

```

; sudo -E helm install --replace my-release spark-operator/spark-operator --namespace spark-
install.go:178: [debug] Original chart version: ""
install.go:195: [debug] CHART PATH: /home/asari/.cache/helm/repository/spark-operator-1.1.19

client.go:128: [debug] creating 1 resource(s)
install.go:151: [debug] CRD scheduledsparkapplications.sparkoperator.k8s.io is already present
client.go:128: [debug] creating 1 resource(s)
install.go:151: [debug] CRD sparkapplications.sparkoperator.k8s.io is already present. Skipping
client.go:299: [debug] Starting delete for "my-release-spark-operator" ServiceAccount
client.go:128: [debug] creating 1 resource(s)
client.go:299: [debug] Starting delete for "my-release-spark-operator" ClusterRole
client.go:128: [debug] creating 1 resource(s)
client.go:299: [debug] Starting delete for "my-release-spark-operator" ClusterRoleBinding
client.go:128: [debug] creating 1 resource(s)
client.go:299: [debug] Starting delete for "my-release-spark-operator-webhook-init" Job
client.go:128: [debug] creating 1 resource(s)
client.go:529: [debug] Watching for changes to Job my-release-spark-operator-webhook-init with
client.go:557: [debug] Add/Modify event for my-release-spark-operator-webhook-init: ADDED
client.go:596: [debug] my-release-spark-operator-webhook-init: Jobs active: 0, jobs failed:
client.go:557: [debug] Add/Modify event for my-release-spark-operator-webhook-init: MODIFIED
client.go:596: [debug] my-release-spark-operator-webhook-init: Jobs active: 1, jobs failed:
client.go:557: [debug] Add/Modify event for my-release-spark-operator-webhook-init: MODIFIED
client.go:596: [debug] my-release-spark-operator-webhook-init: Jobs active: 0, jobs failed:
client.go:557: [debug] Add/Modify event for my-release-spark-operator-webhook-init: MODIFIED
client.go:128: [debug] creating 5 resource(s)
NAME: my-release
LAST DEPLOYED: Fri Apr  8 19:40:12 2022
NAMESPACE: spark-operator
STATUS: deployed
REVISION: 2
TEST SUITE: None
USER-SUPPLIED VALUES:
sparkJobNamespace: spark-apps
webhook:
  enable: true

COMPUTED VALUES:
affinity: {}
batchScheduler:
  enable: false
controllerThreads: 10
fullnameOverride: ""
image:
  pullPolicy: IfNotPresent
  repository: ghcr.io/googlecloudplatform/spark-operator

```

```

    tag: ""
    imagePullSecrets: []
    ingressUrlFormat: ""
    istio:
      enabled: false
    labelSelectorFilter: ""
    leaderElection:
      lockName: spark-operator-lock
      lockNamespace: ""
    logLevel: 2
    metrics:
      enable: true
      endpoint: /metrics
      port: 10254
      portName: metrics
      prefix: ""
    nameOverride: ""
    nodeSelector: {}
    podAnnotations: {}
    podLabels: {}
    podMonitor:
      enable: false
      jobLabel: spark-operator-podmonitor
      labels: {}
      podMetricsEndpoint:
        interval: 5s
        scheme: http
    podSecurityContext: {}
    rbac:
      create: false
      createClusterRole: true
      createRole: true
    replicaCount: 1
    resourceQuotaEnforcement:
      enable: false
    resources: {}
    resyncInterval: 30
    securityContext: {}
    serviceAccounts:
      spark:
        annotations: {}
        create: true
        name: ""
      sparkoperator:
        annotations: {}
        create: true

```

```

    name: ""
sparkJobNamespace: spark-apps
tolerations: []
uiService:
  enable: true
volumeMounts: []
volumes: []
webhook:
  cleanupAnnotations:
    helm.sh/hook: pre-delete, pre-upgrade
    helm.sh/hook-delete-policy: hook-succeeded
  enable: true
  initAnnotations:
    helm.sh/hook: pre-install, pre-upgrade
    helm.sh/hook-weight: "50"
  namespaceSelector: ""
  port: 8080
  timeout: 30

