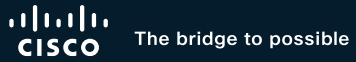


CI/CD pipelines for infrastructure automation

Adrian Iliesiu, Senior Developer Advocate
@aidevnet

Cisco Webex App

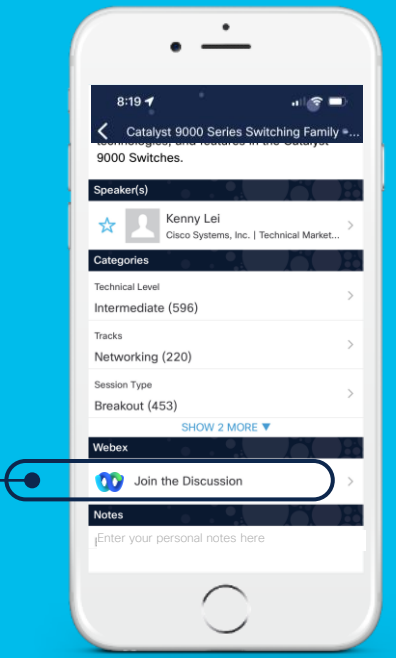
Questions?

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

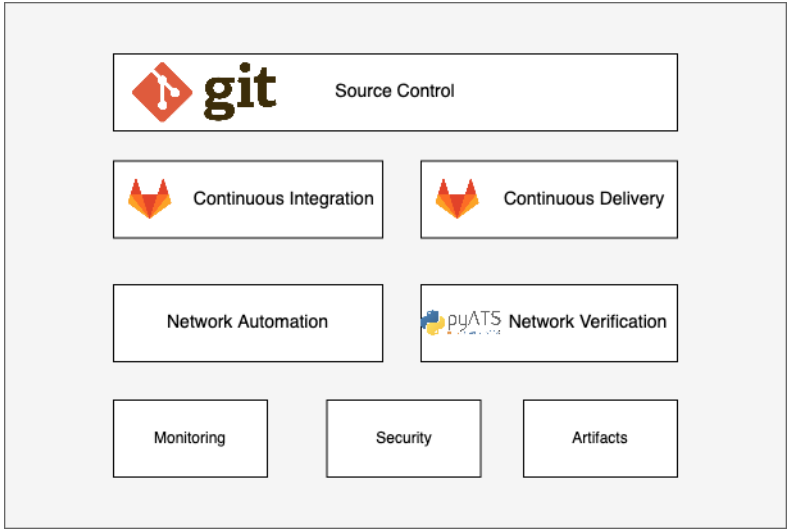
Continuous Deployment – Software

- 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

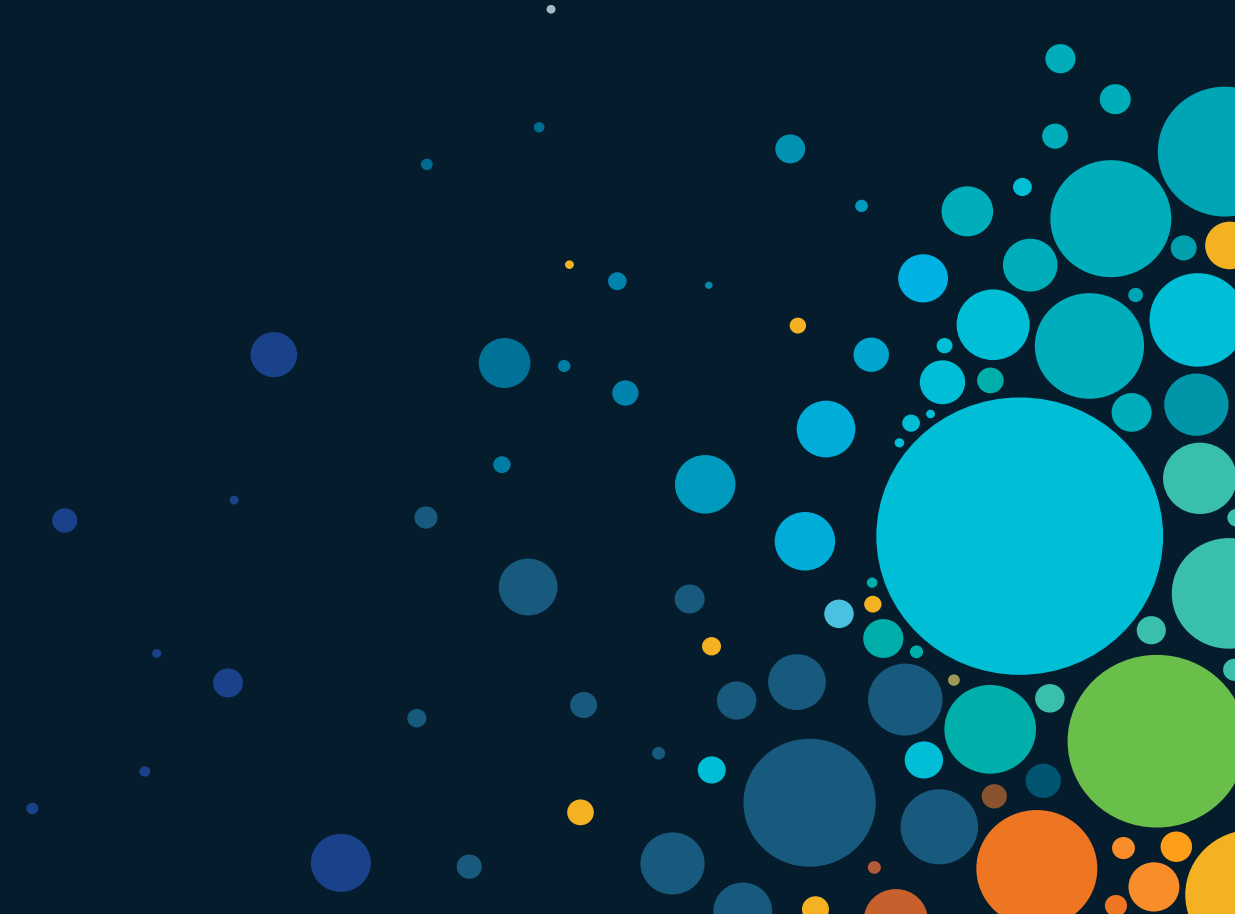


CI/CD Pipelines

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

