GitLab CI - Advanced usage of CI

Environments and Deployments

Environments are used for testing, building and deploying the CI (Continuous Integration) jobs and control the Continuous Deployment of software with the GitLab. GitLab CI is capable of tracking your project deployments and also you will come to know what is being deployed on your server.

The name of an environment could be defined by using *environment:name* string and contain the following −

- letters
- digits
- spaces
- -
- _
- /
- $
- {
- }

Using SSH keys with GitLab CI/CD

You can set the SSH (Secure Shell or Secure Socket Shell) keys to provide a reliable connection between the computer and GitLab. The SSH keys can be used with GitLab CI/CD when −

- You need to checkout internal sub modules.
- You need to download private packages using package manager.
- You need to install an application to your own server.
- You execute the SSH commands to remote server from build environment.
- You need to rsync files to a remote server from the build environment.

The SSH key setup is explained in the GitLab SSH Key Setup chapter.

Artifacts

Artifacts are used to attach the list of files and directories to the job after success. The artifacts contain following types −

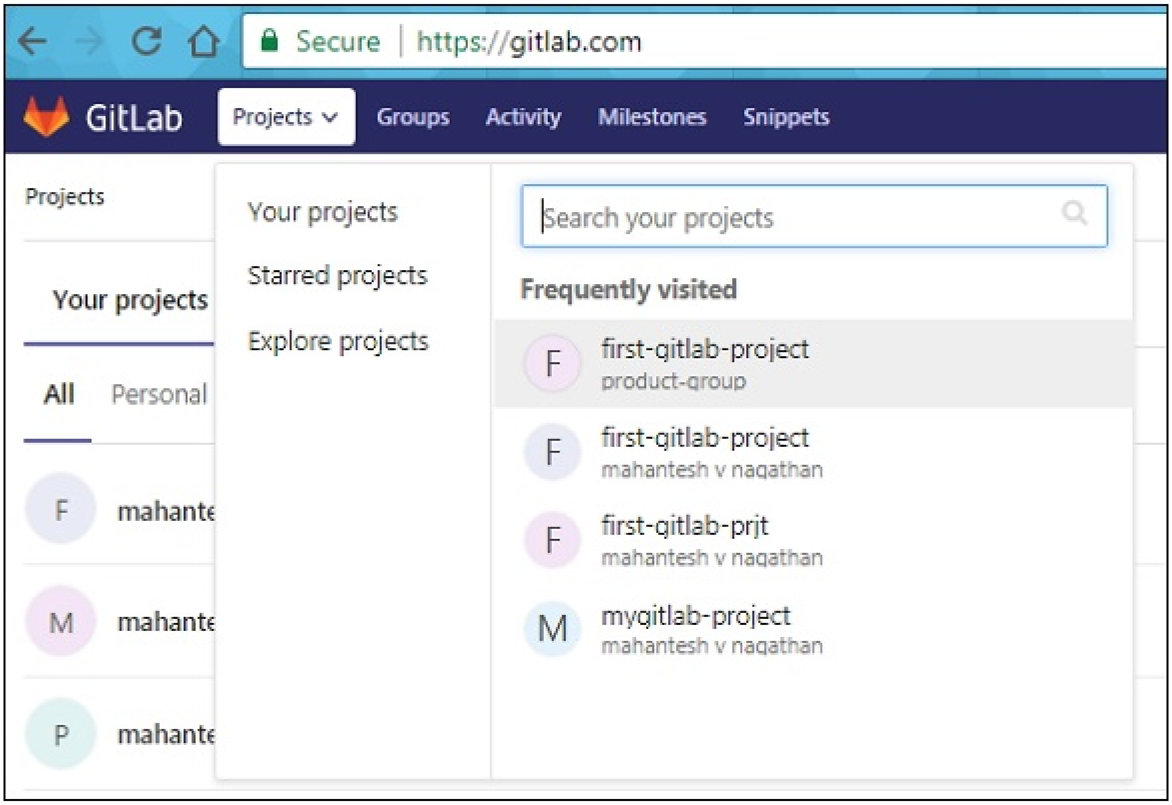
- **artifacts:name** − This directive is used to specify the name of created artifacts archive. It provides unique name for created artifacts archive which is helpful when you are downloading the archive from GitLab.
- **artifacts:when** − This directive is used to upload artifacts when there is a job failure. It contains the following values:
  - **on_success** − It is used to upload the artifacts when there is a job success.
  - **on_failure** − It is used to upload the artifacts when the job fails.
  - **always** − It is used to upload the artifacts regardless of job status.
- **artifacts:expire_in** − It defines that how long artifacts should live before they expire and therefore deleted, since they are uploaded and stored on GitLab

Triggering Pipelines

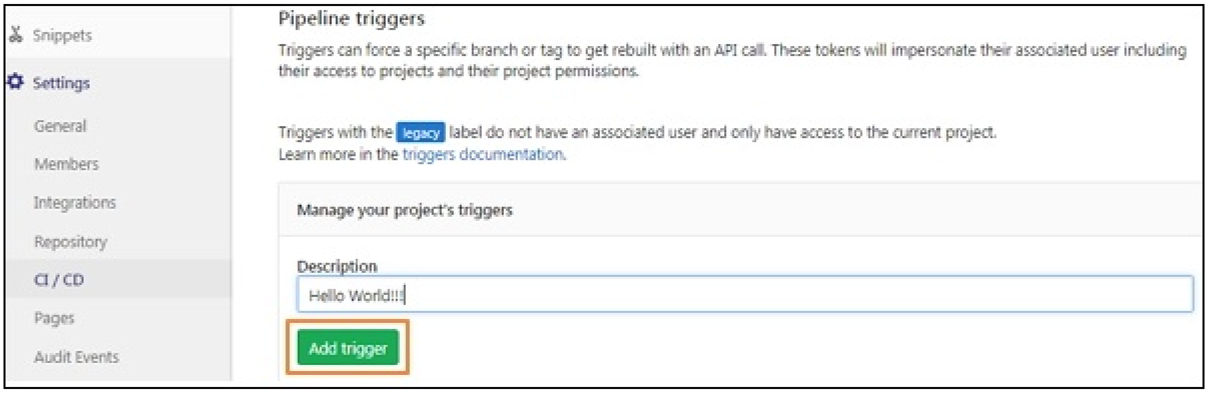
Triggers can force a specific branch or tag to get rebuilt with an API call and triggers with the *legacy* label will have access to the current project.

