



Short Name

Short Description

Offering Type

Introduction

GitHub repository URL

Run GitLab on Kubernetes

Deploy a distributed GitLab leveraging Kubernetes and Docker

0 0 0 0

Last updated Mar 21, 2020

Containers are a major trend in deploying applications in both public and private clouds. The recent widespread adoption of containers is largely due to the development of standards that are aimed at making them easier to use, such as the Docker image format and distribution model. One of the key use cases for containers is moving legacy applications to container orchestration platforms like Kubernetes, which allows for better distribution, scaling, and maintenance.

By Pavan Mada, Arunachal Singh

Get the code

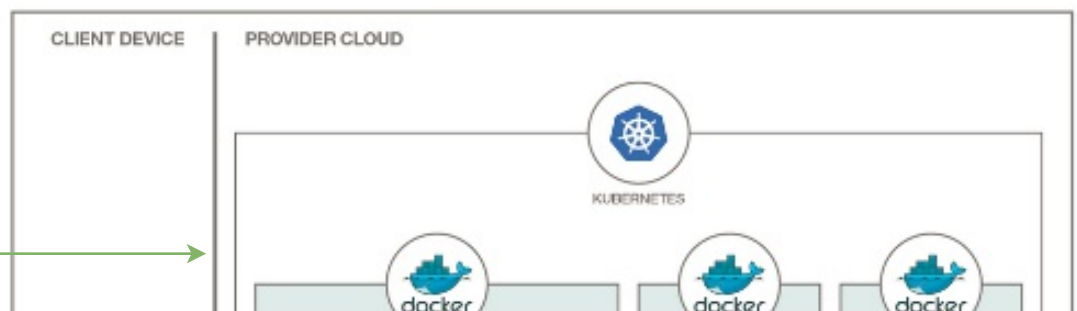
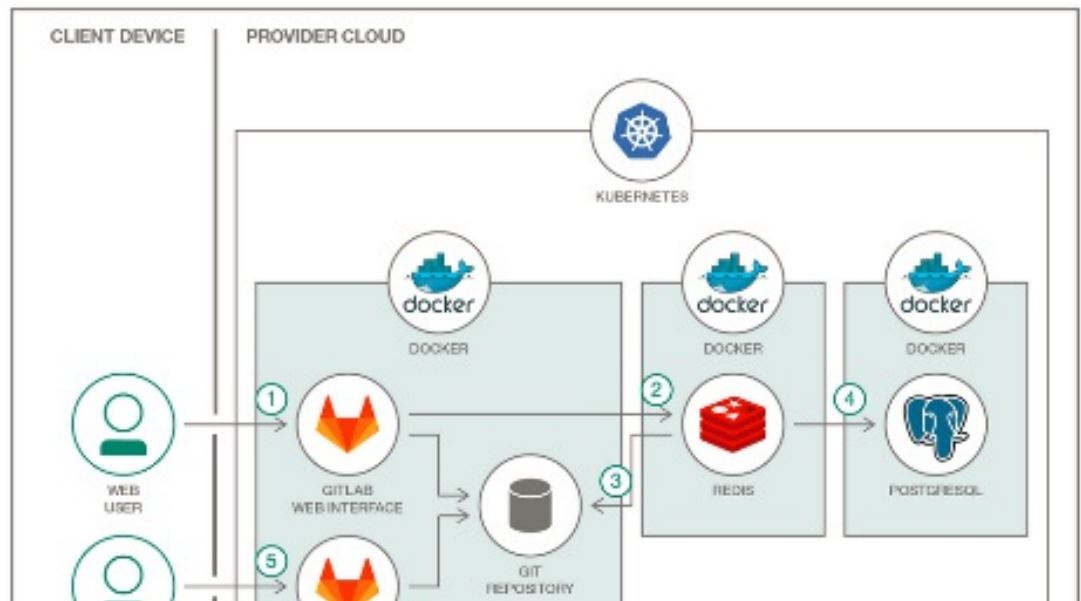
Introduction

