AWS – CodeBuild

By


Keshav Kummari

AWS – CodeBuild

- CodeBuild – CodeBuild is amanaged build service that can compile your source code, run unit tests, and produce deployment artifacts.

- Build and test code with continuous scaling Only pay for the build time you use

- Benefits of CodeBuild:
  - Fully managed by AWS
  - On-demand – AWS scales with your needs
  - Pre-configured environments for many popular programming languages.
- CodeBuild Concepts:
  - Build Project: Defines how CodeBuild will run a build and answers questions like:
    - Where is the source code?
    - What build environment to use?
    - What build commands to run?
    - Where to store the output of the build?

Continue….

- Build Environment – The Operating System language runtime, and tools that CodeBuild uses for the build

- Build Specification – A YAML file that describes the collection of commands and settings for CodeBuild to run a build.
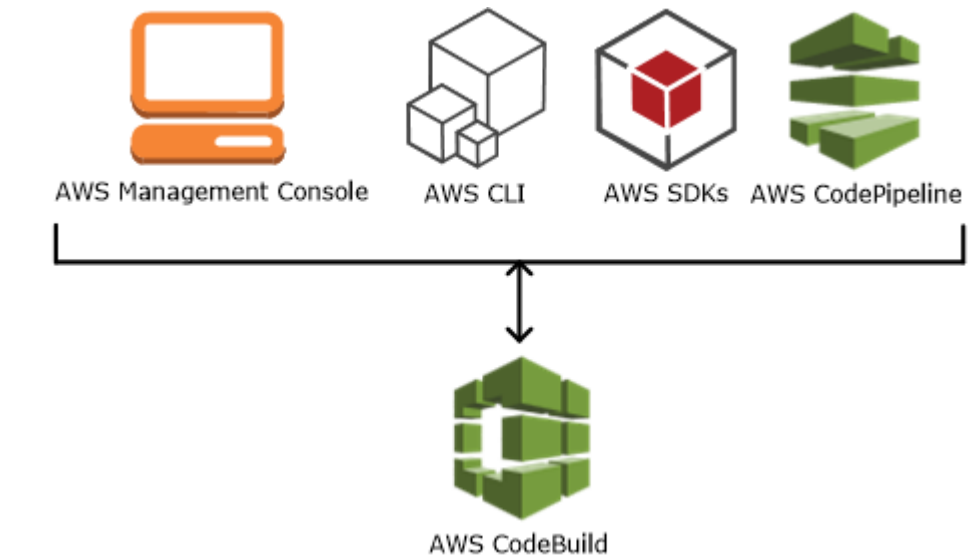
Fill required fields and click on continue

Create project

Configure your project

Specify settings for your build project.

Project name* `pythonProject` ⓘ

Description ⊕ Add description

Source: What to build

Source provider* `GitHub` ▾

Repository* ⦿ Use a public repository
○ Use a repository in my account

Repository URL `https://github.com/devopskeshav/MyFirstWebPrc`

https://github.com/<user-name>/<repository-name>

Git clone depth `1` ▾

Now click on "Start Build"

AWS CodeBuild

Build projects

Build history

Other Resources

Jenkins Integration

Start new build

Choose the build project you want to use. Optionally, you can build a specific version of the source code, and you can override any of the build project's settings for this build only.

Project name* `pythonProject ▾`

Source provider   GitHub

Repository   https://github.com/devopskeshav/MyFirstWebProject.git

⊕ Override source

Source version   `Type your source version`

Git clone depth   `1 ▾`

▸ Override build project settings

▸ Show advanced options

▸ Environment variables

*Required

Cancel   **Start build**

Build is under progress...

Sample Java Build

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: "dockerLoginPassword"

phases:
  install:
    commands:
      - echo Entered the install phase
      - apt-get update -y && apt-get install -y maven
  pre_build:
    commands:
      - echo Entered the pre_build phase
      - docker login -u User -p $LOGIN_PASSWORD
  build:
    commands:
      - echo Build phase started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'
```

Sample Build Spec