KodeKloud

Sprint Goals

01

Creating a new GitHub repo

02

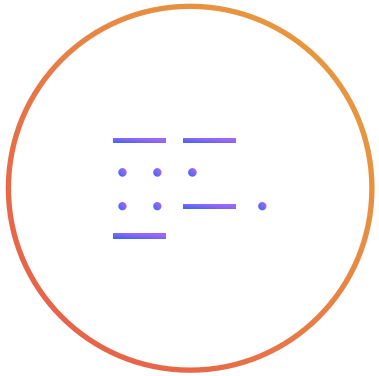
Setting up the repo according to the company's best practices

03

Coding

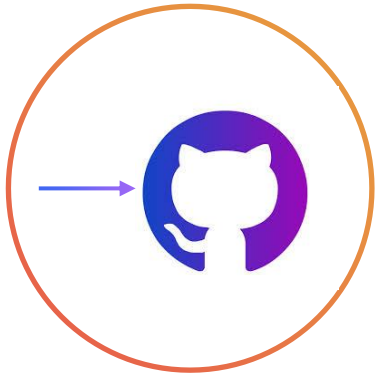
Writing a code for a simple docker image

04

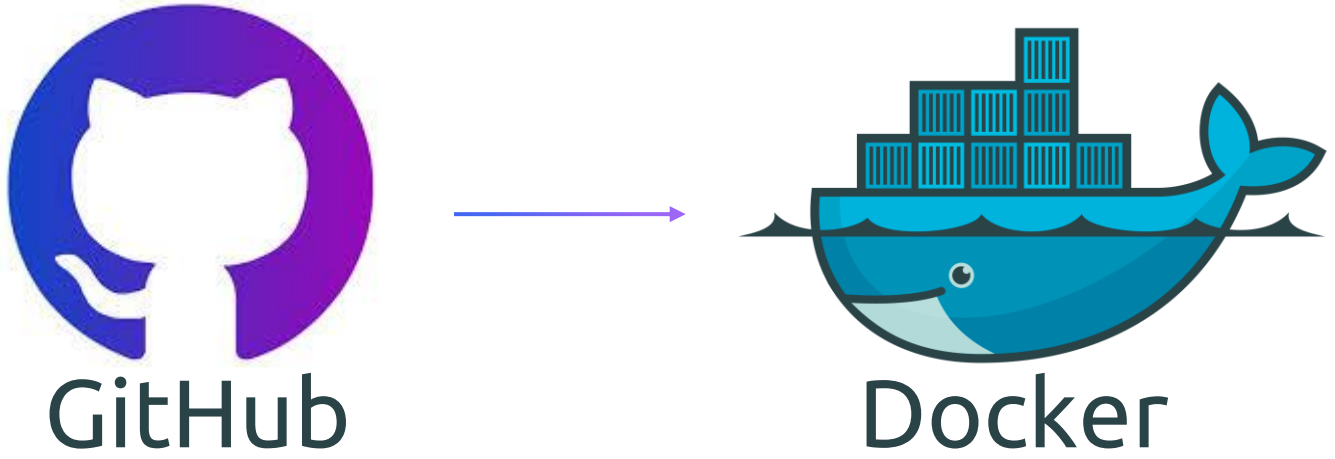


Testing the code locally

05



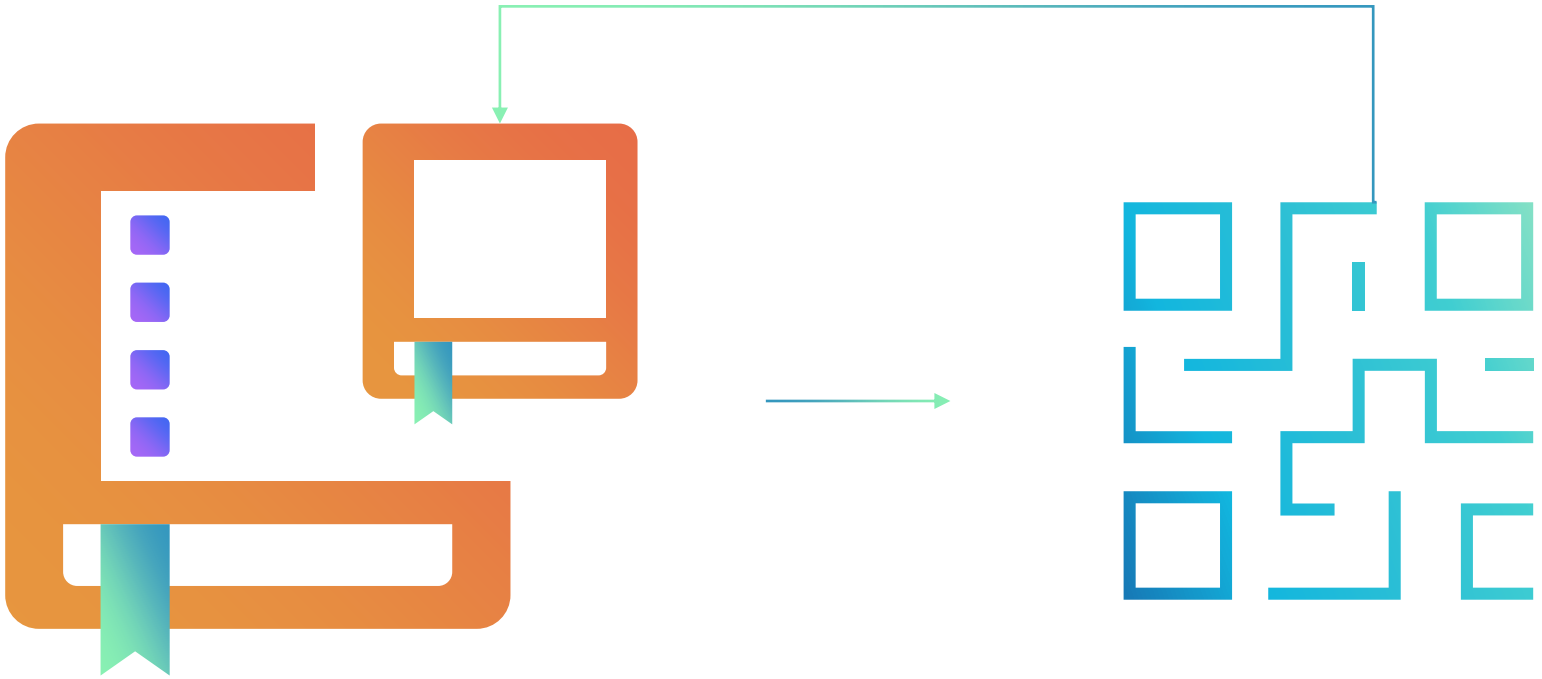
Pushing the code to the GitHub repo



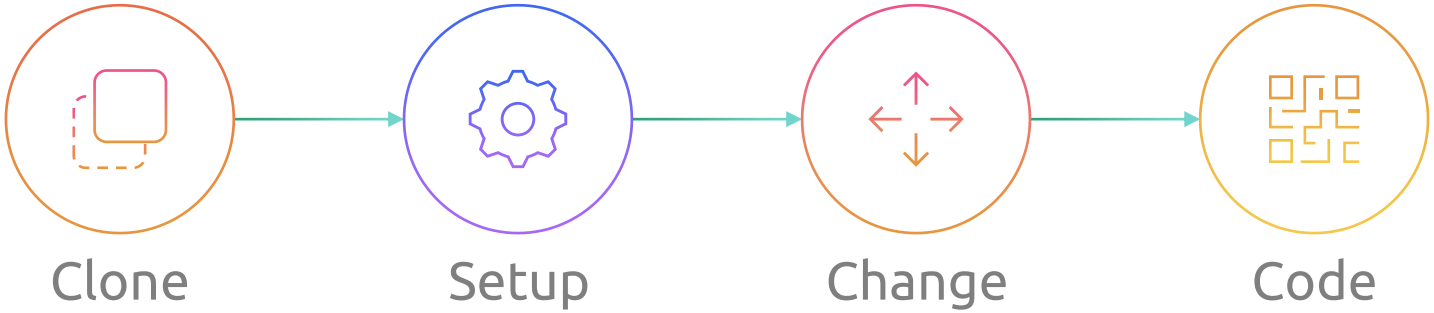
Task 1
Creating a GitHub Repo

Creating a GitHub Repository

Task 2
Cloning the repo and setting up our editor

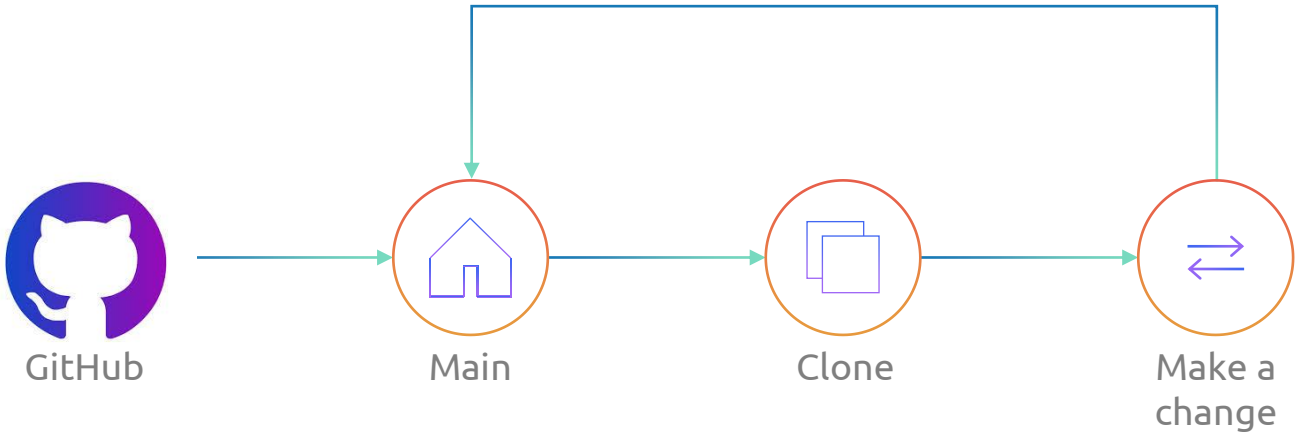


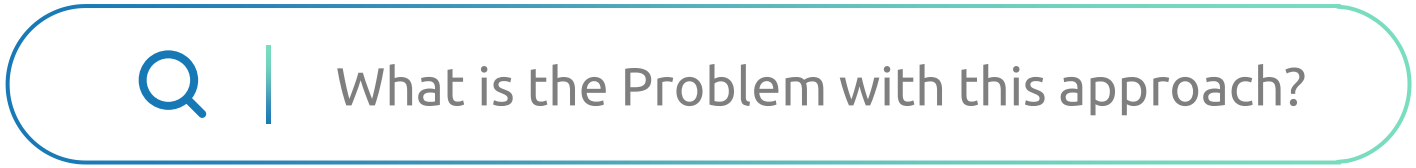
Summary

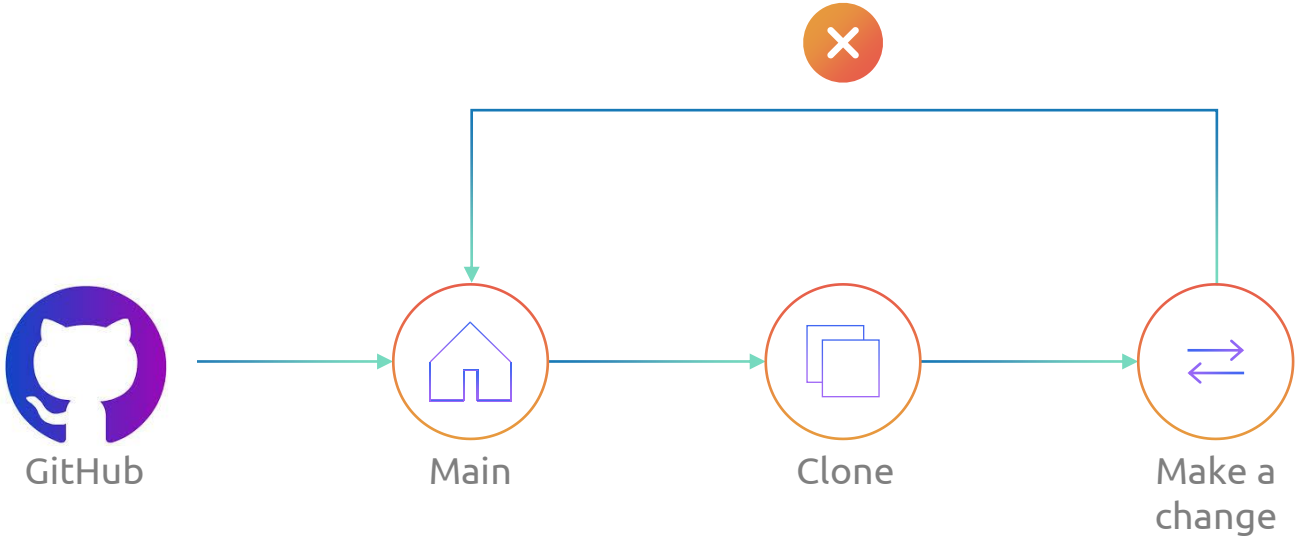


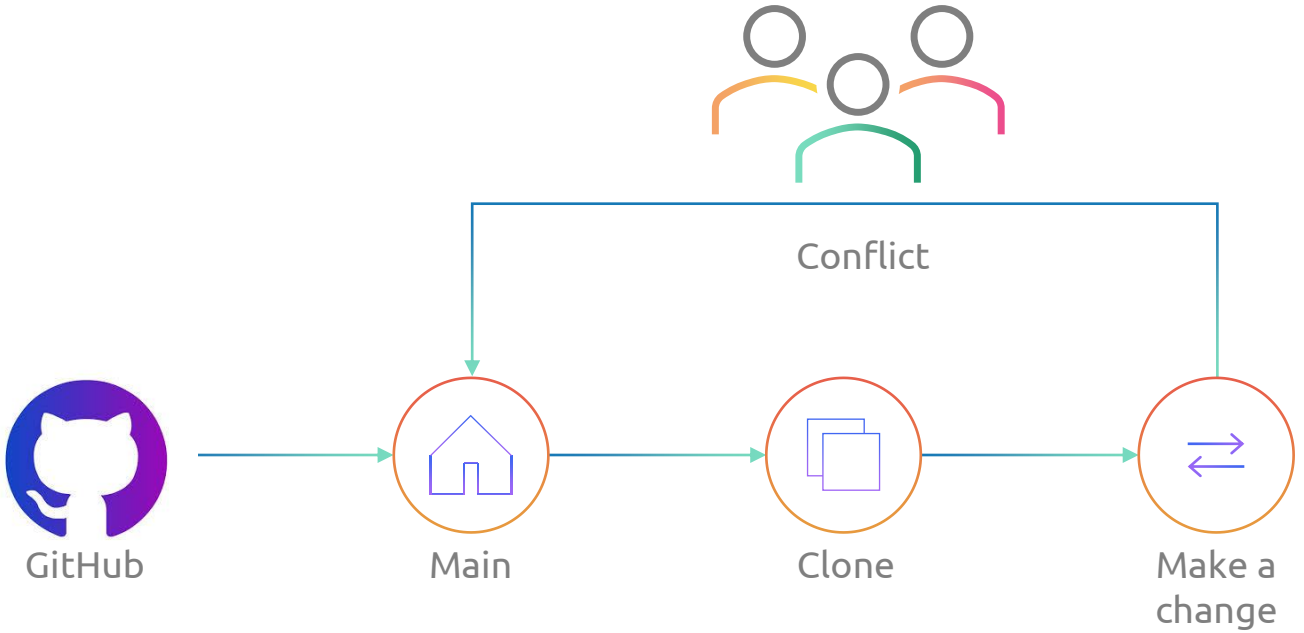
Task 3
Setting up GitHub repo according to DevOps best practices