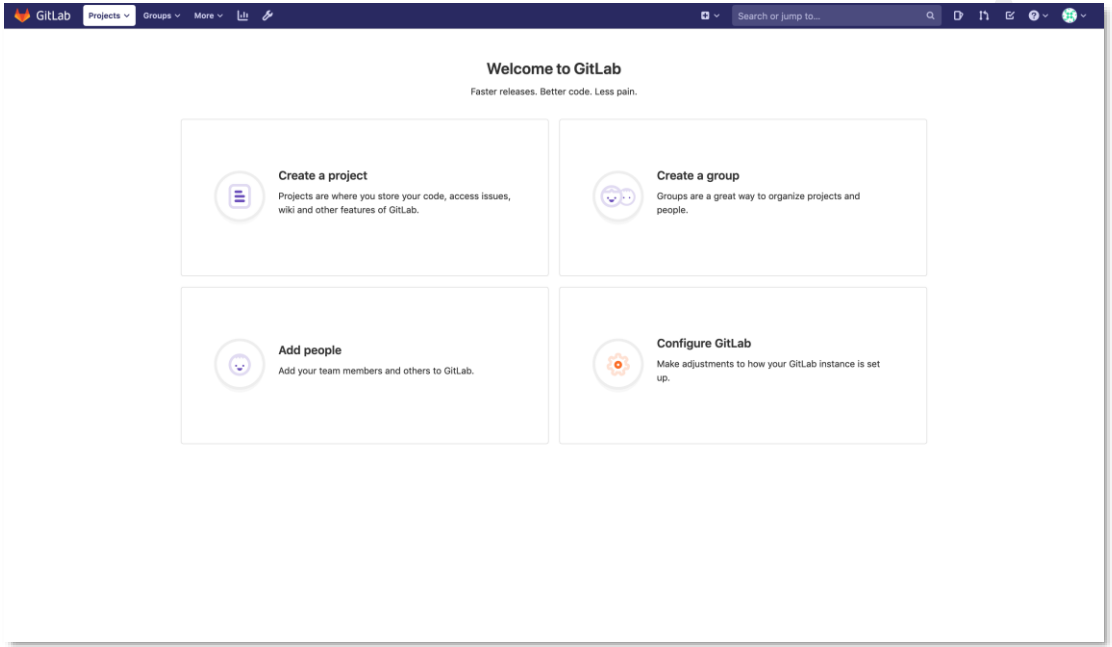
Overview

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

GitLab

Projects

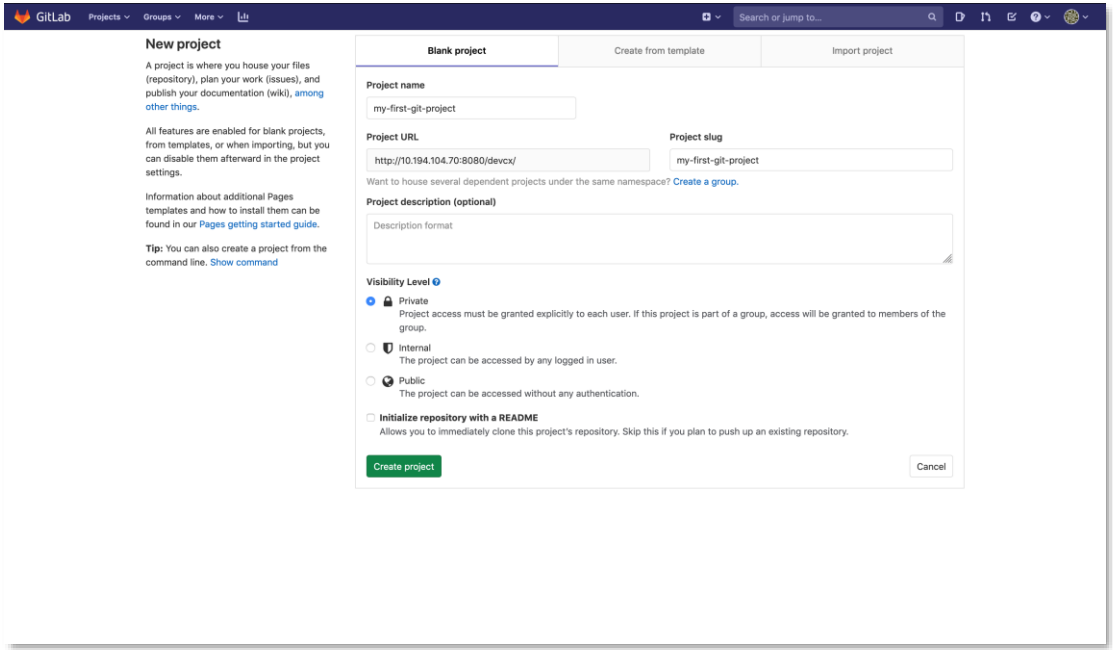
- Code, issues, wikis, etc. are organized into projects



