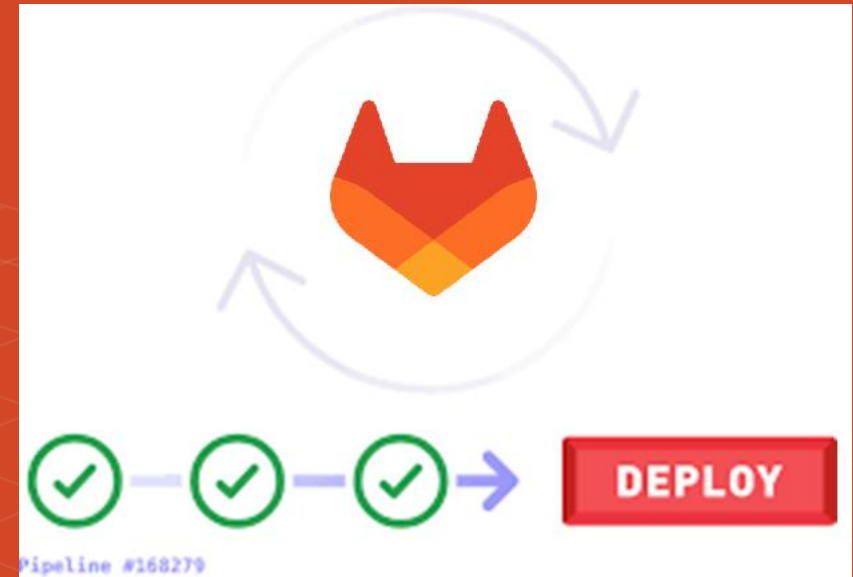


# GitLab and CI

How to get started with GitLab and CI





# GitLab and CI – Create Group

---

- Login to your GitLab account
- Click on  
  
Group -> Create group
- Assign group name “testgr” and click on “Create Group” without changing any setting.




## Create group

Assemble related projects together and grant members access to several projects at once.



# GitLab and CI – Create Project

- Click on  
Project -> Create blank Project
- Assign project name “testprj” and select “testgr” under Project URL for this project.
- Then, click on “Create Project” without changing any setting.



## Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

**Project name**

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain hyphens and slashes.

**Project URL**

testgr7113607

▼

 / 

**Project**

**Project deployment target (optional)**

▼

# GitLab and CI – Create Project



testgr / testprj

Group and Project Name

GitLab Duo Chat

T

testprj



Free



☆ Star

0

🍴 Fork

0



main

testprj /



Used to create  
new files

Find file

Edit

Code



Initial commit

Tejas Joshi authored 7 minutes ago

e8b75387



History

Name

Last commit

Last update

README.md

Initial commit

7 minutes ago

README.md

Files and commit history

Project information

1 Commit

1 Branch

0 Tags

3 KiB Project Storage

README

+ Add LICENSE

+ Add CHANGELOG

+ Add CONTRIBUTING

+ Add Kubernetes cluster

testprj



# GitLab and CI – Create File

- Create a new file with name “.gitlab.yml”.
- And add simple echo statements and click on “Commit Changes”.

## New file

main

.gitlab.yml

```
1 #Execute the simple code
2 build-steps:
3   script:
4     - echo "First Step in build..."
5     - echo "Second Step in build..."
```

## New file

### Commit changes

#### Commit message

The simple YML file with echo statement

#### Branch


- ☒ Commit to the current `main` branch
- ☐ Commit to a new branch

Cancel

Commit changes

# GitLab and CI – After creating a file



 The file has been successfully created.

