```yaml
---
# Source: prometheus/charts/kube-state-metrics/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
automountServiceAccountToken: true
metadata:
  labels:
    helm.sh/chart: kube-state-metrics-5.25.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: kube-state-metrics
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "2.13.0"
  name: prometheus-kube-state-metrics
  namespace: default
---
# Source: prometheus/charts/prometheus-node-exporter/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: prometheus-prometheus-node-exporter
  namespace: default
  labels:
    helm.sh/chart: prometheus-node-exporter-4.37.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: prometheus-node-exporter
    app.kubernetes.io/name: prometheus-node-exporter
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "1.8.2"
automountServiceAccountToken: false
---
# Source: prometheus/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
  namespace: default
  annotations:
    {}
---
# Source: prometheus/templates/cm.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
  namespace: default
data:
  allow-snippet-annotations: "false"
  alerting_rules.yml: |
    {}
  alerts: |
    {}
  prometheus.yml: |
    global:
      evaluation_interval: 1m
      scrape_interval: 1m
      scrape_timeout: 10s
    rule_files:
    - /etc/config/recording_rules.yml
    - /etc/config/alerting_rules.yml
    - /etc/config/rules
    - /etc/config/alerts
    scrape_configs:
    - job_name: prometheus
      static_configs:
      - targets:
        - localhost:9090
    - bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      job_name: kubernetes-apiservers
      kubernetes_sd_configs:
      - role: endpoints
      relabel_configs:
      - action: keep
        regex: default;kubernetes;https
        source_labels:
        - __meta_kubernetes_namespace
        - __meta_kubernetes_service_name
        - __meta_kubernetes_endpoint_port_name
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
    - bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      job_name: kubernetes-nodes
      kubernetes_sd_configs:
      - role: node
      relabel_configs:
      - action: labelmap
        regex: __meta_kubernetes_node_label_(.+)
      - replacement: kubernetes.default.svc:443
        target_label: __address__
      - regex: (.+)
        replacement: /api/v1/nodes/$1/proxy/metrics
        source_labels:
        - __meta_kubernetes_node_name
        target_label: __metrics_path__
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
    - bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      job_name: kubernetes-nodes-cadvisor
      kubernetes_sd_configs:
      - role: node
      relabel_configs:
      - action: labelmap
        regex: __meta_kubernetes_node_label_(.+)
      - replacement: kubernetes.default.svc:443
        target_label: __address__
      - regex: (.+)
        replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
        source_labels:
        - __meta_kubernetes_node_name
        target_label: __metrics_path__
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
    - honor_labels: true
      job_name: kubernetes-service-endpoints
      kubernetes_sd_configs:
      - role: endpoints
      relabel_configs:
      - action: keep
        regex: true
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_scrape
      - action: drop
        regex: true
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_scrape_slow
      - action: replace
        regex: (https?)
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_scheme
        target_label: __scheme__
      - action: replace
        regex: (.+)
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_path
        target_label: __metrics_path__
      - action: replace
        regex: (.+?)(?::\d+)?;(\d+)
        replacement: $1:$2
        source_labels:
        - __address__
        - __meta_kubernetes_service_annotation_prometheus_io_port
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_service_annotation_prometheus_io_param_(.+)
        replacement: __param_$1
      - action: labelmap
        regex: __meta_kubernetes_service_label_(.+)
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: service
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_node_name
        target_label: node
    - honor_labels: true
      job_name: kubernetes-service-endpoints-slow
      kubernetes_sd_configs:
      - role: endpoints
      relabel_configs:
      - action: keep
        regex: true
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_scrape_slow
      - action: replace
        regex: (https?)
