"container kill" refers to deliberately terminating or killing one or more containers within a Kubernetes cluster as part of an experiment designed to test system resilience. This action simulates scenarios where containers fail unexpectedly due to various factors such as hardware failures, software bugs, or resource constraints.

kubectl get deploy -o wide

|  |  |  |
| --- | --- | --- |
| **Variables** | **Description** | **Notes** |
| TARGET\_CONTAINER | The name of container to be killed inside the pod | If the TARGET\_CONTAINER is not provided it will delete the first container |
| CHAOS\_INTERVAL | Time interval b/w two successive container kill (in sec) | If the CHAOS\_INTERVAL is not provided it will take the default value of 10s |
| TOTAL\_CHAOS\_DURATION | The time duration for chaos injection (seconds) | Defaults to 20s |
| PODS\_AFFECTED\_PERC | The Percentage of total pods to target | Defaults to 0 (corresponds to 1 replica), provide numeric value only |
| TARGET\_PODS | Comma separated list of application pod name subjected to container kill chaos | If not provided, it will select target pods randomly based on provided appLabels |
| LIB\_IMAGE | LIB Image used to kill the container | Defaults to litmuschaos/go-runner:latest |
| LIB | The category of lib use to inject chaos | Default value: litmus, supported values: pumba and litmus |
| RAMP\_TIME | Period to wait before injection of chaos in sec |  |
| SEQUENCE | It defines sequence of chaos execution for multiple target pods | Default value: parallel. Supported: serial, parallel |
| SIGNAL | It contains termination signal used for container kill | Default value: SIGKILL |
| SOCKET\_PATH | Path of the containerd/crio/docker socket file | Defaults to /run/containerd/containerd.sock |
| CONTAINER\_RUNTIME | container runtime interface for the cluster | Defaults to containerd, supported values: docker, containerd and crio for litmus and only docker for pumba LIB |

Container run time links to third party projects that provide functionality required by Kubernetes. The Kubernetes project authors aren't responsible for these projects, which are listed alphabetically.