Pod Scenarios using Krkn-hub

This scenario disrupts the pods matching the label in the specified namespace on a Kubernetes/OpenShift cluster.

Run

If enabling Cerberus to monitor the cluster and pass/fail the scenario post chaos, refer docs. Make sure to start it before injecting the chaos and set CERBERUS_ENABLED environment variable for the chaos injection container to autoconnect.

```
Shell
$ podman run --name=<container_name> --net=host --env-host=true -v
<path-to-kube-config>:/home/krkn/.kube/config:Z -d
containers.krkn-chaos.dev/krkn-chaos/krkn-hub:pod-scenarios
$ podman logs -f <container_name or container_id> # Streams Kraken logs
$ podman inspect <container-name or container-id> --format "{{.State.ExitCode}}" #
Outputs exit code which can considered as pass/fail for the scenario
```

Copy

Note

–env-host: This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2) machines. Without the –env-host option you'll have to set each enviornment variable on the podman command line like -e <VARIABLE>=<value>

```
Shell
$ docker run $(./get_docker_params.sh) --name=<container_name> --net=host -v
<path-to-kube-config>:/home/krkn/.kube/config:Z -d
containers.krkn-chaos.dev/krkn-chaos/krkn-hub:pod-scenarios
OR
$ docker run -e <VARIABLE>=<value> --name=<container_name> --net=host -v
<path-to-kube-config>:/home/krkn/.kube/config:Z -d
containers.krkn-chaos.dev/krkn-chaos/krkn-hub:pod-scenarios

$ docker logs -f <container_name or container_id> # Streams Kraken logs
$ docker inspect <container-name or container-id> --format "{{.State.ExitCode}}" #
Outputs exit code which can considered as pass/fail for the scenario
```

Copy

Tip

Because the container runs with a non-root user, ensure the kube config is globally readable before mounting it in the container. You can achieve this with the following commands: kubectl config view --flatten > ~/kubeconfig && chmod 444 ~/kubeconfig && docker run $(./get_docker_params.sh) --name=<container_name> --net=host -v ~kubeconfig:/home/krkn/.kube/config:Z -d containers.krkn-chaos.dev/krkn-chaos/krkn-hub:<scenario>

Supported parameters

The following environment variables can be set on the host running the container to tweak the scenario/faults being injected:
Example if –env-host is used:

```
None
export <parameter_name>=<value>
```

OR on the command line like example:

```
None
-e <VARIABLE>=<value>
```

See list of variables that apply to all scenarios here that can be used/set in addition to these scenario specific variables

Parameter	Description	Default
NAMESPACE	Targeted namespace in the cluster (supports regex)	openshift-.*
POD_LABEL	Label of the pod(s) to target	""
NAME_PATTERN	Regex pattern to match the pods in NAMESPACE when POD_LABEL is not specified	.*

DISRUPTION_COUNT	Number of pods to disrupt	1
KILL_TIMEOUT	Timeout to wait for the target pod(s) to be removed in seconds	180
EXPECTED_RECOVERY_TIME	Fails if the pod disrupted do not recover within the timeout set	120

Note

Set NAMESPACE environment variable to openshift-.* to pick and disrupt pods randomly in openshift system namespaces, the DAEMON_MODE can also be enabled to disrupt the pods every x seconds in the background to check the reliability.

Note

In case of using custom metrics profile or alerts profile when CAPTURE_METRICS or ENABLE_ALERTS is enabled, mount the metrics profile from the host on which the container is run using podman/docker under /home/krkn/kraken/config/metrics-aggregated.yaml and /home/krkn/kraken/config/alerts.
For example:

```
Shell
$ podman run --name=<container_name> --net=host --env-host=true -v
<path-to-custom-metrics-profile>:/home/krkn/kraken/config/metrics-aggregated.yaml -v
<path-to-custom-alerts-profile>:/home/krkn/kraken/config/alerts -v
<path-to-kube-config>:/home/krkn/.kube/config:Z -d
containers.krkn-chaos.dev/krkn-chaos/krkn-hub:container-scenarios
```

Copy