My application is resilient to admin-instigated node drainage

18 July 2018

Abstract

Can my application maintain its minimum resources?

Contents

Summary

Status	completed
Tagged	service, kubernetes
Executed From	Bertrand.local
Platform	Darwin-17.6.0-x86_64-i386-64bit
Started	Wed, 18 Jul 2018 15:25:28 GMT
Completed	Wed, 18 Jul 2018 15:25:29 GMT
Duration	1 second

Experiment

The experiment was made of 1 actions, to vary conditions in your system, and 0 probes, to collect objective data from your system during the experiment.

Steady State Hypothesis

The steady state hypothesis this experiment tried was "**Services are all available and healthy**".

Before Run

The steady state was verified

Probe	Tolerance	Verified
application-must-respond-normally	200	True
pods_in_phase	True	True

After Run

The steady state was verified

Probe	Tolerance	Verified
application-must-respond-normally	200	True
pods_in_phase	True	True

Method

The experiment method defines the sequence of activities that help gathering evidence towards, or against, the hypothesis.

The following activities were conducted as part of the experimental's method:

Type	Name
action	drain_node

Result

The experiment was conducted on Wed, 18 Jul 2018 15:25:28 GMT and lasted roughly 1 second.

Action - drain_node

Status	failed
Background	False

Started	Wed, 18 Jul 2018 15:25:28 GMT
Ended	Wed, 18 Jul 2018 15:25:29 GMT
Duration	1 second

The action provider that was executed:

Type	python
Module	chaosk8s.node.actions
Function	drain_nodes
Arguments	{'name': 'gke-disruption-demo-default-pool-9fa7a856-jrvm', 'delete_pods_with_local_storage': True}

The *drain_node* action raised the following error while running:

Traceback (most recent call last):

Appendix

Action - drain_node

The *action* returned the following result:

```
None
```