Setting up GitHub repo according to DevOps best practices

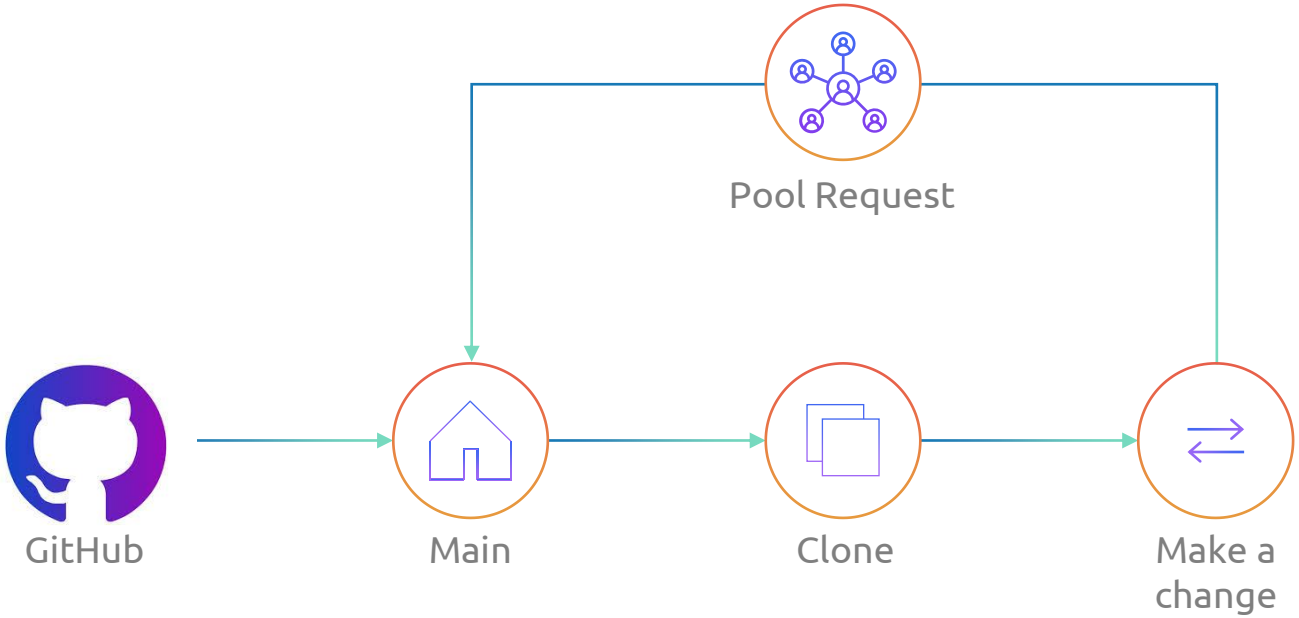


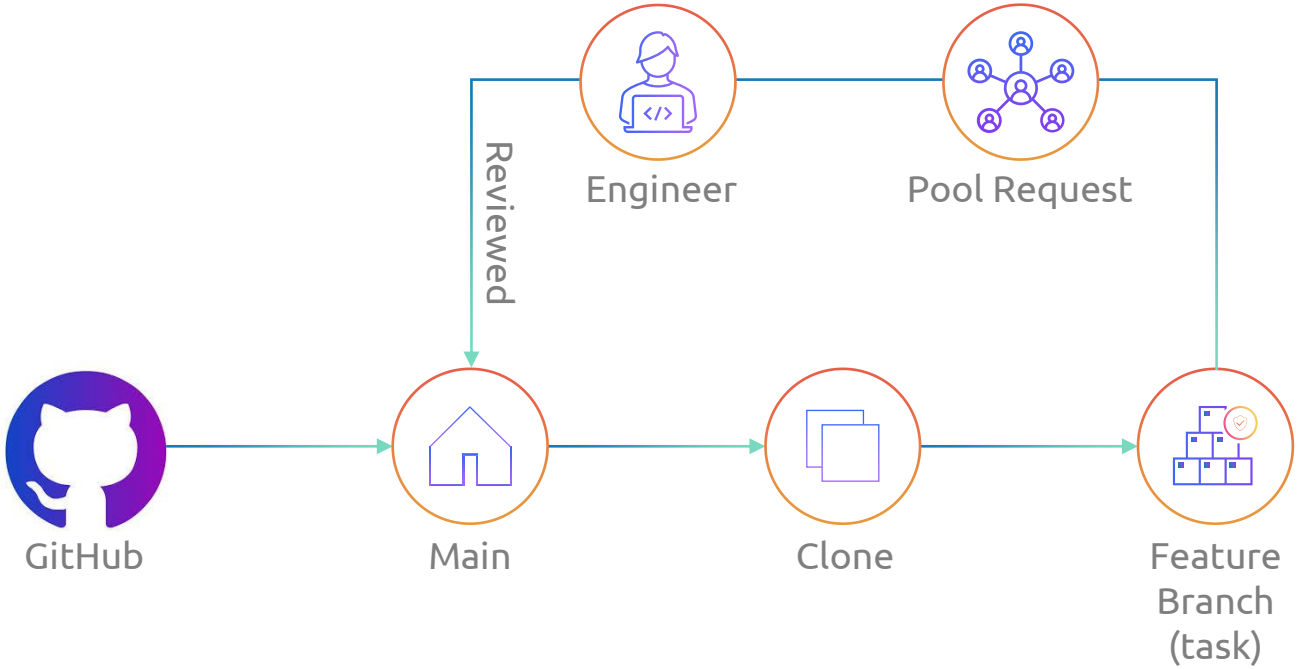




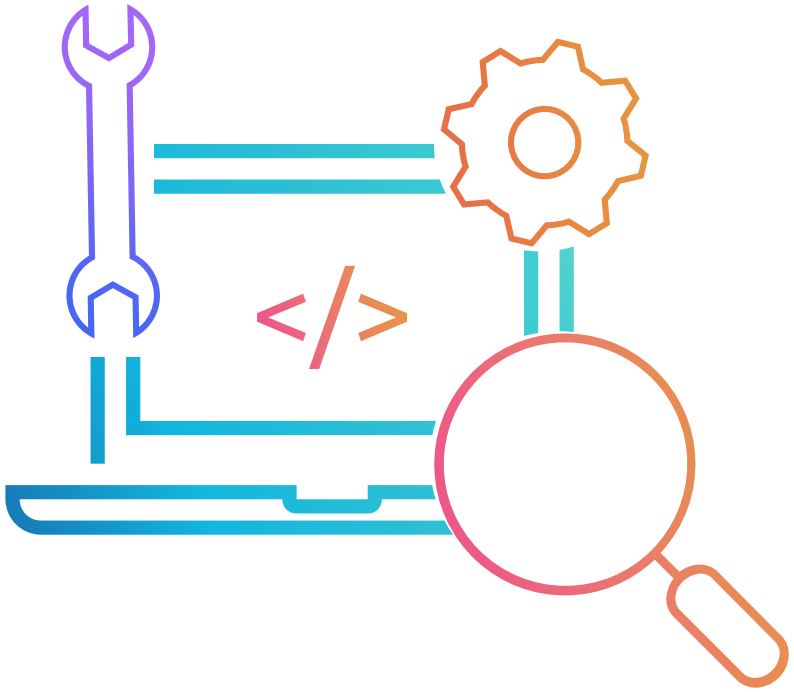


Branch Protection





Task 4
Coding for our application locally



Develop code locally and test it.

VS Code Editor

Demo

Testing, Debugging our code locally

Sprint-01
Review

Sprint Review in Organizations

- Creating a new GitHub repo
- Setting up the repo according to the company's best practices
- Coding: Writing a code for a simple docker image
- Testing the code locally
- Pushing the code to the GitHub repo

KodeKloud

Sprint-02

Sprint Goals

01

Creating a GCP account

02

Understanding the fundamentals of GKE in GCP

03

Setting up a GKE cluster

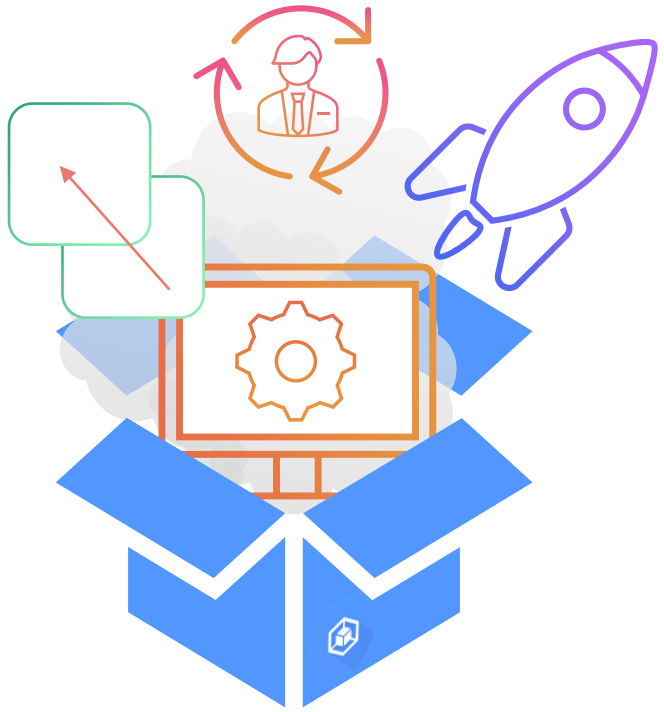
Sprint Goals

- Creating a GCP account
- Understanding the fundamentals of GKE in GCP
- Setting up a GKE cluster

Kubernetes Refresher

Google Cloud Platform

Definition of Kubernetes



Kubernetes, also known as K8s, is an open-source system for automating the

- deployment,
- scaling and
- management of containerized applications.

