Many ways to deploy ELK on AKS : Helm charts or using the latest Kubernetes Operator called ELK Cloud

1) Create an Azure Kubernetes Service cluster or use existing AKS cluster a add new nodepool

2) Install Elastic Cloud on Kubernetes

3) Create an Elasticsearch cluster

4) Deploy Kibana

5) Create an Azure VM for us to monitor

6) Deploy File/Metric beat to collect logs, metrics and events

**Step 1: Create a new nodepool in AKS cluster**

az aks nodepool add \

--resource-group rg-hcfg \

--cluster-name eu-hcfg-stg \

--name elk-pool \

--node-count 3 \

--node-vm-size Standard\_D2as\_v4 \

--no-wait

**or create a new cluster if required:**

az aks create --resource-group rg-hcfg --name elk-cluster --node-count 3 --generate-ssh-keys

**Step 2: Connect to the AKS cluster**

az aks get-credentials --resource-group rg-hcfg --name elk-cluster

**Step 3: Install the ECK operator**

kubectl apply -f https://download.elastic.co/downloads/eck/1.1.2/all-in-one.yaml

elastic-system namespace created

kubectl -n elastic-system logs -f statefulset.apps/elastic-operator

**Step 4: Create an Elasticsearch cluster with an external IP**

We're using the default load balancer that is available with Azure Kubernetes Service. Please refer to actual deployment file shared along with the documentation

**# Edit the elastic-cloud.yaml**

apiVersion: elasticsearch.k8s.elastic.co/v1

kind: Elasticsearch

metadata:

name: quickstart

spec:

version: 7.7.0 #Make sure you use the version of your choice

http:

service:

spec:

type: LoadBalancer #Adds a External IP

nodeSets:

- name: default

count: 1

config:

node.master: true

node.data: true

node.ingest: true

node.store.allow\_mmap: false

kubectl apply -f elastic-cloud.yaml

elasticsearch.elasticsearch.k8s.elastic.co/quickstart created

**Step 5: Monitor the cluster creation**

Provides the state of Elasticsearch usually turns green after the pods are deployed:

kubectl get elasticsearch

Provides information on Elasticsearch pod creation:

kubectl get pods –w

**Step 6: Check the logs of the pod created**

kubectl logs -f quickstart-es-default-0

kubectl get service quickstart-es-http

**Step 7: Retrieve the password of Elasticsearch cluster**

PASSWORD=$(kubectl get secret quickstart-es-elastic-user -o=jsonpath='{.data.elastic}' | base64 --decode)

uE2h13O6m6xP13R16WQ4SKLR

curl -u "elastic:$PASSWORD" -k "https://52.137.58.173:9200" # External IP adress of our ELK cluster

We get the following response from Elasticsearch and confirming the Elasticsearch deployment is working fine:

"name" : "quickstart-es-default-0",

"cluster\_name" : "quickstart",

"cluster\_uuid" : "v2sGCaEKRd-mLScrHl6j2g",

"version" : {

"number" : "7.9.2",

"build\_flavor" : "default",

"build\_type" : "docker",

"build\_hash" : "d34da0ea4a966c4e49417f2da2f244e3e97b4e6e",

"build\_date" : "2020-09-23T00:45:33.626720Z",

"build\_snapshot" : false,

"lucene\_version" : "8.6.2",

"minimum\_wire\_compatibility\_version" : "6.8.0",

"minimum\_index\_compatibility\_version" : "6.0.0-beta1"

"tagline" : "You Know, for Search"

**Step 8: Deploy Kibana**

# Edit the Kibana.yaml Please refer to the actual kibana.yaml file shared with the documentation

apiVersion: kibana.k8s.elastic.co/v1

kind: Kibana

metadata:

name: quickstart

spec:

version: 7.7.0 #Make sure Kibana and Elasticsearch are on the same version.

