What are Gitlab CI Artifacts?

Artifacts are files generated through a process that we use Job or Pipeline. Of course, it’s not all Jobs that create files, but if your projects generate files, we can use the GitLab Artifacts to manage the files for us.

Gitlab will store that artifact, and we can download it using the UI or calling the API.

The Artifact Keyword

We need to use the keyword **artifacts** on our **gitlab-ci.yml** file to use the Gitlab CI Artifacts, specifying which files we would like to store as job artifacts.

What are Job Artifacts?

Job artifacts are a list of directories and files linked to the Job regardless of whether the Job succeeds or fails.

At the end of the Job, the GitLab stores the artifact, and the file will be available for us to download using the UI, only if the artifact size is smaller than the maximum size that we configure to allow it.

Gitlab CI Artifact Example

```
pre_build:
  stage: Pre Build
  image: 123456789011.dkr.ecr.us-east-
1.amazonaws.com/bitslovers/blog:latest
  tags:
    - fargate
  only:
    - master
  script:
    - echo "export BRANCH=\"$(echo $BRANCH)\"" >> variables
    - echo "export RELEASE_VERSION=\"$(echo $RELEASE_VERSION)\"" >>
variables
    - echo "export DB_HOST=\"$(echo $DB_HOST)\"" >> variables
    - echo "export DNS=\"$(echo $DNS)\"" >> variables
  artifacts:
    paths:
      - vars_file
  retry:
    max: 2
    when:
      - runner_system_failure
```

Gitlab Artifacts Between Stages

By default, jobs in later stages automatically download all the artifacts created by jobs in earlier stages. However, you can control artifact download behavior in jobs with **dependencies**.

```
build_app:
  stage: build_app
  script: make build:app
  artifacts:
    paths:
      - bin/
test:
  stage: test_app
  script: make test:app
  dependencies:
    - build_app
```

When using the **needs** keyword, jobs can only download artifacts from the positions defined in the needs configuration.

Job artifacts are collected for successful jobs by default and restored after caches.

Where are artifacts stored in GitLab?

The default location that Gitlab uses to store the files are **/var/opt/gitlab/gitlab-rails/shared/artifacts**.

If you change this location, you need to restart Gitlab.

GitLab Artifact Expiration

Sometimes we don’t want to fill our server with many files. For example, managing files for a big company with many projects and developers could be a big problem. Occasionally, we know that we should not keep some files for a long time on the server, and we could tell GitLab when this file expires or mark to delete.

To delete files after a specific creation period, we can use the **expire_in** keyword to define how long GitLab holds the artifacts. We can also use the Gitlab UI to store artifacts from expiring. If **expire_in** is not specified, the files are deleted after 30 days.

GitLab Artifact Path

We can use multiple paths and wildcards if various files are on the same path.

```
build-job:
  script:
    - mvn package -U
  artifacts:
    paths:
      - target/*.war
```

GitLab Artifact Size Limit

The default size limit for the artifact on Gitlab is 100 MB. However, we can easily change it on Admin Area:

Admin -> Settings -> CI/CD -> Continuous Integration and Deployment