Dockerfile to Domino

Benjamin Allan-Rahill 7/2/2019

[TOC]

Steps

Tag Image

If you have already done this, proceed. This is to setup the push to our private registry.

Where you are running Docker, run the following command with your specified tag. For this example the tag is **gen201902_rev2** and the repository is **rr**.

```
docker tag rr:gen201903_rev2 docker.rdcloud.bms.com:443/rr:gen201903_rev2
```

Push Image

The next step after tagging is to push the image to our registry.

Use this command with the tag you used above

```
docker push docker.rdcloud.bms.com:443/rr:gen201903_rev2
```

Add To Domino

Navigate to the Environments tab

Click the **Environments** tab in the upper left

Create new environment

Click on the **Create Environment** button

Add Dockerfile Location

Select **Custom Image** and then enter the url we specified above.

For our example:

```
docker.rdcloud.bms.com:443/rr:gen201903_rev2
```

Change Visibility

If you want this image available to a larger group, make sure to select that under visibility

Add Scripts

Edit Dockerfile

Navigate to the Overview tab of your environment and click the **Edit Dockerfile** button

Dockerfile Instructions

In this box add:

```
ENTRYPOINT []
```

Pulgable Workspace Tools

Add:

```
rstudio:
    title: "RStudio"
    iconUrl: "/assets/images/workspace-logos/Rstudio.svg"
    start: [ "/var/opt/workspaces/rstudio/start" ]
    httpProxy:
        port: 8888
jupyterlab:
    title: "JupyterLab"
    start: [  /var/opt/workspaces/Jupyterlab/start.sh ]
    httpProxy:
        internalPath: /{{ownerUsername}}/{{projectName}}/{{sessionPathComponent}}/{{runId}}
        port: 8888
        rewrite: false
jupyter:
  title: "Jupyter (Python, R, Julia)"
  iconUrl: "https://raw.github.com/dominodatalab/workspace-configs/develop/workspace-logos/Jupyter.svg?sanitize=true"
  start: [ "/var/opt/workspaces/jupyter/start" ]
  httpProxy:
    port: 8888
    rewrite: false
    internalPath: "/tree/"
  supportedFileExtensions: [ ".ipynb" ]
```

Run Setup Scripts

Pre Run Script

```
smount -v
if [ -f $DOMINO_WORKING_DIR/.setup/prerun.sh ]
then
    source $DOMINO_WORKING_DIR/.setup/prerun.sh
    echo "Ran project prerun.sh"
else
    echo "No prerun.sh found in " $DOMINO_WORKING_DIR/.setup
fi
```

Post Run Script

```
/usr/local/bin/save-settings

if [ -f $DOMINO_WORKING_DIR/.setup/postrun.sh ]
then
    source $DOMINO_WORKING_DIR/.setup/postrun.sh
    echo "Ran project postrun.sh"
else
    echo "No postrun.sh found in " $DOMINO_WORKING_DIR/.setup
fi
```

Advance Settings

Pre Setup Run Script

```
/usr/local/bin/load-settings
```

Docker Arguments

You have to request to support@dominodatalab.com to add the following docker args to the environment:

```
--cap-add=SYS_ADMIN
--device=/dev/fuse
--security-opt=apparmor:unconfined
--cap-add=DAC_READ_SEARCH
```

Test

Now that you have made the environment, start up a project using it to verify that it is complete.