

Continuous Delivery with Containers

Elizabeth K. Joseph
@pleia2

All Things Open 2018

Elizabeth K. Joseph

@pleia2

lyz@princessleia.com

Spent the past 2 years working
containers with Apache Mesos
and Kubernetes

4 years working on CI/CD for
OpenStack

10+ years in Linux systems
administration and engineering
roles

Author of The Official Ubuntu
Book and Common OpenStack
Deployments

Definition: Continuous Delivery

Continuous Delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.

Via https://en.wikipedia.org/wiki/Continuous_delivery

Traditional Delivery

Months (or years!) between releases

Customers don't see new features quickly

Developers lose track of features they worked on

Goal: A Modern Release Process with CD

Week 1: Project planning and release

Develop > Test > Stage > Release

Week 2: Customer Feedback and release

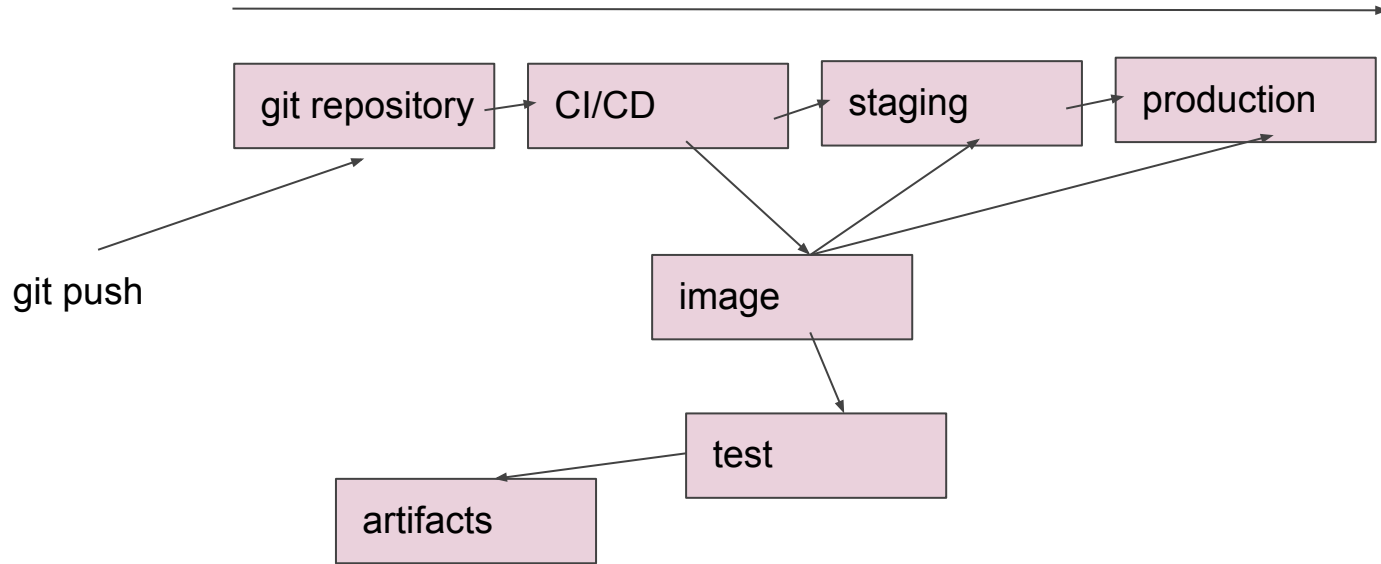
Develop > Test > Stage > Release

Week 3: Customer Feedback and release

Develop > Test > Stage > Release

...

The CI/CD Pipeline



How?

Run in containers!



Organize everything efficiently!



CC BY-ND 2.0: <https://www.flickr.com/photos/96227967@N05/24954030641/>

Sidebar: Can't I just use VMs for testing?

Sure! But consider...

