In chaos engineering, a "memory hog" refers to intentionally creating a scenario where an application or process within a system consumes an excessive amount of memory, leading to resource exhaustion or performance degradation. This action is meant to simulate real-world scenarios where memory-intensive applications or memory leaks cause unexpected spikes in memory usage.

kubectl top pod -n litmus

|  |  |  |
| --- | --- | --- |
| MEMORY\_CONSUMPTION | The amount of memory used of hogging a Kubernetes pod (megabytes) | Defaults to 500MB |
| NUMBER\_OF\_WORKERS | The number of workers used to run the stress process | Defaults to 1 |
| TOTAL\_CHAOS\_DURATION | The time duration for chaos insertion (seconds) | Defaults to 60s |
| LIB | The chaos lib used to inject the chaos. Available libs are litmus and pumba | Defaults to litmus |
| LIB\_IMAGE | Image used to run the helper pod. | Defaults to litmuschaos/go-runner:1.13.8 |
| STRESS\_IMAGE | Container run on the node at runtime by the pumba lib to inject stressors. Only used in LIB pumba | Default to alexeiled/stress-ng:latest-ubuntu |
| TARGET\_PODS | Comma separated list of application pod name subjected to pod memory hog chaos | If not provided, it will select target pods randomly based on provided appLabels |
| TARGET\_CONTAINER | Name of the target container under chaos. | If not provided, it will select the first container of the target pod |
| 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 |
| SOCKET\_PATH | Path of the containerd/crio/docker socket file | Defaults to /run/containerd/containerd.sock |
| PODS\_AFFECTED\_PERC | The Percentage of total pods to target | Defaults to 0 (corresponds to 1 replica), provide numeric value only |
| 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 |