## .gitlab-ci.yml

Find file

Blame

F



Add new .yml file with echo statements

Tejas Joshi authored 7 minutes ago

Status of job  
running on  
runner





85aea04e



F

✓ This GitLab CI configuration is valid. [Learn more](#)

 .gitlab-ci.yml  133 B

Edit ▾

Replace

Delete



```
1 #Execute the simple code
2 build-steps:
3   script:
4     - echo "First Step in build..."
5     - echo "Second Step in build.."
```

Files and commit history

# GitLab and CI – After creating a file



## .gitlab-ci.yml

[Find file](#)[Blame](#)[Permalink](#)

Green tick shows success



Add new .yml file with echo statements

Tejas Joshi authored 7 minutes ago



85aea04e



History

Click green tick to get the details

testgr / testprj / Pipelines / #1664180545

testgr / testprj / Jobs / #9093539467

## Add new .yml file with echo statements



Passed

Tejas Joshi created pipeline for commit 85aea04e 11 minutes ago, finished 9 r

For main

latest

1 job 1.81 1 minute 48 seconds, queued for 0 seconds

Pipeline

Jobs 1

Tests 0

test



build-steps



Click green tick to get the runner log

Search visible log output



```
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/testgr7113607/testprj/.git/
12 Created fresh repository.
13 Checking out 85aea04e as detached HEAD (ref is main)...
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}"
16 Executing "step_script" stage of the job script 00:01
17 Using docker image sha256:9b7e6c941969a4c237e2279c56b241deeab02099eeb6623d2
3d8dacf46c5b0cc for ruby:3.1 with digest ruby@sha256:da5c30fb27e6691500e7fa
1505f447748cfab455aee155087ff7d06e912ef79a ...
18 $ echo "First Step in build..."
19 First Step in build...
20 $ echo "Second Step in build.."
21 Second Step in build..
22 Cleaning up project directory and file based variables 00:00
23 Job succeeded
```



# GitLab and CI – Adding multiple stages

- Click on Project and select the .yml file for editing in pipeline editor.
- Add two more jobs in the file as shown.
- Add commit message and then click on Commit Changes.

[CI/CD Catalog](#) [Help](#)

```
1  #Execute the simple code
2  build-steps:
3    |   script:
4    |     - echo "First Step in build..."
5    |     - echo "Second Step in build.."
6  testing-job:
7    |   script:
8    |     - echo "First Step in testing..."
9    |     - echo "Second Step in testing.."
10 deploy-job:
11 |   script:
12 |     - echo "This is deploy Step..."
13
```

**Commit message**  

Update .gitlab-ci.yml file for two more jobs.

**Branch**  

main

[Commit changes](#) [Reset](#)






# GitLab and CI – Adding multiple stages

testgr / testprj / Pipeline editor


  main ▾

 Pipeline #1664207886 running  
Pipeline running for b52ee2f1

Click the pipeline id to get the status.  
All the three jobs are running in concurrent.

testgr / testprj / Pipelines / #1664207886

## Update .gitlab-ci.yml file for two more jobs.

 Passed Tejas Joshi created pipeline for commit b52ee2f1  9 min

For main

latest  3 jobs  1.52  31 seconds, queued for 1 seconds

Pipeline

Jobs 3

Tests 0

test

 build-steps



 deploy-job



 testing-job





# GitLab and CI – Adding multiple stages

- Click on Project and select the .yaml file for editing in pipeline editor.
- Add “stages” in the file as shown.
- Add commit message and then click on Commit Changes.
- Check the pipeline editor (as shown in previous slide).

```
CI/CD Catalog  Help



1  #Execute the simple code
2
3  stages:
4  |   - build
5  |   - test
6  |   - deploy
7
8  build-steps:
9  |   stage: build
10 |   script:
11 |       - echo "First Step in build..."
12 |       - echo "Second Step in build.."
13 testing-job:
14 |   stage: test
15 |   script:
16 |       - echo "First Step in testing..."
17 |       - echo "Second Step in testing.."
18 deploy-job:
19 |   stage: deploy
20 |   script:
21 |       - echo "This is deploy Step..."
```



# GitLab and CI – Adding multiple stages

testgr / testprj / Pipelines / #1664223169

## Update .gitlab-ci.yml file for stages.

 Running Tejas Joshi created pipeline for commit `fc511a3e`  7 minutes ago

For `main`

`latest`  3 jobs  In progress, queued for 1 seconds

Pipeline Jobs 3 Tests 0

build



build-steps



test



testing-job



deploy




deploy-job





# GitLab and CI – Adding before and after script

- Click on Project and select the .yml file for editing in pipeline editor.
- Add “before\_script” and “after\_script” in the file as shown.
- Add commit message and then click on Commit Changes.
- Check the pipeline editor (as shown in previous slide).