- VMs take longer to provision
- You may not need all they provide (kernel, system libraries...)
- An identical container image is simple to run in development, testing, staging and production



Bare Metal/VMs vs. Containers

Traditional Datacenter

Siloed servers

Low utilization

12-15% for bare metal

30% for virtual machines

Containerization Platform

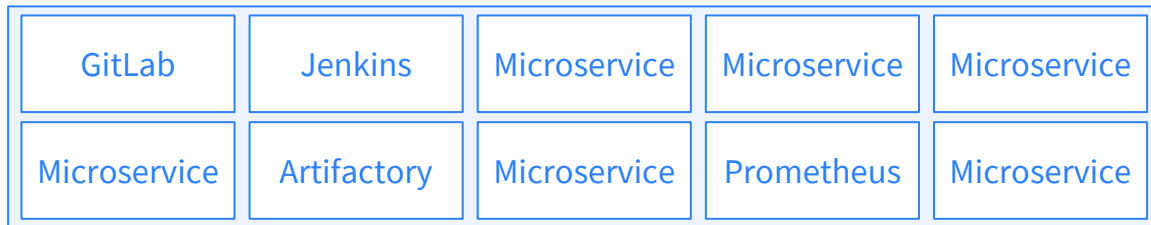
Integrated cluster (service discovery, authentication, etc)

Workload multiplexing on the same machines

Does everything go in containers? Up to you.

Everything Running in Containers

Services & Containers



Kubernetes, Docker Swarm, Apache Mesos



Bare Metal, OpenStack, AWS, Azure, GCP

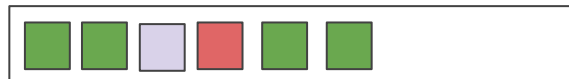


Everything Running in Containers

Traditional Datacenter



Containerization Platform

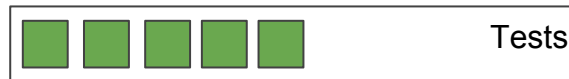


Legacy Infrastructure + Containers

Continue running your legacy systems on Bare Metal or VMs



Send all tests to a new, independent containerization platform



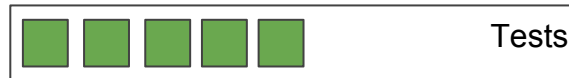
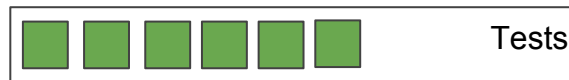
Legacy Infrastructure + hosted + Containers

Continue running your legacy systems on Bare Metal or VMs



GitHub

Send all tests to a new, independent containerization platform



Fully hosted with GitHub

Hosted CI/CD system

CircleCI

Travis CI

TeamCity

...

GitHub

Test on cloud-hosted
Kubernetes service:

- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)
- Amazon Elastic Container Service for Kubernetes (Amazon EKS)

Fully hosted with GitLab.com

Use GitLab.com (hosted GitLab) for repository, artifact store, test runner



GitLab

And connect it to a cloud-hosted Kubernetes service:

- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)
- Amazon Elastic Container Service for Kubernetes (Amazon EKS)

Or a combination of these!

Walkthrough:



Setting up a pipeline on GitLab with Auto DevOps and Kubernetes

Visit https://docs.gitlab.com/ee/topics/autodevops/quick_start_guide.html

Select a project template

Projects




New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose **CI/CD for external repo**.

Tip: You can also create a project from the command line. [Show command](#)

Blank project	Create from template	Import project	CI/CD for external repo
 Ruby on Rails Includes an MVC structure, Gemfile, Rakefile, along with many others, to help you get started.	<input type="button" value="Use template"/>	<input type="button" value="Preview"/>	
 Spring Includes an MVC structure, mvnw and pom.xml to help you get started.	<input type="button" value="Use template"/>	<input type="button" value="Preview"/>	
 NodeJS Express Includes an MVC structure to help you get started.	<input type="button" value="Use template"/>	<input type="button" value="Preview"/>	

Set up project with template

Projects

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).


All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose **CI/CD for external repo**.

Tip: You can also create a project from the command line. [Show command](#)

Blank project	Create from template	Import project	CI/CD for external repo
---------------	-----------------------------	----------------	-------------------------

Template

 Ruby on Rails Change template

Project name

rails-autodevops

Project URL Project slug

https://gitlab.com/ pleia2 rails-autodevops

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Description format

Visibility Level ⓘ

- Private
Project access must be granted explicitly to each user.
- Internal
The project can be accessed by any logged in user.
- Public
The project can be accessed without any authentication.

Create project Cancel

Enable Kubernetes in Google Cloud

Quickstart

This quickstart shows you how to deploy a containerized application with Google Kubernetes Engine.

