



The bridge to possible

CI/CD pipelines for infrastructure automation

Adrian Iliesiu, Senior Developer Advocate
@aidevnet

Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.





Agenda

- CI/CD overview
- GitLab
- Example of a CI/CD pipeline workflow
- Demo
- Resources

CI/CD overview



CI/CD Pipelines

Continuous Integration - Software

- Practice of integrating **code** that developers are producing
- Code is developed in separate **branches**
 - Multiple developers
 - Multiple features
- When code is **merged** into a branch
 - Testing validates changes
 - Flush out potential bugs
 - Ensure quality and no **integration** issues

CI/CD Pipelines

Continuous Delivery – Software

- Releases can be created and delivered at any time
- Changes have met integration requirements?
 - Push to a “stage” environment
 - Run additional tests and checks
- All tasks are automated

CI/CD Pipelines

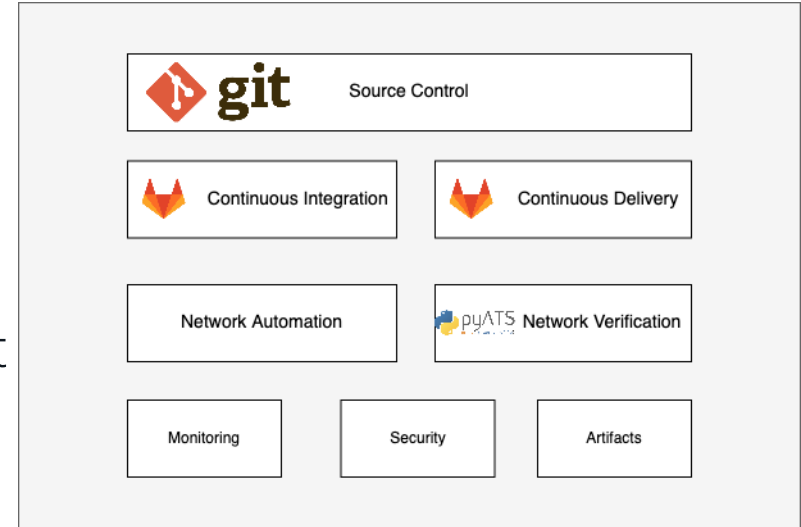
Continuous Deployment – Software

- Takes delivery a step further
- Deploys changes to a production environment
 - Many different deployment strategies available
 - Blue/Green, Canary, Rolling

CI/CD Pipelines

Infrastructure Automation

- Network as Code and Configuration Management strategies
- Continuous Delivery approach to network changes
- Continuous health and improvement approach to monitoring