 .gitlab-ci.yml 525 B

```
1  #Execute the simple code
2
3  stages:
4    - build
5    - test
6    - deploy
7
8  before_script:
9    - echo "this is before script"
10
11 after_script:
12   - echo "this is after script"
13
14 build-steps:
15   stage: build
16   script:
17     - echo "First Step in build..."
18     - echo "Second Step in build.."
19 testing-job:
20   stage: test
21   script:
22     - echo "First Step in testing..."
23     - echo "Second Step in testing.."
24 deploy-job:
25   stage: deploy
```



# GitLab and CI – Adding before and after script

## Update `.gitlab-ci.yml` file before and after script added.

✓ Passed Tejas Joshi created pipeline for commit `3cf3a8ac` 6 minutes ago, finished 5 minutes ago

For `main`

3 jobs 1.45 1 minute 26 seconds, queued for 0 seconds

Pipeline

Jobs

3

Tests

0

build



build-steps



test



testing-job

passed

deploy



deploy-job





# GitLab and CI – Adding before and after script

```
Search visible
12 Created fresh repository.
13 Checking out 3cf3a8ac as detached HEAD (ref is
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}"
16 Executing "step_script" stage of the job script
17 Using docker image sha256:9b7e6c941969a4c237e2279c
b0cc for ruby:3.1 with digest ruby@sha256:da5c30fb
55087ff7d06e912ef79a ...
18 $ echo "this is before script"
19 this is before script
20 $ echo "First Step in build..."
21 First Step in build...
22 $ echo "Second Step in build.."
23 Second Step in build..
24 Running after_script
25 Running after script...
26 $ echo "this is after script"
27 this is after script
28 Cleaning up project directory and file based variables
29 Job succeeded
```

```
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/testgro
12 Created fresh repository.
13 Checking out 3cf3a8ac as detached HEAD (ref is mai
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}"
16 Executing "step_script" stage of the job script
17 Using docker image sha256:9b7e6c941969a4c237e2279c
b0cc for ruby:3.1 with digest ruby@sha256:da5c30fb
55087ff7d06e912ef79a ...
18 $ echo "this is before script"
19 this is before script
20 $ echo "This is deploy Step..."
21 This is deploy Step...
22 Running after_script
23 Running after script...
24 $ echo "this is after script"
25 this is after script
26 Cleaning up project directory and file based variables
27 Job succeeded
```



# GitLab and CI – Running a bash script

- Now create a “try.sh” bash script file as shown using new file option inside the project.

 **try.sh**

 **Edit try.sh**  
Tejas Joshi authored 1 minute ago

 **try.sh**  137 B

1	<code>#!/bin/bash</code>
2	<code>echo "this is try bash script."</code>
3	<code>touch myfile.txt</code>
4	<code>echo "Sample text added" &gt; myfile.txt</code>
5	<code>echo "this is end of bash script."</code>



# GitLab and CI – Running a bash script

- Now update the .yml file for executing the bash script as shown.
- Add commit message and then click on Commit Changes.
- Check the pipeline editor (as shown in previous slide).

A screenshot of the GitLab CI/CD pipeline editor interface. At the top, there are tabs for 'Edit', 'Visualize', 'Validate', 'NEW' (highlighted in blue), and 'Full configuration'. Below the tabs, there are two buttons: 'CI/CD Catalog' and 'Help'. The main area shows a YAML configuration for a pipeline job. The lines are numbered 1 to 5 on the left. The content is as follows:

```
1  # To run the bash script
2
3  bash_execution:
4    script:
5    - bash ./try.sh
```

The line containing the script command is highlighted in yellow.





