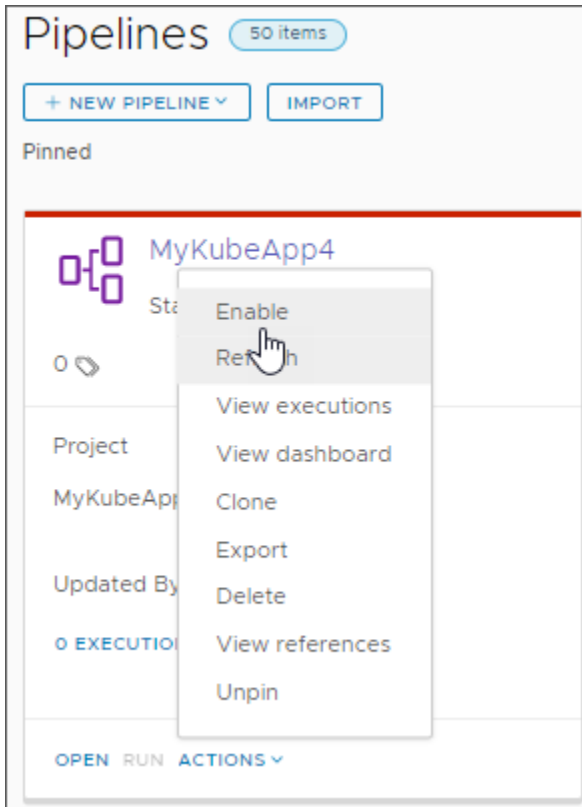
Workspace: The workspace shows a visual representation of the pipeline with two stages: 'Stage0' and 'Stage1'. 'Stage0' contains a 'Task0' (Kubernetes) and a 'Parallel Task'. 'Stage1' contains a 'Task0' (Jenkins) and a 'Parallel Task'. A 'Stage' label is also visible at the bottom of the workspace.

Task Configuration Panel (Task: Task0):

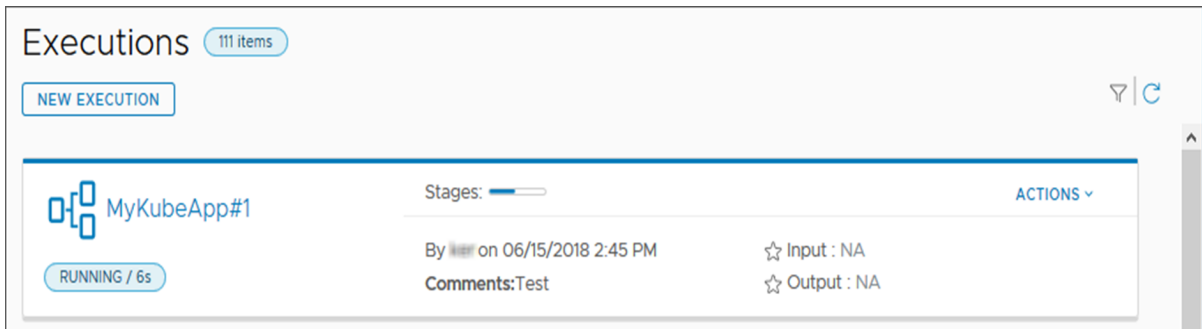
- Task name:** Task0
- Type:** Kubernetes
- Continue on failure:** ☐
- Execute task:** ☒ Always ☐ On condition
- Kubernetes Task Properties:**
 - Kubernetes cluster:** Dev-AWS-Cluster
 - Timeout (in mins):** 5
 - Action:** ☒ Get ☐ Create ☐ Apply ☐ Delete ☐ Rollback
 - Filter by label:** Enter label to filter
 - Source type:** ☒ Source Control ☐ Local definition
 - Git:** GitLab-OnPrem
 - File path:** Enter filepath
- Parameters:** parameter name | parameter value (+)
- Output Parameters:** status

At the bottom of the interface, there are buttons for 'SAVE', 'RUN', and 'CLOSE', along with the text 'Last saved: a year ago'. A 'VALIDATE TASK' button is also present in the top right of the task configuration panel.

- 3 Save your pipeline, then click **Enable**, which enables it to run.



- 4 After you enable the pipeline, click **Run**.
- 5 Click **Executions**, and observe your pipeline as it runs.




- 6 If the pipeline fails, correct the problem and run it again.
- 7 Click **Dashboards**, and select your pipeline dashboard so that you can monitor the pipeline activity.

Results

Your pipeline ran, and downloaded the developer file from a GitLab instance. The pipeline task deployed the application to a Kubernetes cluster, and you monitored all the activity on the pipeline dashboard.

What to do next

If you don't find the information you need here, you can get more help in the product. 

- To get the context-specific information, when and where you need it, click and read the signposts and tooltips in the user interface.
- Open the In-product support panel and read the topics that appear for the active user interface page. To get answers to questions, you can also search in the panel.

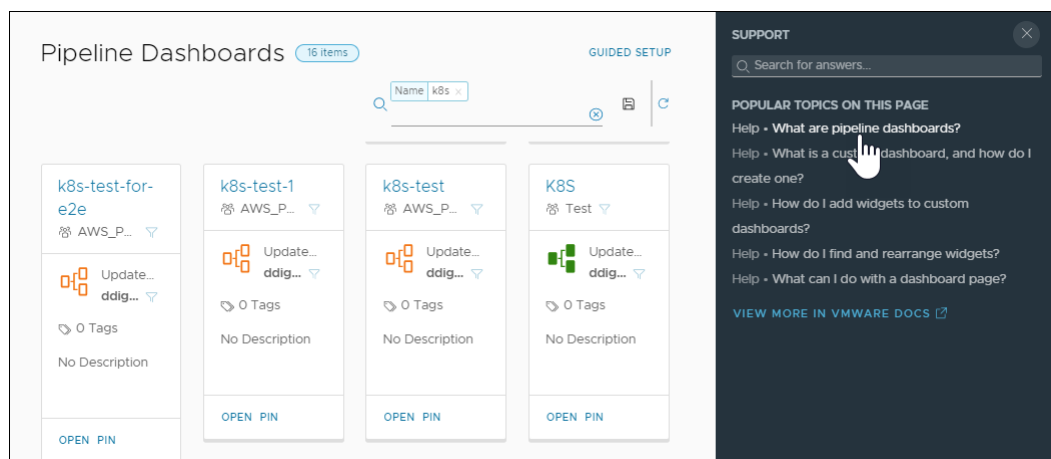
What Else Can I Do with Code Stream

5

As a Code Stream administrator or developer, you can create more complex pipelines that model and automate the steps in your DevOps release lifecycle.

To learn about...	See Using and Managing
What it is and what you can do	What is it and how does it work
Modeling your release process	Setting up to model my release process
Configuring the pipeline workspace	Configuring the Pipeline Workspace
Planning a CICD, CI, or CD native build, and creating the pipeline	Planning to natively build, integrate, and deliver your code
Use cases and tutorials	For example: How do I continuously integrate code from my GitHub or GitLab repository into my pipeline And more!

For more information about what you can do with Code Stream, use the in-product help in the in-product support panel in Code Stream.



By using the in-product help, you can learn about these features and much more:

- Communicating with endpoints
- Working with restricted resources and variables

- Working with projects
- Triggering a pipeline when developers update code
- Tracking KPIs for your pipelines on custom dashboards

To learn more about what you can do with Code Stream, see [Using and Managing VMware Code Stream](#).