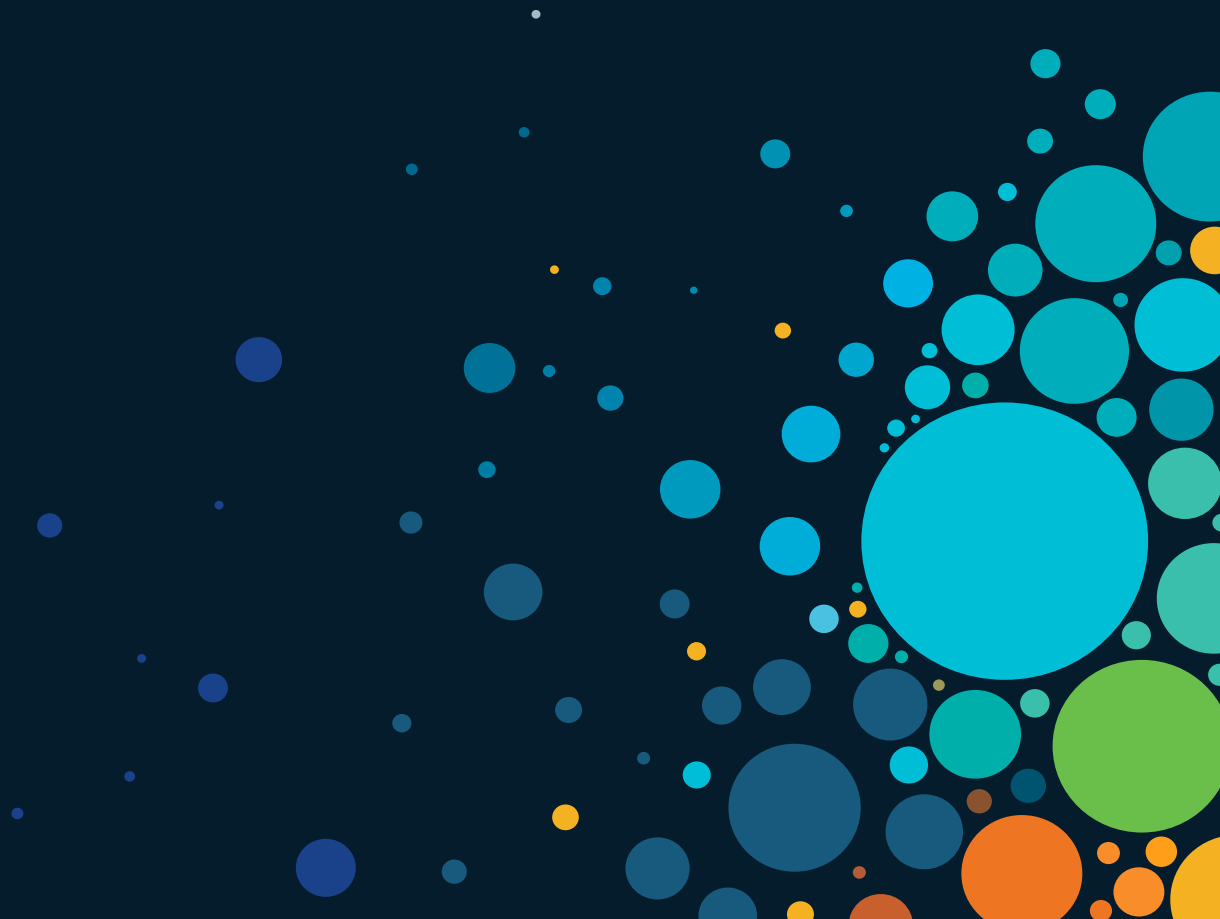


CI/CD Pipelines

Infrastructure Automation

- Configuration changes are done through automation
- Historical insight into changes made to the network
- Validate changes prior to configuring the network
- If an issue is introduced into the network, those changes can be rolled back to a known good state
- Integration with change and ticketing systems for true Continuous Deployment

GitLab



GitLab

Overview

- Complete DevOps platform
- Project Management
- Hosted Git repositories
- Built-in CI/CD