Before you begin

Take the following steps to enable the Kubernetes Engine API:

1. Visit the [Kubernetes Engine page](#) in the Google Cloud Platform Console.
2. Create or select a project.
3. Wait for the API and related services to be enabled. This can take several minutes.
4. Make sure that billing is enabled for your project.

[LEARN HOW TO ENABLE BILLING](#)

“Add Kubernetes Cluster”

 rails-autodevops  Public  Add license

Project ID: 8964321

0  Star 0  Fork SSH  git@gitlab.com:pleia2/r    +   Global 

[Readme](#) [Files \(143 KB\)](#) [Commit \(1\)](#) [Branch \(1\)](#) [Tags \(0\)](#) [Security Dashboard](#)

Add Changelog

Add Contribution guide

Add Kubernetes cluster

Set up CI/CD



Create new Cluster on GKE

Elizabeth K. Joseph > rails-autodevops > **Kubernetes**

Kubernetes cluster integration

With a Kubernetes cluster associated to this project, you can use review apps, deploy your applications, run your pipelines, and much more in an easy way.

Learn more about [Kubernetes](#).

If you are setting up multiple clusters and are using Auto DevOps, [read this first](#).

Create new Cluster on GKE

Add existing cluster

Enter the details for your Kubernetes cluster

Please make sure that your Google account meets the following requirements:

- Your account must have [access to Google Kubernetes Engine](#)
- Make sure your account [meets the requirements](#) to create Kubernetes clusters
- This account must have permissions to create a Kubernetes cluster in the [Google Kubernetes Engine project](#) specified below



Sign in with Google

or create a new Google account

Enter the details for your Kubernetes cluster

Enter the details for your Kubernetes cluster

Please make sure that your Google account meets the following requirements:

- Your account must have [access to Google Kubernetes Engine](#)
- Make sure your account [meets the requirements](#) to create Kubernetes clusters
- This account must have permissions to create a Kubernetes cluster in the [Google Kubernetes Engine project](#) specified below

Read our [help page](#) on Kubernetes cluster integration.

[Select a different Google account](#)

Kubernetes cluster name

Environment scope

Google Cloud Platform project

To use a new project, first create one on [Google Cloud Platform](#).

Zone

Learn more about [zones](#).

Number of nodes

Machine type

Learn more about [machine types](#) and [pricing](#).

The cluster exists!

My Project 71888



Kubernetes clusters

[+ CREATE CLUSTER](#)

[+ DEPLOY](#)

[REFRESH](#)

[DELETE](#)

A Kubernetes cluster is a managed group of uniform VM instances for running Kubernetes. [Learn more](#)

Filter by label or name

<input type="checkbox"/> Name ^	Location	Cluster size	Total cores	Total memory	Notifications	Labels
<input type="checkbox"/>  autodevops	us-central1-a	3	6 vCPUs	22.50 GB	 Low resource requests	Connect  

Enable Applications

Applications

Choose which applications to install on your Kubernetes cluster. Helm Tiller is required to install any of the following applications. [More information](#)



Helm Tiller

Helm streamlines installing and managing Kubernetes applications. Tiller runs inside of your Kubernetes Cluster, and manages releases of your charts.

Installed



Ingress

Ingress gives you a way to route requests to services based on the request host or path, centralizing a number of services into a single endpoint.

Installing

Note: This will add some extra resources like a load balancer, which may incur additional costs depending on the hosting provider your Kubernetes cluster is installed on. If you are using Google Kubernetes Engine, you can [check the pricing here](#).

After installing Ingress, you will need to point your wildcard DNS at the generated external IP address in order to view your app after it is deployed. [More information](#)



Prometheus

Prometheus is an open-source monitoring system with [GitLab Integration](#) to monitor deployed applications.

Install



GitLab Runner

GitLab Runner connects to this project's repository and executes CI/CD jobs, pushing results back and deploying, applications to production.

Install



JupyterHub

JupyterHub, a multi-user Hub, spawns, manages, and proxies multiple instances of the single-user Jupyter notebook server. JupyterHub can be used to serve notebooks to a class of students, a corporate data science group, or a scientific research group.

Install

Enable GitLab Auto DevOps

R rails-autodevops

- Project
- Repository
- Issues 0
- Merge Requests 0
- CI / CD
- Operations
- Registry
- Packages
- Wiki
- Snippets
- Settings**
 - General
 - Members
 - Integrations
 - Repository
 - CI / CD**
 - Pages
 - Audit Events

Elizabeth K. Joseph > rails-autodevops > CI / CD Settings

General pipelines

Expand

Customize your pipeline configuration, view your pipeline status and coverage report.