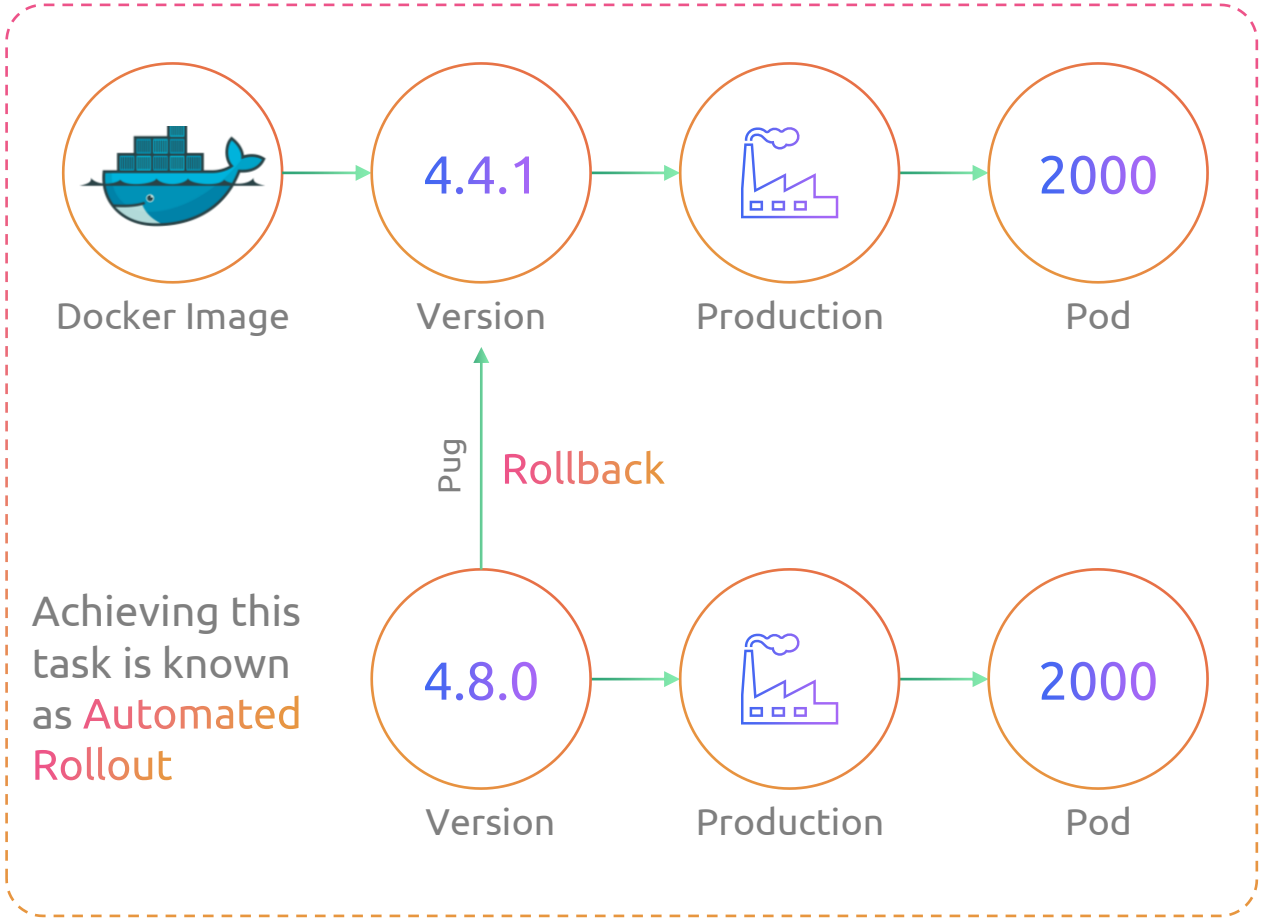
Deploy it

Scale it

Manage it

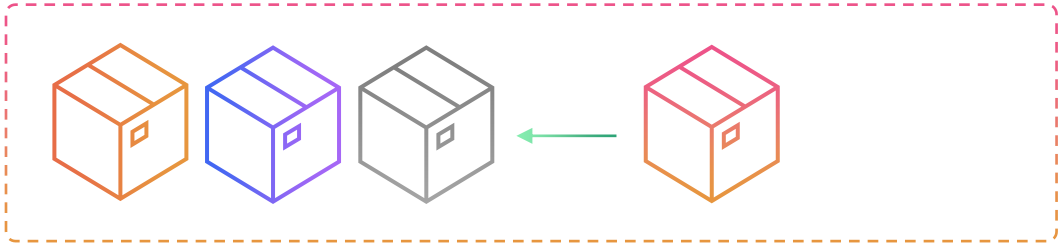
Kubernetes Features

01 Automated Rollouts and Rollbacks

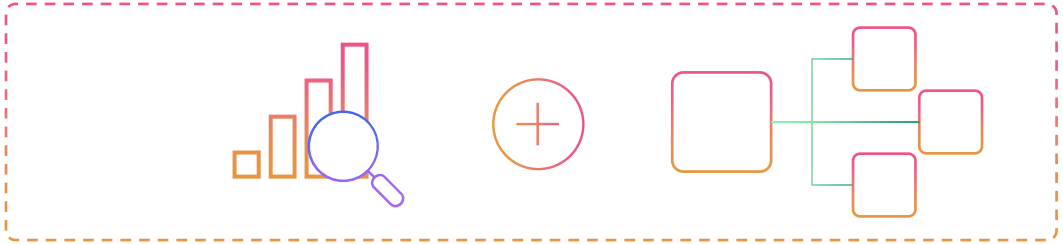


Kubernetes Features

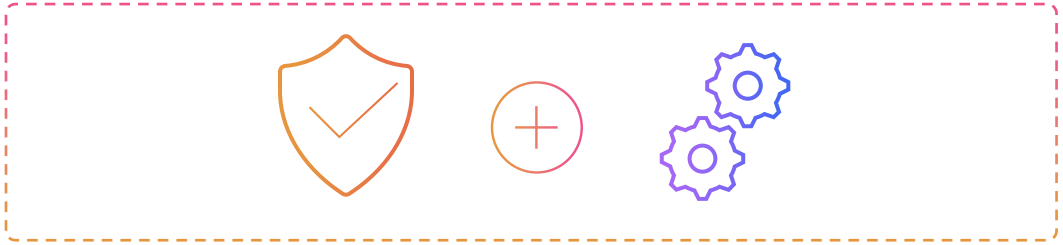
02 Self - Healing



03 Service Discovery and Load Balancing



04 Security and Configuration Management



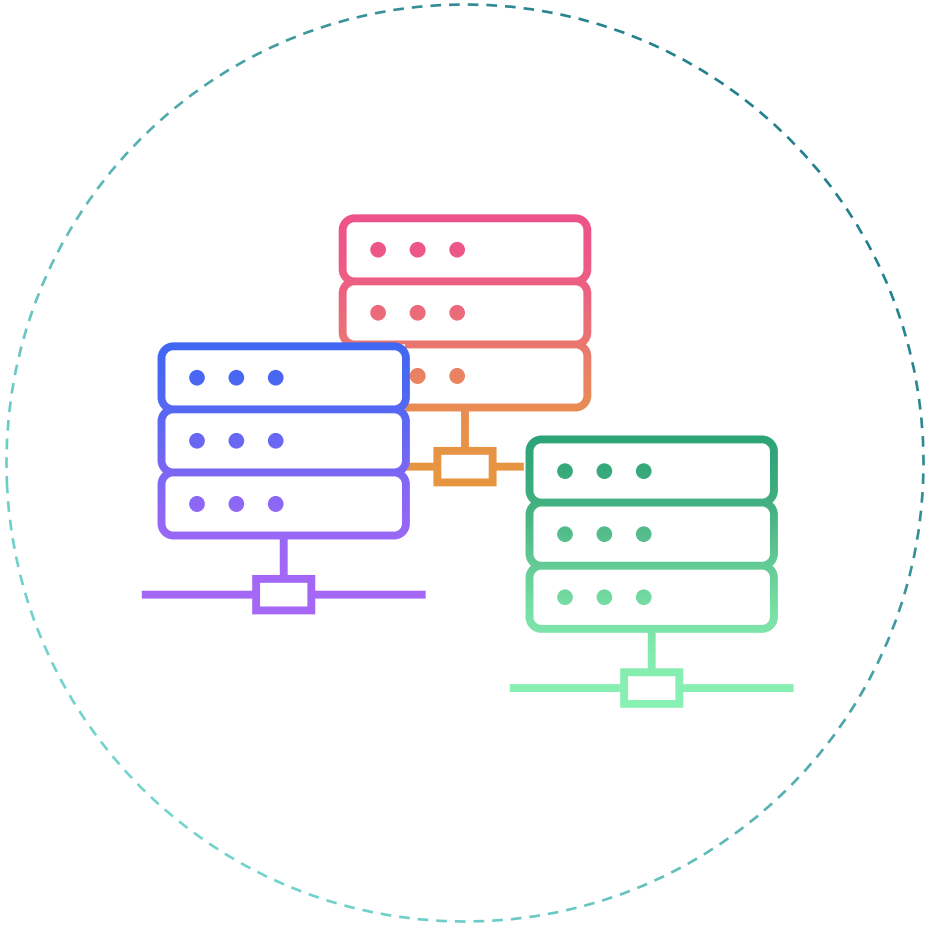
Cloud Options

AWS

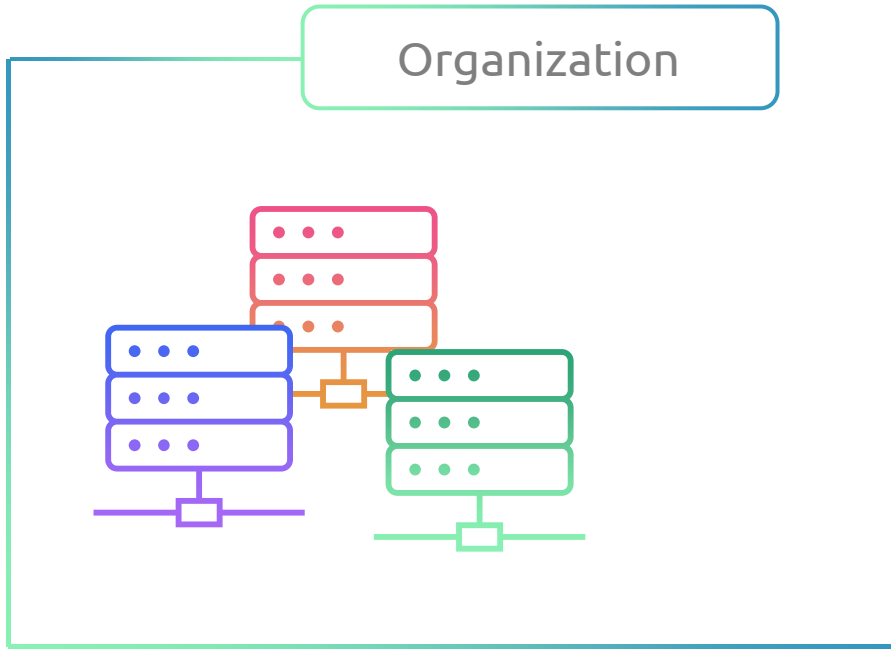
GCP

Microsoft Azure

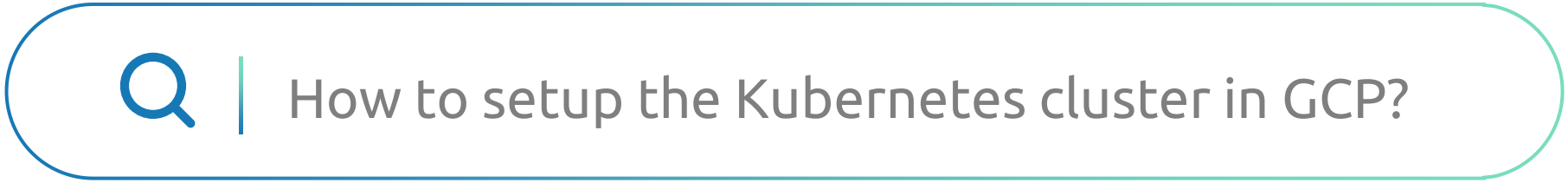
Own Datacentres



Pros & Cons



- Cost
- Big Team



Sprint-02 review

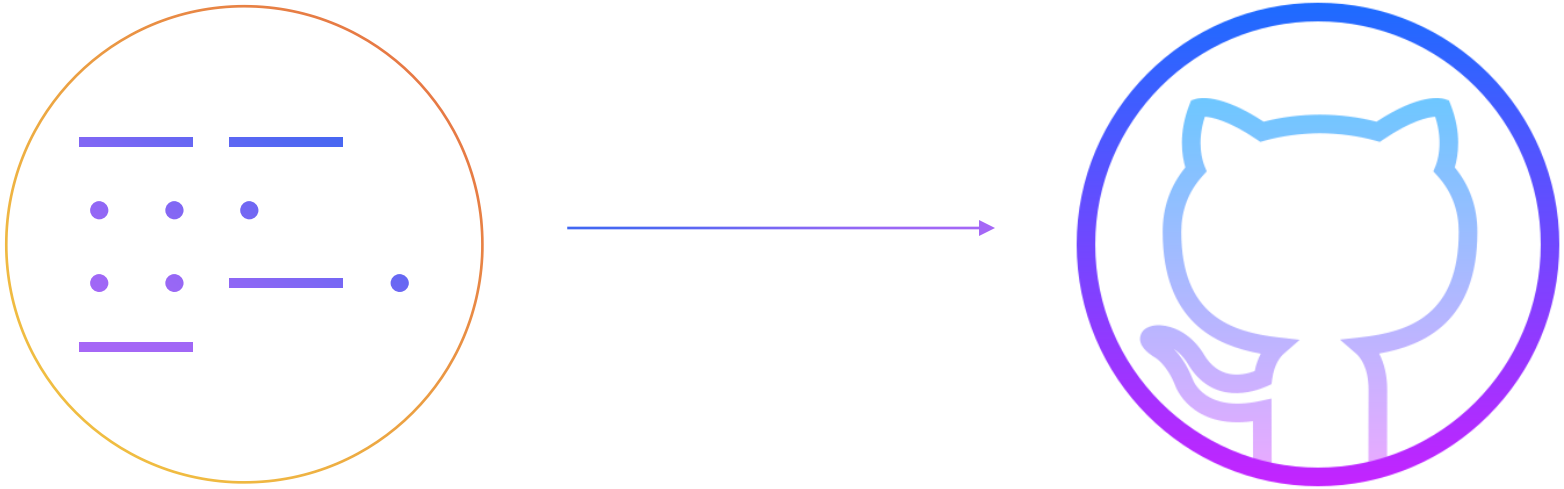
Sprint Review

- Creating a GCP account
- Understanding the fundamentals of GKE in GCP
- Setting up a GKE cluster