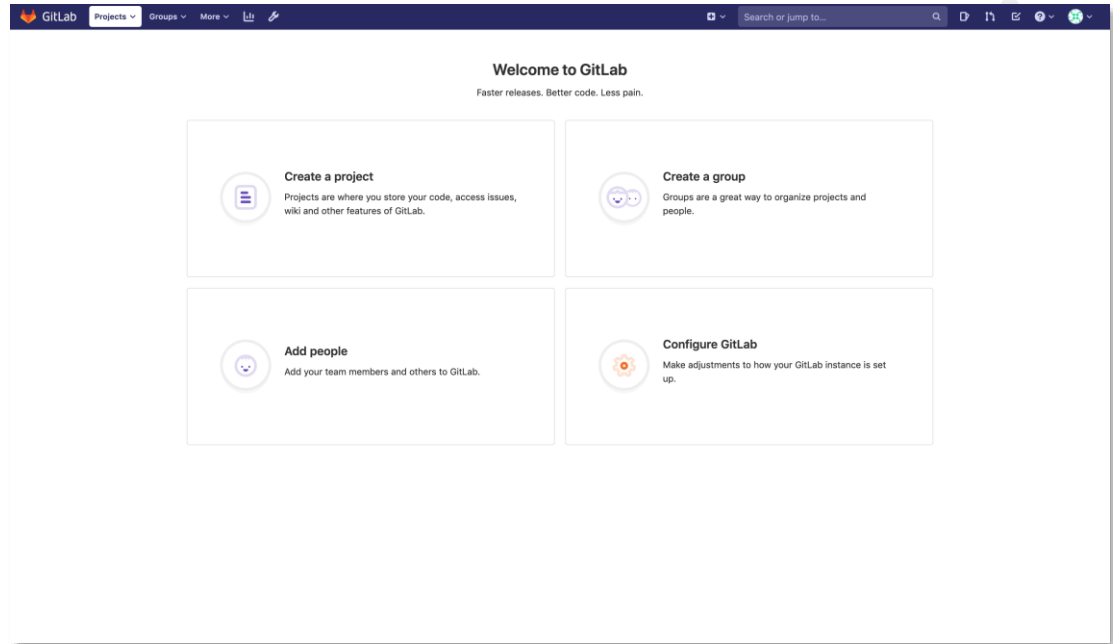
Stages of the DevOps lifecycle

Manage Plan Create Verify Package Secure Release Configure Monitor Defend

GitLab

Projects

- Code, issues, wikis, etc. are organized into projects



GitLab

- Projects

- Create new projects from scratch, use a template, or import an existing project
- Configurable visibility levels: private, internal, public

The screenshot shows the 'New project' page in the GitLab web interface. The page has a dark blue header with the GitLab logo and navigation links. The main content area is white and contains a form for creating a new project. The form is titled 'New project' and has three tabs: 'Blank project' (selected), 'Create from template', and 'Import project'. The 'Blank project' tab contains the following fields and options:

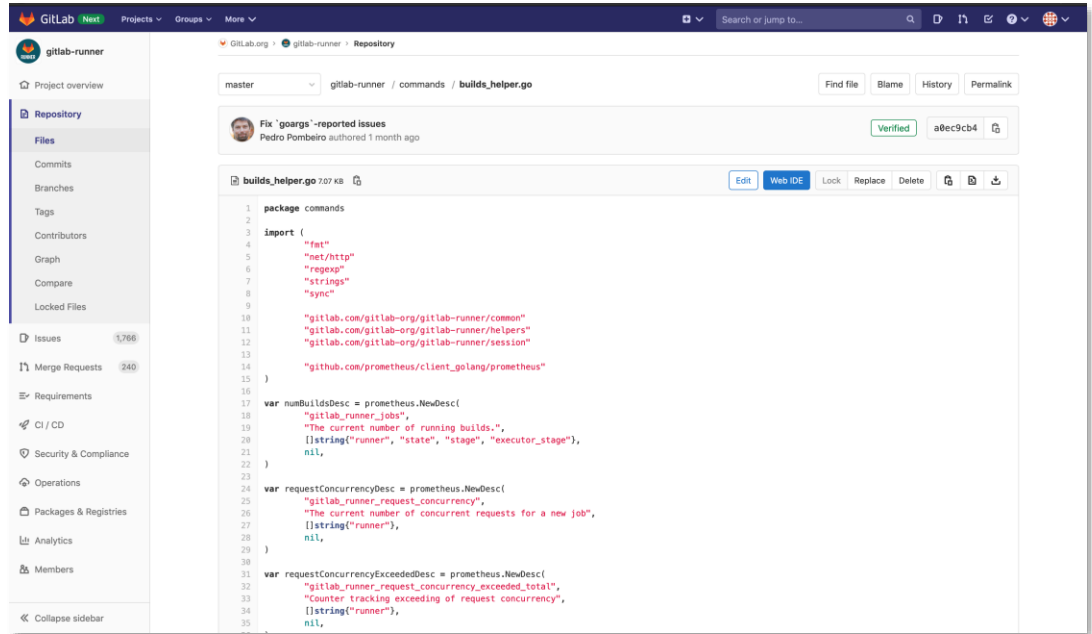
- Project name:** A text input field with the value 'my-first-git-project'.
- Project URL:** A text input field with the value 'http://10.194.104.70:8080/devcx/'.
- Project slug:** A text input field with the value 'my-first-git-project'.
- Project description (optional):** A text area with the placeholder 'Description format'.
- Visibility Level:** A section with three radio button options:
 - ☒ **Private**: Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
 - ☐ **Internal**: The project can be accessed by any logged in user.
 - ☐ **Public**: The project can be accessed without any authentication.
- ☐ **Initialize repository with a README**: Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

At the bottom of the form, there is a green 'Create project' button and a 'Cancel' button.