GitLab

- Projects

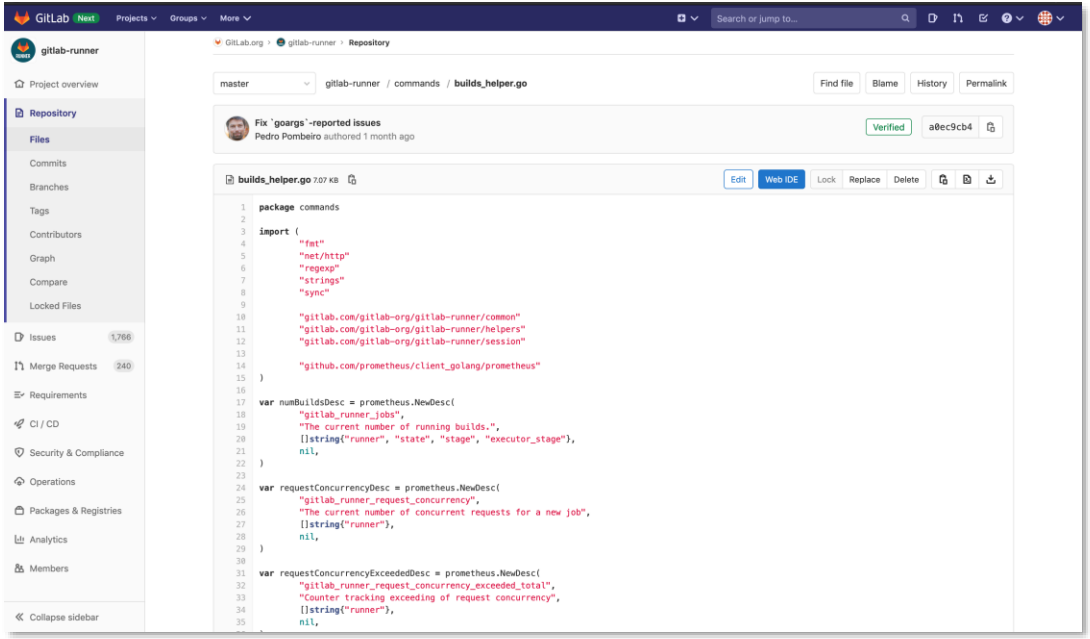
- Create new projects from scratch, use a template, or import an existing project
- Configurable visibility levels: private, internal, public



GitLab

Files

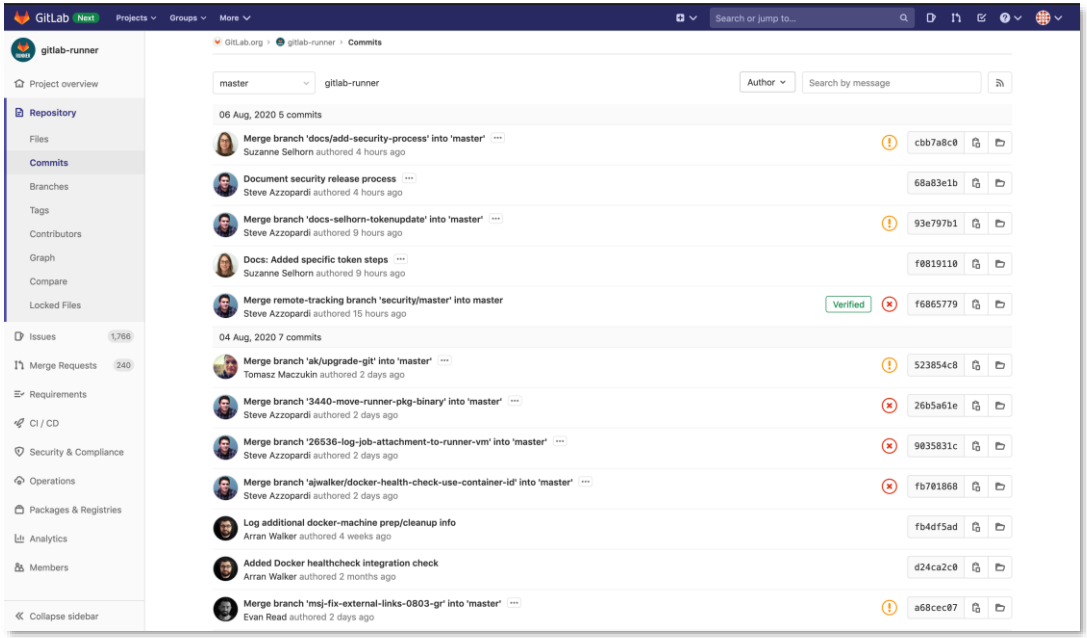
- View, edit, create files under or in your project (repository)



