Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'podman-generate-kube.1'

$ man podman-generate-kube.1

?% podman?kube?generate(1)

podman?kube?generate ? Generate Kubernetes YAML based on
containers, pods or volumes

podman kube generate [options] container... | pod... | volume...

podman kube generate will generate Kubernetes YAML (v1
specification) from Podman containers, pods or volumes.
Regardless of whether the input is for containers or pods, Podman
will always generate the specification as a Pod. The input may be
in the form of one or more containers, pods or volumes names or
IDs.

Podman Containers or Pods

Volumes appear in the generated YAML according to two different
volume types. Bind?mounted volumes become hostPath volume types
and named volumes become persistentVolumeClaim volume types.
Generated hostPath volume types will be one of three subtypes
depending on the state of the host path: DirectoryOrCreate when
no file or directory exists at the host, Directory when host path

is a directory, or File when host path is a file. The value for claimName for a persistentVolumeClaim is the name of the named volume registered in Podman.

Potential name conflicts between volumes are avoided by using a standard naming scheme for each volume type. The hostPath volume types are named according to the path on the host machine, replacing forward slashes with hyphens less any leading and trailing forward slashes. The special case of the filesystem root, /, translates to the name root. Additionally, the name is suffixed with ?host to avoid naming conflicts with persistentVolumeClaim volumes. Each persistentVolumeClaim volume type uses the name of its associated named volume suffixed with ?pvc.

Note that if an init container is created with type once and the pod has been started, the init container will not show up in the generated kube YAML as once type init containers are deleted after they are run. If the pod has only been created and not started, it will be in the generated kube YAML. Init containers created with type always will always be generated in the kube YAML as they are never deleted, even after running to completion.

Note: When using volumes and generating a Kubernetes YAML for an unprivileged and rootless podman container on an SELinux enabled system, one of the following options must be completed:

 * Add the "privileged: true" option to the pod spec

 * Add type: spc_t under the securityContext seLinuxOptions in the pod spec

 * Relabel the volume via the CLI command chcon ?t container_file_t ?R <directory>

Once completed, the correct permissions will be in place to access the volume when the pod/container is created in a Kubernetes cluster.

Note that the generated Kubernetes YAML file can be used to re?run the deployment via podman?play?kube(1).

Output to the given file, instead of STDOUT. If the file already

exists, kube generate will refuse to replace it and return an

error.

Generate a Kubernetes service object in addition to the Pods.

Used to generate a Service specification for the corresponding

Pod output. In particular, if the object has portmap bindings,

the service specification will include a NodePort declaration to

expose the service. A random port is assigned by Podman in the

specification.

Create Kubernetes Pod YAML for a container called some?mariadb.

$ sudo podman kube generate some?mariadb

Save the output of this file and use kubectl create ?f to import

it into Kubernetes.

#

Created with podman?0.11.2?dev

apiVersion: v1

kind: Pod

metadata:

  creationTimestamp: 2018?12?03T19:07:59Z

  labels:

    app: some?mariadb

  name: some?mariadb?libpod

spec:

  containers:

  ? command:

   ? docker?entrypoint.sh

   ? mysqld

   env:

   ? name: HOSTNAME

   ? name: GOSU_VERSION

    value: "1.10"

   ? name: GPG_KEYS

    value: "199369E5404BD5FC7D2FE43BCBCB082A1BB943DB \t177F4010FE56CA3336300305F1656F24C74CD1D8

```
                \t430BDF5C56E7C94E848EE60C1C4CBDCDCD2EFD2A \t4D1BB29D63D98E422B2113B19334A25F8507EFA5"
    ? name: MARIADB_MAJOR
      value: "10.3"
    ? name: MARIADB_VERSION
      value: 1:10.3.10+maria~bionic
    ? name: MYSQL_ROOT_PASSWORD
      value: x
    image: quay.io/baude/demodb:latest
    name: some?mariadb
    ports:
    ? containerPort: 3306
      hostPort: 36533
    tty: true
```

Create Kubernetes Pod YAML for a container with the directory /home/user/my?data on the host bind?mounted in the container to /volume.

```
$ podman kube generate my?container?with?bind?mounted?data
# Save the output of this file and use kubectl create ?f to import
# it into Kubernetes.
#
# Created with podman?3.1.0?dev
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2021?03?18T16:26:08Z"
  labels:
    app: my?container?with?bind?mounted?data
  name: my?container?with?bind?mounted?data
spec:
  containers:
  ? command:
    ? /bin/sh
    image: docker.io/library/alpine:latest
```

```yaml
    name: test?bind?mount
    volumeMounts:
    ? mountPath: /volume
      name: home?user?my?data?host
  restartPolicy: Never
  volumes:
  ? hostPath:
      path: /home/user/my?data
      type: Directory
    name: home?user?my?data?host
```

Create Kubernetes Pod YAML for a container with the named volume

priceless?data mounted in the container at /volume.

```
$ podman kube generate my?container?using?priceless?data
# Save the output of this file and use kubectl create ?f to import
# it into Kubernetes.
#
# Created with podman?3.1.0?dev
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2021?03?18T16:26:08Z"
  labels:
    app: my?container?using?priceless?data
  name: my?container?using?priceless?data
spec:
  containers:
  ? command:
    ? /bin/sh
    image: docker.io/library/alpine:latest
    name: test?bind?mount
    volumeMounts:
    ? mountPath: /volume
      name: priceless?data?pvc
```

```yaml
    restartPolicy: Never
  volumes:
  ? name: priceless?data?pvc
    persistentVolumeClaim:
      claimName: priceless?data
```

Create Kubernetes Pod YAML for a pod called demoweb and include a service.

```
$ sudo podman kube generate ?s demoweb
# Save the output of this file and use kubectl create ?f to import
# it into Kubernetes.
#
# Created with podman?0.12.2?dev
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: 2018?12?18T15:16:06Z
  labels:
    app: demoweb
  name: demoweb?libpod
spec:
  containers:
  ? command:
    ? python3
    ? /root/code/graph.py
    image: quay.io/baude/demoweb:latest
    name: practicalarchimedes
    tty: true
    workingDir: /root/code
???
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: 2018?12?18T15:16:06Z
```

```
    labels:
      app: demoweb
    name: demoweb?libpod
  spec:
    ports:
    ? name: "8050"
      nodePort: 31269
      port: 8050
      targetPort: 0
    selector:
      app: demoweb
    type: NodePort
  status:
    loadBalancer: {}
```

podman(1), podman?container(1), podman?pod(1), podman?kube?play(1), podman?kube?down(1)

December 2018, Originally compiled by Brent Baude (bbaude at redhat dot com)