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_scheme
        target_label: __scheme__
      - action: replace
        regex: (.+)
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_path
        target_label: __metrics_path__
      - action: replace
        regex: (.+?)(?::\d+)?;(\d+)
        replacement: $1:$2
        source_labels:
        - __address__
        - __meta_kubernetes_service_annotation_prometheus_io_port
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_service_annotation_prometheus_io_param_(.+)
        replacement: __param_$1
      - action: labelmap
        regex: __meta_kubernetes_service_label_(.+)
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: service
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_node_name
        target_label: node
      scrape_interval: 5m
      scrape_timeout: 30s
    - honor_labels: true
      job_name: prometheus-pushgateway
      kubernetes_sd_configs:
      - role: service
      relabel_configs:
      - action: keep
        regex: pushgateway
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_probe
    - honor_labels: true
      job_name: kubernetes-services
      kubernetes_sd_configs:
      - role: service
      metrics_path: /probe
      params:
        module:
        - http_2xx
      relabel_configs:
      - action: keep
        regex: true
        source_labels:
        - __meta_kubernetes_service_annotation_prometheus_io_probe
      - source_labels:
        - __address__
        target_label: __param_target
      - replacement: blackbox
        target_label: __address__
      - source_labels:
        - __param_target
        target_label: instance
      - action: labelmap
        regex: __meta_kubernetes_service_label_(.+)
      - source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - source_labels:
        - __meta_kubernetes_service_name
        target_label: service
    - honor_labels: true
      job_name: kubernetes-pods
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
      - action: keep
        regex: true
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_scrape
      - action: drop
        regex: true
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_scrape_slow
      - action: replace
        regex: (https?)
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_scheme
        target_label: __scheme__
      - action: replace
        regex: (.+)
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_path
        target_label: __metrics_path__
      - action: replace
        regex: (\d+);(([A-Fa-f0-9]{1,4}::?){1,7}[A-Fa-f0-9]{1,4})
        replacement: '[$2]:$1'
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_port
        - __meta_kubernetes_pod_ip
        target_label: __address__
      - action: replace
        regex: (\d+);((([0-9]+?)(\.|$)){4})
        replacement: $2:$1
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_port
        - __meta_kubernetes_pod_ip
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_pod_annotation_prometheus_io_param_(.+)
        replacement: __param_$1
      - action: labelmap
        regex: __meta_kubernetes_pod_label_(.+)
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_name
        target_label: pod
      - action: drop
        regex: Pending|Succeeded|Failed|Completed
        source_labels:
        - __meta_kubernetes_pod_phase
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_node_name
        target_label: node
    - honor_labels: true
      job_name: kubernetes-pods-slow
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
      - action: keep
        regex: true
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_scrape_slow
      - action: replace
        regex: (https?)
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_scheme
        target_label: __scheme__
      - action: replace
        regex: (.+)
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_path
        target_label: __metrics_path__
      - action: replace
```

```
        regex: (\d+);(([A-Fa-f0-9]{1,4}::?){1,7}[A-Fa-f0-9]{1,4})
        replacement: '[$2]:$1'
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_port
        - __meta_kubernetes_pod_ip
        target_label: __address__
      - action: replace
        regex: (\d+);((([0-9]+?)(\.|$)){4})
        replacement: $2:$1
        source_labels:
        - __meta_kubernetes_pod_annotation_prometheus_io_port
        - __meta_kubernetes_pod_ip
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_pod_annotation_prometheus_io_param_(.+)
        replacement: __param_$1
      - action: labelmap
        regex: __meta_kubernetes_pod_label_(.+)
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_name
        target_label: pod
      - action: drop
        regex: Pending|Succeeded|Failed|Completed
        source_labels:
        - __meta_kubernetes_pod_phase
      - action: replace
        source_labels:
        - __meta_kubernetes_pod_node_name
        target_label: node
      scrape_interval: 5m
      scrape_timeout: 30s
  recording_rules.yml: |
    {}
  rules: |
    {}
---