KodeKloud

Sprint-03

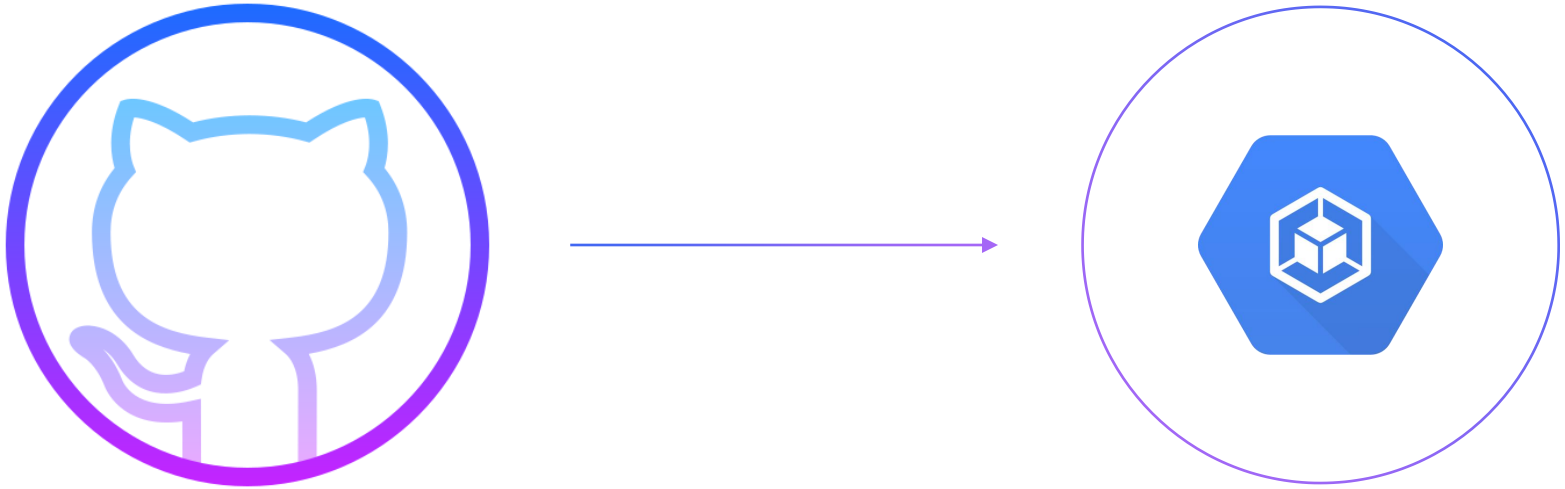
Design Discussion on CI/CD

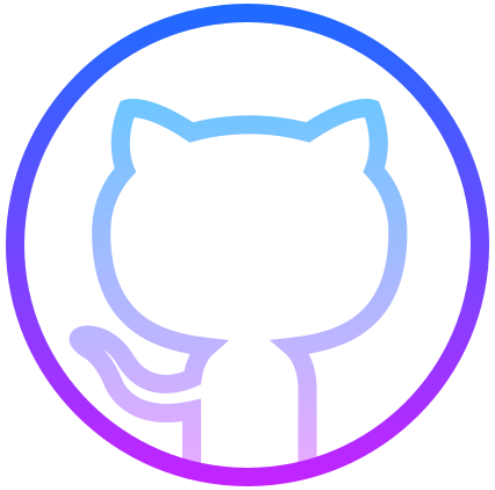


Google Cloud Platform

GKE

KodeKloud

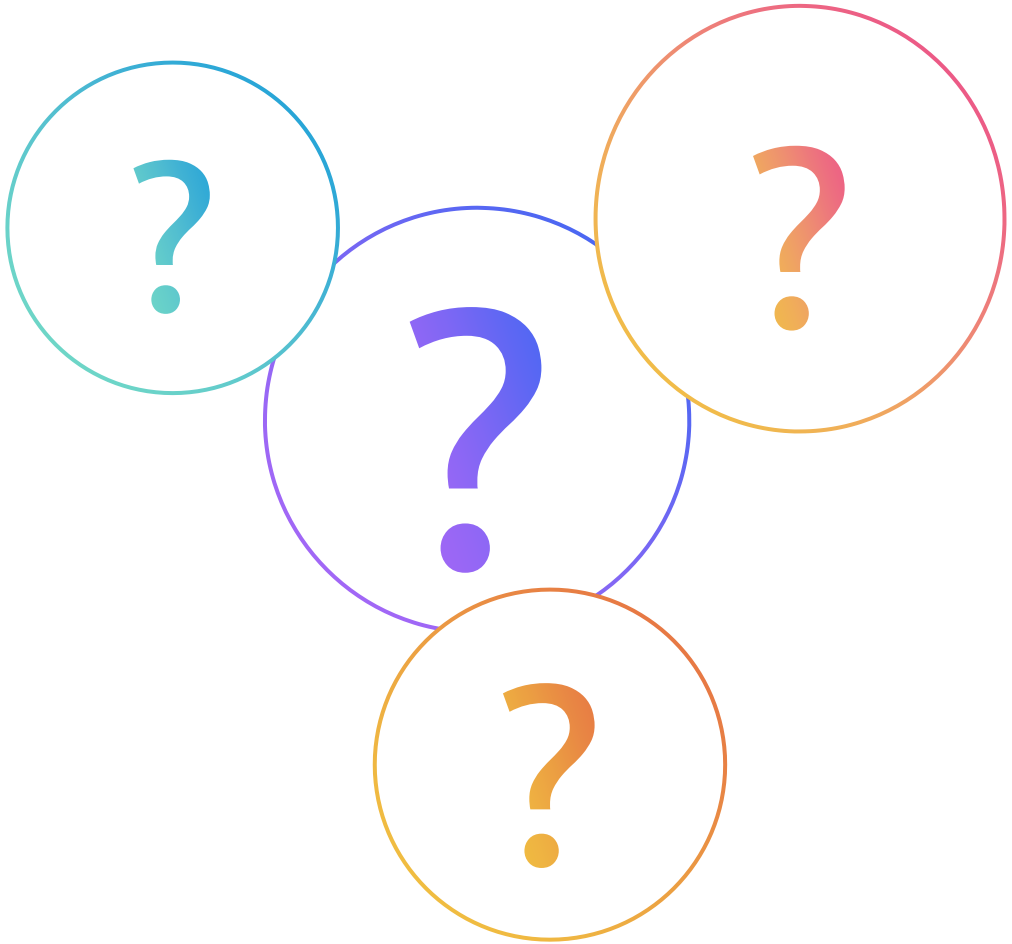




CI/CD

Design Discussion

CI/CD Design Discussion

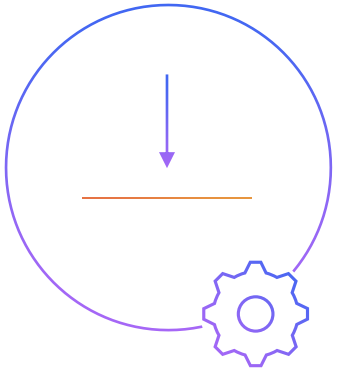


GitHub

Automated way to build the docker image.

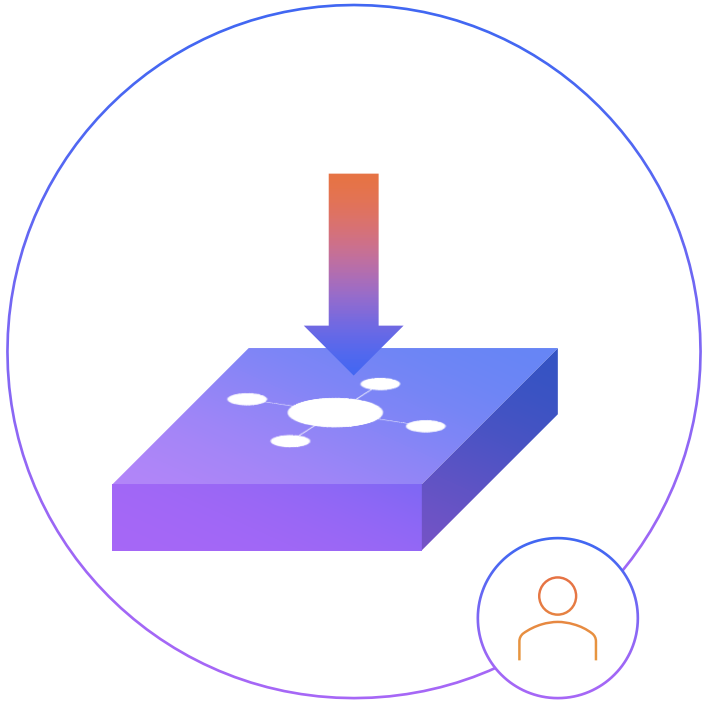
GKE

GitHub

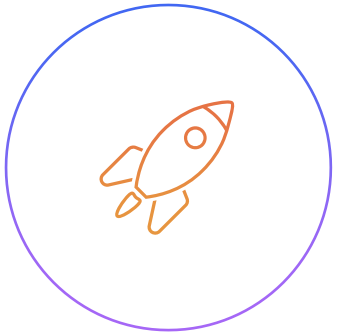


We have to store the Docker image in an artifactory.

GKE

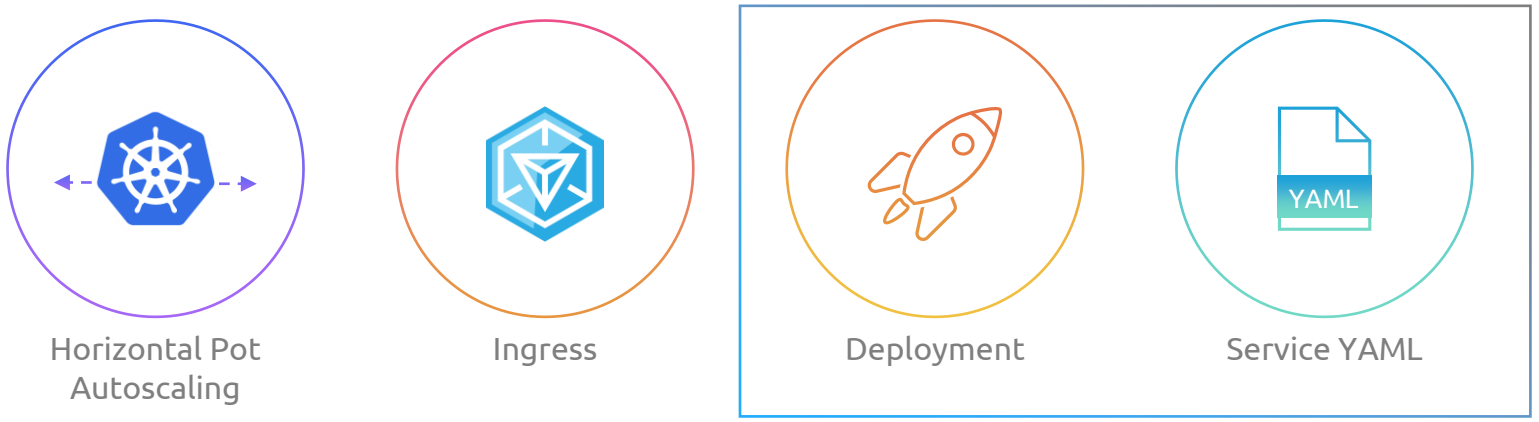


GitHub

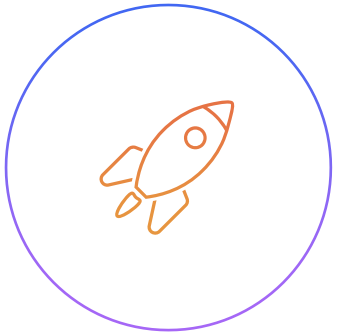


We have to write a deployment / service YAML files for K8 deployment.

GKE



GitHub



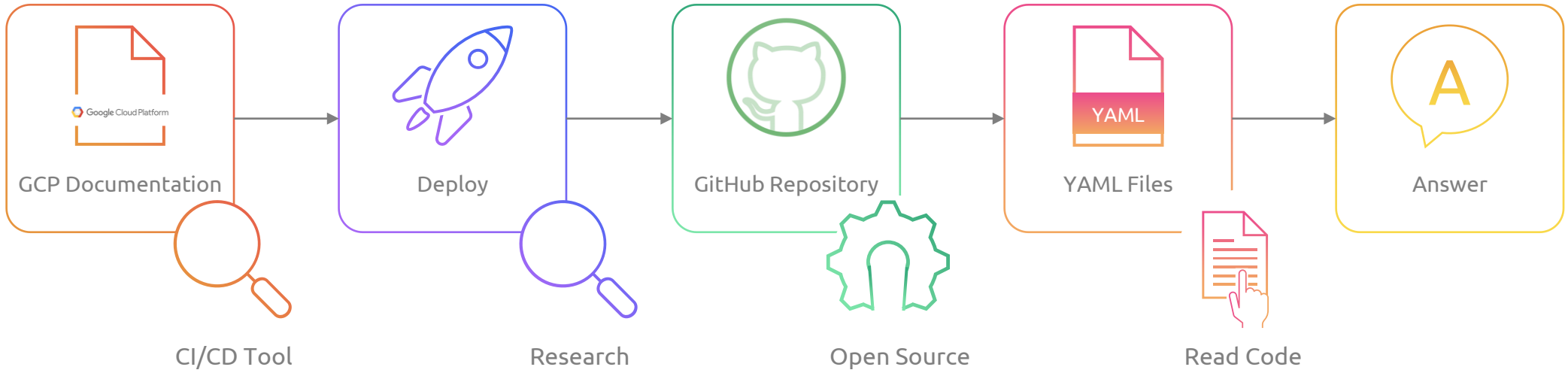
We have to setup CD to deploy this code to GKE using the docker image stored in artifactory.

GKE

GitHub

- Automated way to build the docker image.
- We have to store the Docker image in an artifactory.
- We have to write a deployment/service yaml files for K8 deployment.
- We have to setup CD to deploy this code to GKE using the docker image stored in artifactory.

GKE



GitHub

- Automated way to build the docker image.
- We have to store the Docker image in an artifactory.
- We have to write a deployment/service yaml files for K8 deployment.
- We have to setup CD to deploy this code to GKE using the docker image stored in artifactory.

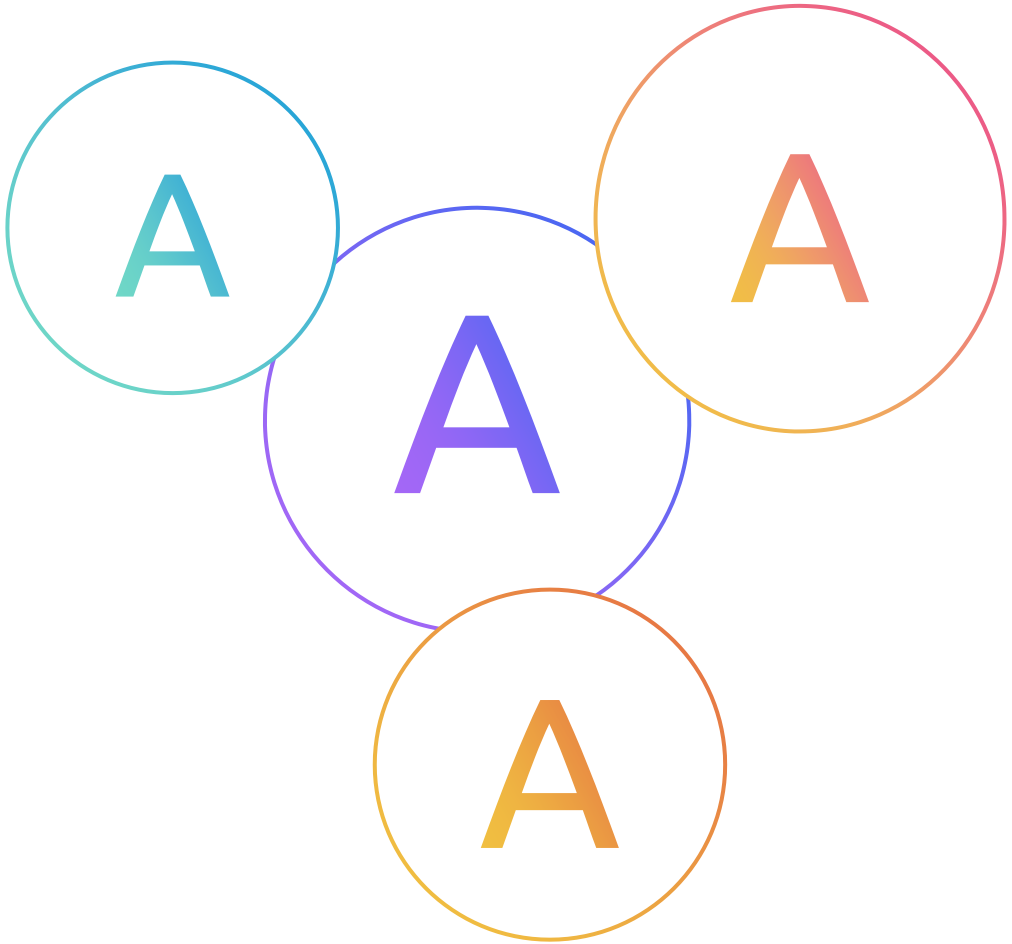
GKE

What tools in GCP can help us achieve this ?

GitHub

- Automated way to build the docker image.
- We have to store the Docker image in an artifactory.
- We have to write a deployment/service yaml files for K8 deployment.
- We have to setup CD to deploy this code to GKE using the docker image stored in artifactory.

GKE



Q

Automated way to
build the docker image.