GitLab

Files

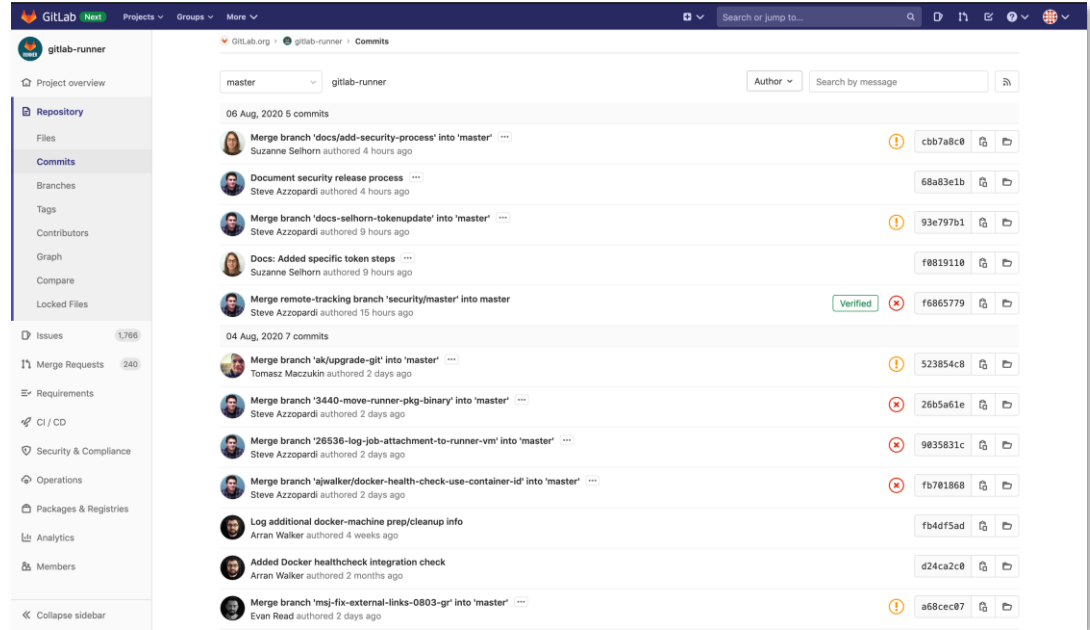
- View, edit, create files under or in your project (repository)



GitLab

Commits

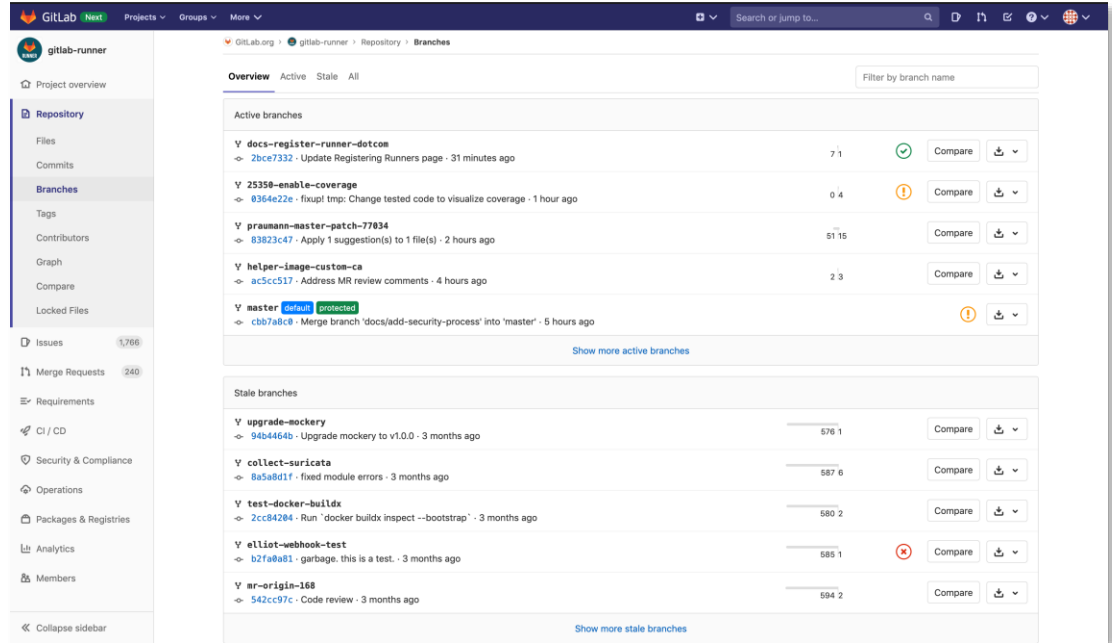
- View and manage commits within your project



