Open in app

Dennis Zielke

Following   301 Followers   About

Dashboard and notifications on AKS for required worker nodes reboots

tl;dr: overview on required reboots of your worker nodes, scheduling restarts automatically and getting dashboards and notifications

Dennis Zielke · Apr 23, 2019 · 4 min read

As mentioned in the official documentation your AKS worker nodes will be receiving automatic unattended upgrades as configured in /etc/apt/apt.conf.d/50unattended-upgrades on each worker node.

Most of them will be installed automatically and can be applied without interruption — but some these updates require you to reboot the node to complete. As of today azure will NOT automatically reboot your machine. I want to use that as an example for a regular maintenance task (there might be other things you want done to your nodes) that you can execute on your cluster while also getting dashboards for overview and notifications to your operators.

That obviously leaves you with the problem of figuring out when a reboot needs to happen on which machines and how to execute the reboot in a production environment. Would it not be nice to have azure look after this and send you notifications for this?
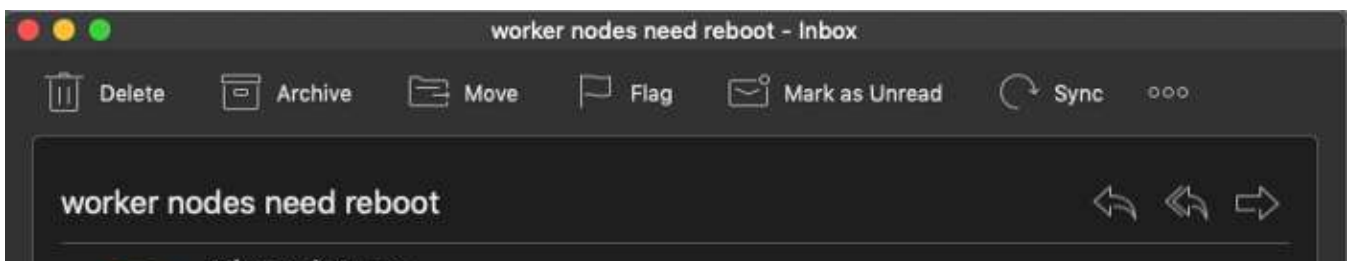
Open in app

Email notification send to inform you about nodes that need a reboot

The reboot requirement is detectable by looking for the existence of the file */var/run/reboot-required* on each host. There is the manual way of doing it for example by deploying a DaemonSet into your cluster that will mount the file give and use a script to give you an indication of its existence on each host by using standard Kubernetes features:

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: check-reboot
  labels:
    app: check-reboot
spec:
  template:
    metadata:
      labels:
        name: check-reboot
    spec:
      containers:
        - name: check-reboot
          securityContext:
            privileged: true
          image: busybox
          imagePullPolicy: Always
          env:
            - name: MY_NODE_NAME
              valueFrom:
                fieldRef:
                  fieldPath: spec.nodeName
          command: [ "sh", "-c"]
          args:
          - while true; do
            [ -f /host/reboot/reboot-required ] && echo "check-
```

Open in app ●●|

```
                                 done;
                          resources:
                            requests:
                                cpu: 5m
                                memory: 50Mi
                          volumeMounts:
                            - name: host-reboot
                              mountPath: /host/reboot
                              readOnly: true
                  volumes:
                  - name: host-reboot
                    hostPath:
                          path: /var/run/
                          type: Directory
```
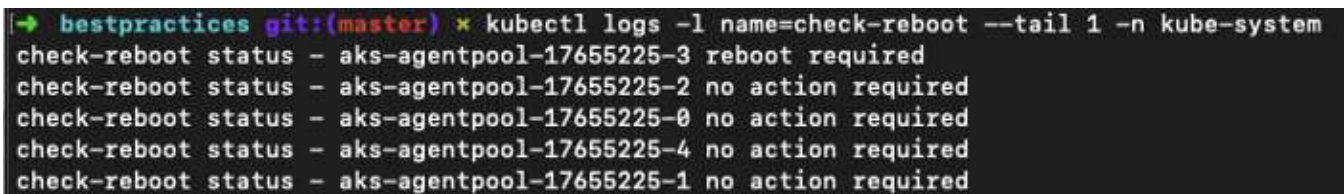
In this case we make sure that we mount the directory with the lock file and also make sure that the host name is also available as an environment variable inside the container. You can simply take the file from my github:

