GitLab - Documentation on the Project

> *This page will be used as a base line for Migrating the pipeline(s) to ADO*

Source Code and Pipeline:

BEA Team is using the GitLab as Source Code repository and using .gitlab-ci.yml file for executing the code in GitLab.

Purpose of the Pipeline:

The pipeline is used to perform CI and CD with in the same pipeline. This Pipeline uses multiple Tasks/Jobs to execute the code which would be explained in detailed in the "Tasks Explanation" section below.

There are few specific parameters which would differentiate the tasks/jobs execution based on the keywords.

What are required(Pre-Requisites):

Docker Images availability in container

Accessibility from ADO to Containers and vice versa

Required Software's installed on the ADO Agent machines

Global Parameters of the Pipeline:

Below are the different variables used in the pipeline. Few of the variables are using the predefined variables from GitLab

Variables:

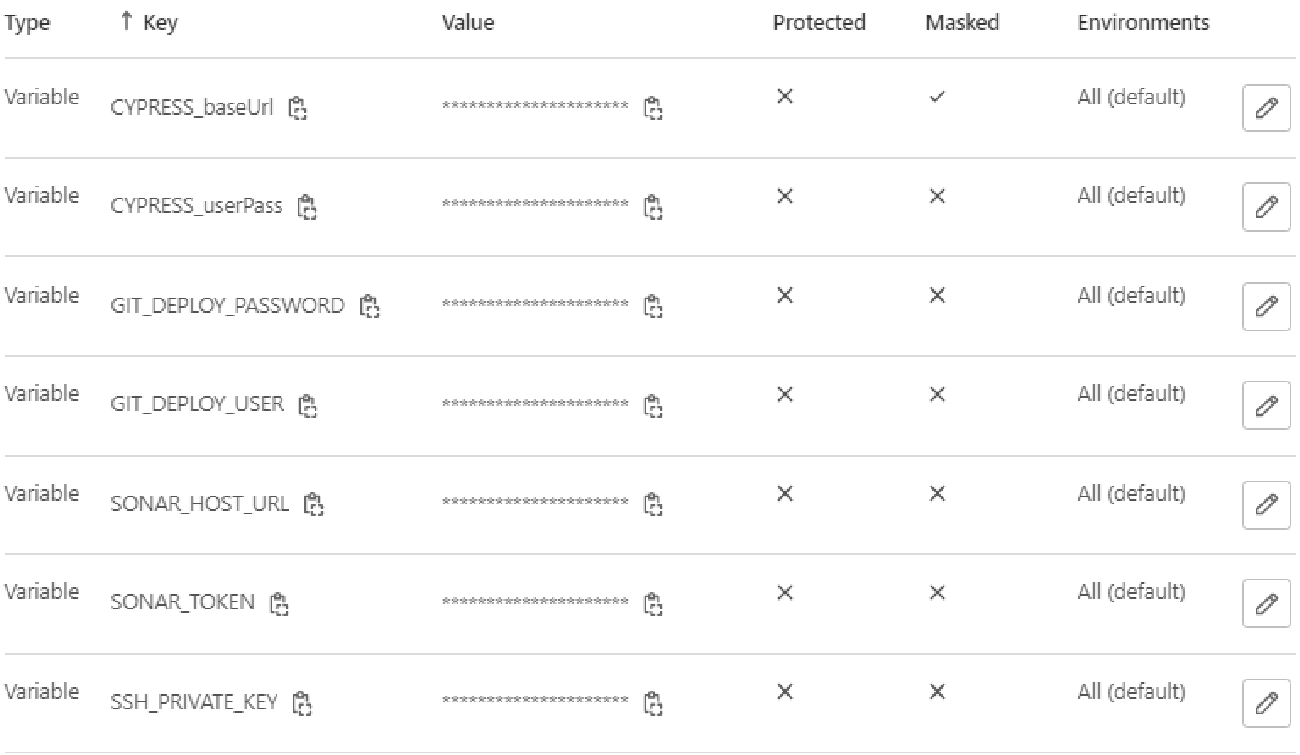
```
    MYSQL_HOST: db
    MYSQL_DATABASE: abbviebea
    MYSQL_USER: root
    MYSQL_PASSWORD: root
    MEMCACHED_HOST: memcached
    MEMCACHED_PORT: 11211
    SOLR_HOST: solr
    SOLR_PORT: 8983
    DB_IMAGE_TAG: $CI_REGISTRY_IMAGE/bd
    BEA_BASE_IMAGE_TAG: $CI_REGISTRY_IMAGE/bea-base
    CYPRESS_IMAGE_TAG: $CI_REGISTRY_IMAGE/cypress
    BEA_IMAGE_TAG: $CI_REGISTRY_IMAGE/bea
    SOLR_IMAGE_TAG: $CI_REGISTRY_IMAGE/solr
    MEMCACHED_IMAGE_TAG: ${CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX}/memcached:1.5.6
    RUBY_IMAGE_TAG: ${CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX}/ruby:2.6.6-buster
    DOCKER_IMAGE_TAG: ${CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX}/docker:19.03.13
    COMPOSER_IMAGE_TAG: ${CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX}/composer:2.0.6
    COMPOSER_MEMORY_LIMIT: 3G
```

Predefined Variables (yaml) from GitLab:

CI_REGISTRY_IMAGE : The address of the project's Container Registry. Only available if the Container Registry is enabled for the project.

CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX : The top-level group image prefix for pulling images through the Dependency Proxy.

Protected or Masked Variable for Pipeline:



Type	↑ Key	Value	Protected	Masked	Environments
Variable	CYPRESS_baseUrl	**********************	✕	✓	All (default)
Variable	CYPRESS_userPass	**********************	✕	✕	All (default)
Variable	GIT_DEPLOY_PASSWORD	**********************	✕	✕	All (default)
Variable	GIT_DEPLOY_USER	**********************	✕	✕	All (default)
Variable	SONAR_HOST_URL	**********************	✕	✕	All (default)
Variable	SONAR_TOKEN	**********************	✕	✕	All (default)
Variable	SSH_PRIVATE_KEY	**********************	✕	✕	All (default)

Stages:

test
sonar
deploy
build-e2e
e2e

Tasks under the stages of Pipeline:

Below are the different Tasks/Jobs executing in different stages as part of the pipeline. All tasks are the script based.

test:

All the jobs under test stage uses the bea:latest docker image at registry.gitlab.com/abbviebea and used the QA GitLab Runner at the given path. These jobs executes only when the merge requests created and run the shell script.

csslinter

This job runs the npm install and stylelint.

jslinter

This job runs the npm install and eslint.

e2e_eslint

This job runs the npm install and eslint.

php_unit

This job has before script defined. So, this job runs the composer, patch and php commands once pecl and docker-php-ext-enable runs.

phpcs

This job runs the composer and php

sonar:

`sonarqube-check`

SonarQube check job uses the `sonarsource/sonar-scanner-cli:latest` docker image and runs the SonarQube analysis.

deploy:

All the jobs under test stage uses the RUBY_IMAGE_TAG Docker image which defined as variable. `deploy_preview` and `deploy_pro` used the pro Runner and rest all uses the QA.

Depends on the keyword except these jobs runs manually or as per schedule

`deploy_dev`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano dev deploy

`deploy_qa`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano qa deploy

`deploy_uat`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano uat deploy

`deploy_drupal9`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano drupal9 deploy

`deploy_preview`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano preview deploy

`deploy_pro`

This job adds a private key, creates a directory and provides the right permissions to install bundle and executes the capistrano pro deploy

build-e2e:

`.build-template`

This job uses the `DOCKER_IMAGE_TAG` Docker image which defined as variable and runs the docker login command. This job runs on Qa Runner.

`build-db`, `build-bea-base`, `build-bea`, `build-cypress`, `build-solr` jobs are dependent on `.build-template` and executes the docker pull and push commands.

e2e

`.e2e-template:`

This job uses the $CYPRESS_IMAGE_TAG:latest Docker image defined as variable and runs the shell commands

Keywords:

Keyword	**Description**
after_script	Override a set of commands that are executed after job.
allow_failure	Allow job to fail. A failed job does not cause the pipeline to fail.
artifacts	List of files and directories to attach to a job on success.
before_script	Override a set of commands that are executed before job.
cache	List of files that should be cached between subsequent runs.
coverage	Code coverage settings for a given job.
dast_configuration	Use configuration from DAST profiles on a job level.
dependencies	Restrict which artifacts are passed to a specific job by providing a list of jobs to fetch artifacts from.
environment	Name of an environment to which the job deploys.
except	Control when jobs are not created.
extends	Configuration entries that this job inherits from.
image	Use Docker images.
inherit	Select which global defaults all jobs inherit.

interruptible	Defines if a job can be canceled when made redundant by a newer run.
needs	Execute jobs earlier than the stage ordering.
only	Control when jobs are created.
pages	Upload the result of a job to use with GitLab Pages.
parallel	How many instances of a job should be run in parallel.
release	Instructs the runner to generate a release object.
resource_group	Limit job concurrency.
retry	When and how many times a job can be auto-retried in case of a failure.
rules	List of conditions to evaluate and determine selected attributes of a job, and whether or not it’s created.
script	Shell script that is executed by a runner.
secrets	The CI/CD secrets the job needs.
services	Use Docker services images.
stage	Defines a job stage.
tags	List of tags that are used to select a runner.
timeout	Define a custom job-level timeout that takes precedence over the project-wide setting.
trigger	Defines a downstream pipeline trigger.
variables	Define job variables on a job level.
when	When to run job.