Cloud Build

CD tool in GCP

Use it to build docker Image

Q

We have to store the Docker image in an artifactory.

GCP Artifact Registry

Store Docker Image

Q

We have to write a deployment/service yaml files for K8 deployment.

Cloud Build

Use it to deploy our image to GKE cluster.

Q

We have to setup CD to deploy this code to GKE using the docker image stored in artifactory.

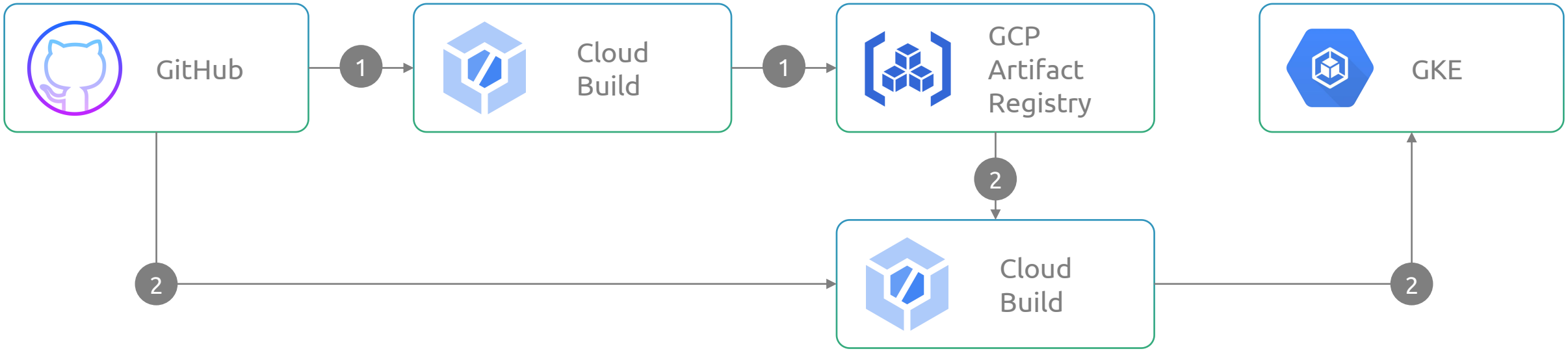
Cloud Build

Use it to deploy our image to GKE cluster.

Services in GCP

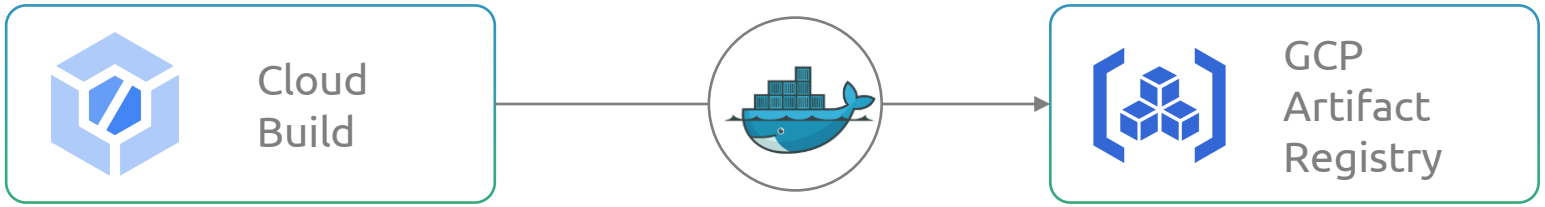
1.  Cloud Build

2.  GCP Artifact Registry



Sprint-03 review

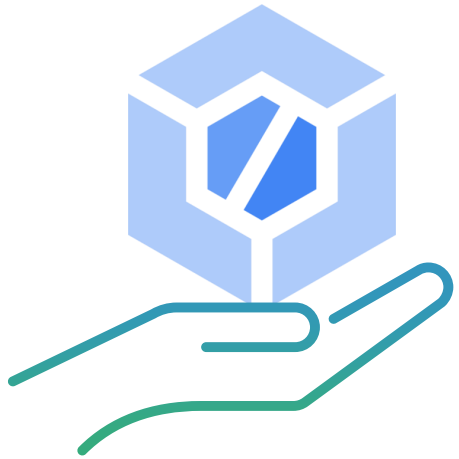
Design Discussion on CI/CD

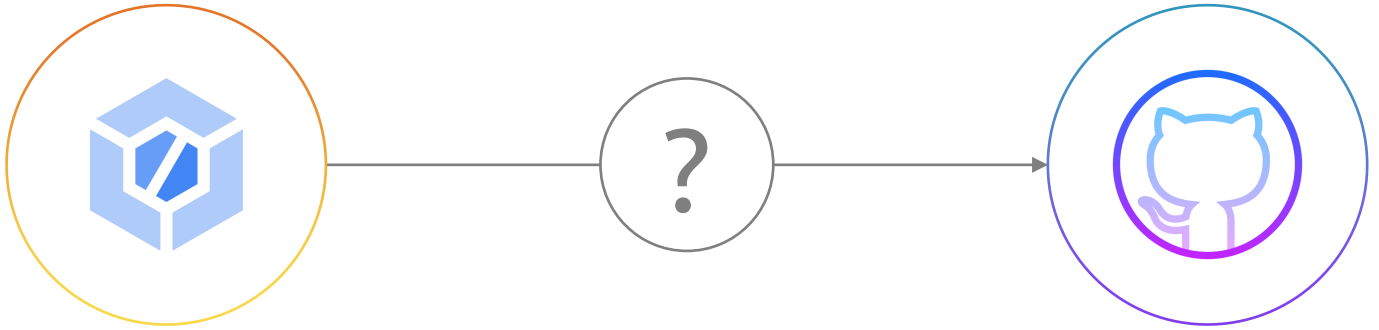


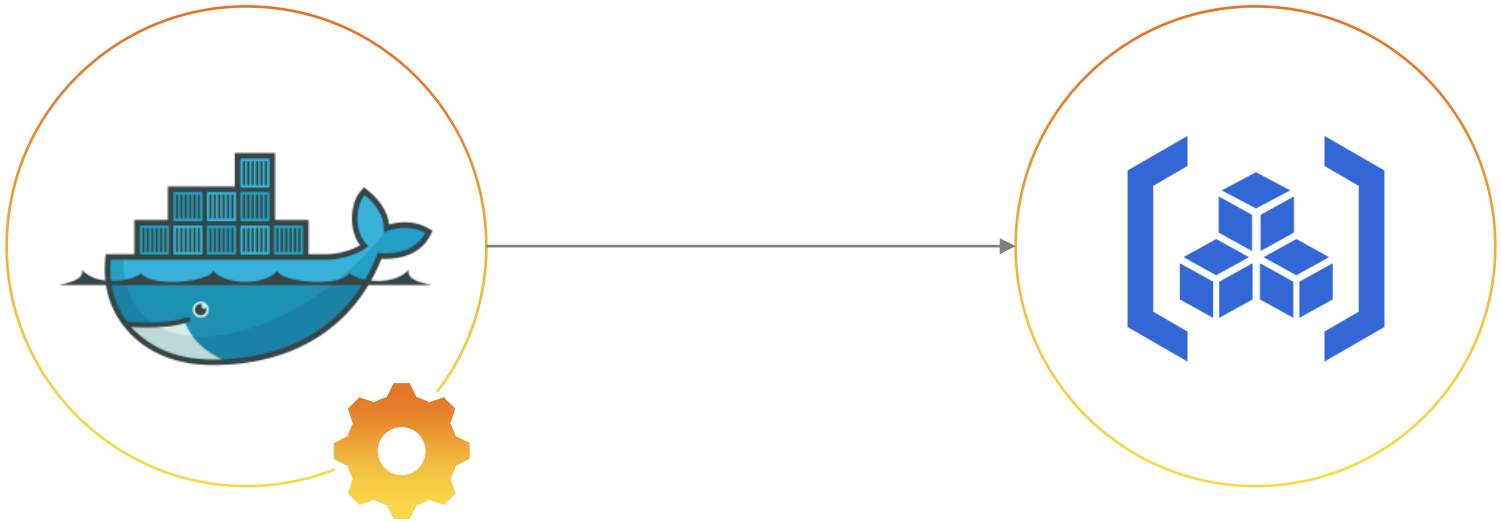
KodeKloud

Sprint-04

Cloud Build







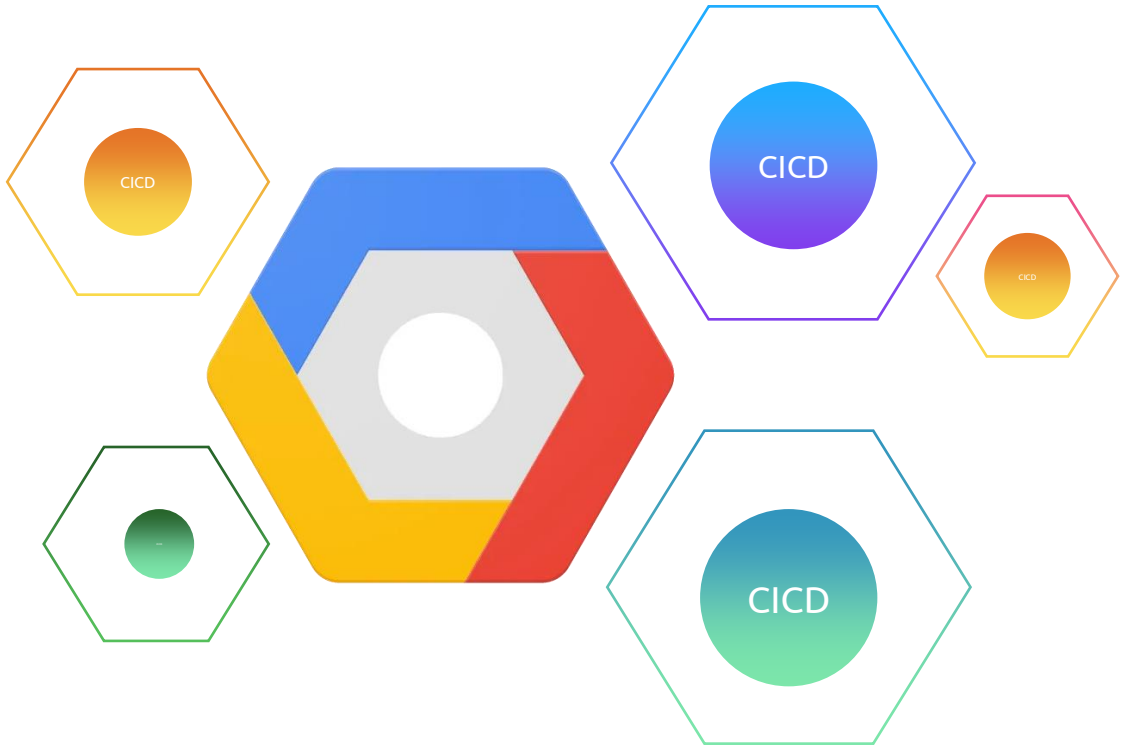
Automate Docker Image Build Process

GCP Artifact Registry

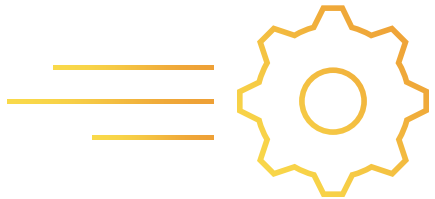
Sprint Goals

- Understanding Cloud Build in details
- How to connect Cloud Build to GitHub Repository?
- Automate Docker Image build process

Cloud Build in detail

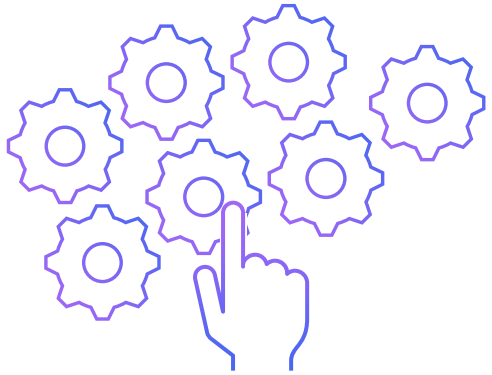


1



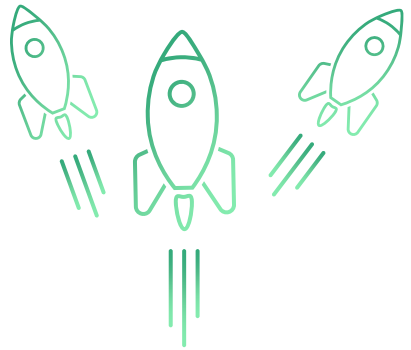
Build software quickly across all programming languages, including Java, Go, Node.js and more.

2



Choose from 15 machine types and run hundreds of concurrent builds per pool.

3



Deploy across multiple environments such as VMs, serverless, Kubernetes, or Firebase.

4

Access cloud-hosted, fully managed CI/CD workflows within your private network.

5

Keep your data at rest within a geographical region or specific location with data residency.

