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-kube-apply.1'

$ man podman-kube-apply.1

?% podman?kube?apply(1)

podman?kube?apply ? Apply Kubernetes YAML based on containers,

pods, or volumes to a Kubernetes cluster

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

podman kube apply will deploy a podman container, pod, or volume

to a Kubernetes cluster. Use the ??file flag to deploy a

Kubernetes YAML (v1 specification) to a kubernetes cluster as

well.

Note that the Kubernetes YAML file can be used to run the

deployment in Podman via podman?play?kube(1).

The path to the CA cert file for the Kubernetes cluster. Usually

the kubeconfig has the CA cert file data and generate kube

automatically picks that up if it is available in the kubeconfig.

If no CA cert file data is available, set this to insecure to

bypass the certificate verification.

Path to the kubernetes yaml file to deploy onto the kubernetes

cluster. This file can be generated using the podman kube generate command. The input may be in the form of a yaml file, or stdin. For stdin, use ??file=?.

Path to the kubeconfig file to be used when deploying the generated kube yaml to the Kubernetes cluster. The environment variable KUBECONFIG can be used to set the path for the kubeconfig file as well. Note: A kubeconfig can have multiple cluster configurations, but kube generate will always only pick the first cluster configuration in the given kubeconfig.

The namespace or project to deploy the workloads of the generated kube yaml to in the Kubernetes cluster.

Used to create a service for the corresponding container or pod being deployed to the cluster. In particular, if the container or pod has portmap bindings, the service specification will include a NodePort declaration to expose the service. A random port is assigned by Podman in the service specification that is deployed to the cluster.

Apply a podman volume and container to the "default" namespace in a Kubernetes cluster.

```
$ podman kube apply ??kubeconfig /tmp/kubeconfig myvol vol?test?1
Deploying to cluster...
Successfully deployed workloads to cluster!
$ kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
vol?test?1?pod   1/1     Running   0          9m
```

Apply a Kubernetes YAML file to the "default" namespace in a Kubernetes cluster.

```
$ podman kube apply ??kubeconfig /tmp/kubeconfig ?f vol.yaml
Deploying to cluster...
Successfully deployed workloads to cluster!
$ kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
vol?test?2?pod   1/1     Running   0          9m
```

Apply a Kubernetes YAML file to the "test1" namespace in a

Kubernetes cluster.

```
$ podman kube apply ??kubeconfig /tmp/kubeconfig ??ns test1 vol?test?3
Deploying to cluster...
Successfully deployed workloads to cluster!
$ kubectl get pods ??namespace test1
NAME            READY   STATUS    RESTARTS   AGE
vol?test?3?pod   1/1     Running   0          9m
```

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

September 2022, Originally compiled by Urvashi Mohnani (umohnani at redhat dot com)