Pipeline structure from Config.yml
===================================

- ORBS recipes for node, eb , aws-cli

```
node: circleci/node@5.0.2
eb: circleci/aws-elastic-beanstalk@2.0.1
aws-cli: circleci/aws-cli@3.1.1
```

- Installing node

```
    - node/install:
        node-version: "14.15"
```

- Seting up aws-cli and eb

```
    - run: node --version
    - checkout
    - aws-cli/setup
    - eb/setup
```

- Installing front end dependencies

```
        name: Install Front-End Dependencies
        command: |
          echo "NODE --version"
          echo $(node --version)
          echo "NPM --version"
          echo $(npm --version)
          npm run frontend:install
```

Installation script: **`"frontend:install": "cd udagram/udagram-frontend && npm install -f"`**

- Installing API dependencies

```
        name: Install API Dependencies
        command: |
          echo "NODE --version"
          echo $(node --version)
          echo "NPM --version"
          echo $(npm --version)
          npm run api:install
```

Installation script: **`"api:install": "cd udagram/udagram-api && npm install ."`**

- Front-end build

```
    name: Front-End Build
    command: |
      npm run frontend:build
```

Build Script:
"frontend:build": "cd udagram/udagram-frontend && npm run build"
Build script from frontend
"build": "ng build"

- API build

```
    name: API Build
    command: |
      npm run api:build
```

Build Script:
"api:build": "cd udagram/udagram-api && npm run build"
Build script from API
"build": "npm install . && npm run clean && tsc && cp -rf src/config www/config   && cp .npmrc www/.npmrc && cp package.json www/package.json && cd www && zip -r Archive.zip . && cd .."

- API deploy

```
- run:
    name: API Deploy
    command: |
      npm run api:deploy
```

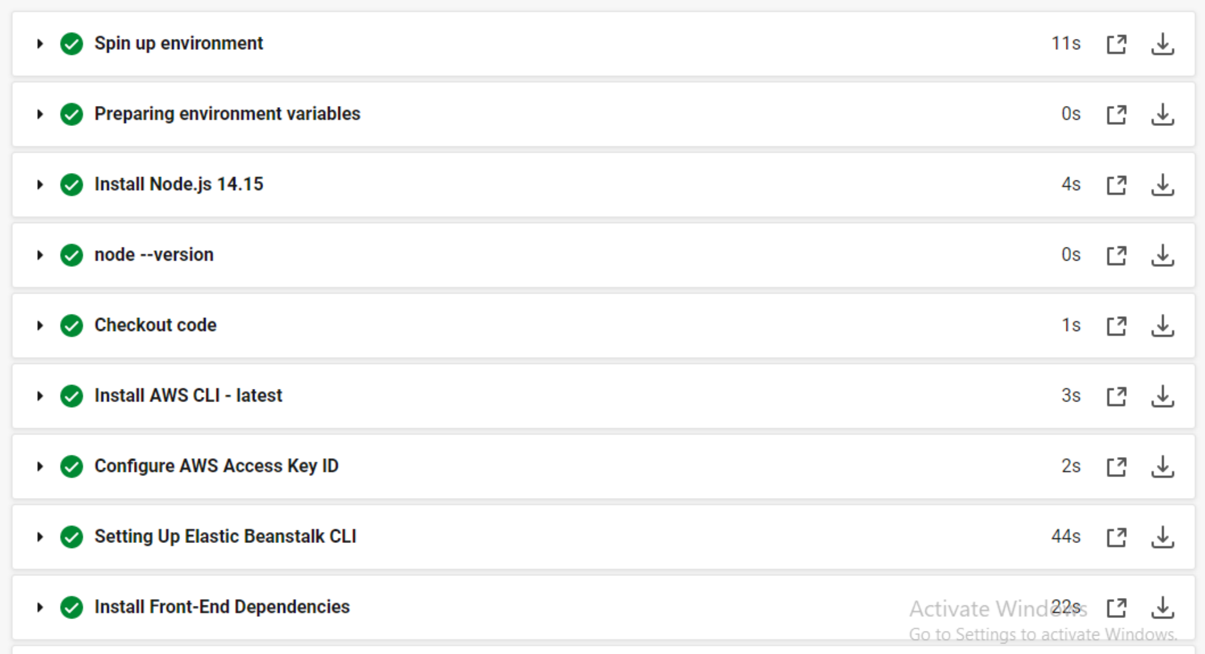
Deploy Script
"api:deploy": "cd udagram/udagram-api && npm run deploy"
Deploy Script from API (includes setting env variables) -> this is in package.json of API
"deploy": "eb use Udagramapi-env && eb deploy && eb setenv AWS_ACCESS_KEY_ID=$AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY=$AWS_SECRET_ACCESS_KEY AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION POSTGRES_USERNAME=$POSTGRES_USERNAME POSTGRES_PASSWORD=$POSTGRES_PASSWORD PORT_DB=$PORT_DB PORT=$PORT POSTGRES_HOST=$POSTGRES_HOST AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION AWS_BUCKET=$AWS_BUCKET URL=$URL JWT_SECRET=$JWT_SECRET POSTGRES_PASSWORD=$POSTGRES_PASSWORD --profile default && cp -R .elasticbeanstalk www/.elasticbeanstalk"