GitLab

Commits

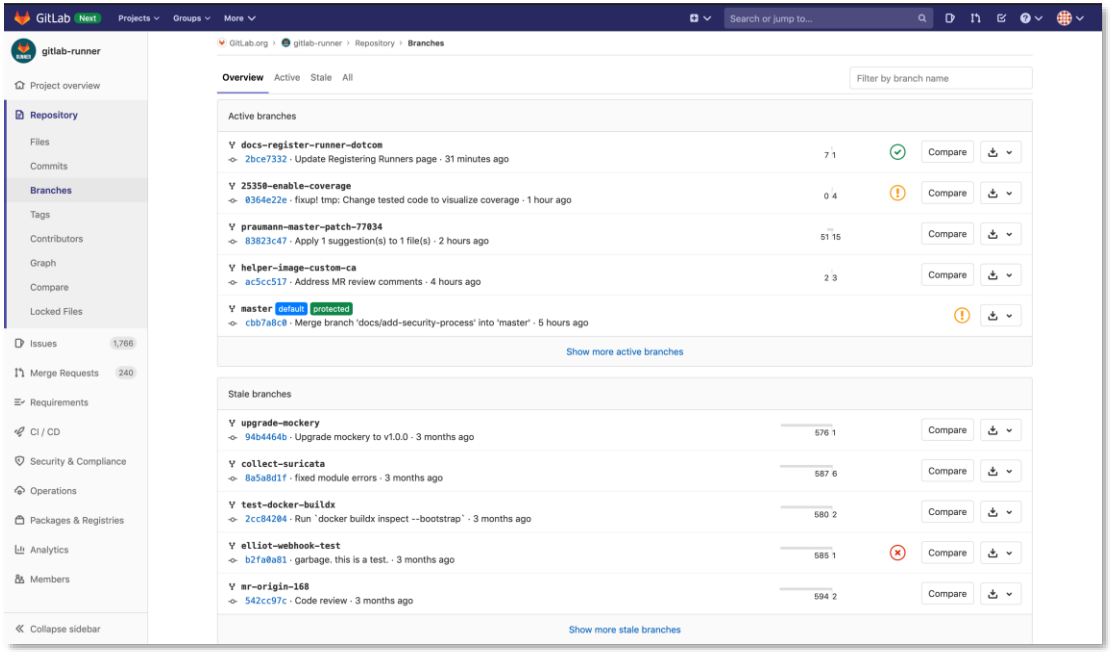
- View and manage commits within your project



GitLab

Branches

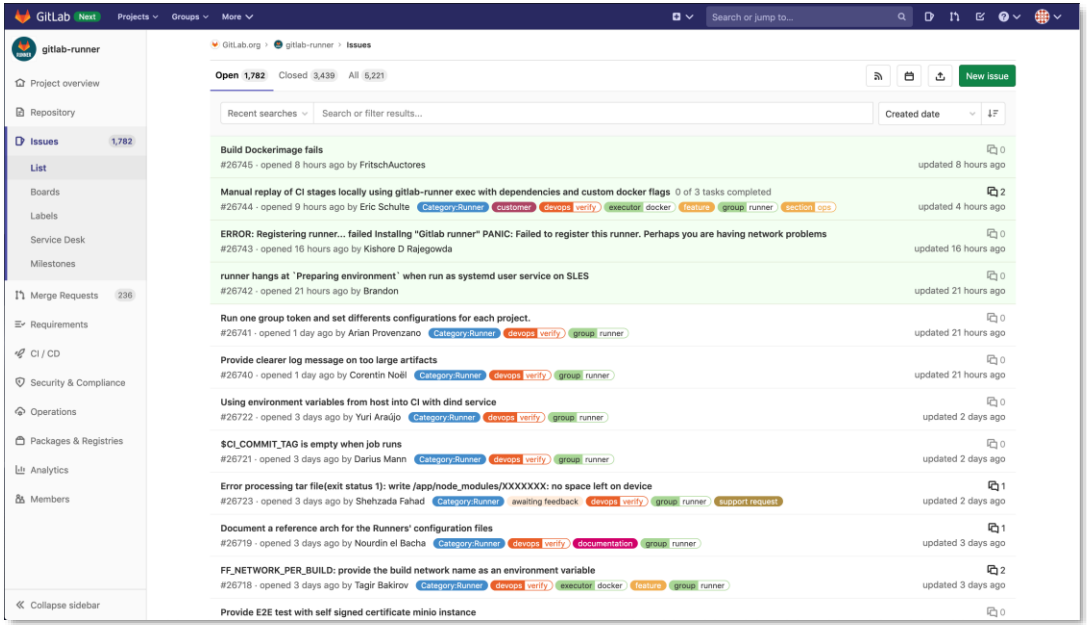
- A version of a project's working tree
- Create merge requests
- Perform inline code review