GitLab

Branches

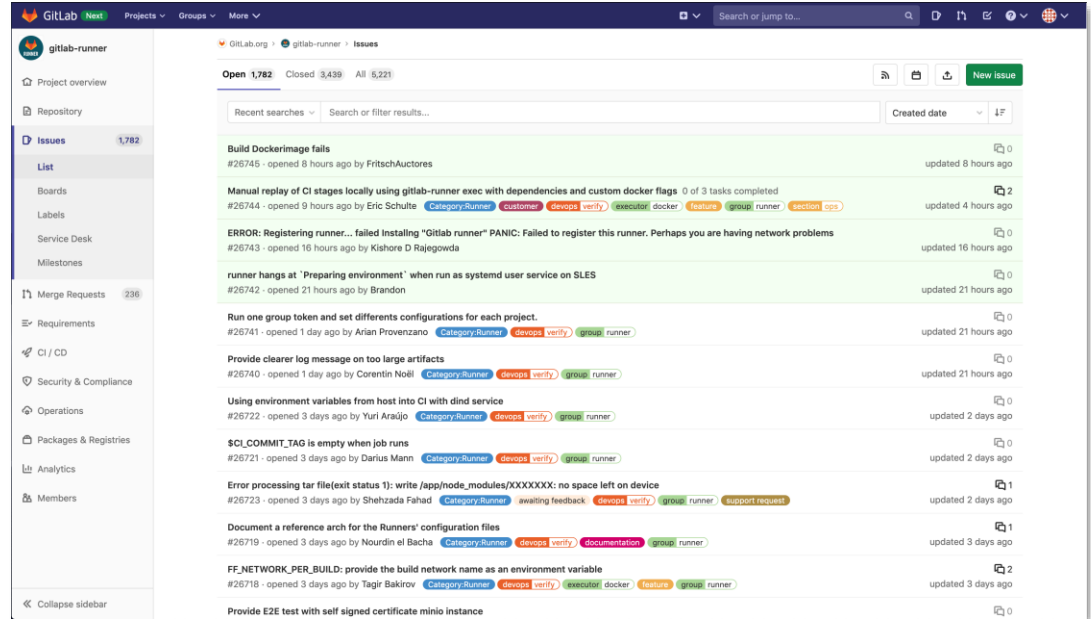
- A version of a project's working tree
- Create merge requests
- Perform inline code review



GitLab

Issues

- Used for collaborating on ideas and planning work
- Enable sharing and discussion of ideas/proposals
- Track status of work and tasks



GitLab

Merge Requests

- Request to merge one branch into another

The screenshot shows the GitLab Merge Requests interface for the `gitlab-runner` repository. The left sidebar contains navigation links: Project overview, Repository, Issues (1,782), Merge Requests (236), Requirements, CI / CD, Security & Compliance, Operations, Packages & Registries, Analytics, and Members. The main content area displays a list of merge requests. The top summary shows 236 Open, 1,632 Merged, 476 Closed, and All 2,343 merge requests. Below this is a search bar and a table of merge requests. Each entry includes the title, author, date, and various status icons (CI build, devops, verify, group, runner, etc.).

Title	Author	Date	Status
clarify which cluster k8s executor is referring to	Muhammed Hussein	opened 12 hours ago	Open
Replace golint linter with stylecheck	Pedro Pombeiro	opened 4 days ago	Open
WIP: Add build status and stage environment variables	Arran Walker	opened 4 days ago	Open
Document how to access IBM Z/OS VM	Pedro Pombeiro	opened 1 week ago	Open
Replace whitelist terminology with allowlist	Pedro Pombeiro	opened 1 week ago	Open
WIP: Update Registering Runners page	Greg Myers	opened 1 week ago	Open
Add labels to docker cache volumes	Adrien Kohlbecker	opened 1 week ago	Open
Disable syslogging by default for systemd systems	Matthias Baur	opened 1 week ago	Open
Use configured users mode for services	Lukáš Brzobohatý	opened 1 week ago	Open
Add indentation to YAML code	Pedro Pombeiro	opened 1 week ago	Open
Upgrade docker (19.03.12), buildx (0.4.1)	Arran Walker	opened 2 weeks ago	Open
Add support k8s runtime class			Open