- Front end deployment

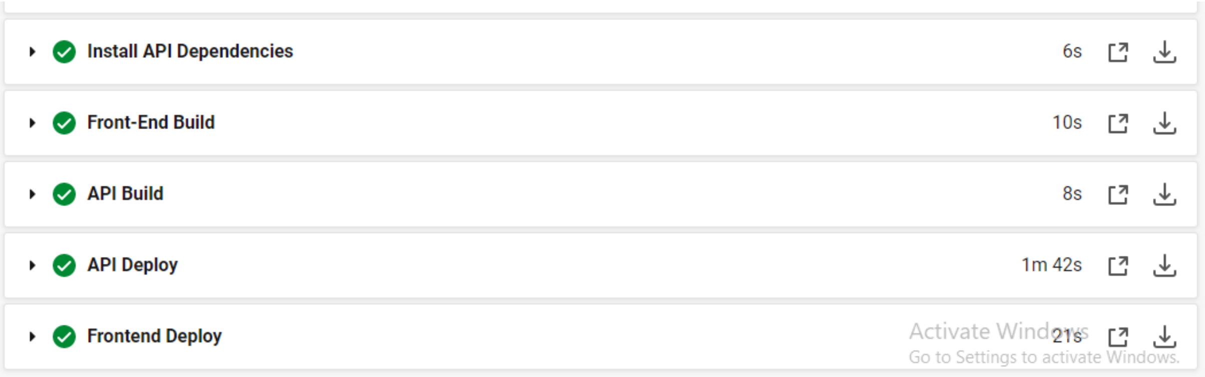
```
      - run:
          name: Frontend Deploy
          command: |
            npm run frontend:deploy
```

Deploy Script
"frontend:deploy": "cd udagram/udagram-frontend && npm run deploy"
Deploy Script from Frontent package.json
"deploy": "npm install -f && npm run build && aws s3 cp --recursive ./www s3://salmazbucket/ "

