Rocky Enterprise Linux 9.2 Manual Pages on command 'containers-auth.json.5'

$ man containers-auth.json.5

containers-auth.json(5)   File Formats Manual   containers-auth.json(5)

NAME

   containers-auth.json - syntax for the registry authentication file

DESCRIPTION

   A credentials file in JSON format used to authenticate against container image registries.

   The primary (read/write) file is stored at ${XDG_RUNTIME_DIR}/containers/auth.json on

   Linux; on Windows and macOS, at $HOME/.config/containers/auth.json.

   When searching for the credential for a registry, the following files will be read in se?

   quence until the valid credential is found: first reading the primary (read/write) file,

   or the explicit override using an option of the calling application. If credentials are

   not present, search in ${XDG_CONFIG_HOME}/containers/auth.json, $HOME/.docker/con?

   fig.json, $HOME/.dockercfg.

   Except the primary (read/write) file, other files are read-only, unless the user use an

   option of the calling application explicitly points at it as an override.

FORMAT

   The auth.json file stores encrypted authentication information for the user to container

   image registries.   The file can have zero to many entries and is created by a login com?

   mand from a container tool such as podman login, buildah login or skopeo login. Each entry

   either contains a single hostname (e.g. docker.io) or a namespace (e.g. quay.io/user/im?

   age) as a key and an auth token in the form of a base64 encoded string as value of auth.

   The token is built from the concatenation of the username, a colon, and the password. The

   registry name can additionally contain a repository name (an image name without tag or di?

   gest) and namespaces. The path (or namespace) is matched in its hierarchical order when

checking for available authentications. For example, an image pull for my-registry.lo?

cal/namespace/user/image:latest will result in a lookup in auth.json in the following or?

der:

 ? my-registry.local/namespace/user/image

 ? my-registry.local/namespace/user

 ? my-registry.local/namespace

 ? my-registry.local

This way it is possible to setup multiple credentials for a single registry which can be

distinguished by their path.

The following example shows the values found in auth.json after the user logged in to

their accounts on quay.io and docker.io:

```
{
    "auths": {
        "docker.io": {
            "auth": "erfi7sYi89234xJUqaqxgmzcnQ2rRFWM5aJX0EC="
        },
        "quay.io": {
            "auth": "juQAqGmz5eR1ipzx8Evn6KGdw8fEa1w5MWczmgY="
        }
    }
}
```

This example demonstrates how to use multiple paths for a single registry, while preserv?

ing a fallback for my-registry.local:

```
{
    "auths": {
        "my-registry.local/foo/bar/image": {
            "auth": "?"
        },
        "my-registry.local/foo": {
            "auth": "?"
        },
        "my-registry.local": {
            "auth": "?"
```

```
        },

      }

    }
```

An entry can be removed by using a logout command from a container tool such as podman lo?

gout or buildah logout.

In addition, credential helpers can be configured for specific registries and the creden?

tials-helper software can be used to manage the credentials in a more secure way than de?

pending on the base64 encoded authentication provided by login. If the credential helpers

are configured for specific registries, the base64 encoded authentication will not be used

for operations concerning credentials of the specified registries.

When the credential helper is in use on a Linux platform, the auth.json file would contain

keys that specify the registry domain, and values that specify the suffix of the program

to use (i.e. everything after docker-credential-). For example:

```
  {

    "auths": {

      "localhost:5001": {}

    },

    "credHelpers": {

        "registry.example.com": "secretservice"

    }

  }
```

For more information on credential helpers, please reference the GitHub docker-creden?

tial-helpers project ?https://github.com/docker/docker-credential-helpers/releases?.

SEE ALSO

        buildah-login(1), buildah-logout(1), podman-login(1), podman-logout(1), skopeo-login(1), skopeo-logout(1)

HISTORY

        Feb 2020, Originally compiled by Tom Sweeney tsweeney@redhat.com ?mailto:tsweeney@red?

        hat.com?