GitLab

Pipelines

- Fundamental building blocks for CI/CD

The screenshot shows the GitLab interface for a project named 'gitlab-runner'. The left sidebar contains navigation links: Project overview, Repository, Issues (1,782), Merge Requests (236), Requirements, CI / CD (selected), Jobs, Schedules, Security & Compliance, Operations, Packages & Registries, Analytics, and Members. The main content area is titled 'Pipelines' and shows a list of pipeline runs. The table has columns for Status, Pipeline, Triggerer, Commit, and Stages. The first four pipelines are in a 'running' state, while the last two are 'canceled' and the last one is 'passed'.

Status	Pipeline	Triggerer	Commit	Stages
running	#177869707	[Avatar]	!2346 -> 8effc580 Merge branch 'k8sexectorclust...	[Icons]
running	#177869404	[Avatar]	!2346 -> d3b5331f Merge branch 'k8sexectorclust...	[Icons]
running	#177606098	[Avatar]	!2325 -> cfa77e77 Merge branch 'patch-4' into 'm...	[Icons]
running	#177604493	[Avatar]	!2325 -> 3b94d486 Updated Windows steps	[Icons]
canceled	#177590399	[Avatar]	!2325 -> 2846f438 Merge branch 'patch-4' into 'm...	[Icons]
canceled	#177535883	[Avatar]	!2325 -> 2846f438 Merge branch 'patch-4' into 'm...	[Icons]
passed	#177412421	[Avatar]	!26632-gitlab->9a6585be Add helper binary	[Icons]
passed	#177408271	[Avatar]	!26632-gitlab->825f9495 Add helper binary	[Icons]

GitLab

Pipelines

- Composed of:
 - Jobs – What to do
 - Stages – When to run jobs
- A typical pipeline would contain four stages
 - Build (compile)
 - Test (unit-tests)
 - Staging (deploy-to-stage, integration-tests)
 - Production (deploy-to-prod)

GitLab

Runners

- Used to run the jobs and send results back to GitLab
- When a pipeline is triggered, a Runner will `git clone` the repo and then execute the instructions inside of `.gitlab-ci.yml`
- Implements a variety of executors
 - Jobs can run in different scenarios
 - SSH, Shell, Parallels, VirtualBox, Docker, Kubernetes, Custom

GitLab

.gitlab-ci.yml

```
1  stages:
2    - build
3    - test
4    - deploy
5
6  image: alpine
7
8  compile_a:
9    stage: build
10   script:
11     - echo "This job builds something."
12
13  compile_b:
14    stage: build
15    script:
16      - echo "This job builds something else."
17
18  unit_test_a:
19    stage: test
20    script:
21      - echo "This job tests something. It will only run when all jobs in the"
22        - echo "build stage are complete."
23
24  unit_test_b:
25    stage: test
26    script:
27      - echo "This job tests something else. It will only run when all jobs in the"
28        - echo "build stage are complete too. It will start at about the same time as test_a."
29
30  deploy_a:
31    stage: deploy
32    script:
33      - echo "This job deploys something. It will only run when all jobs in the"
34        - echo "test stage complete."
35
36  deploy_b:
37    stage: deploy
38    script:
39      - echo "This job deploys something else. It will only run when all jobs in the"
40        - echo "test stage complete. It will start at about the same time as deploy_a."
```

GitLab

Schedules

- Pipelines normally run when conditions are met
 - When a branch is pushed to a repository
- Schedules enable pipeline runs at specific intervals
 - Once every day
 - Every Friday
 - Any schedule as needed