The new trigger can be added as shown in the below steps −

Step 1 − Login to your GitLab account and go to your project −



Step 2 − Click on the *CI/CD* option under *Settings* tab and expand the *Pipeline triggers* option −



Pipeline triggers

Triggers can force a specific branch or tag to get rebuilt with an API call. These tokens will impersonate their associated user including their access to projects and their project permissions.

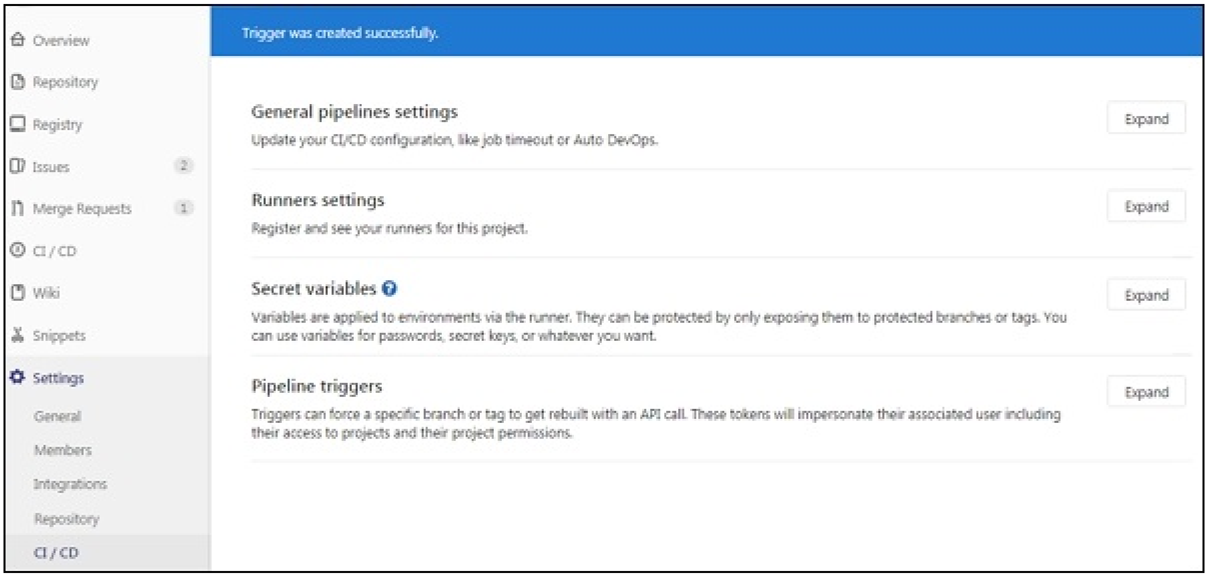
Triggers with the legacy label do not have an associated user and only have access to the current project.
Learn more in the triggers documentation.

Manage your project's triggers

Description: Hello World!!!

Enter the description for the trigger and click on the *Add Trigger* button.

Step 3 − Next, it will display the success message after creating the trigger −



Trigger was created successfully.

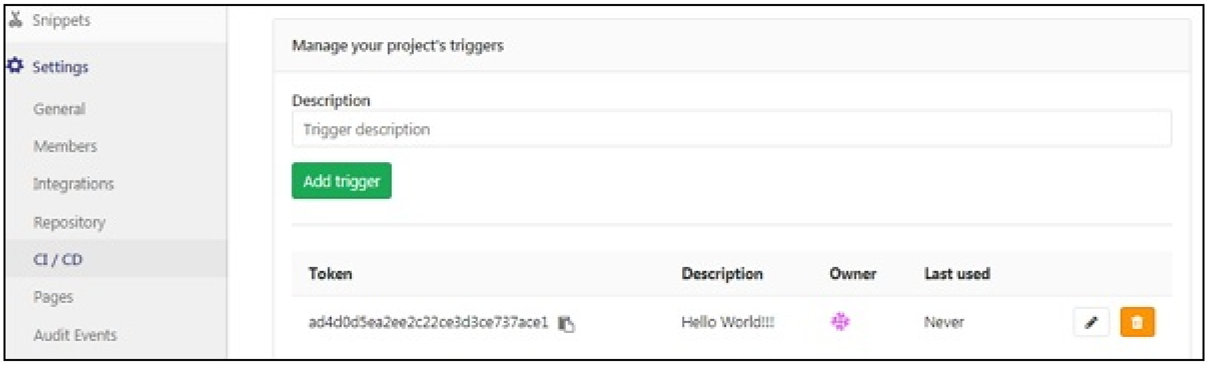
General pipelines settings
Update your CI/CD configuration, like job timeout or Auto DevOps.

Runners settings
Register and see your runners for this project.

Secret variables
Variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. You can use variables for passwords, secret keys, or whatever you want.