# GitLab and CI – Running a bash script

## Update .gitlab-ci.yml file to run bash script.

✓ Passed Tejas Joshi created pipeline for commit 2b5cc029 1 min

For main

latest 1 job 1.1 1 minute 5 seconds, queued for 3 seconds

Pipeline Jobs 1 Tests 0

test

✓ bash\_execution

```
55087ff7d06e912ef79a ...
7 Preparing environment
8 Running on runner-jhcjxvh-project-66979527-con
4-1739253961-76c83eec...
9 Getting source from Git repository
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/te
12 Created fresh repository.
13 Checking out 269b086c as detached HEAD (ref is
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_U
16 Executing "step_script" stage of the job scrip
17 Using docker image sha256:9b7e6c941969a4c237e2
b0cc for ruby:3.1 with digest ruby@sha256:da5c
55087ff7d06e912ef79a ...
18 $ bash ./try.sh
19 this is try bash script.
20 this is end of bash script.
21 Cleaning up project directory and file based v
22 Job succeeded
```



# GitLab and CI – Running a bash script

- Now update the .yml file for few more commands as shown.
- Add commit message and then click on Commit Changes.
- Check the pipeline editor (as shown in previous slide).

The screenshot displays the GitLab CI/CD Pipeline Editor. At the top, it indicates 'Pipeline #1665021420 running' for commit '51e5daf6'. Below this, a green checkmark confirms 'Pipeline syntax is correct.' with a 'Learn more' link. A navigation bar includes 'Edit' (selected), 'Visualize', 'Validate', 'NEW' (highlighted), and 'Full configuration'. Below the navigation bar are buttons for 'CI/CD Catalog' and 'Help'. The main area shows a YAML configuration for a job named 'bash\_execution' with a 'script' section containing four commands: 'mkdir myfolder', 'cd myfolder', 'bash ./try.sh', and 'touch newtext.txt'. The last command is highlighted in yellow.

```
1  # To run the bash script
2
3  bash_execution:
4    script:
5      - mkdir myfolder
6      - cd myfolder
7      - bash ./try.sh
8      - touch newtext.txt
```



# GitLab and CI – Running a bash script

**Update .gitlab-ci.yml file for few more commands**

✖ Failed Tejas Joshi created pipeline for commit `51e5daf6` 1 minute ago, finished

For `main`

latest 1 job 0.47 27 seconds, queued for 0 seconds

Pipeline Jobs 1 Failed Jobs 1 Tests 0

test

✖ bash\_execution

The .sh file is not in the myfolder and hence giving the error

```
Preparing environment
Running on runner-jlguopmm-project-66979527-concurrent-64-1739254237-e3239c8f...
Getting source from Git repository
Fetching changes with git depth set to 20...
Initialized empty Git repository in /builds/testgr...
Created fresh repository.
Checking out 51e5daf6 as detached HEAD (ref is main)
Skipping Git submodules setup
$ git remote set-url origin "${CI_REPOSITORY_URL}"
Executing "step_script" stage of the job script
Using docker image sha256:9b7e6c941969a4c237e2279c5b0cc for ruby:3.1 with digest ruby@sha256:da5c30fb255087ff7d06e912ef79a ...
$ mkdir myfolder
$ cd myfolder
$ bash ./try.sh
bash: ./try.sh: No such file or directory
Cleaning up project directory and file based variables
ERROR: Job failed: exit code 1
```



# GitLab and CI – Running a artifacts and path

- Now update the .yml file for “artifacts & path” as shown. This will preserve the files and folders created by script.
- Add commit message and then click on Commit Changes.
- Check the pipeline editor (as shown in previous slide).

