Kavan Dalwadi

AWS EKS monitoring

horizontal line

31MAY2022

# TOPIC:

We will be monitoring **kubernetes cluster** using **node exporter** to scrape the metrics from the cluster

And we will be deploying **Prometheus and Grafana** on **another server**.

Our node exporter will send metrics to the prometheus using scrape\_config jobs.

From there we will have a Data source of prometheus in grafana and import node exporter dashboard in Grafana.

If there is any misconfiguration in your cluster, your prometheus and grafana will be securely monitoring the cluster as they are on other server.

# PREREQUISITES:

# Install AWS Cli:

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

Aws configure

# Install kubectl:

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

curl -LO "https://dl.k8s.io/release/$(curl -L -s <https://dl.k8s.io/release/stable.txt>)/bin/linux/amd64/kubectl"

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

# Install eksctl :

<https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html>

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

# Create new cluster using eksctl:

eksctl create cluster --name python-app \

--version 1.22 \

--region ap-south-1 \

--nodegroup-name my-nodes \

--node-type t3.small \

--managed --nodes 4 \

--ssh-access \

--ssh-public-key dev\_iam \

--node-ami-family Ubuntu2004

# Scaling Deployment:

kubectl scale deployment/nginx-deployment --replicas=10

# Connect to AWS EKS cluster:

aws configure

aws eks --region <region> update-kubeconfig --name <cluster\_name>

~/.kube/config

# Prometheus Download on Different Server :

<https://prometheus.io/download/> (linux,darwin,windows)

# Grafana Download on Different Server:

<https://grafana.com/docs/grafana/latest/installation/rpm/> (linux)

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-grafana-on-ubuntu-20-04> (ubuntu)

1. **Resources needed to be created on EKS Cluster:**

**DeamonSet YAML:**

apiVersion: apps/v1

kind: DaemonSet

metadata:

labels:

app.kubernetes.io/component: exporter

app.kubernetes.io/name: node-exporter

name: node-exporter

namespace: monitoring

spec:

selector:

matchLabels:

app.kubernetes.io/component: exporter

app.kubernetes.io/name: node-exporter

template:

metadata:

labels:

app.kubernetes.io/component: exporter

app.kubernetes.io/name: node-exporter

spec:

containers:

- args:

- --path.sysfs=/host/sys

- --path.rootfs=/host/root

- --no-collector.wifi

- --no-collector.hwmon

- --collector.filesystem.ignored-mount-points=^/(dev|proc|sys|var/lib/docker/.+|var/lib/kubelet/pods/.+)($|/)

- --collector.netclass.ignored-devices=^(veth.\*)$

name: node-exporter

image: prom/node-exporter

ports:

- containerPort: 9100

protocol: TCP

resources:

limits:

cpu: 250m

memory: 180Mi

requests:

cpu: 102m

memory: 180Mi

volumeMounts:

- mountPath: /host/sys

mountPropagation: HostToContainer

name: sys

readOnly: true

- mountPath: /host/root

mountPropagation: HostToContainer

name: root

readOnly: true

volumes:

- hostPath:

path: /sys

name: sys

- hostPath:

path: /

name: root

**Node Exporter SVC YAML:**

---

kind: Service

apiVersion: v1

metadata:

name: node-exporter

namespace: monitoring

annotations:

prometheus.io/scrape: 'true'

prometheus.io/port: '9100'

spec:

type: NodePort

selector:

app.kubernetes.io/component: exporter

app.kubernetes.io/name: node-exporter

ports:

- name: node-exporter

protocol: TCP

port: 9100

targetPort: 9100

nodePort: 30080

1. **Change prometheus.yaml to get data from node exporter.**
2. **Create prometheus data source in grafana and import node exporter dashboard(1860).**

**Ref:**

[**https://devopscube.com/node-exporter-kubernetes/**](https://devopscube.com/node-exporter-kubernetes/)

eksctl-efk-eks-nodegroup-my-nodes

{

"AWSTemplateFormatVersion": "2010-09-09",

"Description": "EKS Managed Nodes (SSH access: true) [created by eksctl]",

"Mappings": {

"ServicePrincipalPartitionMap": {

"aws": {

"EC2": "ec2.amazonaws.com",

"EKS": "eks.amazonaws.com",

"EKSFargatePods": "eks-fargate-pods.amazonaws.com"

},

"aws-cn": {

"EC2": "ec2.amazonaws.com.cn",

"EKS": "eks.amazonaws.com",

"EKSFargatePods": "eks-fargate-pods.amazonaws.com"

},

"aws-us-gov": {

"EC2": "ec2.amazonaws.com",

"EKS": "eks.amazonaws.com",

"EKSFargatePods": "eks-fargate-pods.amazonaws.com"

}

}

},

"Resources": {

"LaunchTemplate": {

"Type": "AWS::EC2::LaunchTemplate",

"Properties": {

"LaunchTemplateData": {

"BlockDeviceMappings": [

{

"DeviceName": "/dev/sda1",

"Ebs": {

"Encrypted": false,

"Iops": 3000,

"Throughput": 125,

"VolumeSize": 80,

"VolumeType": "gp3"