Pipeline triggers
Triggers can force a specific branch or tag to get rebuilt with an API call. These tokens will impersonate their associated user including their access to projects and their project permissions.

Step 4 − Now go to *CI/CD* option under *Settings* tab and expand the *Pipeline triggers* option. You will see the newly created trigger along with the token as shown in the image below −



Manage your project's triggers

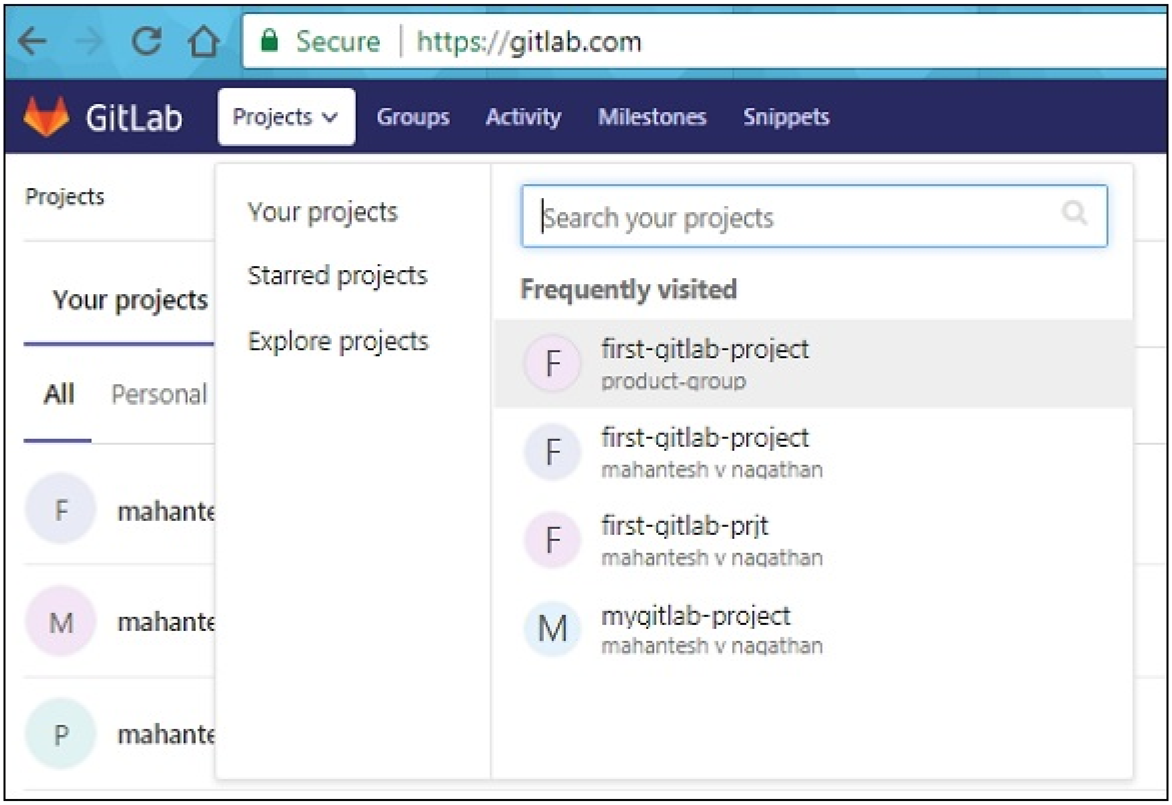
Description: Trigger description

Token	Description	Owner	Last used
ad4d0d5ea2ee2c22ce3d3ce737ace1	Hello World!!!		Never

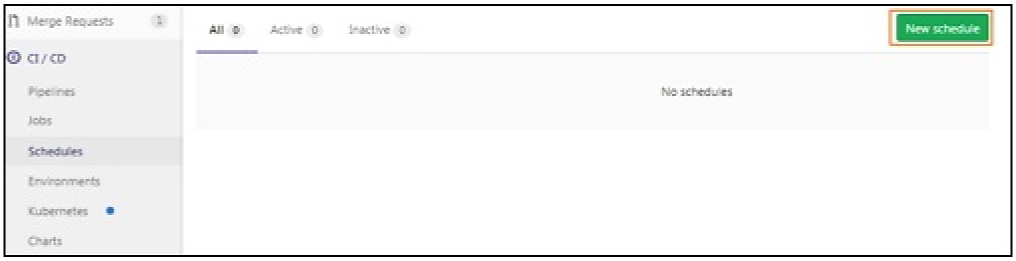
Pipeline Schedules

You can run the pipeline by using the pipeline schedules at specific intervals. To create pipeline schedule, use the below steps −

