**OPA policies for hardening EKS cluster**

Here are some OPA policies that can be used to harden an EKS cluster:

1. Ensure that all EKS worker nodes are running the latest recommended Amazon EKS-optimized Amazon Machine Image (AMI):

package eks.hardening

import data.aws.ec2

latest\_ami := "ami-0c2f20f168de4de4d" # Replace with the latest recommended AMI ID

default allow = false

ami\_id[node\_id] {

node := aws.ec2.instances[\_]

node.iam\_instance\_profile.roles[\_] == "worker"

node.image\_id != latest\_ami

node\_id := node.instance\_id

}

1. Ensure that all worker nodes have the required minimum set of EC2 instance tags:

package eks.hardening

import data.aws.ec2

required\_tags := {

"KubernetesCluster": "my-cluster",

"Environment": "prod",

}

default allow = false

tag\_exists[node\_id] {

node := aws.ec2.instances[\_]

node.iam\_instance\_profile.roles[\_] == "worker"

all(tag\_key, tag\_value := required\_tags) {

node.tags[tag\_key] == tag\_value

}

node\_id := node.instance\_id

}

1. Ensure that all worker nodes have the required security group rules:

package eks.hardening

import data.aws.ec2

required\_ingress\_rules := {

"tcp": [

{

"from\_port": 22,

"to\_port": 22,

"cidr\_blocks": ["0.0.0.0/0"]

}

],

"udp": []

}

default allow = false

ingress\_rule\_exists[node\_id, protocol, from\_port, to\_port, cidr\_block] {

node := aws.ec2.instances[\_]

node.iam\_instance\_profile.roles[\_] == "worker"

node.security\_groups[\_].ip\_permissions[\_].ip\_protocol == protocol

node.security\_groups[\_].ip\_permissions[\_].from\_port == from\_port

node.security\_groups[\_].ip\_permissions[\_].to\_port == to\_port

some(cidr\_block := required\_ingress\_rules[protocol][\_].cidr\_blocks[\_]) {

node.security\_groups[\_].ip\_permissions[\_].ip\_ranges[\_].cidr\_ip == cidr\_block

}

node\_id := node.instance\_id

}

1. Ensure that the Kubernetes API server is accessible only from known IP addresses:

package eks.hardening

import data.kubernetes

default allow = false

allowed\_ips := ["10.0.0.0/8", "192.168.0.0/16"] # Replace with your own list of allowed IP addresses

ip\_allowed {

request := kubernetes.admission.request.object.metadata

all(ip := allowed\_ips) {

net.cidr\_contains(ip, request.annotations["eks.amazonaws.com/ingress-access"])

}

}

1. Ensure that RBAC policies are correctly configured:

package eks.hardening

import data.kubernetes

default allow = false

api\_groups := ["", "apps", "extensions", "batch", "networking.k8s.io"]

allow {

rule := kubernetes.admission.request.object.spec.template.spec.containers[\_].securityContext.privileged

rule == false

}

allow {

rule := kubernetes.admission.request.object.spec.template.spec.initContainers[\_].securityContext.privileged

rule == false

}

allow {

rule := kubernetes.admission.request.object.spec.template.spec.securityContext.seLinuxOptions.type == "spc\_t"

rule == true

}

allow {

rule := kubernetes.admission.request.object.kind