}

}

],

"ImageId": "ami-030b284bf4040238b",

"KeyName": "user\_kavan",

"MetadataOptions": {

"HttpPutResponseHopLimit": 2,

"HttpTokens": "optional"

},

"SecurityGroupIds": [

{

"Fn::ImportValue": "eksctl-efk-eks-cluster::ClusterSecurityGroupId"

},

{

"Ref": "SSH"

}

],

"TagSpecifications": [

{

"ResourceType": "instance",

"Tags": [

{

"Key": "Name",

"Value": "efk-eks-my-nodes-Node"

},

{

"Key": "alpha.eksctl.io/nodegroup-name",

"Value": "my-nodes"

},

{

"Key": "alpha.eksctl.io/nodegroup-type",

"Value": "managed"

}

]

},

{

"ResourceType": "volume",

"Tags": [

{

"Key": "Name",

"Value": "efk-eks-my-nodes-Node"

},

{

"Key": "alpha.eksctl.io/nodegroup-name",

"Value": "my-nodes"

},

{

"Key": "alpha.eksctl.io/nodegroup-type",

"Value": "managed"

}

]

},

{

"ResourceType": "network-interface",

"Tags": [

{

"Key": "Name",

"Value": "efk-eks-my-nodes-Node"

},

{

"Key": "alpha.eksctl.io/nodegroup-name",

"Value": "my-nodes"

},

{

"Key": "alpha.eksctl.io/nodegroup-type",

"Value": "managed"

}

]

}

],

"UserData": ""

},

"LaunchTemplateName": {

"Fn::Sub": "${AWS::StackName}"

}

}

},

"ManagedNodeGroup": {

"Type": "AWS::EKS::Nodegroup",

"Properties": {

"ClusterName": "efk-eks",

"InstanceTypes": [

"t3.small"

],

"Labels": {

"alpha.eksctl.io/cluster-name": "efk-eks",

"alpha.eksctl.io/nodegroup-name": "my-nodes"

},

"LaunchTemplate": {

"Id": {

"Ref": "LaunchTemplate"

}

},

"NodeRole": {

"Fn::GetAtt": [

"NodeInstanceRole",

"Arn"

]

},

"NodegroupName": "my-nodes",

"ScalingConfig": {

"DesiredSize": 2,

"MaxSize": 2,

"MinSize": 2

},

"Subnets": {

"Fn::Split": [

",",

{

"Fn::ImportValue": "eksctl-efk-eks-cluster::SubnetsPublic"

}

]

},

"Tags": {

"alpha.eksctl.io/nodegroup-name": "my-nodes",

"alpha.eksctl.io/nodegroup-type": "managed"

}

}

},

"NodeInstanceRole": {

"Type": "AWS::IAM::Role",

"Properties": {

"AssumeRolePolicyDocument": {

"Statement": [

{

"Action": [

"sts:AssumeRole"

],

"Effect": "Allow",

"Principal": {

"Service": [

{

"Fn::FindInMap": [

"ServicePrincipalPartitionMap",

{

"Ref": "AWS::Partition"

},

"EC2"

]

}

]

}

}

],

"Version": "2012-10-17"

},

"ManagedPolicyArns": [

{

"Fn::Sub": "arn:${AWS::Partition}:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"

},

{

"Fn::Sub": "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSWorkerNodePolicy"

},

{

"Fn::Sub": "arn:${AWS::Partition}:iam::aws:policy/AmazonEKS\_CNI\_Policy"

},

{

"Fn::Sub": "arn:${AWS::Partition}:iam::aws:policy/AmazonSSMManagedInstanceCore"

}

],

"Path": "/",

"Tags": [

{

"Key": "Name",

"Value": {

"Fn::Sub": "${AWS::StackName}/NodeInstanceRole"

}

}

]

}

},

"SSH": {

"Type": "AWS::EC2::SecurityGroup",

"Properties": {

"GroupDescription": "Allow SSH access",

"GroupName": {

"Fn::Sub": "${AWS::StackName}-remoteAccess"

},

"SecurityGroupIngress": [

{

"CidrIp": "0.0.0.0/0",

"Description": "Allow SSH access to managed worker nodes in group my-nodes",

"FromPort": 22,

"IpProtocol": "tcp",

"ToPort": 22

},

{

"CidrIpv6": "::/0",

"Description": "Allow SSH access to managed worker nodes in group my-nodes",

"FromPort": 22,

"IpProtocol": "tcp",

"ToPort": 22

}

],

"Tags": [

{

"Key": "Name",

"Value": {

"Fn::Sub": "${AWS::StackName}/SSH"

}

}

],

"VpcId": {

"Fn::ImportValue": "eksctl-efk-eks-cluster::VPC"

}

}

}

}

}