# Source: prometheus/charts/kube-state-metrics/templates/role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    helm.sh/chart: kube-state-metrics-5.25.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: kube-state-metrics
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "2.13.0"
  name: prometheus-kube-state-metrics
rules:

- apiGroups: ["certificates.k8s.io"]
  resources:
  - certificatesigningrequests
  verbs: ["list", "watch"]

- apiGroups: [""]
  resources:
  - configmaps
  verbs: ["list", "watch"]

- apiGroups: ["batch"]
  resources:
  - cronjobs
```

```
    verbs: ["list", "watch"]

  - apiGroups: ["extensions", "apps"]
    resources:
    - daemonsets
    verbs: ["list", "watch"]

  - apiGroups: ["extensions", "apps"]
    resources:
    - deployments
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - endpoints
    verbs: ["list", "watch"]

  - apiGroups: ["autoscaling"]
    resources:
    - horizontalpodautoscalers
    verbs: ["list", "watch"]

  - apiGroups: ["extensions", "networking.k8s.io"]
    resources:
    - ingresses
    verbs: ["list", "watch"]

  - apiGroups: ["batch"]
    resources:
    - jobs
    verbs: ["list", "watch"]

  - apiGroups: ["coordination.k8s.io"]
    resources:
    - leases
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - limitranges
    verbs: ["list", "watch"]

  - apiGroups: ["admissionregistration.k8s.io"]
    resources:
      - mutatingwebhookconfigurations
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - namespaces
    verbs: ["list", "watch"]

  - apiGroups: ["networking.k8s.io"]
    resources:
    - networkpolicies
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - nodes
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - persistentvolumeclaims
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
      - persistentvolumes
    verbs: ["list", "watch"]

  - apiGroups: ["policy"]
    resources:
      - poddisruptionbudgets
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - pods
    verbs: ["list", "watch"]

  - apiGroups: ["extensions", "apps"]
    resources:
    - replicasets
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - replicationcontrollers
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - resourcequotas
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - secrets
    verbs: ["list", "watch"]

  - apiGroups: [""]
    resources:
    - services
    verbs: ["list", "watch"]

  - apiGroups: ["apps"]
    resources:
    - statefulsets
    verbs: ["list", "watch"]

  - apiGroups: ["storage.k8s.io"]
    resources:
      - storageclasses
    verbs: ["list", "watch"]

  - apiGroups: ["admissionregistration.k8s.io"]
    resources:
      - validatingwebhookconfigurations
    verbs: ["list", "watch"]

  - apiGroups: ["storage.k8s.io"]
    resources:
      - volumeattachments
    verbs: ["list", "watch"]
---
# Source: prometheus/templates/clusterrole.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
rules:
  - apiGroups:
      - ""
    resources:
      - nodes
      - nodes/proxy
      - nodes/metrics
      - services
      - endpoints
      - pods
      - ingresses
      - configmaps
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - "extensions"
      - "networking.k8s.io"
    resources:
      - ingresses/status
      - ingresses
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - "discovery.k8s.io"
    resources:
      - endpointslices
    verbs:
      - get
      - list
      - watch
  - nonResourceURLs:
      - "/metrics"
    verbs:
      - get
---
# Source: prometheus/charts/kube-state-metrics/templates/clusterrolebinding.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  labels:
    helm.sh/chart: kube-state-metrics-5.25.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: kube-state-metrics
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "2.13.0"
  name: prometheus-kube-state-metrics
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus-kube-state-metrics
subjects:
- kind: ServiceAccount
  name: prometheus-kube-state-metrics
  namespace: default
---
# Source: prometheus/templates/clusterrolebinding.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
subjects:
  - kind: ServiceAccount
    name: prometheus-server
    namespace: default
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus-server
---
# Source: prometheus/charts/kube-state-metrics/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: prometheus-kube-state-metrics
  namespace: default
  labels:
    helm.sh/chart: kube-state-metrics-5.25.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: kube-state-metrics
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "2.13.0"
  annotations:
    prometheus.io/scrape: 'true'
spec:
  type: "ClusterIP"
  ports:
  - name: "http"
    protocol: TCP
    port: 8080
    targetPort: 8080

  selector:
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
---
# Source: prometheus/charts/prometheus-node-exporter/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: prometheus-prometheus-node-exporter
  namespace: default
  labels:
    helm.sh/chart: prometheus-node-exporter-4.37.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: prometheus-node-exporter