Overview

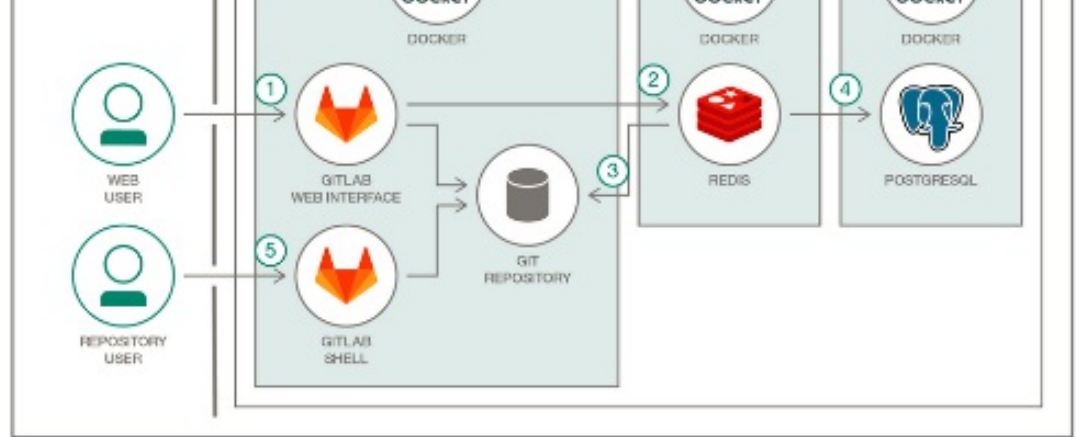
This application showcases the full power of containers and how we can move existing applications to the cloud seamlessly. We provide a full roadmap for developers, helping them to migrate their apps to the cloud and leverage cloud-native packaging by using containers. We show you how a common multi-component application can be deployed on the IBM Bluemix Container Service leveraging Kubernetes.

GitLab represents a typical multi-tier app and each component will have its own container(s). The microservice containers will be for the web tier and the data/DBs will be with Redis and PostgreSQL as the database.

Flow



Architecture Diagram



1. The user interacts with GitLab via the web interface or by pushing code to a GitHub repository. The GitLab container runs the main Ruby on Rails application behind NGINX and gitlab-workhorse, which is a reverse proxy for large HTTP requests like file downloads and Git push/pull. While serving repositories over HTTP(S), GitLab utilizes the GitLab API to receive authorization and access and serves Git objects.
2. After authentication and authorization, the GitLab Rails application puts the incoming jobs,
3. Repositories are created in a local file system.
4. The user creates users, roles, merge requests, groups, and more—all are then stored in PostgreSQL.
5. The user accesses the repository by going through the Git shell.

Flow Steps

Included components



IBM Bluemix Container Service

IBM Bluemix Container Service manages highly available apps inside Docker containers and Kubernetes clusters on the IBM Cloud.

→ [Read more](#)



Kubernetes cluster

Create and manage your own cloud infrastructure and use Kubernetes as your container orchestration engine.

→ [Read more](#)



GitLab

GitLab unifies issues, code review, CI, and CD into a single UI.

→ [Read more](#)



Redis

Redis is an open-source, in-memory data structure store, used as a database, cache and message broker.

→ [Read more](#)



PostgreSQL

Sophisticated open-source Object-Relational DBMS supporting almost all SQL constructs.

→ [Read more](#)

Featured technologies



Containers

Containers are virtual software objects that include all the elements that an app needs to run.

→ [Read more](#)

Related blogs

Deploy a distributed GitLab application on the Bluemix Kubernetes Container Service

by Animesh Singh · on Apr 03, 2017

Going "cloud native" is the trend these days and containers are at the heart of this. At its root, being cloud native means structuring teams, culture, and technology to utilize automation and architectures to manage complexity and unlock velocity. One of the key use cases for being cloud native is moving legacy applications to containers...

[Continue reading](#)

Related links

Kubernetes now available on IBM Bluemix Container Service

The IBM Bluemix Container Service now combines Docker and Kubernetes to deliver powerful tools, an intuitive user experience, and built-in security and isolation.

What's the secret? How to pull an image from a non-default Kubernetes namespace in IBM Bluemix Container Service

Use the IBM Bluemix registry to access IBM-provided public images or to set up your own Docker private image registry in Bluemix.