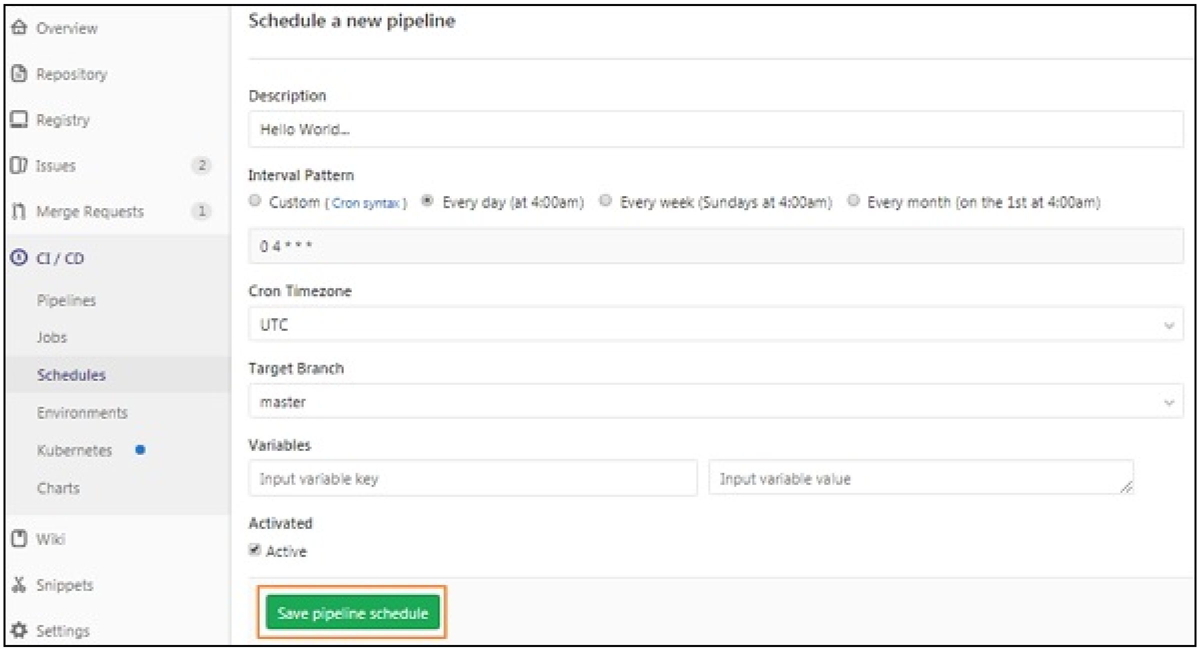
Step 1 − Login to your GitLab account and go to your project −



Step 2 − Click on the *Schedules* option under *CI/CD* tab and click on the *New schedule* button −



Step 3 − Next, it will open the Scheduling new pipeline screen, fill up the fields and click on the *Save pipeline schedule* button −



Schedule a new pipeline

Description

Hello World...

Interval Pattern

○ Custom (Cron syntax) ● Every day (at 4:00am) ○ Every week (Sundays at 4:00am) ○ Every month (on the 1st at 4:00am)

0 4 * * *

Cron Timezone

UTC

Target Branch

master

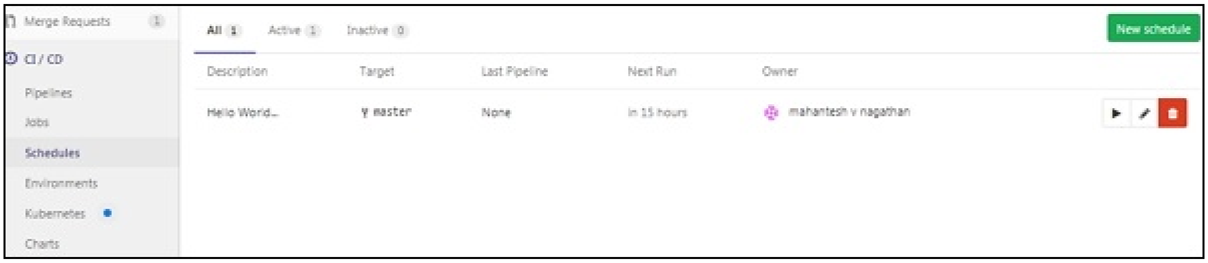
Variables

Input variable key | Input variable value

Activated

☑ Active

Step 4 − Now, you will see the pipeline which is scheduled to run −



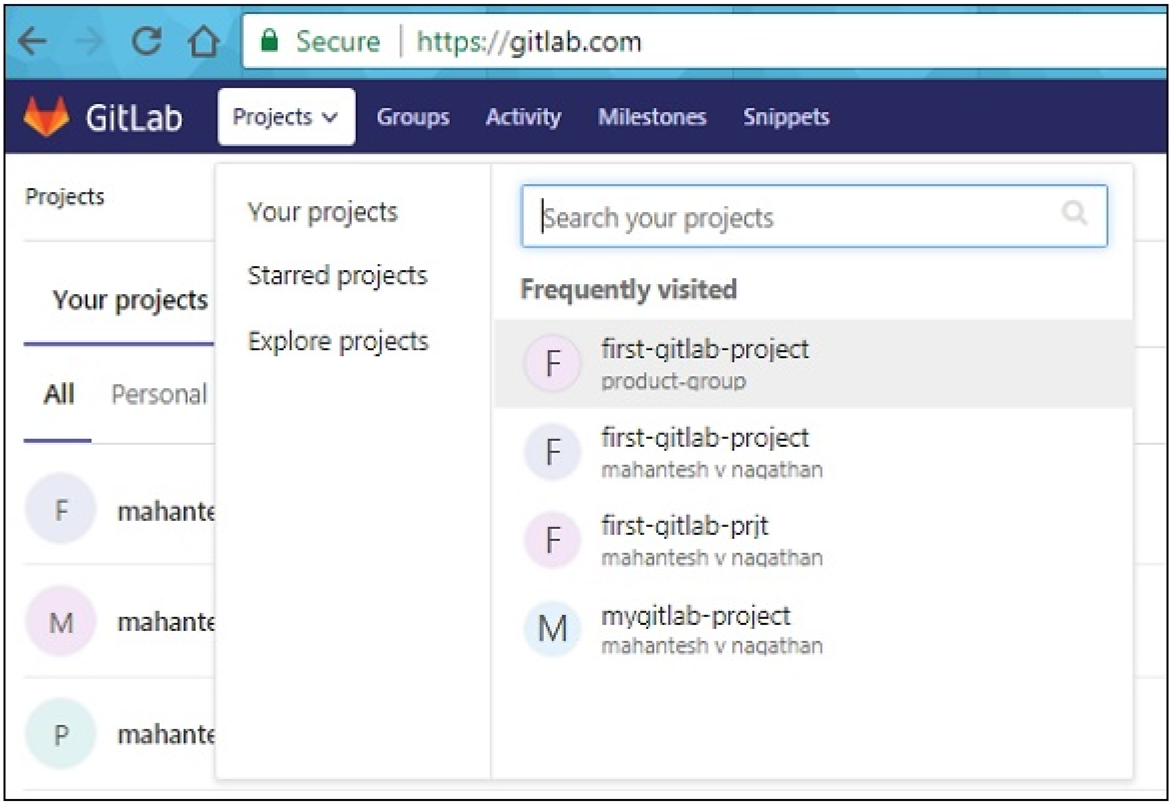
Description	Target	Last Pipeline	Next Run	Owner
Hello World...	master	None	in 15 hours	mahantesh v nagathan