    app.kubernetes.io/name: prometheus-node-exporter
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "1.8.2"
  annotations:
    prometheus.io/scrape: "true"
spec:
  type: ClusterIP
  ports:
    - port: 9100
      targetPort: 9100
      protocol: TCP
      name: metrics
  selector:
    app.kubernetes.io/name: prometheus-node-exporter
    app.kubernetes.io/instance: prometheus
---
# Source: prometheus/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
  namespace: default
spec:
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: 9090
  selector:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
  sessionAffinity: None
  type: "LoadBalancer"
---
# Source: prometheus/charts/prometheus-node-exporter/templates/daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: prometheus-prometheus-node-exporter
  namespace: default
  labels:
    helm.sh/chart: prometheus-node-exporter-4.37.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: prometheus-node-exporter
    app.kubernetes.io/name: prometheus-node-exporter
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "1.8.2"
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: prometheus-node-exporter
      app.kubernetes.io/instance: prometheus
  revisionHistoryLimit: 10
  updateStrategy:
    rollingUpdate:
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      annotations:
        cluster-autoscaler.kubernetes.io/safe-to-evict: "true"
      labels:
        helm.sh/chart: prometheus-node-exporter-4.37.1
        app.kubernetes.io/managed-by: Helm
        app.kubernetes.io/component: metrics
        app.kubernetes.io/part-of: prometheus-node-exporter
        app.kubernetes.io/name: prometheus-node-exporter
        app.kubernetes.io/instance: prometheus
        app.kubernetes.io/version: "1.8.2"
    spec:
      automountServiceAccountToken: false
      securityContext:
        fsGroup: 65534
        runAsGroup: 65534
        runAsNonRoot: true
        runAsUser: 65534
      serviceAccountName: prometheus-prometheus-node-exporter
      containers:
        - name: node-exporter
          image: quay.io/prometheus/node-exporter:v1.8.2
          imagePullPolicy: IfNotPresent
          args:
            - --path.procfs=/host/proc
            - --path.sysfs=/host/sys
            - --path.rootfs=/host/root
            - --path.udev.data=/host/root/run/udev/data
            - --web.listen-address=[$(HOST_IP)]:9100
          securityContext:
            allowPrivilegeEscalation: false
            readOnlyRootFilesystem: true
          env:
            - name: HOST_IP
              value: 0.0.0.0
          ports:
            - name: metrics
              containerPort: 9100
              protocol: TCP
          livenessProbe:
            failureThreshold: 3
            httpGet:
              httpHeaders:
              path: /
              port: 9100
              scheme: HTTP
            initialDelaySeconds: 0
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 1
          readinessProbe:
            failureThreshold: 3
            httpGet:
              httpHeaders:
              path: /
              port: 9100
              scheme: HTTP
            initialDelaySeconds: 0
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 1
          volumeMounts:
            - name: proc
              mountPath: /host/proc
              readOnly:  true
            - name: sys
              mountPath: /host/sys
              readOnly: true
            - name: root
              mountPath: /host/root
              mountPropagation: HostToContainer
              readOnly: true
      hostNetwork: true
      hostPID: true
      nodeSelector:
        kubernetes.io/os: linux
      tolerations:
        - effect: NoSchedule
          operator: Exists
      volumes:
        - name: proc
          hostPath:
            path: /proc
        - name: sys
```

```
          hostPath:
            path: /sys
        - name: root
          hostPath:
            path: /
---
# Source: prometheus/charts/kube-state-metrics/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: prometheus-kube-state-metrics
  namespace: default
  labels:
    helm.sh/chart: kube-state-metrics-5.25.1
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/component: metrics
    app.kubernetes.io/part-of: kube-state-metrics
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: "2.13.0"
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: kube-state-metrics
      app.kubernetes.io/instance: prometheus
  replicas: 1
  strategy:
    type: RollingUpdate
  revisionHistoryLimit: 10
  template:
    metadata:
      labels:
        helm.sh/chart: kube-state-metrics-5.25.1
        app.kubernetes.io/managed-by: Helm
        app.kubernetes.io/component: metrics
        app.kubernetes.io/part-of: kube-state-metrics
        app.kubernetes.io/name: kube-state-metrics
        app.kubernetes.io/instance: prometheus
        app.kubernetes.io/version: "2.13.0"
    spec:
      automountServiceAccountToken: true
      hostNetwork: false
      serviceAccountName: prometheus-kube-state-metrics
      securityContext:
        fsGroup: 65534