GitLab

Issues

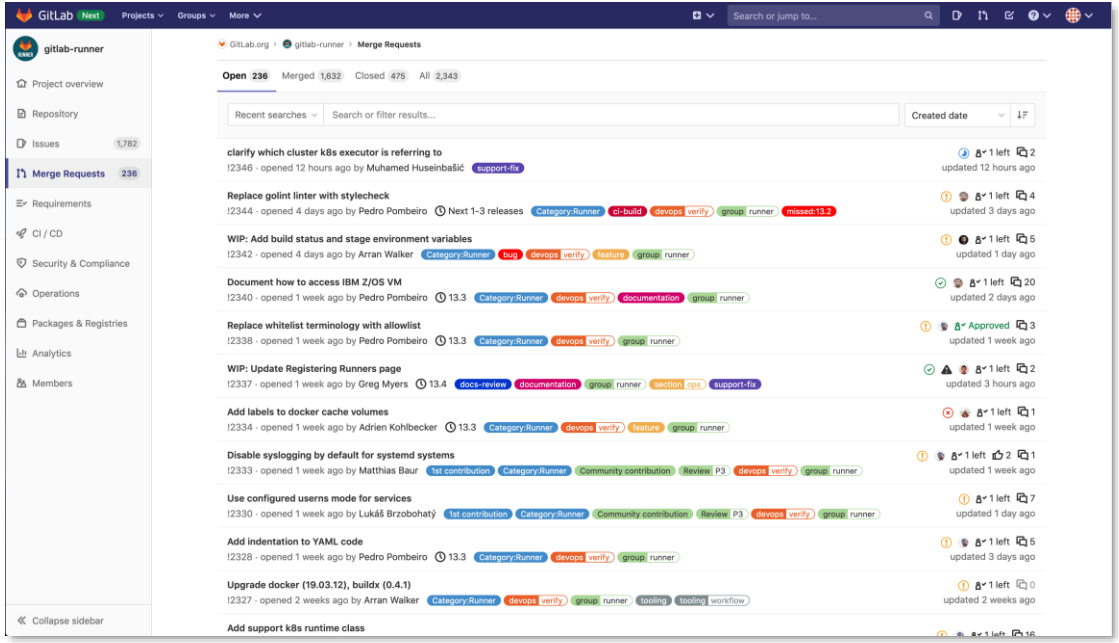
- Used for collaborating on ideas and planning work
- Enable sharing and discussion of ideas/proposals
- Track status of work and tasks