Auto DevOps

Collapse

Auto DevOps will automatically build, test, and deploy your application based on a predefined Continuous Integration and Delivery configuration. [Learn more about Auto DevOps](#)

Auto Review Apps and Auto Deploy need a domain name to work correctly.

Default to Auto DevOps pipeline

The Auto DevOps pipeline will run if no alternative CI configuration file is found. [More information](#)

Domain

35.194.19.243.nip.io

You need to specify a domain if you want to use Auto Review Apps and Auto Deploy stages. `35.194.19.243.nip.io` can be used as an alternative to a custom domain. [?](#)

Do not set up a domain here if you are setting up multiple Kubernetes clusters with Auto DevOps. [?](#)

Deployment strategy

Deployment strategy needs a domain name to work correctly.

- Continuous deployment to production [?](#)
- Continuous deployment to production using timed incremental rollout [?](#)
- Automatic deployment to staging, manual deployment to production [?](#)

Save changes

View pipelines

GitLab Projects Groups Activity Milestones Snippets Search or jump to...

R rails-autodevops

Project
Repository
Issues 0
Merge Requests 0
CI / CD
Pipelines
Jobs
Schedules
Charts
Operations
Registry
Packages

Elizabeth K. Joseph > rails-autodevops > Pipelines

All 1 Pending 0 Running 1 Finished 0 Branches Tags Run Pipeline Clear Runner Caches CI Lint

Status	Pipeline	Commit	Stages
running	#33701282 by latest Auto DevOps	master -> 816a358c Rails template	

A pipeline!

- 0 Merge Requests
- CI / CD
- Pipelines**
- Jobs
- Schedules
- Charts
- Operations
- Registry
- Packages
- Wiki
- Snippets
- Settings

9 jobs from `master` (queued for 2 seconds)

816a358c

Pipeline Jobs 9

Build

✓ build

Test

✓ code_quality

✓ container_scan...

✓ dependency_s...

✓ license_manag...

✓ sast

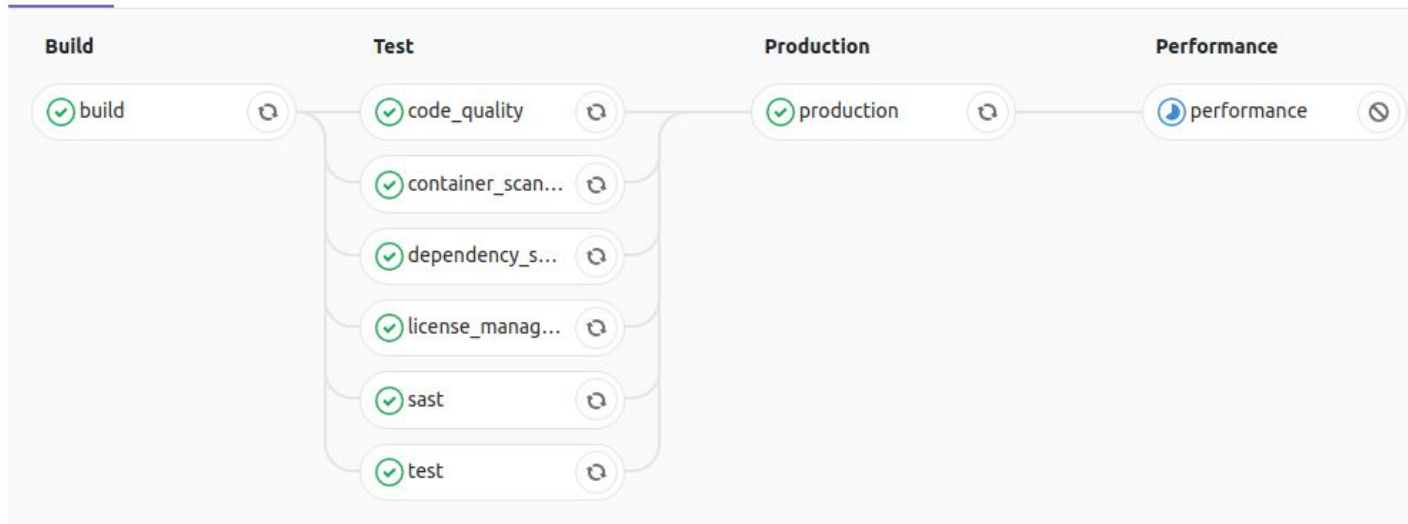
✓ test

Production

✓ production

Performance

🔄 performance



What do these all mean?