```
kubectl apply -f
https://raw.githubusercontent.com/denniszielke/container_demos/maste
r/bestpractices/checkreboot.yaml

kubectl logs -l name=check-reboot --tail 1
```


Checking for pending reboot flags manually might not be the best solution but it works

While that works (maybe you will find a nice way of automating this) it will it be hard to execute this manually at scale. At the end most companies want to make a conscious decision on when to pull the plug on a node so therefore I first want to address the issue of getting notified of pending reboots.

Using azure monitor to get a dashboard

If you are using the azure monitor for containers in your AKS cluster you can use that service to create a query that will return all clusters and worker nodes that have a pending reboot by using the log message from the DaemonSet.

Open in app

```
`check-reboot` | distinct ContainerID;

ContainerLog
| where TimeGenerated > ago(1h) | where ContainerID in
(ContainerIdList) | project LogEntry, TimeGenerated, Computer |
where LogEntry startswith "check-reboot status" and LogEntry
contains "reboot required" | join kind= innerunique
(KubeNodeInventory | where TimeGenerated > ago(1h) |  distinct
Computer, ClusterName ) on Computer | project ClusterName, Computer
| render table
```


You will get the name of the cluster and the nodes that have a pending reboot.

The next step is to turn this into a dashboard that is visible for our operations team by clicking on "Pin" and adding it to our shared operations dashboard. Lets also create a query that will show if any nodes of any of our clusters are not ready and also pin it to our dashboard.

```
KubeNodeInventory | where TimeGenerated > ago (6h) | summarize
dcountif(Computer, Status!="Ready") by bin(TimeGenerated, 30s),
ClusterName | render timechart
```
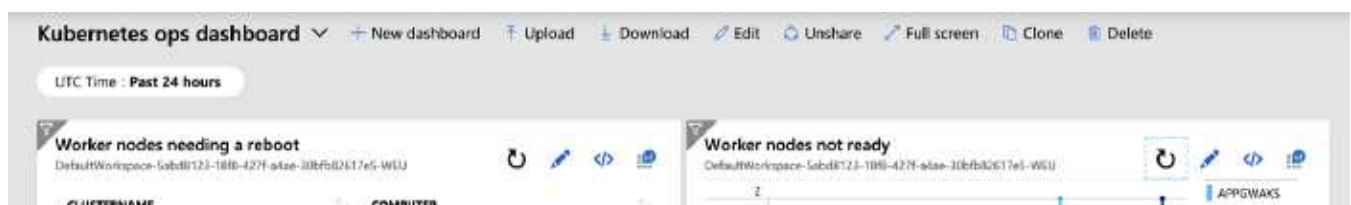
Open in app
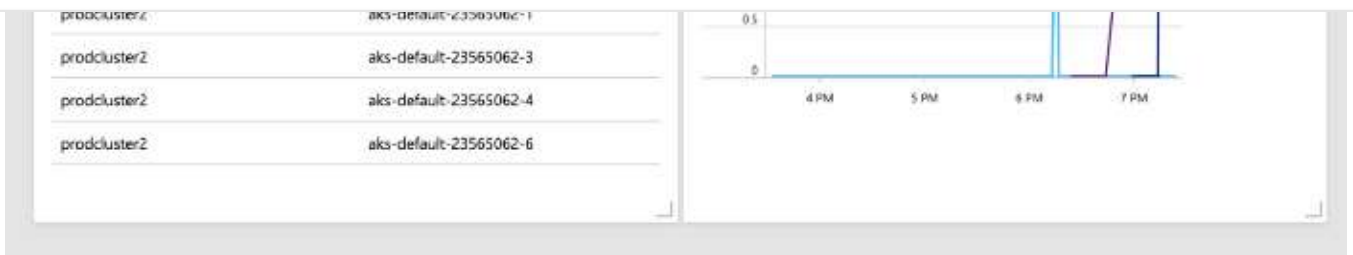
Dashboard will show us all nodes needing a reboot and show also the nodes rebooting by kured

Getting notifications for pending reboots

Go into your log workspace and create an alert rule by using the query from above as a condition. In our case we want the number nodes requiring a reboot to be zero that will be checked on a regular basis.

In this case we use an email notification but SMS, LogicApp, Webhook, ITSM or Runbook can also be triggered.

To make sure we get also the names of the clusters and the names of the nodes as part of the message we include a custom Json payload on the action.

```
{ "text":"There are #searchresultcount worker nodes needing a
reboot", "IncludeSearchResults":true }
```

```
kubectl apply -f
https://github.com/weaveworks/kured/releases/download/1.1.0/kured-
1.1.0.yaml
```

The end ;)

Docker   Azure Kubernetes Service   Reboot

Get the Medium app