Connecting GitLab with a Kubernetes Cluster

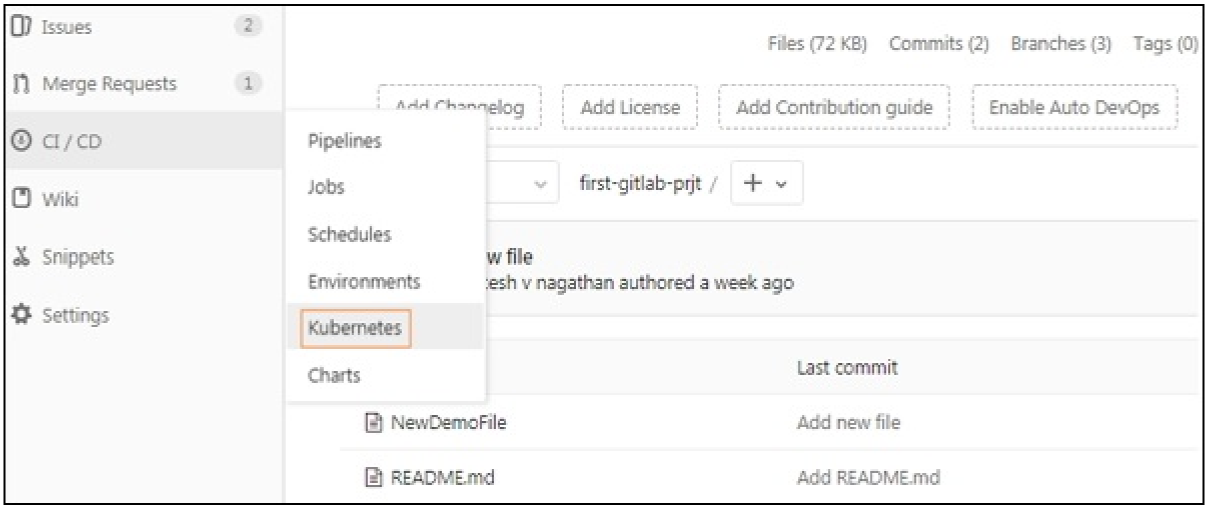
The Kubernetes cluster can be used to review and deploy the applications, running the pipeline etc in an easy method. You can create a new cluster to your project by associating your GitLab account with the Google Kubernetes Engine (GKE).

The new Kubernetes cluster can be created as shown in the below steps −

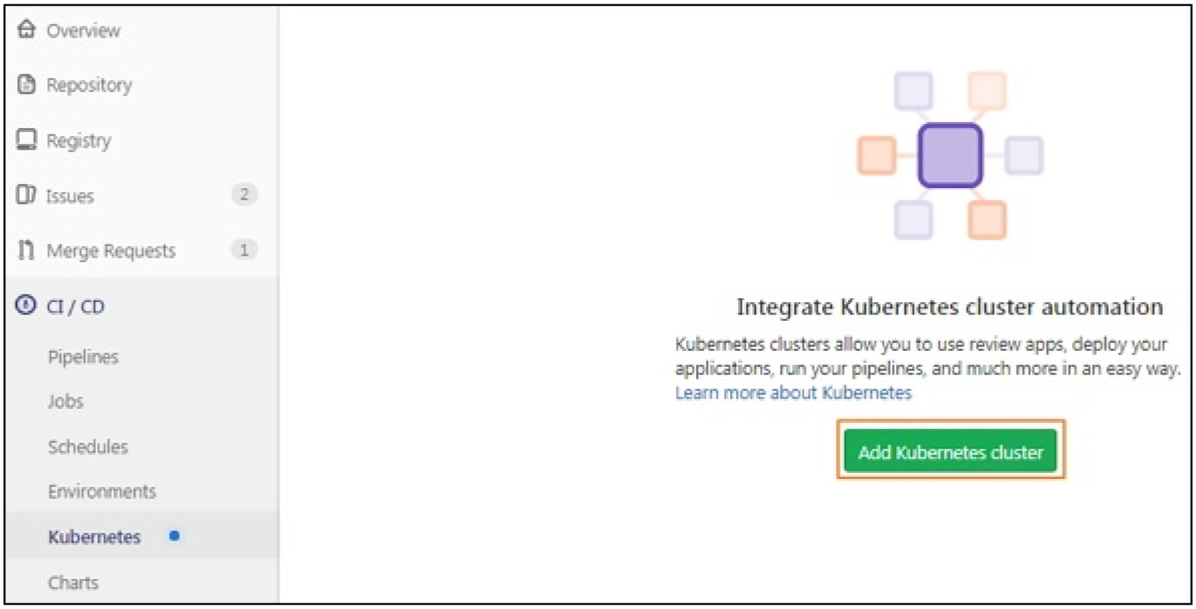
Step 1 − Login to your GitLab account and go to your project −



