

```
























































deploy to preprod:
  stage: deploy
  variables:
    TARGET_ENV: preprod
    MyLogin: "Preprod ROMAN"
  script:
    - echo "Do your deploy here to ${TARGET_ENV}"
    - echo ${DB_SERVER}
    - echo "MyLogin"
    - echo $MyLogin
    - echo "MyPassword"
    - echo $MyPassword
  only:
    - main
  environment:
    name: preprod

```

```

deploy to production:
  stage: deploy
  variables:
    TARGET_ENV: production
    MyLogin: "Production ROMAN"
  script:
    - echo "Do your deploy here to ${TARGET_ENV}"
    - echo ${DB_SERVER}
    - echo "MyLogin"
    - echo $MyLogin
    - echo "MyPassword"
    - echo $MyPassword
  only:
    - main
  environment:
    name: production

```

↑ Key	Value	Environments	Actions
DB_SERVER  Expanded	***** 	production 	 
DB_SERVER  Expanded	***** 	preprod 	 
DB_SERVER  Protected Expanded	***** 	prod 	 
DB_SERVER  Expanded	***** 	staging 	 
GITLAB_CL_PASSWORD  Masked Expanded	***** 	All (default) 	 
GITLAB_CL_USER  Expanded	***** 	All (default) 	 
MyPassword  Protected Expanded	***** 	staging 	 
MyPassword  Protected Expanded	***** 	production 	 
MyPassword  Protected Expanded	***** 	prod 	 
MyPassword  Protected Expanded	***** 	preprod 	 
RUNNER_TOKEN  Masked Expanded	***** 	All (default) 	 

```

18 Successfully extracted cache
19 Downloading artifacts
20 Downloading artifacts for build1 (5669333461)...
21 Runtime platform                                arch=amd64 os=linux pid=27043
   on=f5da3c5a version=16.6.1
22 Downloading artifacts from coordinator... ok      host=cdn.artifacts.gitlab-stat
   id=5669333461 responseStatus=200 OK token=64_bLn4N
23 Executing "step_script" stage of the job script
24 $ echo "Before script section"
25 Before script section
26 $ echo "For example you might run an update here or install a build dependency"
27 For example you might run an update here or install a build dependency
28 $ echo "Or perhaps you might print out some debugging details"
29 Or perhaps you might print out some debugging details
30 $ echo "Do your deploy here to ${TARGET_ENV}"
31 Do your deploy here to preprod
32 $ echo ${DB_SERVER}
33 db-preprod.domain.local
34 $ echo "MyLogin"
35 MyLogin
36 $ echo $MyLogin
37 Preprod P0
38 $ echo "MyPassword"
39 MyPassword
40 $ echo $MyPassword
41 preprod1234
42 Running after_script
43 Running after script...
44 $ echo "After script section"
45 After script section
46 $ echo "For example you might do some cleanup here"
47 For example you might do some cleanup here
48 Saving cache for successful job
49 Creating cache 0_composer-1217e64b9c43c35c95f90914c08cf7a5d2e4f4ac-protected...
50 Runtime platform                                arch=amd64 os=linux pid=27087
   on=f5da3c5a version=16.6.1
51 vendor/: found 2 matching artifact files and directories
52 Archive is up to date!
53 Created cache
54 Cleaning up project directory and file based variables
55 Job succeeded

```

```

# ----- Cancel -----
cancel:
  stage: stop previous jobs
  image: everpeace/curl-jq
  script:
    - |
      if [ "$CI_COMMIT_REF_NAME" == "main" ]
      then
        (
          echo "Cancel old pipelines from the same branch except last"
          OLD_PIPELINES=$( curl -s -H "PRIVATE-TOKEN: $RUNNER_TOKEN" "https://gitlab.com/api/v4/projects/${CI_PROJECT_ID}/pipelines" | jq '.[0] | .id' | tail -n +2 )
          for pipeline in ${OLD_PIPELINES}; \
          do echo "Killing ${pipeline}" && \
              curl -s --request POST -H "PRIVATE-TOKEN: ${RUNNER_TOKEN}" "https://gitlab.com/api/v4/projects/${CI_PROJECT_ID}/pipelines/${pipeline}" \
          ) || echo "Canceling old pipelines (${OLD_PIPELINES}) failed"
        )
      fi

```

```
16 Checking cache for 0_composer-1217e64b9c43c35c95f90914c08cf7a5d2e4f4ac-protected..  
17 Runtime platform                                arch=amd64 os=linux pid=27620  
   on=f5da3c5a version=16.6.1  
18 No URL provided, cache will not be downloaded from shared cache server. Instead a  
   version of cache will be extracted.  
19 Successfully extracted cache  
20 Downloading artifacts  
21 Downloading artifacts for build1 (5669333461)...  
22 Runtime platform                                arch=amd64 os=linux pid=27644  
   on=f5da3c5a version=16.6.1  
23 Downloading artifacts from coordinator... ok      host=cdn.artifacts.gitlab-stat  
   id=5669333461 responseStatus=200 OK token=64_04xuZ  
24 Downloading artifacts for pages (5669333473)...  
25 Runtime platform                                arch=amd64 os=linux pid=27651  
   on=f5da3c5a version=16.6.1  
26 Downloading artifacts from coordinator... ok      host=cdn.artifacts.gitlab-stat  
   id=5669333473 responseStatus=200 OK token=64_D4xuZ  
27 Executing "step_script" stage of the job script  
28 $ echo "Before script section"  
29 Before script section  
30 $ echo "For example you might run an update here or install a build dependency"  
31 For example you might run an update here or install a build dependency  
32 $ echo "Or perhaps you might print out some debugging details"  
33 Or perhaps you might print out some debugging details  
34 $ if [ "$CI_COMMIT_REF_NAME" == "main" ] # collapsed multi-line command  
35 Cancel old pipelines from the same branch except last  
36 bash: line 155: jq: command not found  
37 Running after_script  
38 Running after script...  
39 $ echo "After script section"  
40 After script section  
41 $ echo "For example you might do some cleanup here"  
42 For example you might do some cleanup here  
43 Saving cache for successful job  
44 Creating cache 0_composer-1217e64b9c43c35c95f90914c08cf7a5d2e4f4ac-protected..  
45 Runtime platform                                arch=amd64 os=linux pid=27701  
   on=f5da3c5a version=16.6.1  
46 vendor/: found 2 matching artifact files and directories  
47 Archive is up to date!  
48 Created cache  
49 Cleaning up project directory and file based variables  
50 Job succeeded
```