GitLab

Merge Requests

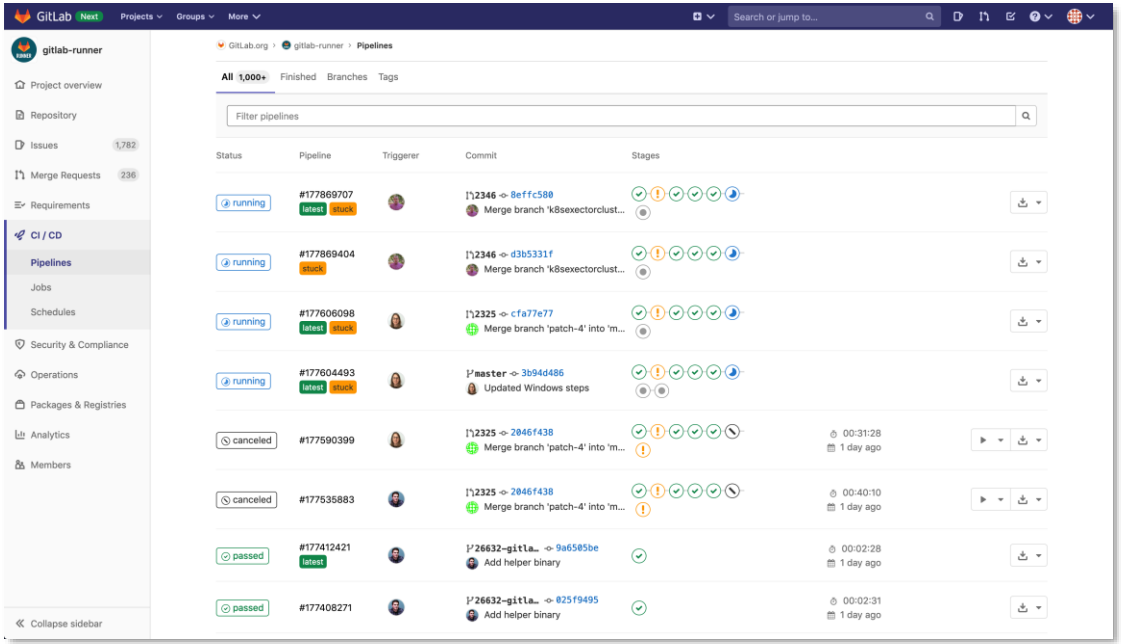
- Request to merge one branch into another