Step 2 − Click on the *Kubernetes* option under *CI/CD* tab −



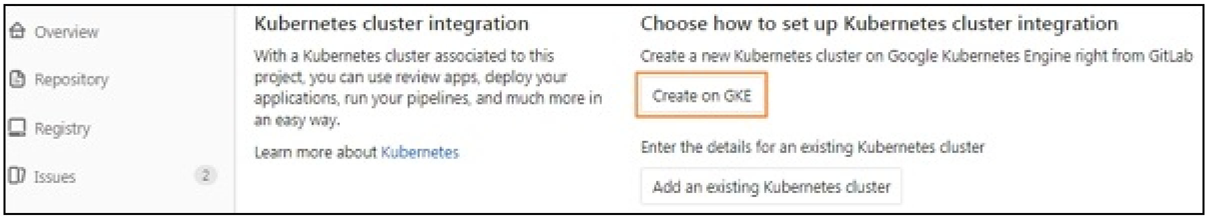
Step 3 − Next, click on *Add Kubernetes cluster* button −



Integrate Kubernetes cluster automation

Kubernetes clusters allow you to use review apps, deploy your applications, run your pipelines, and much more in an easy way. Learn more about Kubernetes

Step 4 − Click on *Create on GKE* button to create a new Kubernetes cluster on Google Kubernetes Engine −



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.

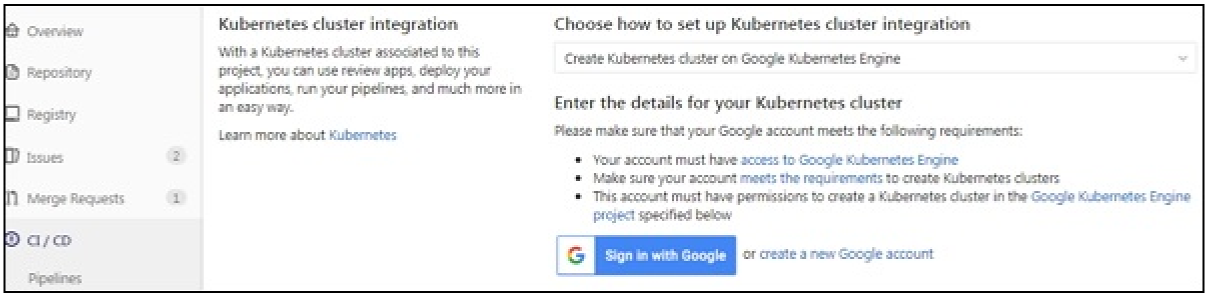
Learn more about Kubernetes

Choose how to set up Kubernetes cluster integration

Create a new Kubernetes cluster on Google Kubernetes Engine right from GitLab

Enter the details for an existing Kubernetes cluster

Step 5 − If you have a Google account, then sign with that account to enter the details for Kubernetes cluster or else create a new Google account −



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

Choose how to set up Kubernetes cluster integration

Create Kubernetes cluster on Google Kubernetes Engine

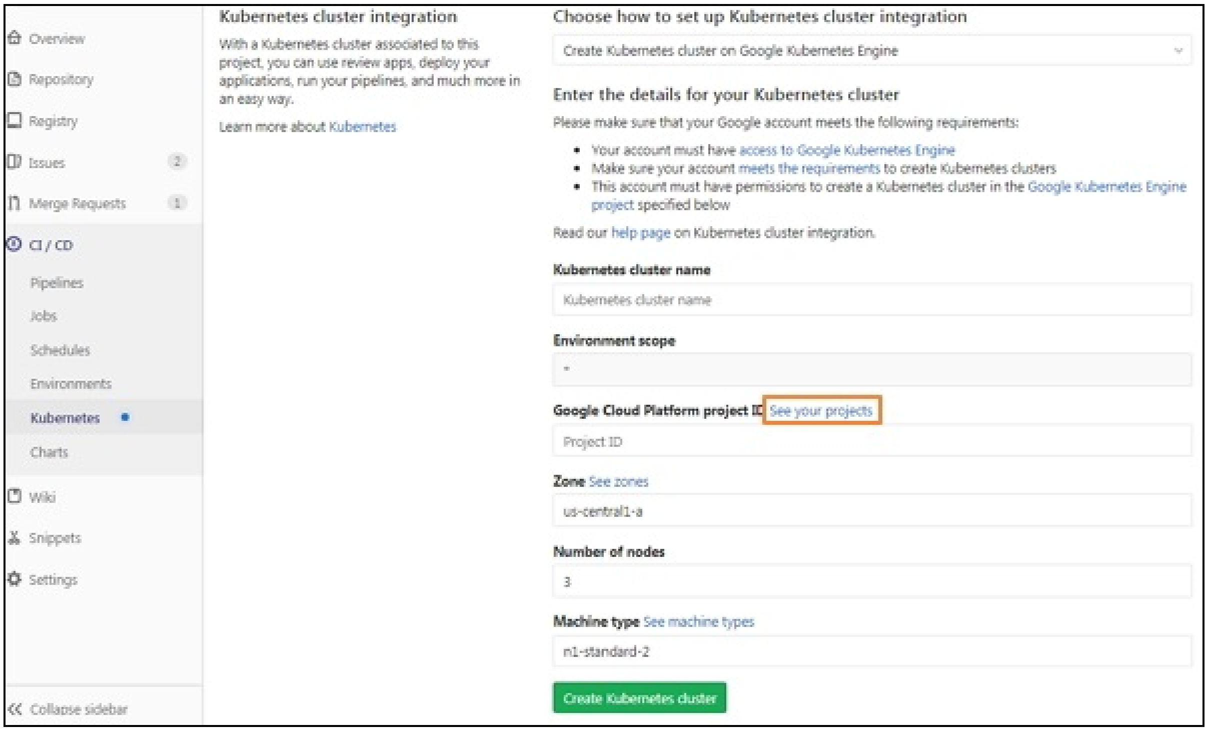
Enter the details for your Kubernetes cluster