http:

service:

spec:

type: LoadBalancer #Adds a External IP

count: 1

elasticsearchRef:

name: quickstart

kubectl apply -f kibana.yaml

**Step 9: Monitor the Kibana deployment**

Provides the state of kibana deployment usually turns green after the pods and services are deployed:

kubectl get kibana

kubectl all service:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 77d

quickstart-es-default ClusterIP None <none> <none> 12m

quickstart-es-http LoadBalancer 10.0.13.14 52.137.58.173 9200:31586/TCP 12m

quickstart-es-transport ClusterIP None <none> 9300/TCP 12m

quickstart-kb-http LoadBalancer 10.0.66.226 51.137.3.150 5601:32416/TCP 5m38s

**Step 10: Adding certificate**

**Create new secret:**

kubectl create secret tls elk-secret --namespace default --key global.key --cert global.crt --from-literal=password=Kibana@123 --from-literal=username=elastic

secret/elk-secret created

**Step 11: Edit the Kibana service and apply for secretName and restart the deployment**

Edit the kibana.yaml and add the TLS information for encrypting traffic coming and going out of Kibana

apiVersion: kibana.k8s.elastic.co/v1

kind: Kibana

metadata:

  name: quickstart

spec:

  version: 7.7.0 #Make sure Kibana and Elasticsearch are on the same version.

  http:

    service:

      spec:

        type: LoadBalancer #Adds a External IP

    tls:

      certificate:

        secretName: elk-secret

  count: 1

  elasticsearchRef:

    name: quickstart

**Step 12: DNS Mapping**

Create a new DNS entry for Kibana external IP

Access the kibana-stg.app.ggotteron.ch:5601

**Step 13 : Genearte and Add the id\_rsa to the nodepool for remote logging to AKS nodes:**

# Genearete SSH keys:

ssh-keygen

# Add the key to the VMSS:

az vmss extension set \

--resource-group MC\_rg-hcfg\_test\_westeurope \

--vmss-name aks-agentpool-20139558-vmss \

--name VMAccessForLinux \

--publisher Microsoft.OSTCExtensions \

--version 1.4 \

--protected-settings "{\"username\":\"azureuser\", \"ssh\_key\":\"$(cat ~/.ssh/id\_rsa.pub)\"}"

# Attach ACR if required:

az aks update -n test -g MC\_rg-hcfg\_test\_westeurope --attach-acr hcfgtest

**STEP 14: Download and Install Filebeat, Metricbeat on the AKS nodes:**

curl -L -O https://raw.githubusercontent.com/elastic/beats/7.10/deploy/kubernetes/filebeat-kubernetes.yaml

# Edit the filebeat-kubernetes.yaml file and change the following:

spec:

serviceAccountName: filebeat

terminationGracePeriodSeconds: 30

hostNetwork: true

dnsPolicy: ClusterFirstWithHostNet

containers:

- name: filebeat

image: docker.elastic.co/beats/filebeat:7.10.0 # check in docker hub for filebeat

namespace = should be the same namespace that elk and kibana are running

output.elasticsearch:

hosts: ['${ELASTICSEARCH\_HOST:elasticsearch}:${ELASTICSEARCH\_PORT:9200}']

username: ${ELASTICSEARCH\_USERNAME}

password: ${ELASTICSEARCH\_PASSWORD}

ssl.certificate\_authorities:

- /home/ganesh-r\_sharma/certificates/global.crt

# Adding a volume that will mount our secret:

volumes:

- name: certs

secret:

secretName: quickstart-es-http-certs-public

# After adding the colume with secret mount it under volumeMounts:

volumeMounts:

- name: certs

mountPath: /home/ganesh-r\_sharma/certificates/global.crt

readOnly: true

subPath: global.crt

# Add env variables in the following:

env:

- name: ELASTICSEARCH\_HOST

value: https://quickstart-es-http

- name: ELASTICSEARCH\_PORT

value: "9200"

- name: ELASTICSEARCH\_USERNAME

value: elastic

- name: ELASTICSEARCH\_PASSWORD

value: uE2h13O6m6xP13R16WQ4SKLR

- name: ELASTIC\_CLOUD\_ID

value:

- name: ELASTIC\_CLOUD\_AUTH

value:

- name: NODE\_NAME

valueFrom:

fieldRef:

fieldPath: spec.nodeName

securityContext:

runAsUser: 0

kubectl apply -f filebeat-kubernetes.yaml

configmap/filebeat-config created

daemonset.apps/filebeat created

clusterrolebinding.rbac.authorization.k8s.io/filebeat created

clusterrole.rbac.authorization.k8s.io/filebeat created

serviceaccount/filebeat created

Please use the attached filebeat manifest file for deployment