Invoke the CICD pipeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dev | Prod (master) |  | Comment |
|  | Run CICD |  |  |  |
|  |  |  |  |  |
| Setup GIT repo on GIT host if required |  |  | lib\modules\run.js | Filter on commit message 'no-cicd' to avoid the build to trigger. |
| Clone GIT |  |  |  |  |
|  |  | Extract files from prod into master branch (refresh in case changes made on prod without using this pipeline) | lib\modules\export-files-from-master.js |  |
| Push master to GIT |  |  |  | Filter on commit message 'no-cicd' to avoid the build to trigger. |
| Create branch for update set if not already exists |  |  | lib\modules\export-update-set.js |  |
| Refresh update set branch with changes made on master. |  |  | To avoid merge collision later in the process. |
|  | Export update set XML |  |  |
|  | Export update set files |  |  |  |
| Configure Lint / JsDoc / ATF |  |  |  |  |
| Push branch to GIT |  |  |  | This will cause CICD pipeline to start if build on branch is enabled. |
| If CICD\_EMBEDDED\_BUILD is 'true' | | | | |
| Build the project locally |  |  | lib\modules\build-project.js |  |

Build project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dev (Source) | Test (Target) |  | Comment |
| Gulp Init |  |  | lib\project-templates\gulpfile.js | Get build information from server (/build/config) |
| Gulp Lint, Doc |  |  | Lint and jsDoc results are zipped and sent to the CICD server (/build/task) |
| Gulp Test | Run ATF suites and tests |  | lib\project-templates\atf-wrapper.js | Start hidden browser as test runner on CICD server.  Send zipped mocha results to CICD server (/build/task) |
| Gulp complete |  |  | lib\project-templates\gulpfile.js | Send build complete info to server (/build/complete). |
| Build complete |  |  | lib\modules\build-complete.js |  |
| If CICD\_GIT\_PR\_ENABLED is 'true'. | | | | |
| Raise pull request |  |  |  | Raise PR in GIT system |
| Else If deploy target information in place and CD\_CD\_DEPLOY\_ON\_BUILD\_PASS is 'true' | | | | |
|  |  | Complete update set |  | Update set can now be moved |
|  |  | Deploy to target | lib\modules\deploy-update-set.js | Deployment can be done either via update set pull from source or pull from GIT (CICD\_CD\_DEPLOY\_FROM\_GIT) |
| Else notify via Slack build has completed | | | | |

Pull request resolved

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dev (Source) | Test (Target) |  | Comment |
| Pull request completed |  |  | lib\cicd.js | PR information coming from GIT system (/pull\_request) |
| If pull request resolved | | | | |
| If CICD\_GIT\_DELETE\_BRANCH\_ON\_MERGE remove local branch | | | | Delete branch locally and remote |
|  | Complete update set |  |  | Update set can now be moved |
| If CICD\_CD\_DEPLOY\_ON\_PR\_RESOLVE is 'true' | | | | |
|  | Deploy to target |  | lib\modules\deploy-update-set.js | Deployment can be done either via update set pull from source or pull from GIT (CICD\_CD\_DEPLOY\_FROM\_GIT) |
|  |  |  |  |  |
|  |  |  |  |  |

**Trigger Build**

Checkout code from GIT

Run

* npm install
* gulp

This will test the application on the default host (source). If configured it will also raise a pull request against master branch.

**Trigger Tests**

Checkout code from GIT

Run

* npm install
* gulp test --commit-id <commit-id> --on-host <test-host.service-now.com>

--commit-id: the commit ID of the build

--on-host: the host on which the ATF test will run. If CICD\_ATF\_RUN\_ON\_PRODUCTION is 'false' it will not allow to run on the master environment.

**Trigger Deliver**

Run

* npm install
* gulp deploy --commit-id <commit-id> --git --deliver-to <target-host.service-now.com> --deliver-from <source-host. service-now.com>

--commit-id: the commit ID of the build

--git: if exists, update set will be taken and deployed from GIT

--deliver-to: the environment to deliver to

-- deliver-from: the environment from which to deliver. If GIT is enabled, this environment will act as a proxy to connect to GIT.

**Trigger Deploy**

Run

* npm install
* gulp deploy --commit-id <commit-id> --git --deploy-to <target-host.service-now.com> --deploy-from <source-host. service-now.com>

--commit-id: the commit ID of the build

--git: if exists, update set will be taken and deployed from GIT

--deploy-to: the environment to deploy to

--deploy-from: the environment from which to deploy. If GIT is enabled, this environment will act as a proxy to connect to GIT.

Trigger Deploy/Deliver via REST call

Method POST

URL /deploy/us

Header "x-access-token": CICD\_DEPLOY\_ACCESS\_TOKEN, "accept": "application/json"

Body {  
 "commitId": <the commit ID of the build>,  
 "from": <source-host>,  
 "to": <target-host>,  
 "deploy": <true = deploy|false = deliver> [false],  
 "git": <true = via git|false = via source> [false]  
}

Rest client must support long polling and follow the redirects in the response header.