Buildah

This cheat sheet covers the buildah command set for the buildah tool.

QUERIES

`buildah containers`

list all working containers

`buildah images`

list all local images

`buildah inspect container-id`

displays details about a container

`buildah inspect --type image image-id`

displays details about an image

CREATING CONTAINERS

`buildah from base`

creates a new working container from a base image; the ID of the container is returned from this call

`buildah from scratch`

creates a new scratch image with no base; the name of the container is returned from this call (see "Scratch Containers" below for more information)

EDITING CONTAINERS

`buildah run container-id— commands`

runs the commands after the `--` inside of the specified container (similar in concept to the `RUN` command in a Dockerfile)

`buildah copy container-id local-file destination`

copies a file to the specified destination within the given container

`buildah config flags container-id`

modifies configuration values that will be saved to a container's image when it is created

Some example flags are as follows (the full list of configuration options can be found by running `buildah config -h`):

`--cmd command-string`	sets the default command run for containers created from this image
`--entrypoint entrypoint-string`	sets the entry point for containers created from this image
`--port port`	adds a port to be exposed by this image's containers

`buildah commit container-id image-name`

creates a new image based on the current state of the provided container

DELETING CONTAINERS

`buildah rm container-id`

deletes the given container; multiple container IDs may be specified, separated by spaces

`buildah rm --all`

delete all containers

DELETING IMAGES

`buildah rmi image-id`

deletes the given image

`buildah rmi --all`

deletes all images

`buildah rmi --prune`

prunes dangling images

SCRATCH CONTAINERS

Once a scatch container has been initialized, the commands under "Editing Containers" can also be used to manipulate them. The following commands are used to establish the set of required packages on the scratch image.

`buildah from scratch`

creates a new scratch image with no base; the name of the container is returned from this call

`buildah mount container-id`

mounts a scratch image to a directory on the local system, returning the full path to the directory

`buildah unshare buildah mount container-id`

mounts a scratch image for non-root users

`dnf install --installroot scratch-mount-dir package-list --setopt install_weak_deps=false -y`

example call to the dnf package manager to install packages to the directory to which the scratch container is mounted

`buildah unmount container-id`

unmount a previously mounted container

`buildah commit container-id image-name`

creates a new image based on the current state of the provided container, including the changes made while the container was mounted

USING DOCKERFILES

`buildah bud -t image-tag`

creates an image from a Dockerfile in the current directory (`bud` is short for `build-using-dockerfile` which may also be used)

INTERACTING WITH REGISTRIES

`buildah login` *`registry`*

logs into the specified registry, such as `docker.io` or `quay.io`

`buildah push` *`image-tag`*

pushes the specified image to a remote repository

`buildah tag` *`existing-image-tag new-image-tag`*

create an additional image tag on an existing image