# Create a terraform template for Azure kubernetes and deploy a Elasticsearch inside the AKS.

# You need to have [Azure Account](https://azure.microsoft.com/en-us/free/?WT.mc_id=elastic-blog-arsaha) and the [Azure CLI](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli) for Microsoft Azure installed to run terraform platform specific commands. This helps you to create your cluster using this Azure CLI command.

### Step 1: Create an AKS cluster using terraform

resource "azurerm\_resource\_group" "aks" {

name = "my-resources"

location = "east us"

}

resource "azurerm\_kubernetes\_cluster" "aks" {

name = "my-aks1"

location = azurerm\_resource\_group.aks.location

resource\_group\_name = azurerm\_resource\_group.aks.name

dns\_prefix = "myaks1"

default\_node\_pool {

name = "default"

node\_count = 1

vm\_size = "Standard\_D2\_v2"

}

identity {

type = "SystemAssigned"

}

tags = {

Environment = "Production"

}

}

output "client\_certificate" {

value = azurerm\_kubernetes\_cluster.example.kube\_config.0.client\_certificate

sensitive = true

}

output "kube\_config" {

value = azurerm\_kubernetes\_cluster.example.kube\_config\_raw

sensitive = true

}

### Step 2: Connect to the AKS cluster using powershell

# az aks get-credentials --resource-group resourceGroupName --name clusterName

### Step 3: Install the ECK operator

# kubectl create -f https://download.elastic.co/downloads/eck/1.7.1/crds.yaml kubectl apply -f <https://download.elastic.co/downloads/eck/1.7.1/operator.yaml>

### Step 4: Create an Elasticsearch cluster with a external IP

# cat <<EOF | kubectl apply -f –

# apiVersion: elasticsearch.k8s.elastic.co/v1

# kind: Elasticsearch

# metadata:

# name: quickstart

# spec:

# version: 7.14.1 #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.store.allow\_mmap: false

# EOF

### Step 5: Monitor the cluster creation

# kubectl get elasticsearch

# 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)

# curl -u "elastic:$PASSWORD" -k https://<IP\_ADDRESS>:9200

# The public IP address of Elasticsearch can be picked by running

# kubectl get "kubectl get svc quickstart-es-http"

### Step 8: Deploy Kibana

# cat <<EOF | kubectl apply -f –

# apiVersion: kibana.k8s.elastic.co/v1

# kind: Kibana

# metadata:

# name: quickstart

# spec:

# version: 7.14.1 #Make sure Kibana and Elasticsearch are on the same version.

# http:

# service:

# spec:

# type: LoadBalancer #Adds a External IP

# count: 1

# elasticsearchRef:

# name: quickstart

# EOF

### Step 9: Monitor the Kibana deployment

# kubectl get kibana