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

or create a new Google account

Step 6 − Now enter the values in the fields for your Kubernetes cluster −



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

Choose how to set up Kubernetes cluster integration

Create Kubernetes cluster on Google Kubernetes Engine

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.

Kubernetes cluster name

Kubernetes cluster name

Environment scope

*

Google Cloud Platform project ID See your projects

Project ID

Zone See zones

us-central1-a

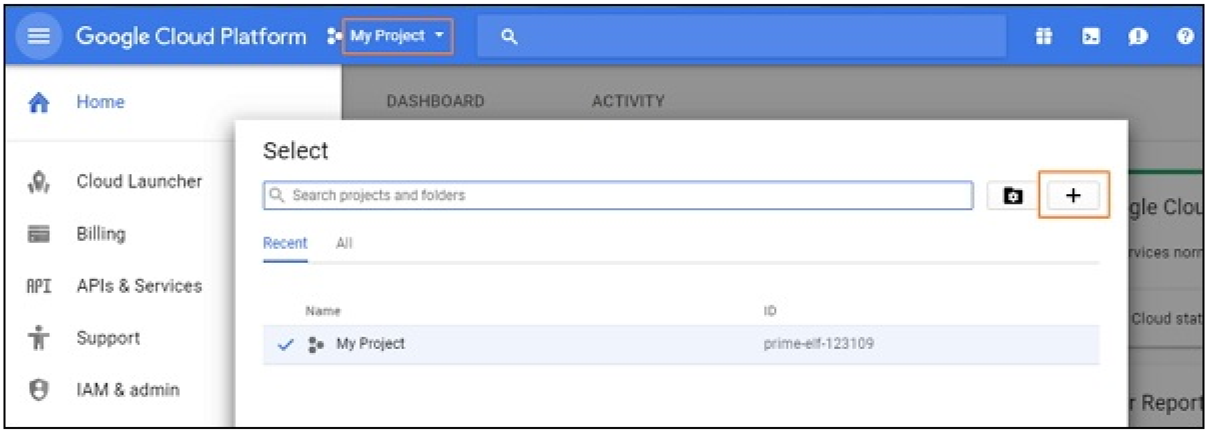
Number of nodes

3

Machine type See machine types

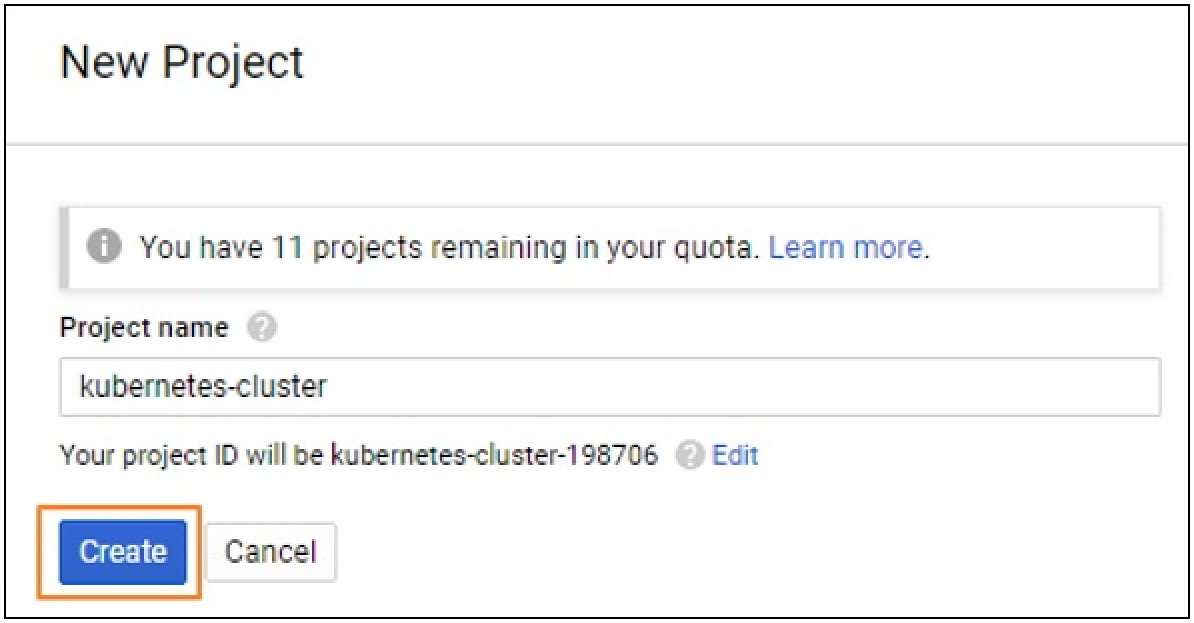
n1-standard-2

Step 7 − Before adding values in the fields, you need ID of the project which is created in the Google Cloud Platform console to host the Kubernetes cluster. To create ID, click on the *See your projects* link which is highlighted in the previous image. It will open the below screen, then click on *My Project* menu and click on the plus (+) icon to create a new project −