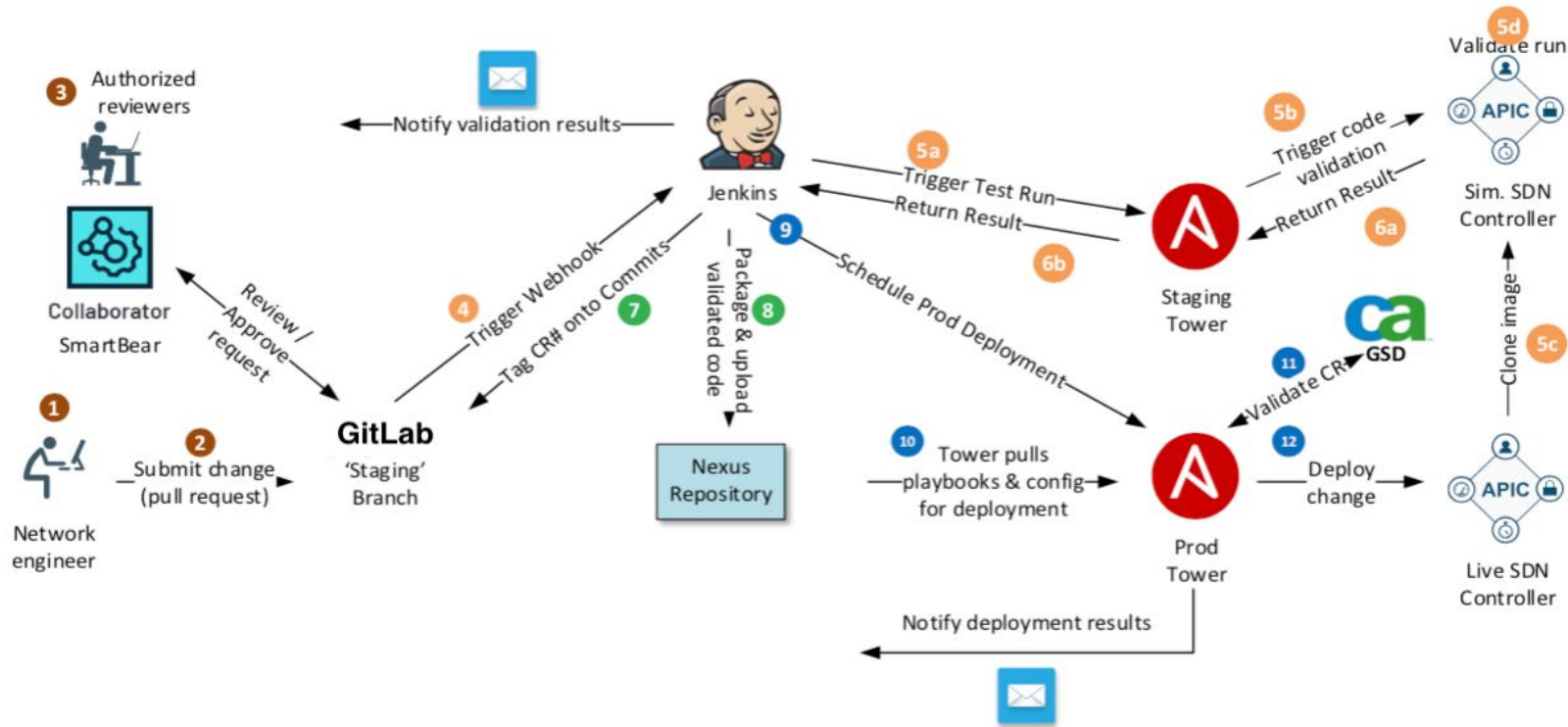
GitLab

Environment Variables

- Useful for customizing your jobs

```
export CI_JOB_ID="50"
export CI_COMMIT_SHA="1ecfd275763eff1d6b4844ea3168962458c9f27a"
export CI_COMMIT_SHORT_SHA="1ecfd275"
export CI_COMMIT_REF_NAME="master"
export CI_REPOSITORY_URL="https://gitlab-ci-token:abcde-1234ABCD5678ef@example.com/gitlab-org/gitlab-foss.git"
export CI_COMMIT_TAG="1.0.0"
export CI_JOB_NAME="spec:other"
export CI_JOB_STAGE="test"
export CI_JOB_MANUAL="true"
export CI_JOB_TRIGGERED="true"
export CI_JOB_TOKEN="abcde-1234ABCD5678ef"
export CI_PIPELINE_ID="1000"
export CI_PIPELINE_IID="10"
export CI_PAGES_DOMAIN="gitlab.io"
export CI_PAGES_URL="https://gitlab-org.gitlab.io/gitlab-foss"
export CI_PROJECT_ID="34"
export CI_PROJECT_DIR="/builds/gitlab-org/gitlab-foss"
export CI_PROJECT_NAME="gitlab-foss"
export CI_PROJECT_TITLE="GitLab FOSS"
export CI_PROJECT_NAMESPACE="gitlab-org"
export CI_PROJECT_ROOT_NAMESPACE="gitlab-org"
export CI_PROJECT_PATH="gitlab-org/gitlab-foss"
export CI_PROJECT_URL="https://example.com/gitlab-org/gitlab-foss"
export CI_REGISTRY="registry.example.com"
export CI_REGISTRY_IMAGE="registry.example.com/gitlab-org/gitlab-foss"
export CI_REGISTRY_USER="gitlab-ci-token"
export CI_REGISTRY_PASSWORD="longalphanumericstring"
export CI_RUNNER_ID="10"
export CI_RUNNER_DESCRIPTION="my runner"
export CI_RUNNER_TAGS="docker, linux"
export CI_SERVER="yes"
export CI_SERVER_URL="https://example.com"