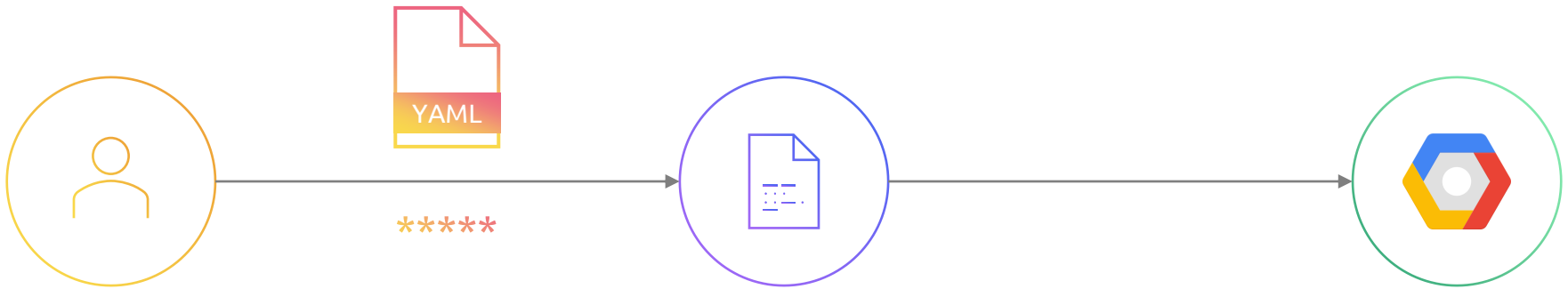
Takeaways

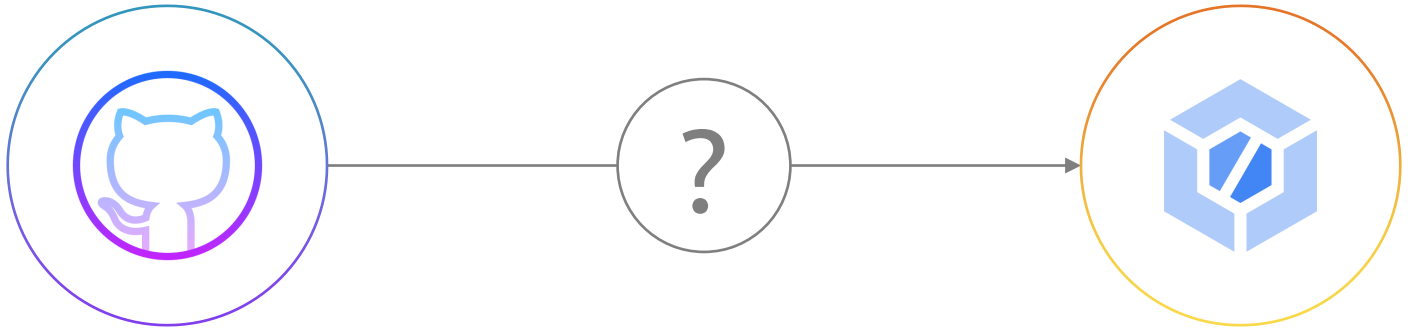
It is a complete serverless CI/CD platform

No Infrastructure to maintain

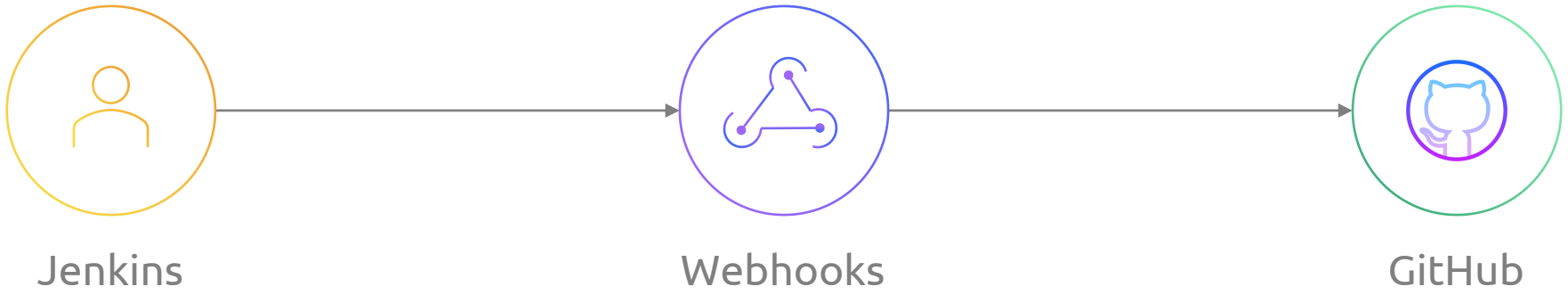
Pricing: E2 – Medium serverless $0.003 per build/minute

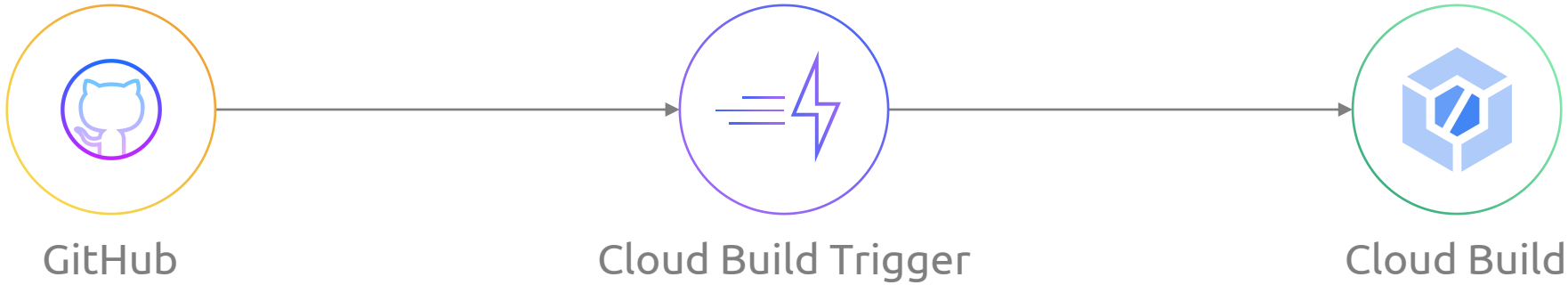
Code: Written in YAML file

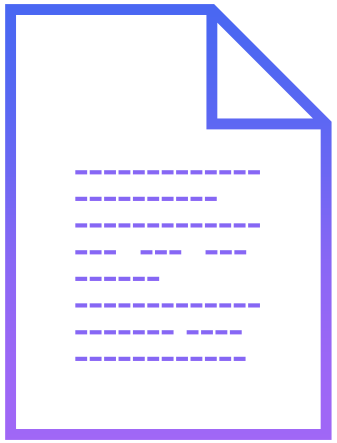




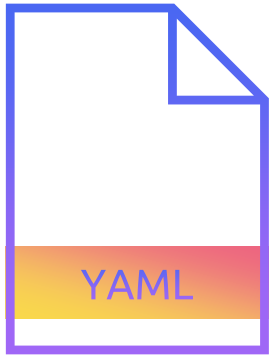
Cloud Build trigger







Any push on the main/master branch will trigger our Cloud Build

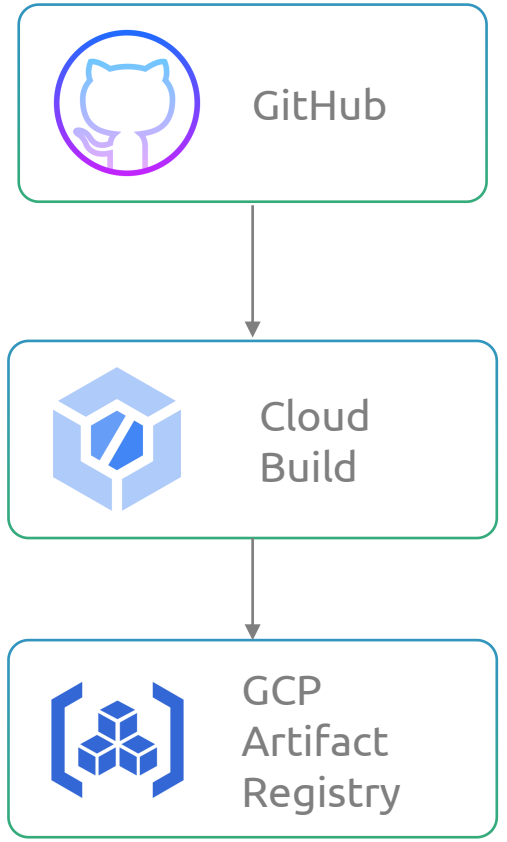


Cloudbuild.yaml file shall contain the CD code

Quick walkthrough

Recap

- Make code changes
- Commit the changes to GitHub
- Open a pull request (PR)
- Merge the PR to the main/master repo
- Cloud Build is triggered
- Cloud Build will store the Docker image to the Artifact Registry



Sprint-04 review

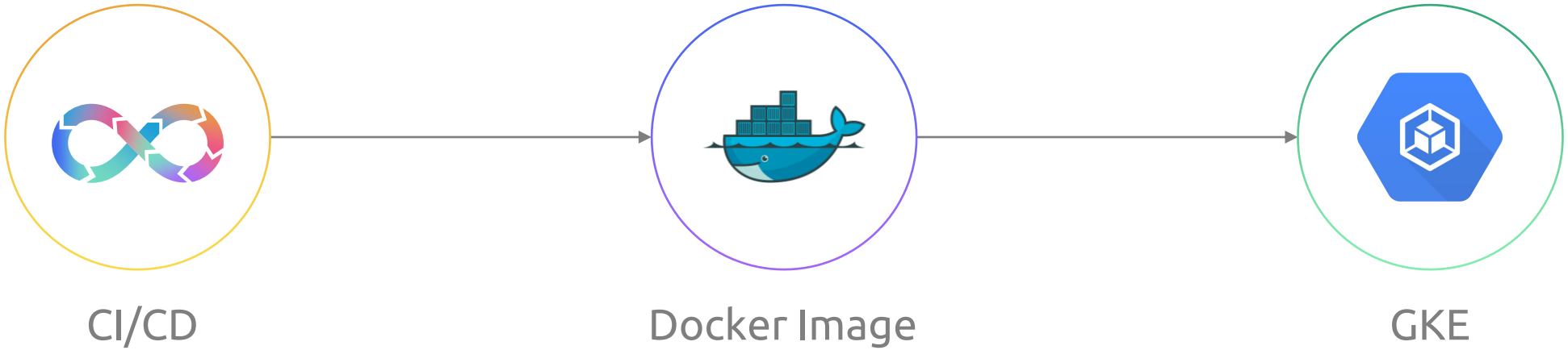
Sprint Goals

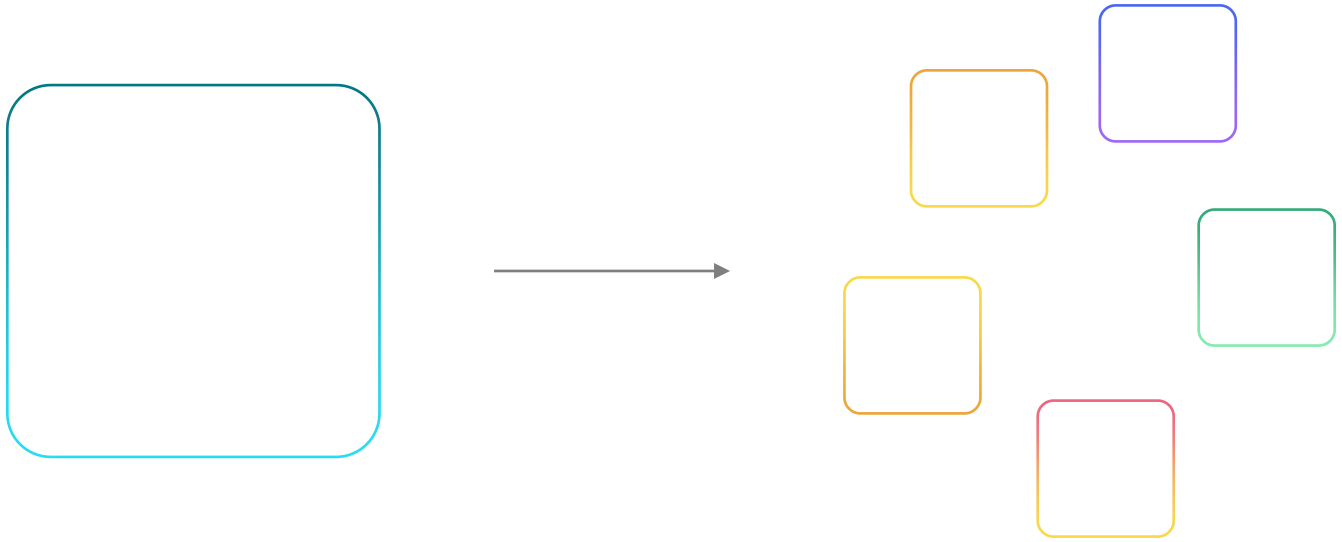
- Understand Cloud Build in detail
- Learn to connect Cloud Build to GitHub repo
- Automate Docker image build process

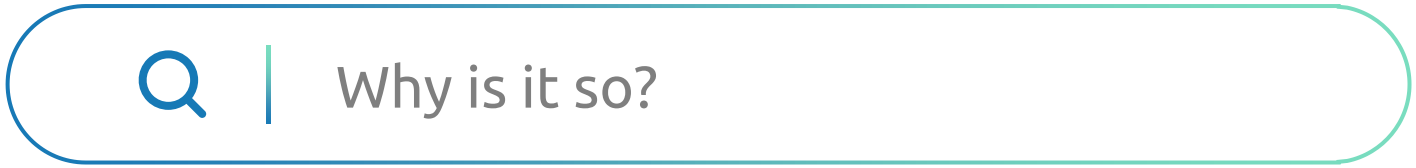
KodeKloud

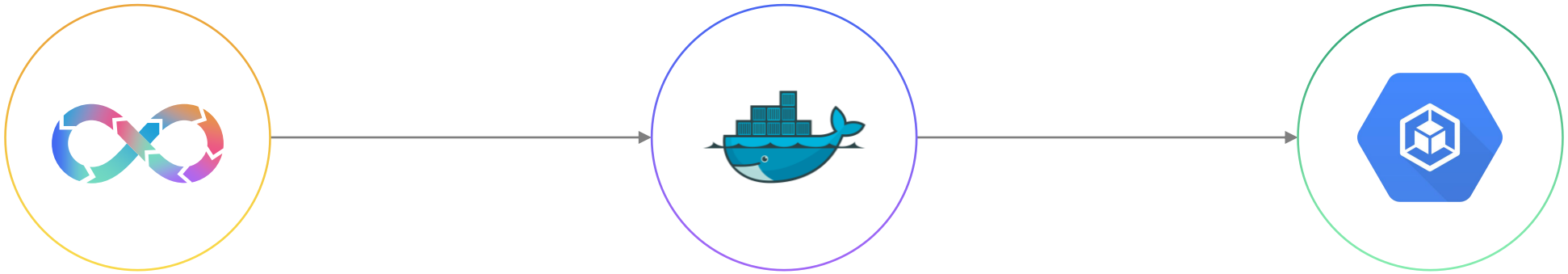
Sprint-05

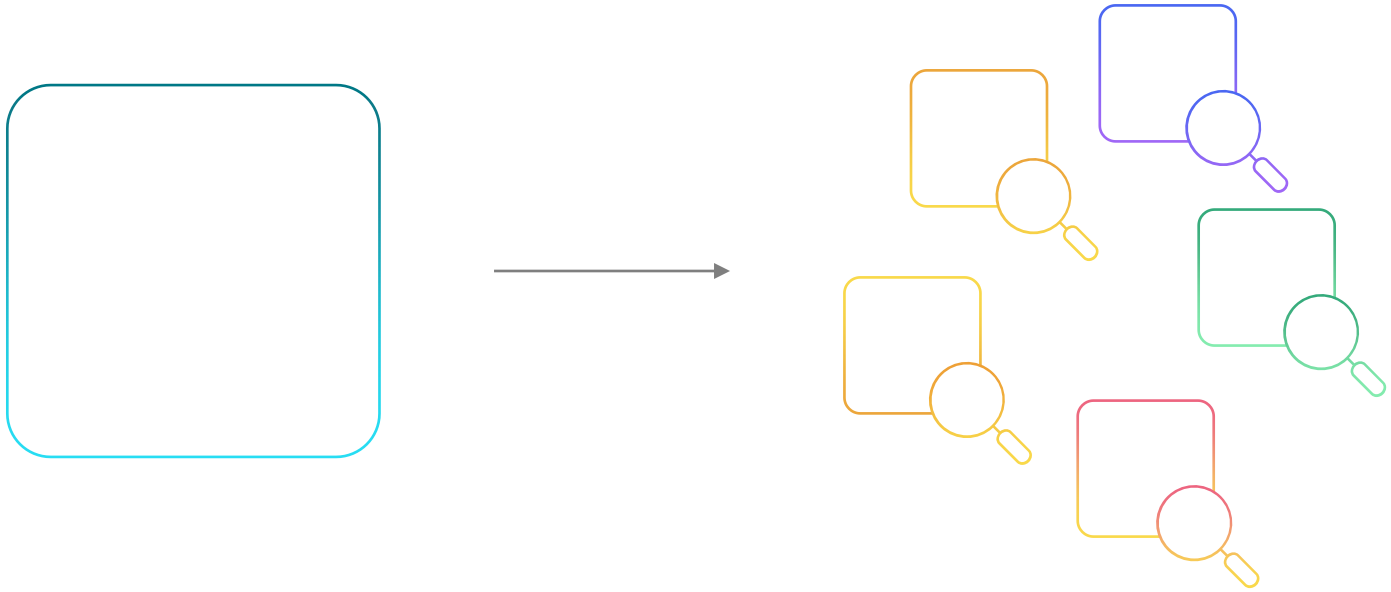
Extending the CI/CD process to deploy our Docker image on a GKE cluster