✓ Pipeline syntax is correct. [Learn more](#)

Edit

Visualize

Validate

NEW

Full configuration

CI/CD Catalog

Help

```
1  # To run the bash script
2
3  bash_execution:
4    script:
5      - mkdir myfolder
6      - cp ./try.sh myfolder/
7      - cd myfolder
8      - bash ./try.sh
9      - touch newtext.txt
10
11    artifacts:
12      paths:
13        - "./myfolder"
14      when: on_success
15      access: all
16      expire_in: 30 days
```



# GitLab and CI – Running a artifacts and path

## Update .gitlab-ci.yml file added artifact and cp command

✓ Passed Tejas Joshi created pipeline for commit 618d019d 1 minute ago, finished just r

For main

latest 1 job 0.52 31 seconds, queued for 0 seconds

Pipeline Jobs 1 Tests 0

test

✓ bash\_execution

The .sh file is not in the myfolder and  
hence giving the error

```
$ mkdir myfolder
$ cp ./try.sh myfolder/
$ cd myfolder
$ bash ./try.sh
this is try bash script.
this is end of bash script.
$ touch newtext.txt
Uploading artifacts for successful job
Uploading artifacts...
./myfolder: found 4 matching artifact files and directories
WARNING: Upload request redirected location=https
4/jobs/9099268690/artifacts?artifact_format=zip&artifact_type=arch
s new-url=https://gitlab.com
WARNING: Retrying... context=artifa
equest redirected
Uploading artifacts as "archive" to coordinator... 201 Created id
Status=201 Created token=glcvt-66
Cleaning up project directory and file based variables
Job succeeded
```



# GitLab and CI – Running a artifacts and path

00:03

stories  
location=https://gitlab.com/api/v  
fact\_type=archive&expire\_in=30+day  
context=artifacts-uploader error=r  
01 Created id=9099268690 response  
es

00:00

Duration: 31 seconds  
Finished: 3 minutes ago  
Queued: 0 seconds  
Timeout: 1h (from project) ?  
Runner: #12270848 (ns46NMmJ) 2-  
green.saas-linux-small-  
amd64.runners-  
manager.gitlab.com/default

Job artifacts ?  
These artifacts are the latest. They will  
not be deleted (even if expired) until  
newer artifacts are available.

KeepDownloadBrowse

Commit 610d019d  
Update .gitlab-ci.yml file added artifact  
and cp command

Pipeline #1665032894 Passed for

Check the right section of log for Job artifacts.  
Click on Browse or Download to get the  
artifacts.



# GitLab and CI – Running pipeline manually

- To run the pipeline manual, use the Build->Pipelines menu which displays all the pipelines previously created with their status.

All 14 Finished Branches Tags [View analytics](#) [Clear runner caches](#) [New pipeline](#)

Filter pipelines  [Show Pipeline ID](#)

Status	Pipeline	Created by	Stages	Actions
✓ Passed 00:00:31 7 minutes ago	<a href="#">Edit try.sh</a> #1665059985 <small>main</small> <small>b1f6722e</small>		✓ latest	
✓ Passed 00:00:32 12 minutes ago	<a href="#">Update file try.sh</a> #1665055448 <small>main</small> <small>6d523859</small>		✓	
✓ Passed 00:00:30	<a href="#">Edit try.sh</a> #1665043356 <small>main</small> <small>4fd97c8a</small>		✓	

Project

T testprj

📌 Pinned

Issues 0

Merge requests 0

👤 Manage

📅 Plan

</> Code

🚀 Build

🛡️ Secure

📦 Deploy

🔧 Operate

📊 Monitor

main testprj /

Edit try.sh  
Tejas Joshi authored 5

**Name**

.gitlab-ci.yml

README.md

try.sh

Pipelines

Jobs

Pipeline editor

Pipeline schedules

Artifacts





# GitLab and CI – Scheduling of pipelines

- To run the pipeline at a particular schedule, use the Build->Pipeline schedules menu and set the appropriate time.

