An application specification file (AppSpec file), which is unique to CodeDeploy, is a [YAML](http://www.yaml.org/)-formatted or [JSON](http://www.json.org/)-formatted file. The AppSpec file is used to manage each deployment as a series of lifecycle event hooks, which are defined in the file.

If your application uses the EC2/On-Premises compute platform, the AppSpec file is always YAML-formatted. The AppSpec file is used to:

Map the source files in your application revision to their destinations on the instance.

Specify custom permissions for deployed files.

Specify scripts to be run on each instance at various stages of the deployment process.

AppSpec File example for an EC2/On-Premises deployment

version: 0.0

os: linux

files:

- source: Config/config.txt

destination: /webapps/Config

- source: source

destination: /webapps/myApp

hooks:

BeforeInstall:

- location: Scripts/UnzipResourceBundle.sh

- location: Scripts/UnzipDataBundle.sh

AfterInstall:

- location: Scripts/RunResourceTests.sh

timeout: 180

ApplicationStart:

- location: Scripts/RunFunctionalTests.sh

timeout: 3600

ValidateService:

- location: Scripts/MonitorService.sh

timeout: 3600

run as: codedeployuser

# BUILDSPEC FILE: A buildspec is a collection of build commands and related settings, in YAML format, that CodeBuild uses to run a build. Without a build spec, CodeBuild cannot successfully convert your build input into build output or locate the build output artifact in the build environment to upload to your output bucket.

Example for buildspec files:

version: 0.2

env:

variables:

JAVA\_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"

parameter-store:

LOGIN\_PASSWORD: /CodeBuild/dockerLoginPassword

phases:

install:

commands:

- echo Entered the install phase...

- apt-get update -y

- apt-get install -y maven

finally:

- echo This always runs even if the update or install command fails

pre\_build:

commands:

- echo Entered the pre\_build phase...

- docker login -u User -p $LOGIN\_PASSWORD

finally:

- echo This always runs even if the login command fails

build:

commands:

- echo Entered the build phase...

- echo Build started on `date`

- mvn install

finally:

- echo This always runs even if the install command fails

post\_build:

commands:

- echo Entered the post\_build phase...

- echo Build completed on `date`

reports:

arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:

files:

- "\*\*/\*"

base-directory: 'target/tests/reports'

discard-paths: no

reportGroupCucumberJson:

files:

- 'cucumber/target/cucumber-tests.xml'

discard-paths: yes

file-format: CUCUMBERJSON # default is JUNITXML

artifacts:

files:

- target/messageUtil-1.0.jar

discard-paths: yes

secondary-artifacts:

artifact1:

files:

- target/artifact-1.0.jar

discard-paths: yes

artifact2:

files:

- target/artifact-2.0.jar

discard-paths: yes

cache:

paths:

- '/root/.m2/\*\*/\*'