Name	ID
✓ My Project	prime-elf-123109

Step 8 − Now enter the project name and click on the *Create* button −



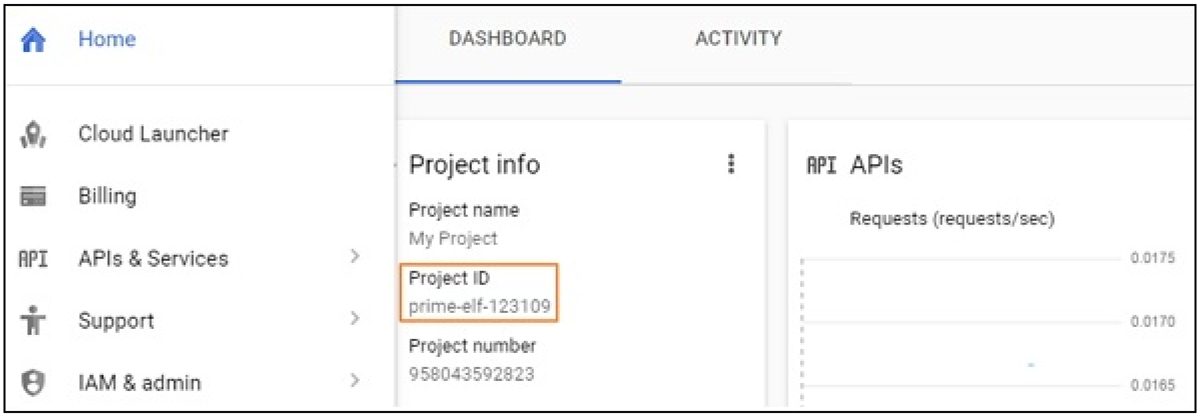
New Project

You have 11 projects remaining in your quota. Learn more.

Project name: kubernetes-cluster

Your project ID will be kubernetes-cluster-198706

Step 9 − You will get the ID of the project which will host the Kubernetes cluster −



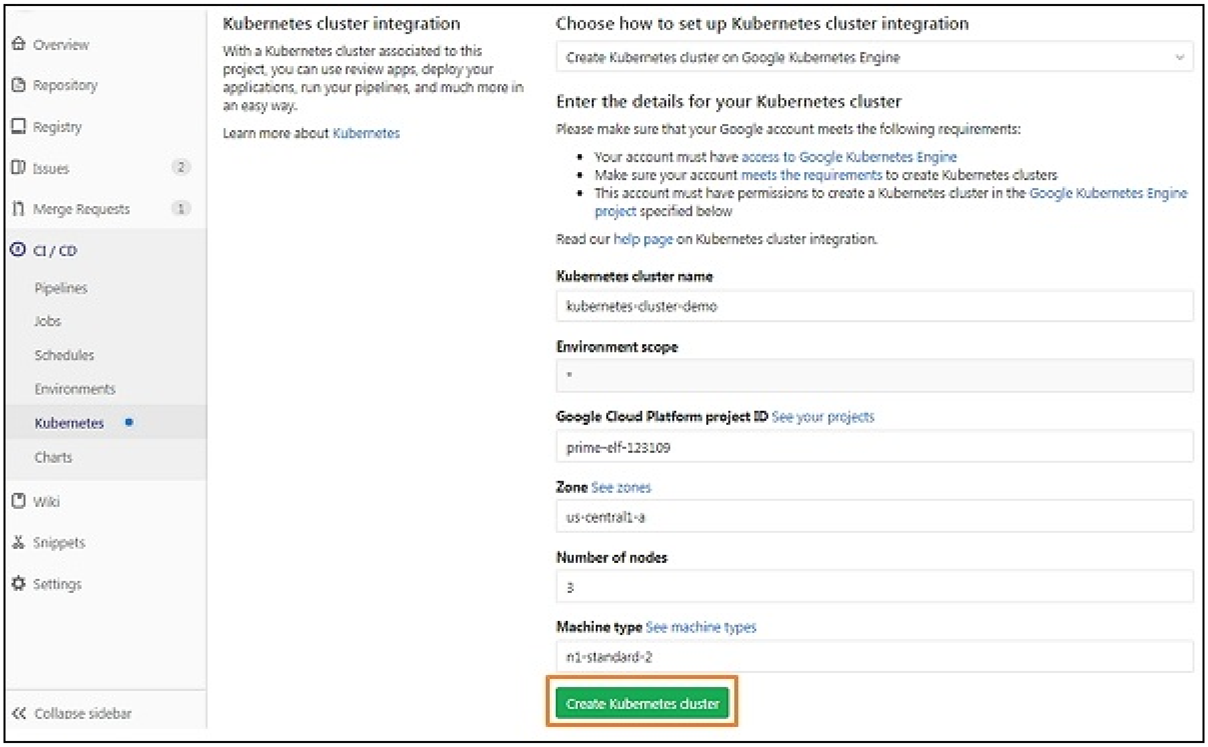
Project info

Project name: My Project

Project ID: prime-elf-123109

Project number: 958043592823

Step 10 − Enter the values in the fields for your Kubernetes cluster along with the Google Cloud Platform project ID and click on the *Create Kubernetes cluster* button −



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

Choose how to set up Kubernetes cluster integration

Create Kubernetes cluster on Google Kubernetes Engine

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.

Kubernetes cluster name

kubernetes-cluster-demo

Environment scope

*

Google Cloud Platform project ID See your projects

prime-elf-123109

Zone See zones

us-central1-a

Number of nodes

3

Machine type See machine types

n1-standard-2