Circleci Pipline Screenshots

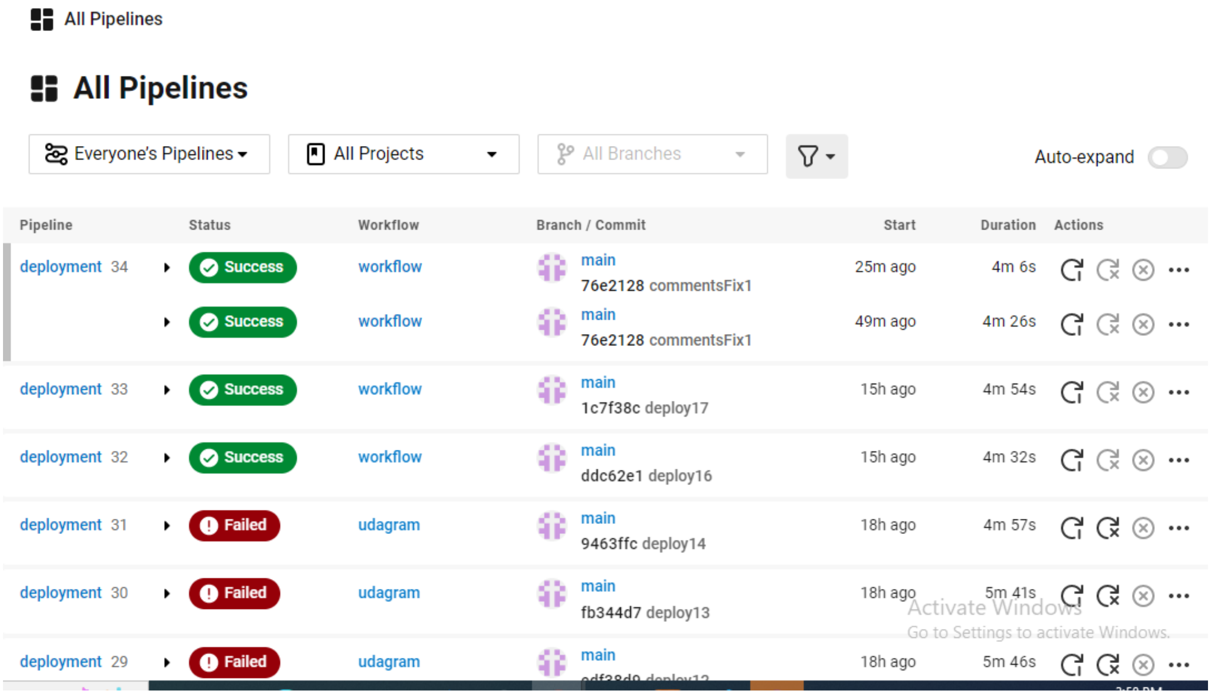


✓	Spin up environment	11s
✓	Preparing environment variables	0s
✓	Install Node.js 14.15	4s
✓	node --version	0s
✓	Checkout code	1s
✓	Install AWS CLI - latest	3s
✓	Configure AWS Access Key ID	2s
✓	Setting Up Elastic Beanstalk CLI	44s
✓	Install Front-End Dependencies	22s



✓	Install API Dependencies	6s
✓	Front-End Build	10s
✓	API Build	8s
✓	API Deploy	1m 42s
✓	Frontend Deploy	21s

Circleci Dashboard



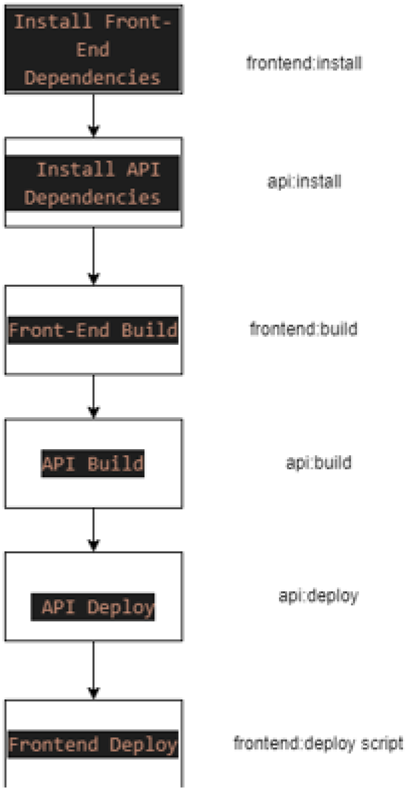
All Pipelines

All Pipelines

Everyone's Pipelines ▾ | All Projects ▾ | All Branches ▾ | ▾ | Auto-expand

Pipeline		Status	Workflow	Branch / Commit	Start	Duration	Actions
deployment 34	▸	✓ Success	workflow	main 76e2128 commentsFix1	25m ago	4m 6s	•••
	▸	✓ Success	workflow	main 76e2128 commentsFix1	49m ago	4m 26s	•••
deployment 33	▸	✓ Success	workflow	main 1c7f38c deploy17	15h ago	4m 54s	•••
deployment 32	▸	✓ Success	workflow	main ddc62e1 deploy16	15h ago	4m 32s	•••
deployment 31	▸	! Failed	udagram	main 9463ffc deploy14	18h ago	4m 57s	•••
deployment 30	▸	! Failed	udagram	main fb344d7 deploy13	18h ago	5m 41s	•••
deployment 29	▸	! Failed	udagram	main edf38d9 deploy12	18h ago	5m 46s	•••

Diagram


Install Front-End Dependencies
frontend:install
Install API Dependencies
api:install
Front-End Build
frontend:build
API Build
api:build
API Deploy
api:deploy
Frontend Deploy
frontend:deploy script