In the **build** stage, the application is built into a Docker image and then uploaded to your project's [Container Registry \(Auto Build\)](#).

In the **test** stage, GitLab runs various checks on the application:

- The `test` job runs unit and integration tests by detecting the language and framework ([Auto Test](#))
- The `code_quality` job checks the code quality and is allowed to fail ([Auto Code Quality](#)) ?
- The `container_scanning` job checks the Docker container if it has any vulnerabilities and is allowed to fail ([Auto Container Scanning](#))
- The `dependency_scanning` job checks if the application has any dependencies susceptible to vulnerabilities and is allowed to fail ([Auto Dependency Scanning](#)) ?
- The `sast` job runs static analysis on the current code to check for potential security issues and is allowed to fail ([Auto SAST](#)) ?
- The `license_management` job searches the application's dependencies to determine each of their licenses and is allowed to fail ([Auto License Management](#)) ?

Note: As you might have noticed, all jobs except `test` are allowed to fail in the test stage.

The **production** stage is run after the tests and checks finish, and it automatically deploys the application in Kubernetes ([Auto Deploy](#)).

Lastly, in the **performance** stage, some performance tests will run on the deployed application ([Auto Browser Performance Testing](#)). ?

Via: https://docs.gitlab.com/ee/topics/autodevops/quick_start_guide.html#deploying-the-application

Navigate to deployed application

GitLab Projects Groups Activity Milestones Snippets Search or jump to...

rails-autodevops


Project
Repository
Issues 0
Merge Requests 0
CI / CD
Operations
Metrics
Environments
Kubernetes
Feature Flags
Registry

Elizabeth K. Joseph > rails-autodevops > Pipelines > Environments

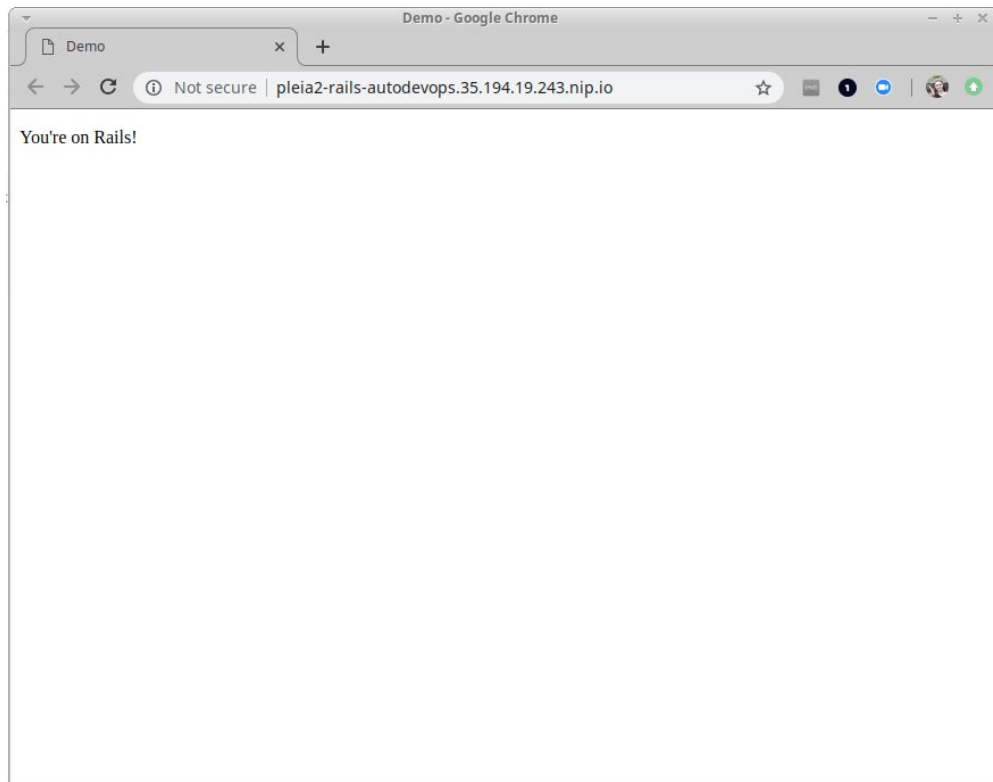
Available 1 Stopped 0 [New environment](#)

