Olá pessoal, hoje venho trazer um simples tutorial de implantação do Elasticsearch no Azure Kubernets AKS, este vai ser um projeto bem simples porem seria legal se você se já tiver algum conhecimento com a stack Elastic.

Vamos lá então.

Entre a CLI do Azure, e vamos usar o Azure Cloud Shell (bash).



Em suma, vamos realizar:

- Crie um cluster AKS
- Instale ECK (definições de recursos personalizados + operador)
- Implantar Elastic Stack Elasticsearch, Kibana

"Antes de prosseguir, lembre-se de que este início rápido faz várias suposições simplificadas que não são recomendadas para uma implantação de produção / séria. Por exemplo, usando credenciais de super usuário para tudo ou desabilitando a validação de TLS, entre outros. Consulte a documentação para obter as melhores práticas."

Criar cluster AKS

Crie o grupo de recursos do Azure seguido por um cluster AKS.

az group create --name eck-quickstart-rg --location westus

az aks create --resource-group eck-quickstart-rg --name eckAKSCluster --nodecount 3 --enable-addons monitoring --generate-ssh-keys --tags "purpose=Elastic Deployment"



Isso leva alguns minutos. Uma vez feito isso, o próximo ainda será obter as credenciais do cluster para que seu kubectl possa trabalhar com ele

az aks get-credentials --resource-group eck-quickstart-rg --name eckAKSCluster

rafael@Asure:-\$ az aks get-oredentials --resource-group eck-quickstart-rg --name eckAKSClu Merged "eckAKSCluster" as current context in /home/rafael/.kube/config

Instale ECK

Documentação da Elastic

https://www.elastic.co/guide/en/cloud-on-k8s/current/k8s-deploy-eck.html

kubectl apply -f https://download.elastic.co/downloads/eck/0.9.0/all-in-one.yaml

rafael8&zure:~\$ kubect1 apply -f https://download.elastic.co/downloads/eck/1.2.1/a11-in-one.yam1]

Aguarde alguns segundos e você poderá monitorar os registros conforme o operador é aplicado usando o comando abaixo.



Para ver os logs durante o processo

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

Crie o arquivo para o Elasticsearch

Começaremos com Elasticsearch – um cluster de 3 nós com todos os nós assumindo múltiplas responsabilidades (mestre, ingestão, dados). Crie o arquivo elasticsearch.yaml com o conteúdo abaixo e salve o arquivo. Sinta-se à vontade para escolher o editor de texto (vi ou nano) de sua escolha.



```
apiVersion: elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
   name: quickstart
   labels:
      component: elasticsearch
spec:
   version: 7.9.3
   http:
      service:
```

```
spec:
    type: LoadBalancer
nodeSets:
- name: default
count: 3
config:
    node.master: true
    node.data: true
    node.ingest: true
    node.store.allow_mmap: false
```

O valor LoadBalancer para a propriedade type atribui um endereço external-ip ao serviço elasticsearch para que você possa testá-lo em seu navegador.

Isso implantará um cluster Elasticsearch de 3 nós, portanto, pode levar alguns minutos (levou cerca de 3 minutos para mim). Espere até que a coluna HEALTH mude de vermelho para verde .

Depois verifique o status do Elasticsearch

```
watch kubectl get elasticsearch
```

```
Every 2.0s: kubectl get elasticsearch
NAME HEALTH NODES VERSION PHASE AGE
quickstart green 1 7.9.3 Ready 113s
```

Agora vamos pegar o IP da nossa instalação

kubectl get service quickstart-es-http

rafacl@Azurc:=% kubectl get service quickstart=es=http NAME 0YPE CLUSTER=IP EXTERNAL-IP PORT(8) AGE quickstart=es=http LoadBalancer 10.0.59.242 9200:32657/TCP 2m26s

Bem antes de abrir o navegador e ver o seu Elasticsearch temos de pegar o usuário e senha

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

Sucesso!

OBS. Como o certificado é auto assinado o seu navegador vai apresentar a mensagem se deseja continuar ok...



Agora vamos criar o Kibana para visualizar as nossas métricas



piVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
 name: quickstart
 labels:
 component: kibana
spec:

```
version: 7.9.3
count: 1
elasticsearchRef:
   name: quickstart
http:
   service:
    spec:
     type: LoadBalancer
tls:
     selfSignedCertificate:
```

disabled: true

Rodando o comando para criar.

kubectl apply -f kibana.yaml



Validando o status

watch kubectl get kibana



Agora vamos obter o endereço do Kibana (precisamos ver os dados não é)

Use o comando:

kubectl get service quickstart-kb-http



Acessando:

OBS. Lembrem que tirei o https ok.

Stack:Monitoring-log	PICHL: M 🗣 Decover-Dettic	🗙 🔥 edsAGClaster - Mic	roadh Aour 🕺 😒 Durtic	× +		- 0 ×
€⇒ ୯ ۵	© # S601/A	ogir Toest e Mill F		© ☆	IN ED 38 🗢 4	• • • =
		Welcome	e to Elastic			
		Desmane Pessword				
		Log in				
🔮 Stat Meetering - Sop	ACHLI X Shower-Bade	X 🗠 edittitione - Mo	needlaam 🗙 🔇 Home-Static	× +		- 0 ×
= 🕹 🖪 Hem	• • • • • • • • •	(p), e = e)				
Obse	rvabilty			ą	Security	
APM APM automatic depth performs arrors from insi applications. Add APM	ally collects in ingr ance metrics and source de your proc	5 at logs from popular data toes and easily visualize in configured dishiboards.	Motrics Collect metrics from the ope system and services numbry your derivers. Add metric data	noting inform thread one of the second	+ Endpoint Security of hosts, analyze security station and events, bunt is, automate detections, an or cases.	,
Load a dat	Add sample data a set and a Kibare dashboard	Uplead det	from log file DJSON, or log file	Use S Connect to	Jastiesearch data your Flasticsearch index	
Visualize and	d Explore Data		Manage and Admin	nister the Elast	ic Stack	
APM	(n)	App search evenue distilioants	Stip (LEE and up	e this	Summarize and state	

Bem agora vamos encaminhar alguns logs

Aqui vou encaminhar algumas informações do filebeat (Já existe um artigo ensinando como usar <u>AQUI!</u>).

Uma nota você precisa desabilitar a verificação de ssl ok

Adicione essa linha na configuração para o Elasticsearch

 $ssl.verification_mode: none$

Segue exemplo do meu log "bem simples"



Então é isso pessoal espero que possa ter ajudado um pouco com esse simples tutorial e qualquer dúvida não deixe de comentar será um prazer ajudar.

Sem mais fiquem com Deus e com objetivos claros. \square