# Software Development Life Cycle (SDLC)

**Note**

To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst Controller**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

# Architecture of Software Development Life Cycle Pipelines

There are two Cisco Catalyst SD-WAN for government Software Development Life Cycle (SDLC) pipelines:

- Cisco vOrchestrator and Cisco vMonitor SDLC pipeline: The SDLC pipeline used to develop and deploy control components in the Amazon management VPC.

- Cisco Catalyst SD-WAN controllers and Cisco SD-WAN Manager SDLC pipeline: The SDLC pipeline used to develop and deploy control components in the Amazon customer VPC.

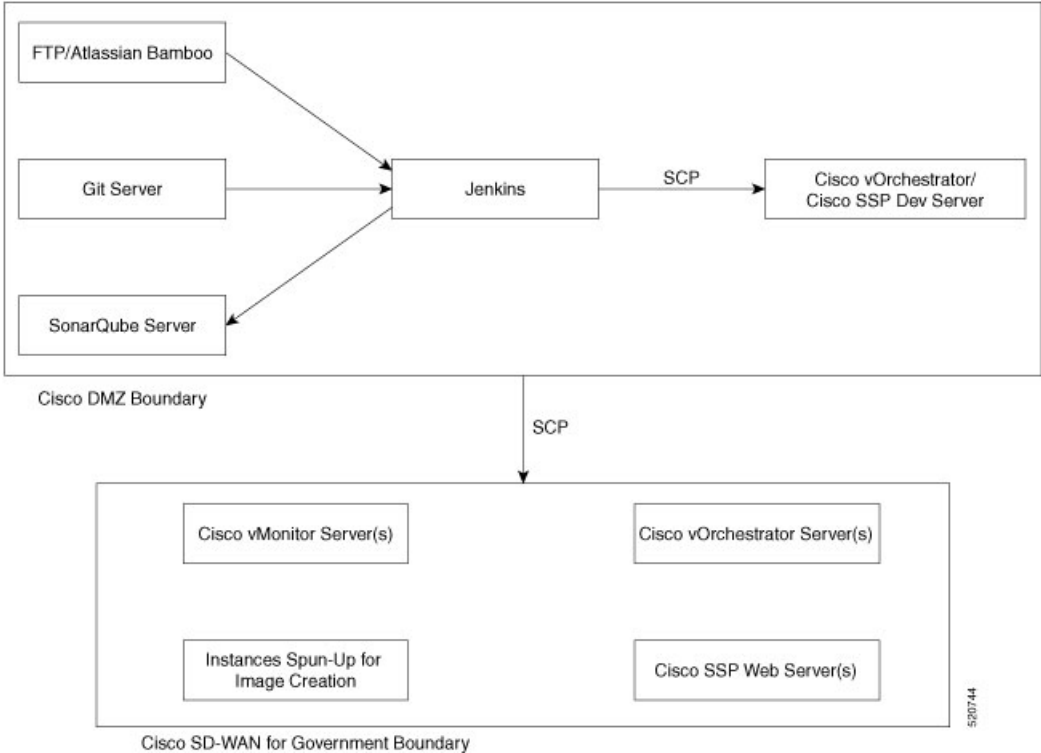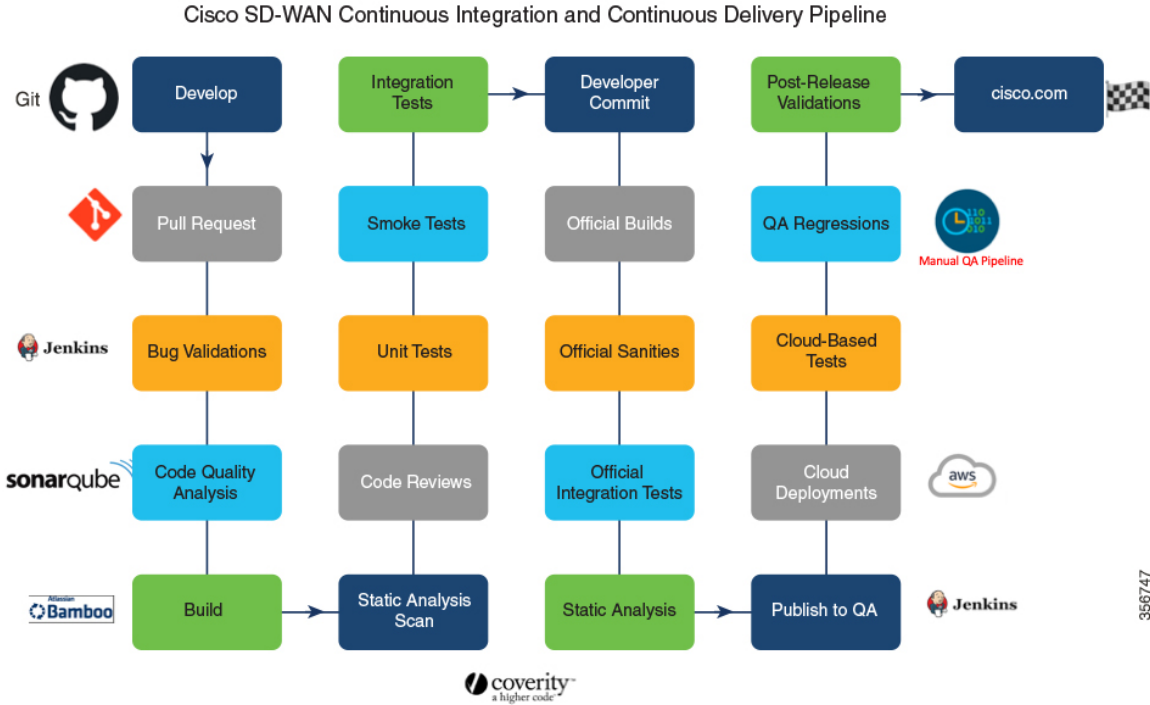*Figure 1: Cisco vOrchestrator and Cisco vMonitor SDLC pipeline*