Environment	Deployment	Job	Commit	Updated
production	#1 by	production #110618148	816a358c Rails template	21 minutes ago

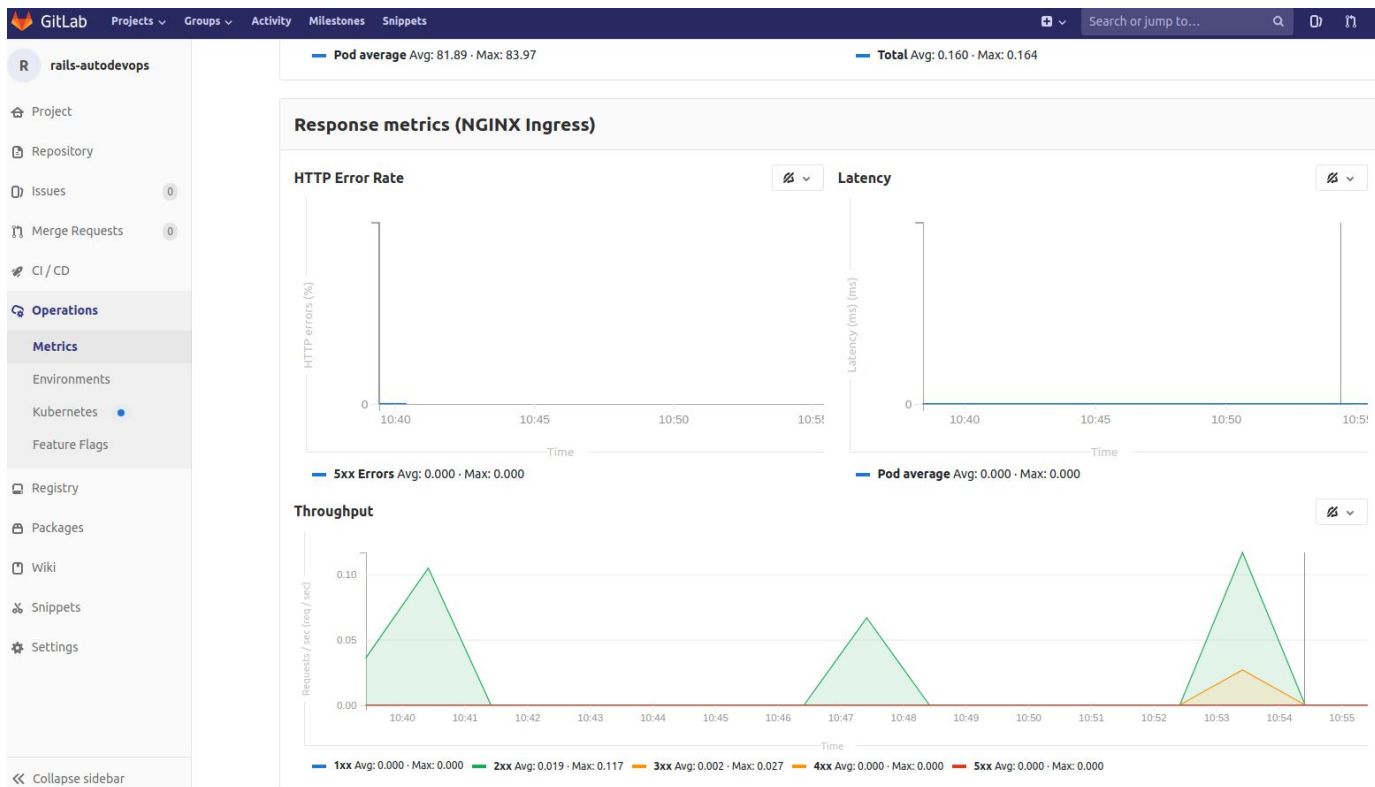
Instance (1)
100% Complete



Simple site in production!



Metrics! Via Prometheus in the GitLab UI



Make a change

Elizabeth K. Joseph > rails-autodevops > Merge Requests > 11

Open Opened in 28 seconds by  Elizabeth K. Joseph

Edit

Close merge request

Webtext

Update the web text to welcome ATO attendees.