export CI_SERVER_HOST="example.com"
export CI_SERVER_PORT="443"
export CI_SERVER_PROTOCOL="https"
export CI_SERVER_NAME="GitLab"
export CI_SERVER_REVISION="70606bf"
export CI_SERVER_VERSION="8.9.0"
export CI_SERVER_VERSION_MAJOR="8"
export CI_SERVER_VERSION_MINOR="9"
export CI_SERVER_VERSION_PATCH="0"
export GITLAB_USER_EMAIL="user@example.com"
export GITLAB_USER_ID="42"
```


Example of a CI/CD Pipeline Workflow



Demo

Test environment

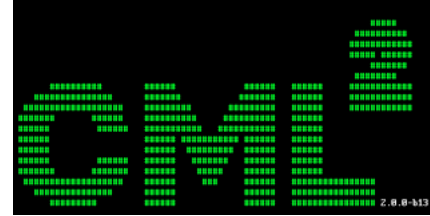


**Dev
Workstation**

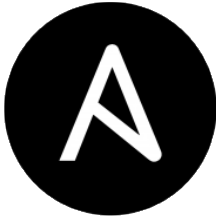


CentOS

**Hosting
Server**



**Network
Simulation**



**Infrastructure
as Code**

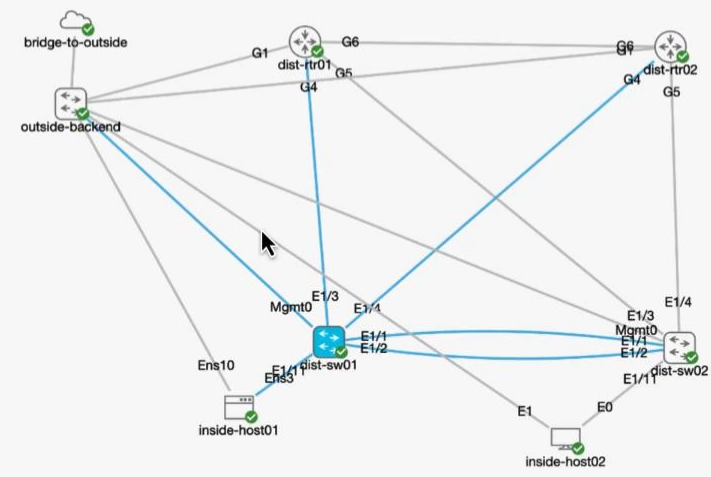


GitLab

**Source
Control**



**Testing
Framework**



NODE INFO SIMULATE CONNECTIVITY CONSOLE VNC EDIT CONFIG INTERFACES

CLONE NODE DELETE NODE

NX-OS 9000
Node Name:
dist-sw01

CPU 28.56% MEMORY 18.82% DISK 8.38%



Resources



- <https://blogs.cisco.com/author/adrianiliesiu>
- <https://github.com/CiscoDevNet>

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at <https://www.ciscolive.com/emea/learn/sessions/session-catalog.html>



Continue Your Education



Visit the Cisco Showcase for related demos.



Book your one-on-one Meet the Engineer meeting.



Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.



Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.



The bridge to possible

Thank you

CISCO *Live!*

CISCO *Live!*

ALL IN