QA Consulting

Docker .dockerignore

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

When we are building Docker Images it's convenient that we can copy files like applications or config by referring to the relative location, this is thanks to how the context works. The docker build command requires a context to build, meaning a folder that it can send to the Docker daemon.

There are going to be times when we don't want every file that is in the context to be sent to the Docker daemon. For instance files that contain credentials and aren't needed in the Docker context, there is no need to copy them anywhere, it can only create a potential security risk. Another, more likely example would be when there are large files in the context that the context that aren't going to be used, sending them to the Docker daemon will cause the image build to take longer.

Dockerignore is a file that can be placed at the root of the context, called **.dockerignore**. The entries listed in this file will be taken into account before the files are sent over. All files in the context that match any of the expressions in the ignore file, will be ignored and not sent to the Docker daemon.

Usage

All entries will need to be put in a file called **.dockerignore** at the root of the context. You can usually ignore the files that you need to by either providing the file itself or by using wildcards.

.dockerignore

```
# this is a dockerignore file
# comments can be made the with conventional '#'

# ignore a sensitive file:
secrets/passwords
# ignore all markdown files:
*.md
# ignore any text files in a sub directory:
*/*.txt
# all markdown files have been ignored,
# but make an exception for the README.md:
!README.md
```

Tasks

We'll be looking at how to utilise a .dockerignore file in this exercise. We can use a Dockerfile to test how the ignore file is behaving and see which files are not being sent to the Docker daemon. Start by making a new folder for this exercise: **~/docker/dockerignore_exercise**

Create the Dockerfile

First we can make the Dockerfile which is going to serve as a way to demonstrate how the ignore file is taking effect.

Dockerfile

```
# build from the latest alpine image
# alpine is a very lightweight distribution of Linux
FROM alpine:latest
# copy everything from the context to the container
COPY . /context
# display everything that has been copied to the container
RUN ls -al /context
```

Create the Project Files

Now we need to make some files that can be copied to the container, they don't have to contain anything, it's just the file names and directories that we are concerned about here.

```
app.py
Dockerfile
docs/
my-notes.md
README.md
```

Build without the .dockerignore File

Build the Docker Image now, the files and folder that you just created should be listed in the output. There are ways that the ignore file could improve our build, for instance the docs folder on and actual folder may contain many images and files that are not necessary for running the application.

Build with the .dockerignore File

Create the .dockerignore file and try building the image again, you should see that the ignored files are not sent to the container.

.dockerignore

```
# ignore the docs folder
docs
# ignore all markdown files
*.md
# in this case, we can make an exception for the README
!README.md
```

Delete all of the Docker Images created during this to clean up.