Chapter 9. Managing a container network

The chapter provides information about how to communicate among containers.

9.1. Listing container networks

In Podman, there are two network behaviors - rootless and rootful:

- Rootless networking - the network is setup automatically, the container does not have an IP address.
- Rootful networking - the container has an IP address.

Prerequisites

- The `container-tools` meta-package is installed.

Procedure

- List all networks as a root user:

  ```
  # podman network ls
  NETWORK ID    NAME        VERSION     PLUGINS
  2f259bab93aa  podman      0.4.0
  bridge,portmap,firewall,tuning
  ```

  - By default, Podman provides a bridged network.
  - List of networks for a rootless user is the same as for a rootful user.

ADDITIONAL RESOURCES

`podman-network-ls` man page

9.2. Inspecting a network

Display the IP range, enabled plugins, type of network, and so on, for a specified network listed by the `podman network ls` command.

Prerequisites

- The `container-tools` meta-package is installed.

Procedure

- Inspect the default `podman` network:

```
$ podman network inspect podman
[
    {
        "cniVersion": "0.4.0",
        "name": "podman",
        "plugins": [
            {
                "bridge": "cni-podman0",
                "hairpinMode": true,
                "ipMasq": true,
                "ipam": {
                    "ranges": [
                        [
                            {
                                "gateway": "10.88.0.1",
                                "subnet": "10.88.0.0/16"
                            }
                        ]
                    ],
                    "routes": [
                        {
                            "dst": "0.0.0.0/0"
                        }
```

You can see the IP range, enabled plugins, type of network, and other network settings.

ADDITIONAL RESOURCES

`podman-network-inspect` man page

9.3. Creating a network

Use the `podman network create` command to create a new network.

> **Note**
>
> By default, Podman creates an external network. You can create an internal network using the `podman network create --internal` command. Containers in an internal network can communicate with other containers on the host, but cannot connect to the network outside of the host nor be reached from it.

Prerequisites

- The `container-tools` meta-package is installed.

Procedure

- Create the external network named `mynet` :

```
# podman network create mynet
/etc/cni/net.d/mynet.conflist
```

Verification

- List all networks:

```
# podman network ls
NETWORK ID    NAME        VERSION     PLUGINS
2f259bab93aa  podman      0.4.0
bridge,portmap,firewall,tuning
11c844f95e28  mynet       0.4.0
bridge,portmap,firewall,tuning,dnsname
```

  You can see the created `mynet` network and default `podman` network.

> **Note**
>
> Beginning with Podman 4.0, the DNS plugin is enabled by default if you create a new external network using the `podman network create` command.

ADDITIONAL RESOURCES

`podman-network-create` man page

9.4. Connecting a container to a network

Use the `podman network connect` command to connect the container to the network.

Prerequisites

- The `container-tools` meta-package is installed.
- A network has been created using the `podman network create` command.
- A container has been created.

Procedure

- Connect a container named `mycontainer` to a network named `mynet` :

  ```
  # podman network connect mynet mycontainer
  ```

Verification

- Verify that the `mycontainer` is connected to the `mynet` network:

  ```
  # podman inspect --format='{{.NetworkSettings.Networks}}'
  mycontainer
  map[podman:0xc00042ab40 mynet:0xc00042ac60]
  ```

  You can see that `mycontainer` is connected to `mynet` and `podman` networks.

ADDITIONAL RESOURCES

`podman-network-connect` man page

9.5. Disconnecting a container from a network

Use the `podman network disconnect` command to disconnect the container from the network.

Prerequisites

- The `container-tools` meta-package is installed.
- A network has been created using the `podman network create` command.
- A container is connected to a network.

Procedure

- Disconnect the container named `mycontainer` from the network named `mynet` :

```
# podman network disconnect mynet mycontainer
```

Verification

- Verify that the `mycontainer` is disconnected from the `mynet` network:

```
# podman inspect --format='{{.NetworkSettings.Networks}}'
mycontainer
map[podman:0xc000537440]
```

You can see that `mycontainer` is disconnected from the `mynet` network, `mycontainer` is only connected to the default `podman` network.

ADDITIONAL RESOURCES

`podman-network-disconnect` man page

9.6. Removing a network

Use the `podman network rm` command to remove a specified network.

Prerequisites

- The `container-tools` meta-package is installed.

Procedure

1. List all networks:

```
# podman network ls
NETWORK ID    NAME        VERSION     PLUGINS
2f259bab93aa  podman      0.4.0
bridge,portmap,firewall,tuning
11c844f95e28  mynet       0.4.0
bridge,portmap,firewall,tuning,dnsname
```

2. Remove the `mynet` network:

```
# podman network rm mynet
mynet
```

Note

> If the removed network has associated containers with it, you have to use the `podman network rm -f` command to delete containers and pods.

Verification

- Check if `mynet` network was removed:

```
# podman network ls
NETWORK ID    NAME        VERSION     PLUGINS
2f259bab93aa  podman      0.4.0
bridge,portmap,firewall,tuning
```

ADDITIONAL RESOURCES

`podman-network-rm` man page

9.7. Removing all unused networks

Use the `podman network prune` to remove all unused networks. An unused network is a network which has no containers connected to it. The `podman network prune` command does not remove the default `podman` network.

Prerequisites

- The `container-tools` meta-package is installed.

Procedure

- Remove all unused networks:

```
# podman network prune
WARNING! This will remove all networks not used by at least one
container.
Are you sure you want to continue? [y/N] y
```

Verification

- Verify that all networks were removed:

```
# podman network ls
NETWORK ID    NAME        VERSION     PLUGINS
2f259bab93aa  podman      0.4.0
bridge,portmap,firewall,tuning
```

ADDITIONAL RESOURCES

`podman-network-prune` man page

PREVIOUS NEXT