 Request to merge `webtext` into `master`

Open in Web IDE

Check out branch



 Pipeline #33703055 running for 0518af1e on webtext



 No Approval required

 Merge when pipeline succeeds Remove source branch Squash commits  Modify commit message

You can merge this merge request manually using the [command line](#)

 0  0 

Discussion 0 [Commits](#) 2 [Pipelines](#) 2 [Changes](#) 2



Write [Preview](#)

B *I* **¶** **<>**     

Write a comment or drag your files here...

Tests run again!

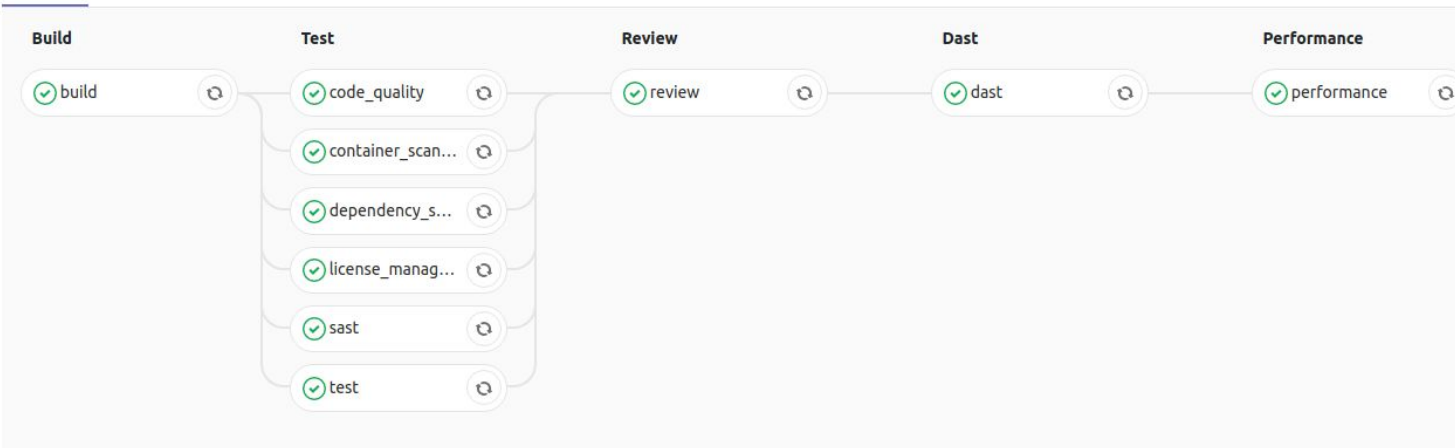
passed Pipeline #33703055 triggered 40 minutes ago by Elizabeth K. Joseph

Update welcome_controller_test.rb

21 jobs from [webtext](#) in 15 minutes and 43 seconds

[0518af1e](#)

Pipeline Jobs 21



View console details of jobs running

The screenshot displays the GitLab CI/CD interface. The left sidebar shows the project structure with 'Jobs' selected. The main console area shows the output of a 'review' job, including deployment status, resource usage, and pod details. The right sidebar provides additional job information such as duration, timeout, and runner details.

```
Create secret...
$ deploy
Release "review-webtext-ts54de" has been upgraded. Happy Helming!
LAST DEPLOYED: Sat Oct 20 18:14:35 2018
NAMESPACE: rails-autodevops-8964321
STATUS: DEPLOYED

RESOURCES:
==> v1/Secret
NAME                                     AGE
review-webtext-ts54de-postgres          11m

==> v1/PersistentVolumeClaim
review-webtext-ts54de-postgres          11m

==> v1/Service
review-webtext-ts54de-postgres          11m
review-webtext-ts54de-auto-deploy      11m

==> v1beta1/Deployment
review-webtext-ts54de-postgres          11m
review-webtext-ts54de                   11m

==> v1beta1/Ingress
review-webtext-ts54de-auto-deploy      11m

==> v1/Pod(related)

NAME                                     READY   STATUS    RESTARTS   AGE
review-webtext-ts54de-postgres-9447f68cf-bc2cl  1/1     Running   0           11m
review-webtext-ts54de-64bb7db49d-bcctv         0/1     ContainerCreating  0           3s
review-webtext-ts54de-7f6dd9fd49-njhvr         0/1     Terminating  0           11m

NOTES:
Application should be accessible at: http://pleia2-rails-autodevops-review-webtext-ts54de.35.194.19.243.nip.io
Waiting for rollout to finish: 0 of 1 updated replicas are available...
deployment "review-webtext-ts54de" successfully rolled out
$ persist_environment_url
Uploading artifacts...
environment_url.txt: found 1 matching files
Uploading artifacts to coordinator... ok          id=110623382 responseStatus=201 Created token=brBb6Dxw
Job succeeded
```

review [Retry](#)