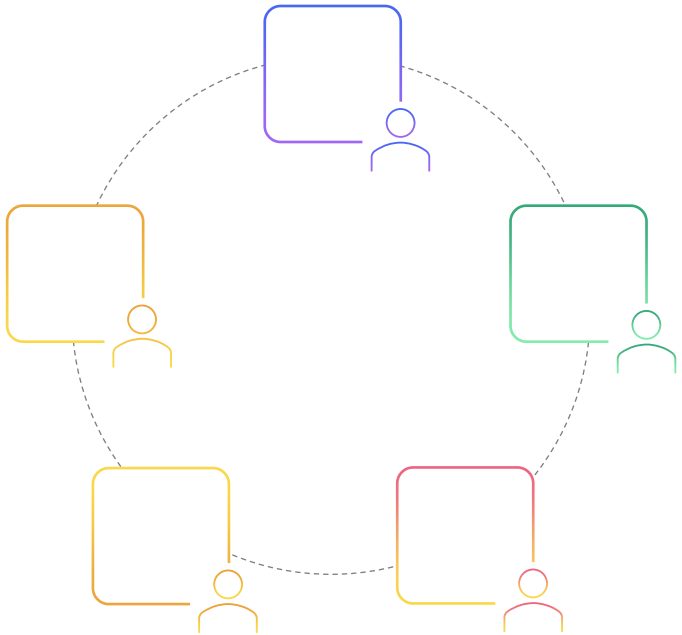


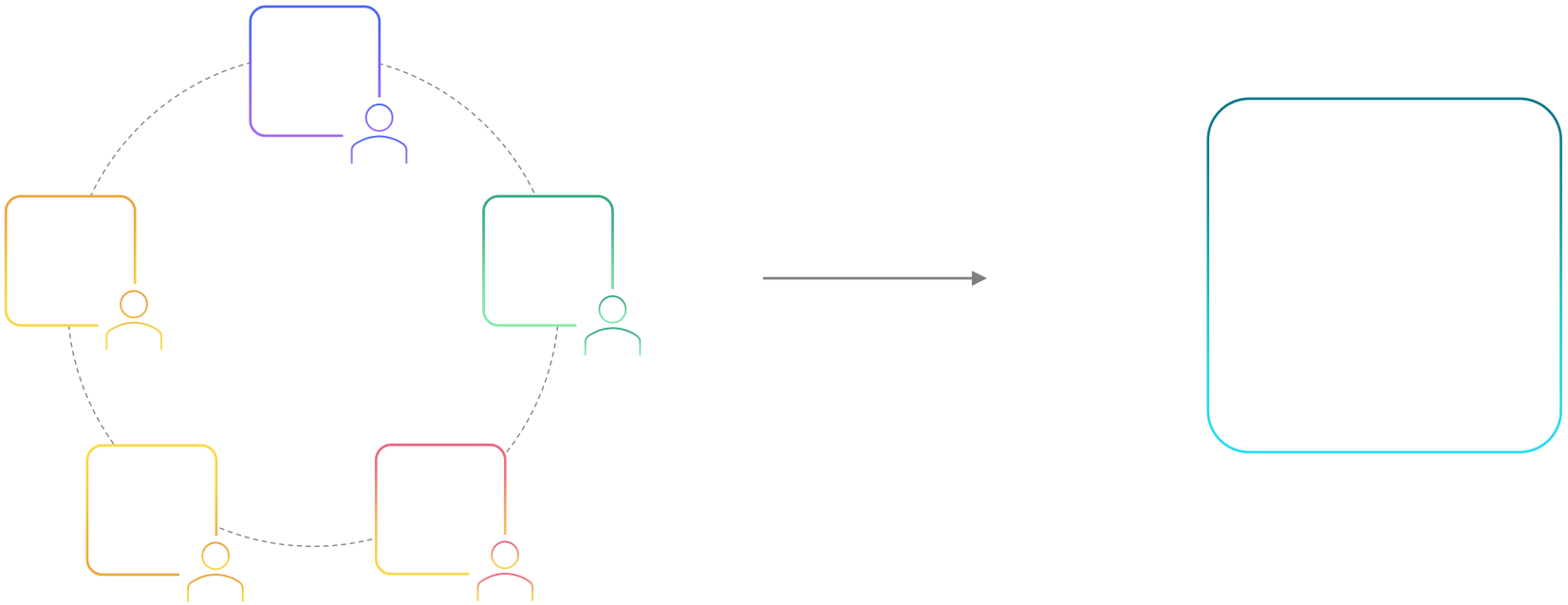








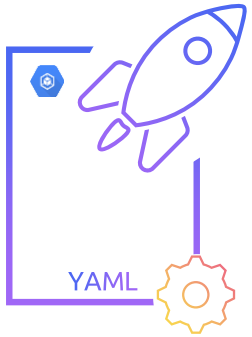




1

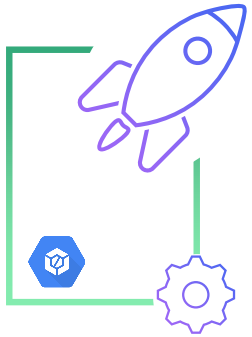
Creating a namespace in our GKE cluster

2



Creating a deployment file

3



Updating the Cloud Build code for deployment

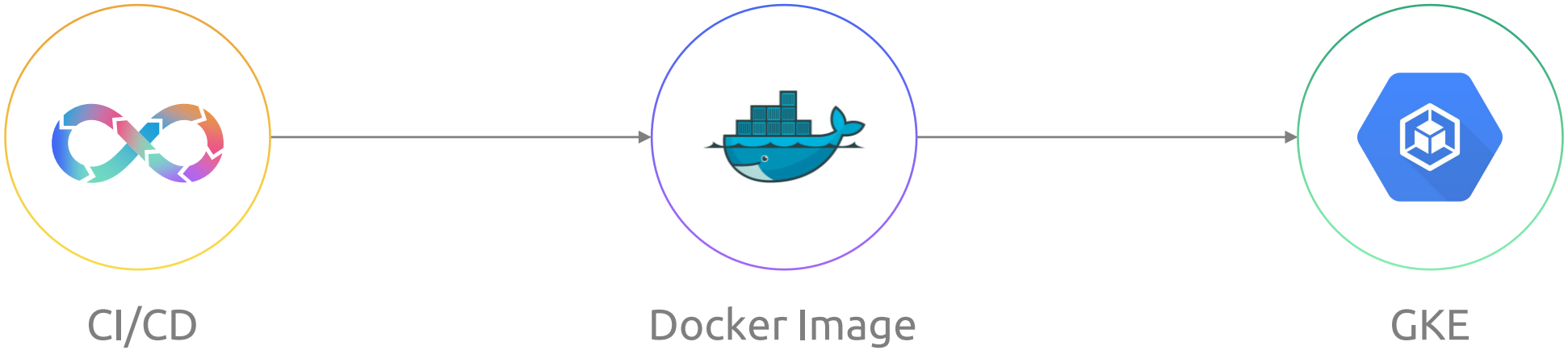
4

Validating the deployment

1. Creating a namespace in our GKE cluster
2. Creating a deployment file
3. Updating the Cloud Build code for deployment
4. Validating the deployment

Sprint-05 review

Extending the CI/CD process to deploy our Docker image on a GKE cluster

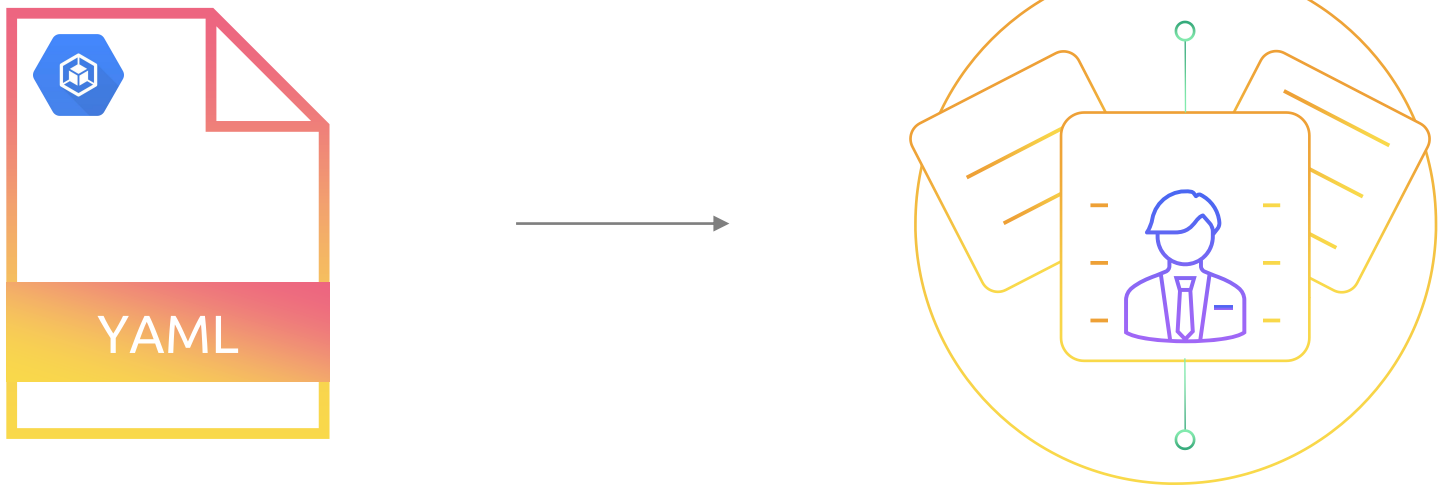


Sprint Goals

- Creating a namespace in our GKE cluster
- Creating a deployment file
- Updating the Cloud Build code for deployment
- Validating the deployment

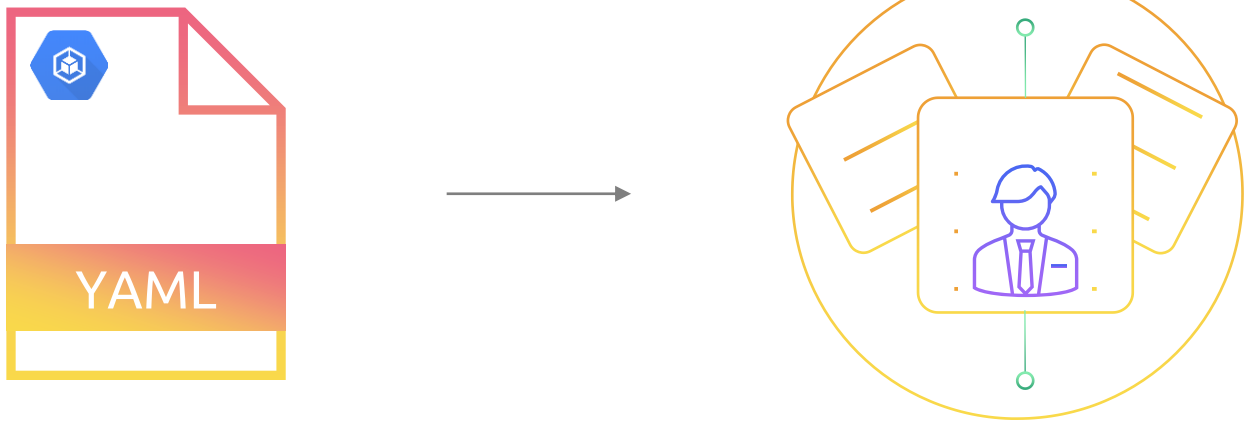
KodeKloud

Sprint-06



Extending our gke.yaml file to also include
code to expose our application via an endpoint