The screenshot illustrates the navigation path to configure pipeline schedules in GitLab. On the left, the 'Build' menu item is highlighted with a blue border, and a dropdown menu is open, showing 'Pipeline schedules' as the selected option. In the center, the 'Pipeline schedules' page is visible, featuring a 'Create a new pipeline schedule' button and a list of pipeline details. On the right, the 'Project' sidebar is shown with the 'Build' option highlighted by a red box. A blue arrow points from this 'Build' option to the 'Pipeline schedules' dropdown in the left sidebar. Another blue arrow points from the 'Pipeline schedules' dropdown to the 'Pipeline schedules' page in the center.

**Project**

- T testprj
- Pinned
- Issues 0
- Merge requests 0
- Manage
- Plan
- Code
- Build**
- Secure
- Deploy
- Operate
- Monitor

**main** testprj /

**Edit try.sh**  
Tejas Joshi authored 5

**Name**

- .gitlab-ci.yml
- README.md
- try.sh

**Pipeline schedules**

A scheduled pipeline starts automatically at regular intervals

The pipeline:

- Runs for a specific branch or tag
- Can have custom CI/CD variable
- Runs with the same project permissions as the

Learn more in the [scheduled pipelines document](#)

[Create a new pipeline schedule](#)

**Pipelines**

- Jobs
- Pipeline editor
- Pipeline schedules**
- Artifacts





# GitLab and CI – Create Project to deal with HelloWorld.java

- Click on

Project -> Create blank Project

- Assign project name “HelloWorldJava” and select “testgr” under Project URL for this project.
- Then, click on “Create Project” without changing any setting.

The screenshot shows the GitLab web interface for creating a new project. On the left, a sidebar menu lists various project management options, with 'Projects' currently selected. The main content area is titled 'Create blank project' and contains several input fields and options. The 'Project name' field is filled with 'HelloWorldJava'. The 'Project URL' field is a dropdown menu showing 'https://gitlab.com/' and 'testgr7113607'. The 'Project slug' field is filled with 'helloworldjava'. The 'Project deployment target (optional)' field is a dropdown menu showing 'Select the deployment target'. The 'Visibility Level' section has three radio buttons: 'Private' (selected), 'Internal', and 'Public'. Below the 'Private' radio button, there is a note: 'Project access must be granted explicitly to each user. If this project is part of a group, access is gr'.



# GitLab and CI – Create Project to deal with HelloWorld.java

- Create a “HelloWorld.java” file and add java code in it.
- Create a configuration file – “.gitlab-ci.yml” for the project.
- Contents of the configurations file are on next slide.

testgr / HelloWorldJava / Repository

## New file

main

HelloWorld.java

```
1 class HelloWorld {  
2     public static void main(String a[]){  
3         System.out.println("Hello World");  
4     }  
5 }
```



# GitLab and CI – Scheduling of pipelines

main

.gitlab-ci.yml

Apply a template ▾

```
1 stages:
2   - build
3   - test
4
5 before_script:
6   - apt-get update && apt-get install -y openjdk-17-jdk # Install Java
7
8 build:
9   stage: build
10  script:
11    - javac HelloWorld.java # Compile the Java program
12    - ls -l
13  artifacts:
14    paths:
15    - HelloWorld.class # Save the compiled file for the next job
16  only:
17    - main
18
19 test:
20   stage: test
21   script:
22     - ls -l # Verify if the class file is present
23     - java HelloWorld # Run the program to check execution
24   only:
25     - main
```