*Figure 2: Cisco SD-WAN Continuous Integration and Continuous Delivery Pipeline*

# Management VPC SDLC Pipeline

Before the images are built and deployed to Cisco Catalyst SD-WAN for government, the code is first analyzed. After analysis, the code is pushed to a development server in the Cisco DMZ network.

The management VPC SDLC pipeline does the following:

## Checks for Regressions and Analysis of Code

To check for regressions and analyze code, Cisco has created the following automated pipeline:

1. Jenkins pulls the code locally from Cisco's Git server.

2. Jenkins (open-source automation tool) securely copies the code to the development server using the scp (secure copy) utility.

3. On the development server, robot regressions are triggered. Regression reports are generated and stored locally on the Jenkins server.

4. The SonarQube scanner scans the local source code. The results of the scans are pushed to the SonarQube server.

5. Clears the local workspace.

## Upgrade and Deploy Apps

To upgrade and deploy applications, Jenkins does the following:

1. Pulls the source code locally.

2. Copies the code using the scp utility to the appropriate server using the PEM key.

3. Performs the required steps to upgrade or deploy the servers.

4. Verifies that all the services are functioning.

5. Sends a notification email of the job status.

6. Clears the local workspace.

## Deploy to Cisco Catalyst SD-WAN for Government

The pipeline creates instances that it uses to create encrypted Amazon Machine Images (AMI) of the new builds. These images are then copied to the federal government environment and added to the image database. The pipeline does this:

1. Pulls the appropriate build file (tar.gz) from the FTP server.

2. Creates instances in the GovCloud enviroment.

3. Uses these instances to create base images.

4. Using the scp utility, and securely copies the code to these instances.

5.   Configures the instances to meet the requirements of each controller.

6.   Creates base images using these instances.

7.   Terminates the instances that were used for image creation.

8.   Creates unencrypted copies of the new image.

9.   Creates encrypted copies of the unencrypted images.

10.  Tags the encrypted images.

11.  Clears the build files from the local server.

12.  Cisco vOrchestrator identifies the encrypted images using the tags.

13.  Cisco vOrchestrator stores the AMI IDs in the database to use when creating an overlay network.

# Customer VPC SDLC Pipeline

To develop and deploy code for Cisco Catalyst SD-WAN controllers, the following is done:

