Rocky Enterprise Linux 9.2 Manual Pages on command 'podman-auto-update.1'

$ man podman-auto-update.1

podman-auto-update(1) General Commands Manual podman-auto-update(1)

NAME

podman-auto-update - Auto update containers according to their auto-up?

date policy

SYNOPSIS

podman auto-update [options]

DESCRIPTION

podman auto-update looks up containers with a specified io.contain?

ers.autoupdate label (i.e., the auto-update policy).

If the label is present and set to registry, Podman reaches out to the

corresponding registry to check if the image has been updated. The la?

bel image is an alternative to registry maintained for backwards com?

patibility. An image is considered updated if the digest in the local

storage is different than the one of the remote image. If an image

must be updated, Podman pulls it down and restarts the systemd unit ex?

ecuting the container.

The registry policy requires a fully-qualified image reference (e.g.,

quay.io/podman/stable:latest) to be used to create the container. This

enforcement is necessary to know which image to actually check and pull. If an image ID was used, Podman would not know which image to check/pull anymore.

Alternatively, if the autoupdate label is set to local, Podman will compare the image a container is using to the image with its raw name in local storage. If an image is updated locally, Podman simply restarts the systemd unit executing the container.

If io.containers.autoupdate.authfile label is present, Podman reaches out to the corresponding authfile when pulling images.

At container-creation time, Podman looks up the PODMAN_SYSTEMD_UNIT en? vironment variable and stores it verbatim in the container's label.

This variable is now set by all systemd units generated by podman-gen? erate-systemd and is set to %n (i.e., the name of systemd unit starting the container). This data is then being used in the auto-update se? quence to instruct systemd (via DBUS) to restart the unit and hence to restart the container.

Note that podman auto-update relies on systemd. The systemd units are expected to be generated with podman-generate-systemd --new, or similar units that create new containers in order to run the updated images. Systemd units that start and stop a container cannot run a new image.

Auto Updates and Kubernetes YAML

Podman supports auto updates for Kubernetes workloads. As mentioned above, podman auto-update requires the containers to be running sys? temd. Podman ships with a systemd template that can be instantiated with a Kubernetes YAML file, see podman-generate-systemd(1).

To enable auto updates for containers running in a Kubernetes workload, set the following Podman-specific annotations in the YAML:

* io.containers.autoupdate: "registry|local" to apply the auto-update policy to all containers

* io.containers.autoupdate/$container: "registry|local" to apply the auto-update policy to $container only

* io.containers.sdnotify: "conmon|container" to apply the sdnotify policy to all containers

* io.containers.sdnotify/$container: "conmon|container" to apply the sdnotify policy to $container only

By default, the autoupdate policy is set to "disabled", the sdnotify policy is set to "conmon".

Systemd Unit and Timer

Podman ships with a podman-auto-update.service systemd unit. This unit is triggered daily at midnight by the podman-auto-update.timer systemd timer. The timer can be altered for custom time-based updates if de? sired. The unit can further be invoked by other systemd units (e.g., via the dependency tree) or manually via systemctl start podman-auto-update.service.

OPTIONS

--authfile=path

Path of the authentication file. Default is ${XDG_RUNTIME_DIR}/contain? ers/auth.json, which is set using podman login. If the authorization state is not found there, $HOME/.docker/config.json is checked, which is set using docker login.

Note: There is also the option to override the default path of the au? thentication file by setting the REGISTRY_AUTH_FILE environment vari? able. This can be done with export REGISTRY_AUTH_FILE=path.

--dry-run

Check for the availability of new images but do not perform any pull operation or restart any service or container. The UPDATED field indi? cates the availability of a new image with "pending".

--format=format

Change the default output format. This can be of a supported type like 'json' or a Go template. Valid placeholders for the Go template are listed below:

??

?Placeholder ? Description ?

??

?.Unit ? Name of the systemd unit ?

??

?.ContainerName ? Name of the container ?

??

?.ContainerID ? ID of the container ?

??

?.Container ? ID and name of the container ?

??

?.Image ? Name of the image ?

??

?.Policy ? Auto-update policy of the ?

? ? container ?

??

?.Updated ? Update status: ?

? ? true,false,failed ?

??

--rollback

If restarting a systemd unit after updating the image has failed, roll?

back to using the previous image and restart the unit another time.

Default is true.

Please note that detecting if a systemd unit has failed is best done by

the container sending the READY message via SDNOTIFY. This way,

restarting the unit will wait until having received the message or a

timeout kicked in. Without that, restarting the systemd unit may suc?

ceed even if the container has failed shortly after.

For a container to send the READY message via SDNOTIFY it must be cre?

ated with the --sdnotify=container option (see podman-run(1)). The ap?

plication running inside the container can then execute systemd-notify

--ready when ready or use the sdnotify bindings of the specific pro?

gramming language (e.g., sd_notify(3)).

EXAMPLES

Autoupdate with registry policy

Start a container

$ podman run --label "io.containers.autoupdate=registry"

--label "io.containers.autoupdate.authfile=/some/authfile.json"

-d --name=test registry.fedoraproject.org/fedora:latest sleep infinity

bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d

Generate a systemd unit for this container

$ podman generate systemd --new --files bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d

/home/user/container-bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d.service

Load the new systemd unit and start it

$ mv ./container-bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d.service ~/.config/systemd/user/container-test.service

$ systemctl --user daemon-reload

If the previously created containers or pods are using shared resources, such as ports, make sure to remove them before starting the generated systemd units.

$ podman stop bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d

$ podman rm bc219740a210455fa27deacc96d50a9e20516492f1417507c13ce1533dbdcd9d

$ systemctl --user start container-test.service

Check if a newer image is available

$ podman auto-update --dry-run --format "{{.Image}} {{.Updated}}"

registry.fedoraproject.org/fedora:latest pending

Autoupdate the services

$ podman auto-update

UNIT CONTAINER IMAGE POLICY UPDATED

container-test.service 08fd34e533fd (test) registry.fedoraproject.org/fedora:latest registry false

Autoupdate with local policy

Start a container

$ podman run --label "io.containers.autoupdate=local"

-d busybox:latest top

be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338

Generate a systemd unit for this container

$ podman generate systemd --new --files be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338

/home/user/container-be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338.service

Load the new systemd unit and start it

$ mv ./container-be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338.service

~/.config/systemd/user

$ systemctl --user daemon-reload

If the previously created containers or pods are using shared resources, such as ports, make sure to remove them before starting the generated systemd units.

$ podman stop be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338

$ podman rm be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338

$ systemctl --user start container-be0889fd06f252a2e5141b37072c6bada68563026cb2b2649f53394d87ccc338.service

Get the name of the container

$ podman ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

01f5c8113e84 docker.io/library/busybox:latest top 2 seconds ago Up 3 seconds inspiring_galileo

Modify the image

$ podman commit --change CMD=/bin/bash inspiring_galileo busybox:latest

Auto-update the container

$ podman auto-update

[...]

SEE ALSO

podman(1), podman-generate-systemd(1), podman-run(1), sd_notify(3), systemd.unit(5)

podman-auto-update(1)