        runAsGroup: 65534
        runAsNonRoot: true
        runAsUser: 65534
        seccompProfile:
          type: RuntimeDefault
      containers:
      - name: kube-state-metrics
        args:
        - --port=8080
        - --
resources=certificatesigningrequests,configmaps,cronjobs,daemonsets,deployments,endpoints,horizont
alpodautoscalers,ingresses,jobs,leases,limitranges,mutatingwebhookconfigurations,namespaces,networ
kpolicies,nodes,persistentvolumeclaims,persistentvolumes,poddisruptionbudgets,pods,replicasets,rep
licationcontrollers,resourcequotas,secrets,services,statefulsets,storageclasses,validatingwebhookc
onfigurations,volumeattachments
        imagePullPolicy: IfNotPresent
        image: registry.k8s.io/kube-state-metrics/kube-state-metrics:v2.13.0
        ports:
        - containerPort: 8080
          name: "http"
        livenessProbe:
          failureThreshold: 3
          httpGet:
            httpHeaders:
```

```
            path: /livez
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 5
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
        readinessProbe:
          failureThreshold: 3
          httpGet:
            httpHeaders:
            path: /readyz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 5
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
        resources:
          {}
        securityContext:
          allowPrivilegeEscalation: false
          capabilities:
            drop:
            - ALL
          readOnlyRootFilesystem: true
---
# Source: prometheus/templates/deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/version: v2.54.0
    helm.sh/chart: prometheus-25.26.0
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: prometheus
  name: prometheus-server
  namespace: default
spec:
  selector:
    matchLabels:
      app.kubernetes.io/component: server
      app.kubernetes.io/name: prometheus
      app.kubernetes.io/instance: prometheus
  replicas: 1
  revisionHistoryLimit: 10
  strategy:
    type: Recreate
    rollingUpdate: null
  template:
    metadata:
      labels:
        app.kubernetes.io/component: server
        app.kubernetes.io/name: prometheus
        app.kubernetes.io/instance: prometheus
        app.kubernetes.io/version: v2.54.0
        helm.sh/chart: prometheus-25.26.0
        app.kubernetes.io/managed-by: Helm
        app.kubernetes.io/part-of: prometheus
    spec:
      enableServiceLinks: true
      serviceAccountName: prometheus-server
      containers:
        - name: prometheus-server-configmap-reload
          image: "quay.io/prometheus-operator/prometheus-config-reloader:v0.75.2"
          imagePullPolicy: "IfNotPresent"
```

```
            args:
              - --watched-dir=/etc/config
              - --listen-address=0.0.0.0:8080
              - --reload-url=http://127.0.0.1:9090/-/reload
            ports:
              - containerPort: 8080
                name: metrics
            livenessProbe:
              httpGet:
                path: /healthz
                port: metrics
                scheme: HTTP
              initialDelaySeconds: 2
              periodSeconds: 10
            readinessProbe:
              httpGet:
                path: /healthz
                port: metrics
                scheme: HTTP
              periodSeconds: 10
            volumeMounts:
              - name: config-volume
                mountPath: /etc/config
                readOnly: true

          - name: prometheus-server
            image: "quay.io/prometheus/prometheus:v2.54.0"
            imagePullPolicy: "IfNotPresent"
            args:
              - --storage.tsdb.retention.time=15d
              - --config.file=/etc/config/prometheus.yml
              - --storage.tsdb.path=/data
              - --web.console.libraries=/etc/prometheus/console_libraries
              - --web.console.templates=/etc/prometheus/consoles
              - --web.enable-lifecycle
            ports:
              - containerPort: 9090
            readinessProbe:
              httpGet:
                path: /-/ready
                port: 9090
                scheme: HTTP
              initialDelaySeconds: 30
              periodSeconds: 5
              timeoutSeconds: 4
              failureThreshold: 3
              successThreshold: 1
            livenessProbe:
              httpGet:
                path: /-/healthy
                port: 9090
                scheme: HTTP
              initialDelaySeconds: 30
              periodSeconds: 15
              timeoutSeconds: 10
              failureThreshold: 3
              successThreshold: 1
            volumeMounts:
              - name: config-volume
                mountPath: /etc/config
              - name: storage-volume
                mountPath: /data
                subPath: ""
        dnsPolicy: ClusterFirst
        securityContext:
          fsGroup: 65534
          runAsGroup: 65534
          runAsNonRoot: true
          runAsUser: 65534
```

```yaml
      terminationGracePeriodSeconds: 300
      volumes:
        - name: config-volume
          configMap:
            name: prometheus-server