GitLab

Pipelines

- Fundamental building blocks for CI/CD



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

```yaml
stages:
  - build
  - test
  - deploy

image: alpine

compile_a:
  stage: build
  script:
    - echo "This job builds something."

compile_b:
  stage: build
  script:
    - echo "This job builds something else."

unit_test_a:
  stage: test
  script:
    - echo "This job tests something. It will only run when all jobs in the"
    - echo "build stage are complete."

unit_test_b:
  stage: test
  script:
    - echo "This job tests something else. It will only run when all jobs in the"
    - echo "build stage are complete too. It will start at about the same time as test_a."

deploy_a:
  stage: deploy
  script:
    - echo "This job deploys something. It will only run when all jobs in the"
    - echo "test stage complete."

deploy_b:
  stage: deploy
  script:
    - echo "This job deploys something else. It will only run when all jobs in the"
    - 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

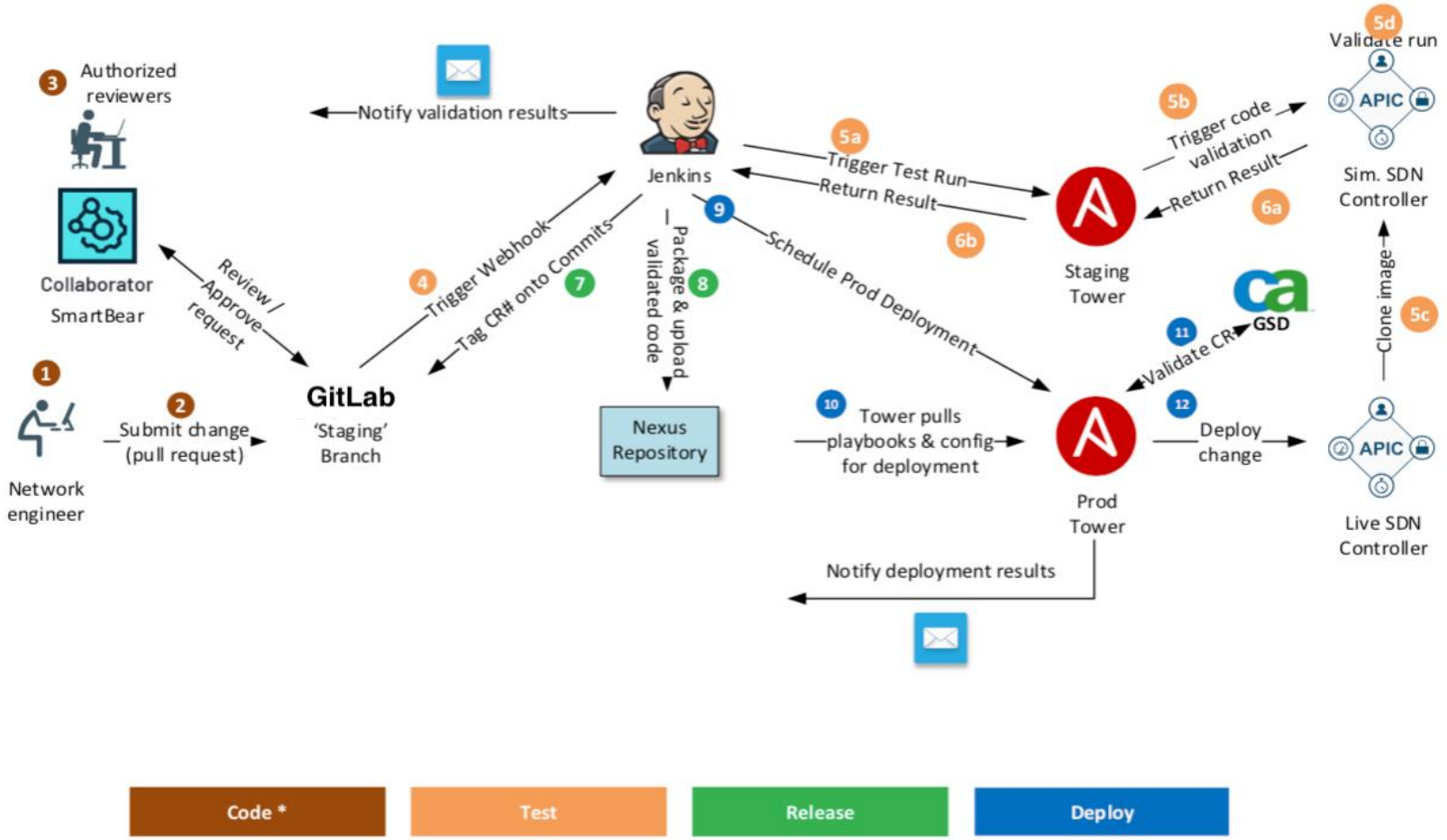
GitLab

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.g
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="longalfanumstring"
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

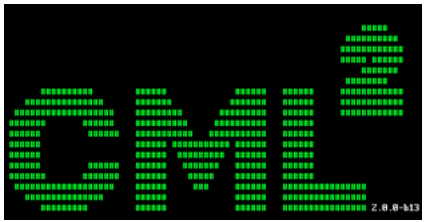


Demo

Test environment

Dev Workstation

Hosting Server



Network Simulation

Infrastructure as Code

Source Control

Testing Framework

Camtasia 2022 File Edit Modify Text View Export Window Help

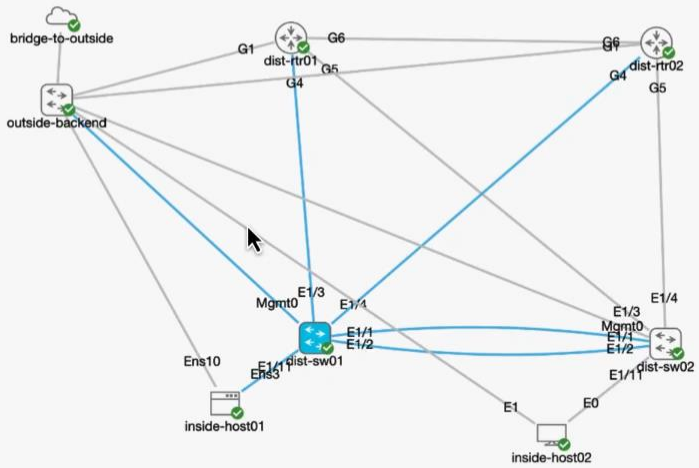
Files · main · developer develop | NXOS/IOSXE Network :: CML²

https://10.10.10.33/lab/8e1463b8-2d5b-4596-92a4-7c8e5af4b2e9

110%

Cisco Modeling Labs Workbench NXOS/IOSXE Network

DASHBOARD TOOLS ADMIN



ADD NODES

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% Notifications 0 Status OK

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.


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN