2019-09-19 - New K8s workers unable to join cluster

Authors

- Sebastian Herzberg
- Clifford Sanders

Status

- Incident resolved with a workaround
- Incident review completed

Summary

Kubernetes worker autoscaling groups in all environments utilize spot instances. The termination of nodes is expected at any given moment and the cluster is able to cope with it, as long as eventually? enough capacity is available. At an unknown point (roughly 2-3 days before this incident) the CentOS package **container-selinux-2.68-1.el7.noarch.rpm** was moved inside the repository mirror?it is installed from. The bootstrap process of a new worker nodes however had the URL to this package hardcoded. As it could not be downloaded and installed, the bootstrap process never finished.?This lead to clusters with insufficient capacity for pods.?

Impact

- All Kubernetes clusters were impacted by the threat of insufficient capacity since the restructuring of the CentOS mirror.
- Pods on clusters with insufficient capacity were not scheduled and stuck in the Pending state
- Developers could not deploy to clusters with insufficient capacity

Largest contributing factor

- Nodeup can't find container-selinux-2.68-1.el7.noarch.rpm when trying to bootstrap a new node to a cluster

Trigger

- CentOS repositories were restructured which made the static download URL for container-selinux package invalid.
- Kops had this URL hardcoded in nodeup and depended on it to run docker-ce

Resolution

- Changing the AMI of the Kubernetes clusters from Amazon Linux 2 to Debian

Detection

- Developers asking in #ask-platform Slack channel for help because their deployments were failing

Action Items

☐ Create an alert that can indicate that nodes can not join the cluster. (Maybe?the cluster-autoscaler has this kind of information.)

☐ Document external dependencies for node startup - think about ways how to reduce them.?

☐ Change AMI back to Amazon Linux 2 Minimal when kops resolves the issue.

☐ Document cluster node troubleshooting better. (What happens during bootstrap and where are the logs?)

Lessons Learned

What went well

- Changing and applying cluster changes worked beautifully with kops + Jenkins

- Troubleshooting quickly revealed that failing deployments are caused by insufficient cluster capacity

What went wrong

- It was not noticed that there are clusters where the number of worker nodes in the ASGs is not equal to the number worker nodes in K8s over a longer period

Where we got lucky

- No large amount of spot instances died on the production system during the issue
- We noticed during office hours

Timeline

All times CEST.

2019-09-19

Time	Description
16:00	Report that deployment is failing on Prelive in #ask-platform
16:06	Another report of failing deployment on Prelive in #ask-platform
16:10	Investigation shows that containers can not be scheduled because there is no capacity
16:15	Autoscaling Group of Prelive has 4 pretty new instances that did not join the cluster. SSHing into the node reveals that container-selinux package can not be installed due to checksum mismatch.
16:20	It is clear that this is https://github.com/kubernetes/kops/issues/7608. Cluster capacity can disappear in all clusters at any given moment and not be replaced.
16:35	Discussion how to work around the problem, as there was no fixed version of kops. Changing the AMI is an unknown path. Loading the correct package during startup from an S3 bucket is selected as a safer option. Downloading the package manually on Prelive nodes lets the bootstrap process finish and the nodes join the cluster.?
16:40	Informing the On-Call Team that at any given moment the production system might get significantly impaired.
16:55	To avoid failing deploys due to capacity problems developers are asked to not deploy to production.
17:21	• Uploaded the package to a public S3 bucket • Added a startup hook to the kops cluster configuration to download the package to the correct location during bootstrap • Committed the changes to the k8s repo • Jenkins runs kops update to apply the changes to Prelive
17:35	Workaround does not work on Prelive. Problems with the syntax of the systemd unit.
18:12	• Build with fixed syntax succeeded • Manually trying the workaround on Prelive
18:20	Realizing that the workaround will not work at all, as the startup hooks are executed too late.
18:39	Approaching the second workaround option: Changing the cluster AMI to Debian instead of Amazon Linux 2 (CentOS)
18:51	Merging the AMI changes to master - applying to Prelive.
18:55	Trying out the AMI change on Prelive reveals that it works fine. New nodes are able to join the cluster.
19:00	Applying the change to all clusters.
19:08	Declaring the incident as resolved.

Supporting Information

Links to logs, dashboards or conversations go here.