Merge Request: 11
Duration: 1 minute 37 seconds
Timeout: 1h (from project) ⓘ
Runner: shared-runners-manager-6.gitlab.com (#380987)

Job artifacts
[Download](#) [Browse](#)

Commit 0518af1e [🔗](#) 11
Update welcome_controller_test.rb

[🔗](#) Pipeline #33703055 from webtext
review

➔ [🟢 review](#)

Successfully deployed!


Status

Pipeline

Commit

Stages

passed

#33703055 by 

latest Auto DevOps

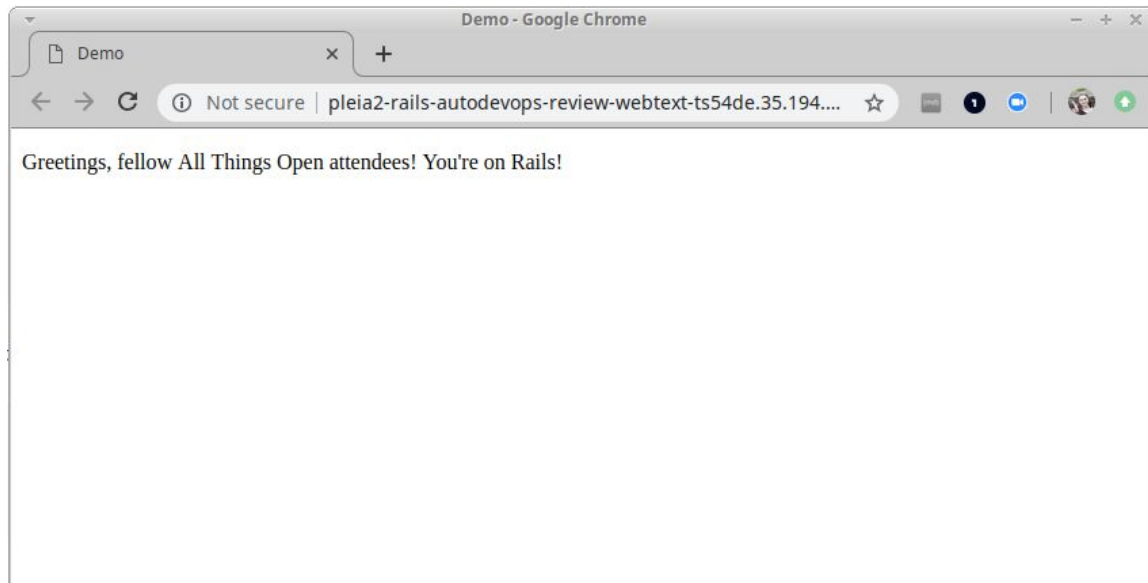
webtext -> 0518af1e

 Update welcome_controller...



00:15:43

19 seconds ago



Advanced CD Strategies

Advanced Strategies: Canary Deployments

“Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.”

<https://martinfowler.com/bliki/CanaryRelease.html>

Advanced Strategies: Blue/Green Deployments

— — —

“One of the challenges with automating deployment is the cut-over itself, taking software from the final stage of testing to live production. You usually need to do this quickly in order to minimize downtime. The blue-green deployment approach does this by ensuring you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live. As you prepare a new release of your software you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the router so that all incoming requests go to the green environment - the blue one is now idle.”

<https://martinfowler.com/bliki/BlueGreenDeployment.html>

Advanced tooling exists!

Many containerization platforms allow for enough deployment strategies to define a canary or blue/green strategy yourself.

Tooling like Vamp can also help.

<https://vamp.io/documentation/installation/v1.0.0/kubernetes/>

<https://vamp.io/documentation/installation/v1.0.0/dcos/>

Questions?

Elizabeth K. Joseph

@pleia2

lyz@princessleia.com

<https://princessleia.com/>