1.  Developers write and integrate code:

   a.   Writes code and pushes to Git.

   b.   Jenkins validates the code for bugs.

   c.   SonarQube analyzes the code for quality.

   d.   Atlassian Bamboo builds the code

   e.   A static analysis scan is performed.

   f.   Other developers review the code.

   g.   Other standard code tests such as unit, smoke, and integration tests are performed.

   h.   The developer commits the code.

   i.   An official build is generated.

   j.   Official sanity and integration tests are conducted.

   k.   Static analysis scans are run on the build.

   l.   The builds are published to Quality Assurance.

2.  Quality Assurance tests the code:

   a.   Quality Assurance deploys the builds to cloud deployments.

   b.   Quality Assurance runs cloud-based tests.

   c.   Quality Assurance runs regression tests and other tests that are a part of the manual Quality Assurance pipeline.

   d.   Run postrelease validations.

3. The build is published to Cisco.com.

# Code Analysis Reporting

Whenever one of the Cisco cloud applications (Cisco vOrchestrator, Cisco vMonitor, Cisco Catalyst SD-WAN Portal, AWS bastion host, and Data Center Services [DCS]) applications are upgraded through Jenkins, a code analysis report is generated. The report is available on the Cisco Catalyst SD-WAN Portal. After every upgrade, a script pushes the report to AWS S3, and saves it based on its version. If there is a request from the Cisco Catalyst SD-WAN Portal, a report is downloaded to AWS S3 directly and served.

*Figure 3: Code Analysis Reporting Workflow*
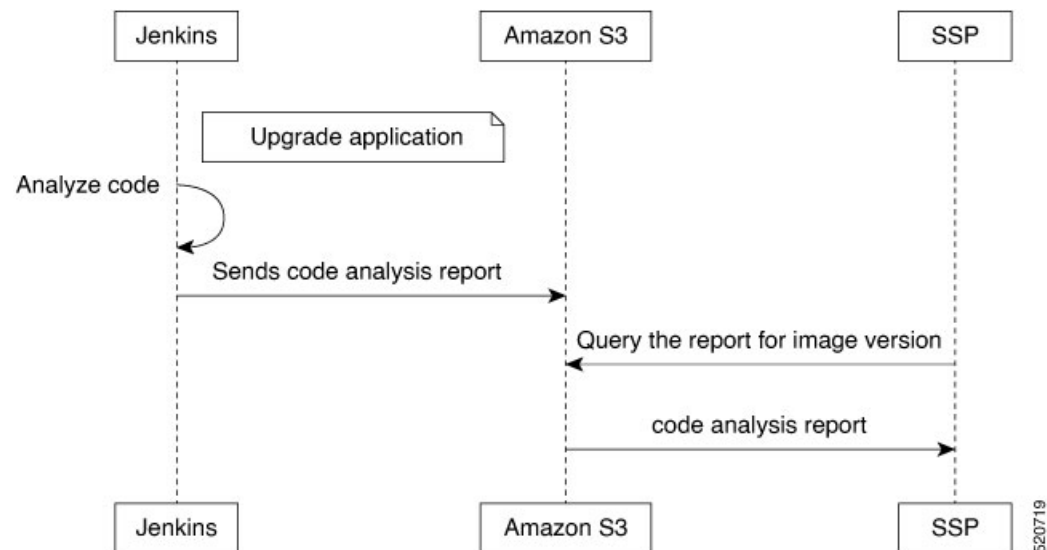


**Accessing the Build Report Using the Cisco Catalyst SD-WAN Portal**

1. Log in to the Cisco Catalyst SD-WAN Portal.

2. Click the sidebar icon in the top left corner of the window.

3. Click **Build Reports**.

4. Download the code analysis reports.