# GitLab and CI – Create Project to deal with HelloWorld.java

- Add another statement in the java program check the output.

```
807 update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstatd (jstatd) in auto mode
808 update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/serialver (serialver) in auto mode
809 update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb (jhsdb) in auto mode
810 Setting up openjdk-17-jre:amd64 (17.0.14+7-1~deb12u1) ...
811 Setting up openjdk-17-jdk:amd64 (17.0.14+7-1~deb12u1) ...
812 update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jconsole (jconsole) in auto mode
813 $ ls -l
814 total 16
815 -rw-r--r-- 1 root root 470 Feb  9 18:17 HelloWorld.class
816 -rw-rw-rw- 1 root root 151 Feb  9 18:17 HelloWorld.java
817 -rw-rw-rw- 1 root root 6194 Feb  9 18:17 README.md
818 $ java HelloWorld
819 Hello World
820 Another line in Hello World
821 Cleaning up project directory and file based variables
822 Job succeeded
```



# GitLab and CI – Create Project to deal with HelloWorld.java

- To manually execute any of the stage of .yml file, one can add “when” as shown here –

```
16   only:
17     - main
18
19   test:
20     stage: test
21     when: manual
22     script:
23       - ls -l # Verify if the class file is present
24       - java HelloWorld # Run the program to check execution
25   only:
26     - main
```

## Update .gitlab-ci.yml file test stage cc

✓ Passed Tejas Joshi created pipeline for commit 78ceb72b

For main

latest 2 jobs 0.99 59 seconds, queued for 0 second

Pipeline

Jobs 2

Tests 0

build

✓ build

test

⚙ test



# GitLab and CI – Create Project to deal with HelloWorld.java

---

- Open Git Bash on local machine and run following commands –

```
admin@DESKTOP-QBTKOPB MINGW64 ~  
$ cd d:  
  
admin@DESKTOP-QBTKOPB MINGW64 /d  
$ mkdir HelloWorldPrj  
  
admin@DESKTOP-QBTKOPB MINGW64 /d  
$ cd HelloWorldPrj  
  
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj  
$ git clone https://gitlab.com/testgr7113607/helloworldjava.git  
Cloning into 'helloworldjava'...  
remote: Enumerating objects: 15, done.  
remote: Counting objects: 100% (15/15), done.  
remote: Compressing objects: 100% (14/14), done.  
remote: Total 15 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (15/15), 4.40 KiB | 562.00 KiB/s, done.  
Resolving deltas: 100% (2/2), done.
```



# GitLab and CI – Create Project to deal with HelloWorld.java

---

- Run following commands –

```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj  
$ cd helloworldjava
```

```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj/helloworldjava (main)  
$ vi HelloWorld.java
```

```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj/helloworldjava (main)  
$ git add .
```

```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj/helloworldjava (main)  
$ git add HelloWorld.java
```

```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj/helloworldjava (main)  
$ git commit -m "Local Machine added a statement.."
```



# GitLab and CI – Create Project to deal with HelloWorld.java

---

- Run following commands –


```
admin@DESKTOP-QBTKOPB MINGW64 /d/HelloWorldPrj/helloworldjava (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 400 bytes | 400.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/testgr7113607/helloworldjava.git
78ceb72..d4c3ec5  main -> main
```







# GitLab and CI – Create Project to deal with HelloWorld.java

testgr / HelloWorldJava





## HelloWorldJava Free


main 


helloworldjava / 


Find file

Edit 




Code 

 Local Machine added a statement..  
Tejas Joshi authored 4 minutes ago



d4c3ec5d 

History

Name	Last commit	Last update
 .gitlab-ci.yml	Update .gitlab-ci.yml file test ...	18 minutes ago
 HelloWorld.java	Local Machine added a state...	4 minutes ago
 README.md	Initial commit	33 minutes ago



# GitLab and CI – Create Project to deal with HelloWorld.java

testgr / HelloWorldJava / Pipelines / #1665221117

**Local Machine added a statement..**

✓ Passed Tejas Joshi created pipeline for commit d4c3ec5d

For main

latest @ 2 jobs ⌚ 0.96 ⌚ 57 seconds, queued for 1 seconds

Pipeline

Jobs 2

Tests 0

build

test

✓ build



⚙ test

