Create a CDK project in Python

Pre-requirements:

* npm install
* npm install -g aws-cdk
* pyenv install <version>

|  |  |
| --- | --- |
| Lib | version |
| node.js | 20.5.1 |
| python | 3.11.5 |
| aws-cdk | 2.126.0 (build fb74c41) |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | Create git repository |  |
|  | Under the desired folder in gitlab, example <https://gitlab.upmc.com/DataEngineering/aws-batch-process>  Click “New project” button |  |
|  | Choose  “Create blank project” |  |
|  | Enter your project name, example  “cdk-batch-zip-job-deployment”  Check: Initialize repository with a README under Project Configuration  Click “Create project” button |  |
| 2 | Clone the project created from git repository  Example:  Git clone https://gitlab.upmc.com/DataEngineering/aws-batch-process/cdk-batch-zip-job-deployment |  |
| 3. | Open the project in Pycharm |  |
|  | Create the following folders  deployments/  src/  tests/ |  |
|  | Add .gitlab-ci.yml or copy from it from an existing project, make sure update it with correct \*\_app.py |  |
|  | Add/create Pipfile |  |
|  | Add .env file, **but not add it to repository** |  |
|  | Add/copy pyproject.toml file, make sure update the following contents:   * [project] name = * [project.urls] “homepage” = |  |
|  | Add.copy .gitignore file |  |
| 4. | Configure virtual environment for your project, example for cdk-batch-zip-\*  Open a Terminal  > pyenv local 3.11.5 |  |
|  | > pyenv exec pip install pipenv --user |  |
|  |  | Install dependencies base on Pipfile  >pyenv exec pipenv install  Or install dependencies from Pipfile.lock  >pyenv exec pipenv sync  To update Lib |
|  | >pyenv exec pipenv install –dev  Or  >pyenv exec pipenv sync –dev  To update lib  > pyenv exec pipenv lock |  |
|  | > pyenv exec pipenv shell  or  ./.venv/Scripts/activate (if $env:PIPENV\_VENV\_IN\_PROJECT=1) |  |
|  | >deactivate or exit  or exit virtual env |  |
| 5. | Update README.md file |  |
|  |  |  |
| 6. | Create python files under src/ |  |
| 7. | Create python test files under test/  Make sure use moto to mock aws s3, etc access |  |
| 8. | Run tests  >pytest --cov=src tests/  >pytest --cov-report html --cov=src tests/ |  |
| 9. | Run analytics  Code Analytic Checks  >mypy src/ --ignore-missing-imports  Code Checks  >pylint src/  >pyflakes src/  >pycodestyle src/ --max-line-length=100  >pipenv check or pyenv exec pipenv check  Code Security Issue Checks  >bandit src/ -r -c ./pyproject.toml  or  >bandit src/ -r |  |
| 10 | Run CDK synth |  |
|  | > $env:OPENSSL\_CONF="C:/Users/brusyj/config/openssl.conf"  Note this is require to access aws through global protect using python 3.11.5+ |  |
|  | > $env:CDK\_DEFAULT\_REGION="us-east-1" |  |
|  | > $env:OPENSSL\_CONF="C:/Users/brusyj/config/openssl.conf"  > $env:CDK\_DEFAULT\_REGION="us-east-1"  Dev  $env:CDK\_DEFAULT\_ACCOUNT="994897293970"  Tst:  > $env:CDK\_DEFAULT\_ACCOUNT="946643926925"  > $env:AWS\_PROFILE="de-prdadmin"  Stage  > $env:CDK\_DEFAULT\_ACCOUNT="138464497820"  > $env:AWS\_PROFILE="dist-stg"  Production  > $env:CDK\_DEFAULT\_ACCOUNT="856866532878"  > $env:AWS\_PROFILE="dist-prd"  > aws sso login |  |
|  | Note 99489466########" is for de-nobaa  > $env:CDK\_DEFAULT\_ACCOUNT="9466########"  $env:CDK\_DEFAULT\_ACCOUNT="9466########"" |  |
|  | > $env:AWS\_PROFILE="de-nonbaa" |  |
|  | Login to Environment you want to working on  >aws sso login |  |
|  | Example:  >cdk --app "python ./src/batch\_zip\_glue\_app.py" synth --require-approval never --context ConfigFile=./deployments/dev-deploy.json --context DataDogKey=######  cdk --app "python ./src/batch\_zip\_glue\_app.py" deploy --require-approval never --context ConfigFile=./deployments/dev-deploy.json --context DataDogKey=###### |  |
|  | Deploy Recipe Stack Example  >  cdk --app "python ./src/databrew\_infrastructure\_app.py" deploy --require-approval never --context ConfigFile=./deployments/dev-deploy.json --context Environment=dev --outputs-file ./cdk.out/recipe-output.json  Publishing Recipes  >  python ./src/publish\_recipes.py -i ./cdk.out/recipe-output.json -r us-east-1  Deploy Job  >  python ./src/create\_job.py -i ./cdk.out/recipe-output.json -c ./deployments/dev-deploy.json |  |

To deploy on gitlab:

> cd C:\app\scripts

C:\app\scripts>$env:AWS\_PROFILE="de-nonbaa"

C:\app\scripts>aws sso login

C:\app\scripts>.\Get-AwsCredentials.ps1

A screenshot of a computer

Description automatically generated