apiVersion: v1

kind: ServiceAccount

metadata:

labels:

k8s-app: metrics-server

name: metrics-server

namespace: kube-system

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

labels:

k8s-app: metrics-server

rbac.authorization.k8s.io/aggregate-to-admin: "true"

rbac.authorization.k8s.io/aggregate-to-edit: "true"

rbac.authorization.k8s.io/aggregate-to-view: "true"

name: system:aggregated-metrics-reader

rules:

- apiGroups:

- metrics.k8s.io

resources:

- pods

- nodes

verbs:

- get

- list

- watch

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

labels:

k8s-app: metrics-server

name: system:metrics-server

rules:

- apiGroups:

- ""

resources:

- pods

- nodes

- nodes/stats

- namespaces

- configmaps

verbs:

- get

- list

- watch

---

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

labels:

k8s-app: metrics-server

name: metrics-server-auth-reader

namespace: kube-system

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: extension-apiserver-authentication-reader

subjects:

- kind: ServiceAccount

name: metrics-server

namespace: kube-system

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

labels:

k8s-app: metrics-server

name: metrics-server:system:auth-delegator

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: system:auth-delegator

subjects:

- kind: ServiceAccount

name: metrics-server

namespace: kube-system

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

labels:

k8s-app: metrics-server

name: system:metrics-server

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: system:metrics-server

subjects:

- kind: ServiceAccount

name: metrics-server

namespace: kube-system

---

apiVersion: v1

kind: Service

metadata:

labels:

k8s-app: metrics-server

name: metrics-server

namespace: kube-system

spec:

ports:

- name: https

port: 443

protocol: TCP

targetPort: https

selector:

k8s-app: metrics-server

---

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

k8s-app: metrics-server

name: metrics-server

namespace: kube-system

spec:

selector:

matchLabels:

k8s-app: metrics-server

strategy:

rollingUpdate:

maxUnavailable: 0

template:

metadata:

labels:

k8s-app: metrics-server

spec:

containers:

- args:

- --cert-dir=/tmp

- --secure-port=4443

- --kubelet-preferred-address-types=InternalIP

- --kubelet-insecure-tls

image: k8s.gcr.io/metrics-server/metrics-server:v0.4.1

imagePullPolicy: IfNotPresent

livenessProbe:

failureThreshold: 3

httpGet:

path: /livez

port: https

scheme: HTTPS

periodSeconds: 10

name: metrics-server

ports:

- containerPort: 4443

name: https

protocol: TCP

readinessProbe:

failureThreshold: 3

httpGet:

path: /readyz

port: https

scheme: HTTPS

periodSeconds: 10

securityContext:

readOnlyRootFilesystem: true

runAsNonRoot: true

runAsUser: 1000

volumeMounts:

- mountPath: /tmp

name: tmp-dir

nodeSelector:

kubernetes.io/os: linux

priorityClassName: system-cluster-critical

serviceAccountName: metrics-server

volumes:

- emptyDir: {}

name: tmp-dir

---

apiVersion: apiregistration.k8s.io/v1

kind: APIService

metadata:

labels:

k8s-app: metrics-server

name: v1beta1.metrics.k8s.io

spec:

group: metrics.k8s.io

groupPriorityMinimum: 100

insecureSkipTLSVerify: true

service:

name: metrics-server

namespace: kube-system

version: v1beta1

versionPriority: 100