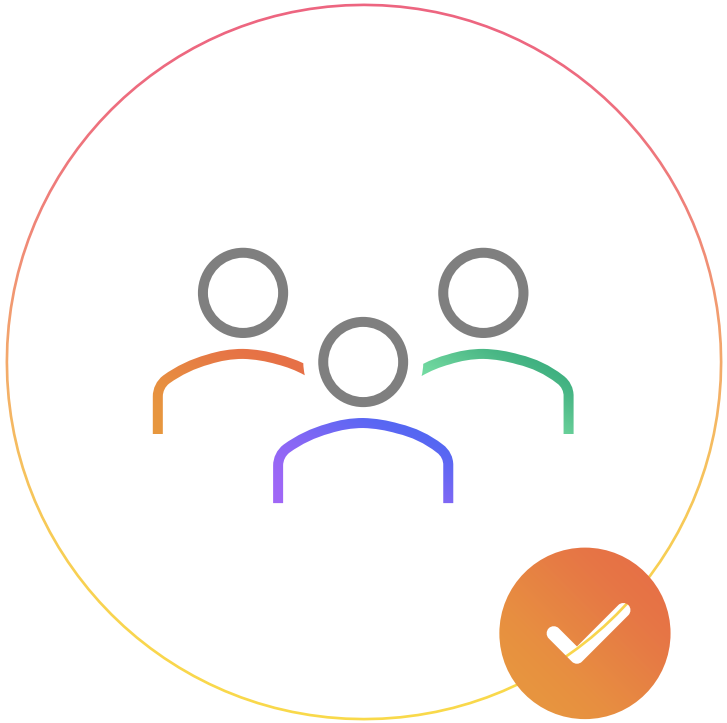
Sprint-06 review

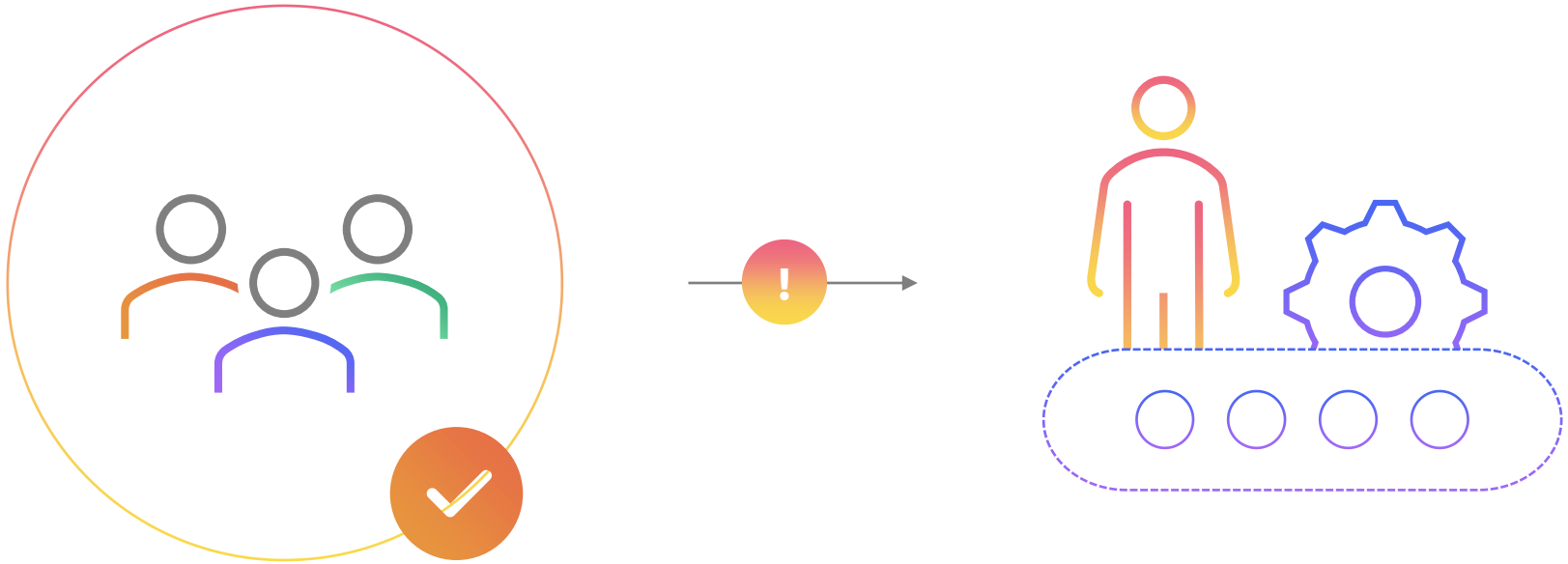


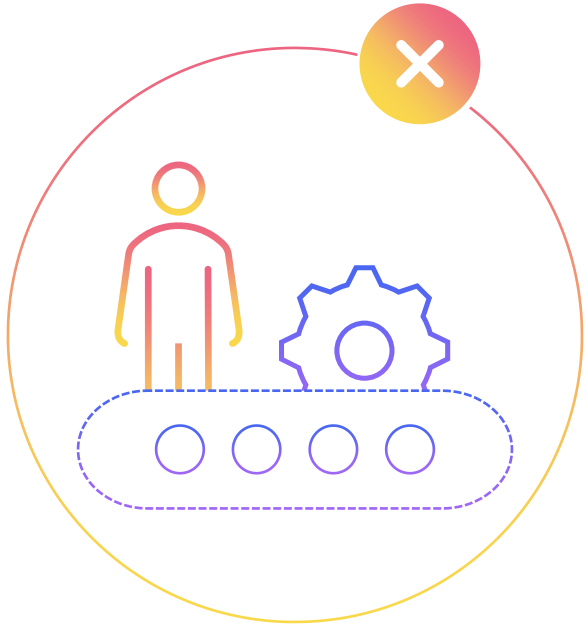
- [x] Extending our gke.yaml file to also include code to expose our application via an endpoint

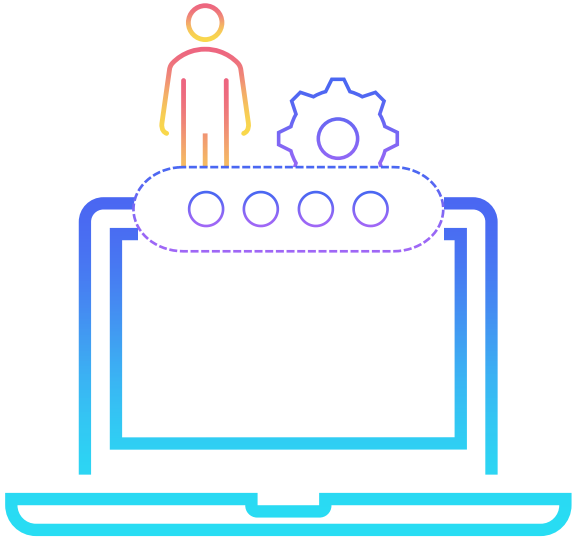
KodeKloud

Sprint-07

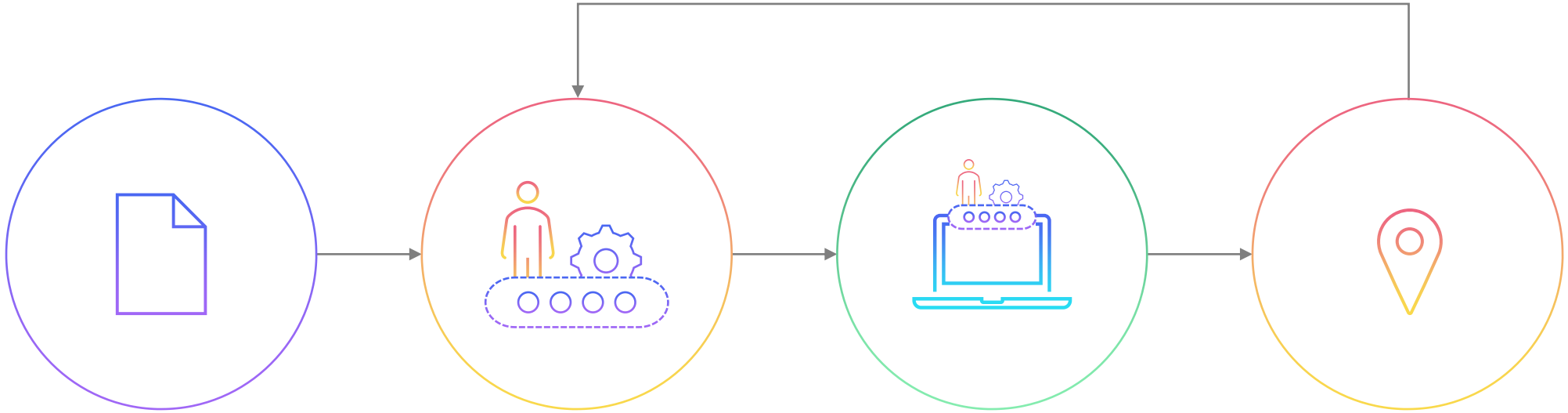




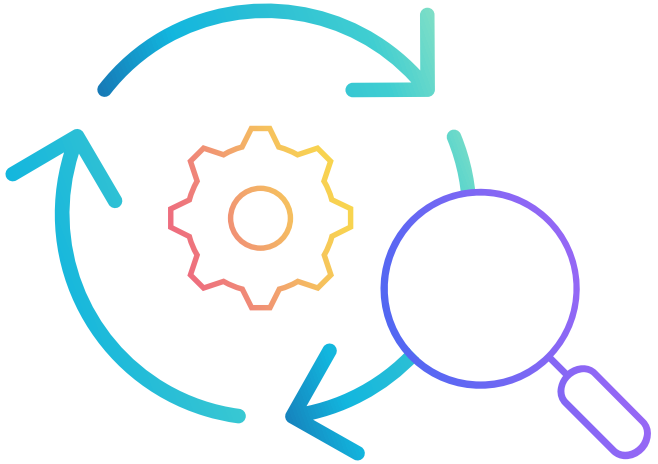




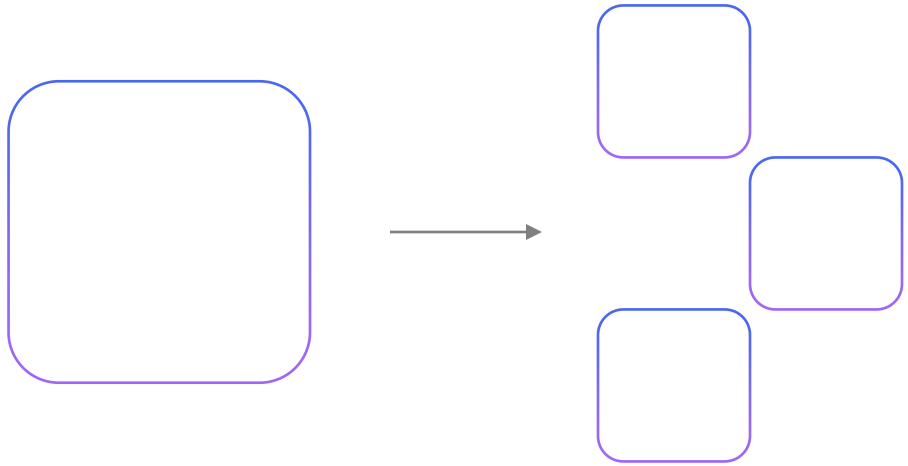
We are requested to setup a development environment



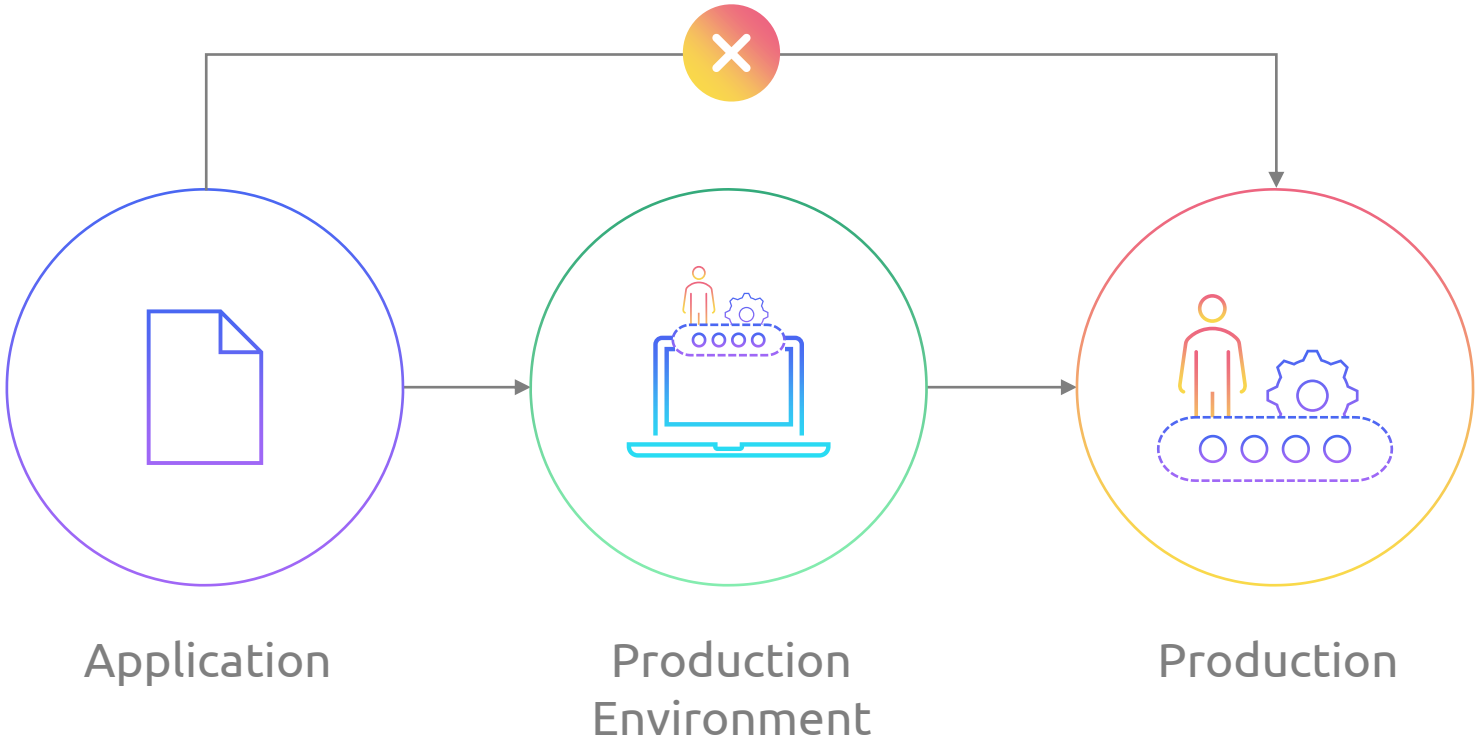
Upgrade replicas using the new flow



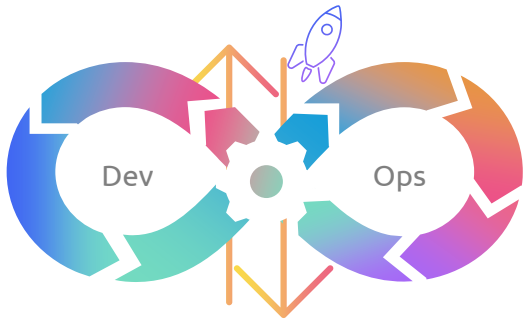
Deployment process using the Development Environment



Deployment process using the Development Environment



Sprint-07 review



Deploy the changes to production using the right DevOps lifecycle

KodeKloud