Rocky Enterprise Linux 9.2 Manual Pages on command 'buildah-commit.1'

$ man buildah-commit.1

```
buildah-commit(1)          General Commands Manual          buildah-commit(1)

NAME

       buildah-commit - Create an image from a working container.

SYNOPSIS

       buildah commit [options] container [image]

DESCRIPTION

       Writes  a new image using the specified container's read-write layer and if it is based on

       an image, the layers of that image.  If image does not begin with a registry  name  compo?

       nent,  localhost will be added to the name.  If image is not provided, the image will have

       no name.  When an image has no name, the buildah images command will display <none> in the

       REPOSITORY and TAG columns.

RETURN VALUE

       The  image  ID  of  the  image that was created.  On error, 1 is returned and errno is re?

       turned.

OPTIONS

       --authfile path

       Path of the authentication file. Default is  ${XDG_\RUNTIME_DIR}/containers/auth.json.  If

       XDG_RUNTIME_DIR  is  not  set, the default is /run/containers/$UID/auth.json. This file is

       created using using buildah login.

       If the authorization state is not found there, $HOME/.docker/config.json is checked, which

       is set using docker login.

       Note:  You  can  also  override the default path of the authentication file by setting the

       REGISTRY_AUTH_FILE environment variable. export REGISTRY_AUTH_FILE=path
```

--cert-dir path

Use certificates at path (*.crt, *.cert, *.key) to connect to the registry.   The default certificates directory is /etc/containers/certs.d.

--creds creds

The [username[:password]] to use to authenticate with the registry if required. If one or both values are not supplied, a command line prompt will appear and the value can be entered. The password is entered without echo.

--disable-compression, -D

Don't compress filesystem layers when building the image unless it is required by the location where the image is being written. This is the default setting, because image layers are compressed automatically when they are pushed to registries, and images being written to local storage would only need to be decompressed again to be stored.   Compression can be forced in all cases by specifying --disable-compression=false.

--encryption-key key

The [protocol:keyfile] specifies the encryption protocol, which can be JWE (RFC7516), PGP (RFC4880), and PKCS7 (RFC2315) and the key material required for image encryption. For instance, jwe:/path/to/key.pem or pgp:admin@example.com or pkcs7:/path/to/x509-file.

--encrypt-layer layer(s)

Layer(s) to encrypt: 0-indexed layer indices with support for negative indexing (e.g. 0 is the first layer, -1 is the last layer). If not defined, will encrypt all layers if encryption-key flag is specified.

--format, -f [oci | docker]

Control the format for the image manifest and configuration data. Recognized formats include oci (OCI image-spec v1.0, the default) and docker (version 2, using schema format 2 for the manifest).

Note: You can also override the default format by setting the BUILDAH_FORMAT environment variable. export BUILDAH_FORMAT=docker

--iidfile ImageIDfile

Write the image ID to the file.

--manifest "listName"

Name of the manifest list to which the built image will be added. Creates the manifest list if it does not exist. This option is useful for building multi architecture images.

--quiet, -q

When writing the output image, suppress progress output.

--rm Remove the working container and its contents after creating the image. Default leaves the container and its content in place.

--sign-by fingerprint

Sign the new image using the GPG key that matches the specified fingerprint.

--squash

Squash all of the new image's layers (including those inherited from a base image) into a single new layer.

--timestamp seconds

Set the create timestamp to seconds since epoch to allow for deterministic builds (de? faults to current time). By default, the created timestamp is changed and written into the image manifest with every commit, causing the image's sha256 hash to be different even if the sources are exactly the same otherwise. When --timestamp is set, the created time? stamp is always set to the time specified and therefore not changed, allowing the image's sha256 to remain the same. All files committed to the layers of the image will be created with the timestamp.

--tls-verify bool-value

Require HTTPS and verification of certificates when talking to container registries (de? faults to true). TLS verification cannot be used when talking to an insecure registry.

EXAMPLE

This example saves an image based on the container.

```
 buildah commit containerID newImageName
```

This example saves an image named newImageName based on the container.

```
 buildah commit --rm containerID newImageName
```

This example saves an image with no name, removes the working container, and creates a new container using the image's ID.

```
 buildah from $(buildah commit --rm containerID)
```

This example saves an image based on the container disabling compression.

```
 buildah commit --disable-compression containerID
```

This example saves an image named newImageName based on the container disabling compres? sion.

```
 buildah commit --disable-compression containerID newImageName
```

This example commits the container to the image on the local registry while turning off tls verification.

```
 buildah commit --tls-verify=false containerID docker://localhost:5000/imageId
```

This example commits the container to the image on the local registry using credentials and certificates for authentication.

```
 buildah commit --cert-dir  /auth  --tls-verify=true --creds=username:password containerID
docker://localhost:5000/imageId
```

This example commits the container to the image on the local registry using credentials from the /tmp/auths/myauths.json file and certificates for authentication.

```
 buildah commit --authfile /tmp/auths/myauths.json --cert-dir    /auth   --tls-verify=true
--creds=username:password containerID docker://localhost:5000/imageName
```

This example saves an image based on the container, but stores dates based on epoch time.

```
buildah commit --timestamp=0 containerID newImageName
```

Building an multi-architecture image using the --manifest option (requires emulation software)

```
#!/bin/sh
build() {
    ctr=$(./bin/buildah from --arch $1 ubi8)
    ./bin/buildah run $ctr dnf install -y iputils
    ./bin/buildah commit --manifest ubi8ping $ctr
}
build arm
build amd64
build s390x
```

ENVIRONMENT

BUILD_REGISTRY_SOURCES

BUILD_REGISTRY_SOURCES, if set, is treated as a JSON object which contains lists of reg?
istry names under the keys insecureRegistries, blockedRegistries, and allowedRegistries.

When committing an image, if the image is to be given a name, the portion of the name that corresponds to a registry is compared to the items in the blockedRegistries list, and if it matches any of them, the commit attempt is denied. If there are registries in the al?
lowedRegistries list, and the portion of the name that corresponds to the registry is not in the list, the commit attempt is denied.

TMPDIR The TMPDIR environment variable allows the user to specify where temporary files are stored while pulling and pushing images. Defaults to '/var/tmp'.

FILES

registries.conf (/etc/containers/registries.conf)

registries.conf is the configuration file which specifies which container registries should be consulted when completing image names which do not include a registry or domain portion.

policy.json (/etc/containers/policy.json)

Signature policy file. This defines the trust policy for container images. Controls which container registries can be used for image, and whether or not the tool should trust the images.

SEE ALSO

buildah(1), buildah-images(1), containers-policy.json(5), containers-registries.conf(5)

buildah March 2017 buildah-commit(1)