        - name: storage-volume
          emptyDir:
            {}
---
# Source: grafana/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
automountServiceAccountToken: false
metadata:
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
  name: grafana
  namespace: default
---
# Source: grafana/templates/secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
type: Opaque
data:

  admin-user: "YWRtaW4="
  admin-password: "bWFhTVBBMUZ0ZlBaZkUwS3dyZnRFSFYwMjVXbnV2N1d5WWRtMlI0dA=="
  ldap-toml: ""
---
# Source: grafana/templates/configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
data:

  grafana.ini: |
    [analytics]
    check_for_updates = true
    [grafana_net]
    url = https://grafana.net
    [log]
    mode = console
    [paths]
    data = /var/lib/grafana/
    logs = /var/log/grafana
    plugins = /var/lib/grafana/plugins
    provisioning = /etc/grafana/provisioning
    [server]
    domain = ''
---
# Source: grafana/templates/clusterrole.yaml
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
  name: grafana-clusterrole
rules: []
---
# Source: grafana/templates/clusterrolebinding.yaml
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: grafana-clusterrolebinding
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
subjects:
  - kind: ServiceAccount
    name: grafana
    namespace: default
roleRef:
  kind: ClusterRole
  name: grafana-clusterrole
  apiGroup: rbac.authorization.k8s.io
---
# Source: grafana/templates/role.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
rules: []
---
# Source: grafana/templates/rolebinding.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: grafana
subjects:
- kind: ServiceAccount
  name: grafana
  namespace: default
```

```
---
# Source: grafana/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
spec:
  type: LoadBalancer
  ports:
    - name: service
      port: 80
      protocol: TCP
      targetPort: 3000
  selector:
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
---
# Source: grafana/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grafana
  namespace: default
  labels:
    helm.sh/chart: grafana-8.5.11
    app.kubernetes.io/name: grafana
    app.kubernetes.io/instance: grafana
    app.kubernetes.io/version: "11.3.0"
    app.kubernetes.io/managed-by: Helm
spec:
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/name: grafana
      app.kubernetes.io/instance: grafana
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app.kubernetes.io/name: grafana
        app.kubernetes.io/instance: grafana
      annotations:
        checksum/config: 0e9cbd0ea8e24e32f7dfca5bab17a2ba05652642f0a09a4882833ae88e4cc4a3
        checksum/sc-dashboard-provider-config:
e70bf6a851099d385178a76de9757bb0bef8299da6d8443602590e44f05fdf24
        checksum/secret: e4dd95cb163f3635b163bb90182d0c9c474af8ab6685f9d1c0c4924ec8aa94f6
        kubectl.kubernetes.io/default-container: grafana
    spec:
      
      serviceAccountName: grafana
      automountServiceAccountToken: true
      securityContext:
        fsGroup: 472
        runAsGroup: 472
        runAsNonRoot: true
        runAsUser: 472
      enableServiceLinks: true
      containers:
        - name: grafana
          image: "docker.io/grafana/grafana:11.3.0"
          imagePullPolicy: IfNotPresent
          securityContext:
            allowPrivilegeEscalation: false
            capabilities:
              drop:
              - ALL
            seccompProfile:
              type: RuntimeDefault
          volumeMounts:
            - name: config
              mountPath: "/etc/grafana/grafana.ini"
              subPath: grafana.ini
            - name: storage
              mountPath: "/var/lib/grafana"
          ports:
            - name: grafana
              containerPort: 3000
              protocol: TCP
            - name: gossip-tcp
              containerPort: 9094
              protocol: TCP
            - name: gossip-udp
              containerPort: 9094
              protocol: UDP
          env:
            - name: POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
            - name: GF_SECURITY_ADMIN_USER
              valueFrom:
                secretKeyRef:
                  name: grafana
                  key: admin-user
            - name: GF_SECURITY_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: grafana
                  key: admin-password
            - name: GF_PATHS_DATA
              value: /var/lib/grafana/
            - name: GF_PATHS_LOGS
              value: /var/log/grafana
            - name: GF_PATHS_PLUGINS
              value: /var/lib/grafana/plugins
            - name: GF_PATHS_PROVISIONING
              value: /etc/grafana/provisioning
          livenessProbe:
            failureThreshold: 10
            httpGet:
              path: /api/health
              port: 3000
            initialDelaySeconds: 60
            timeoutSeconds: 30
          readinessProbe:
            httpGet:
              path: /api/health
              port: 3000
      volumes:
        - name: config
          configMap:
            name: grafana
        - name: storage
          emptyDir: {}
```