```

HOOKS:

```

# Source: spark-operator/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-release-spark-operator
  annotations:
    "helm.sh/hook": pre-install
    "helm.sh/hook-delete-policy": hook-failed, before-hook-creation
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm

```

```

# Source: spark-operator/templates/rbac.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: my-release-spark-operator
  annotations:
    "helm.sh/hook": pre-install
    "helm.sh/hook-delete-policy": hook-failed, before-hook-creation
  labels:
    helm.sh/chart: spark-operator-1.1.19

```

```

    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
rules:
- apiGroups:
  - ""
  resources:
  - pods
  verbs:
  - "*"
- apiGroups:
  - ""
  resources:
  - services
  - configmaps
  - secrets
  verbs:
  - create
  - get
  - delete
  - update
- apiGroups:
  - extensions
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - create
  - get
  - delete
- apiGroups:
  - ""
  resources:
  - nodes
  verbs:
  - get
- apiGroups:
  - ""
  resources:
  - events
  verbs:
  - create
  - update
  - patch
- apiGroups:

```

```

- ""
resources:
- resourcequotas
verbs:
- get
- list
- watch
- apiGroups:
- apiextensions.k8s.io
resources:
- customresourcedefinitions
verbs:
- create
- get
- update
- delete
- apiGroups:
- admissionregistration.k8s.io
resources:
- mutatingwebhookconfigurations
- validatingwebhookconfigurations
verbs:
- create
- get
- update
- delete
- apiGroups:
- sparkoperator.k8s.io
resources:
- sparkapplications
- sparkapplications/status
- scheduledsparkapplications
- scheduledsparkapplications/status
verbs:
- "*"

- apiGroups:
- batch
resources:
- jobs
verbs:
- delete
---
# Source: spark-operator/templates/rbac.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding

```

```

metadata:
  name: my-release-spark-operator
  annotations:
    "helm.sh/hook": pre-install
    "helm.sh/hook-delete-policy": hook-failed, before-hook-creation
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
subjects:
- kind: ServiceAccount
  name: my-release-spark-operator
  namespace: spark-operator
roleRef:
  kind: ClusterRole
  name: my-release-spark-operator
  apiGroup: rbac.authorization.k8s.io
---
# Source: spark-operator/templates/webhook-cleanup-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: my-release-spark-operator-webhook-cleanup
  annotations:
    helm.sh/hook: pre-delete, pre-upgrade
    helm.sh/hook-delete-policy: hook-succeeded
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
spec:
  template:
    metadata:
      name: my-release-spark-operator-webhook-cleanup
    spec:
      serviceAccountName: my-release-spark-operator
      restartPolicy: OnFailure
      containers:
        - name: clean-secret
          image: ghcr.io/googlecloudplatform/spark-operator:v1beta2-1.3.3-3.1.1
          imagePullPolicy: IfNotPresent
          securityContext:

```

```

    {}
  command:
  - "/bin/sh"
  - "-c"
  - "curl -ik \
    -X DELETE \
    -H \"Authorization: Bearer $(cat /var/run/secrets/kubernetes.io/serviceaccount/token)\" \
    -H \"Accept: application/json\" \
    -H \"Content-Type: application/json\" \
    https://kubernetes.default.svc/api/v1/namespaces/spark-operator/secrets/my-release-
    && \
    curl -ik \
    -X DELETE \
    -H \"Authorization: Bearer $(cat /var/run/secrets/kubernetes.io/serviceaccount/token)\" \
    -H \"Accept: application/json\" \
    -H \"Content-Type: application/json\" \
    --data \"{{\\\"kind\\\":\\\"DeleteOptions\\\",\\\"apiVersion\\\":\\\"batch/v1\\\",\\\"
    https://kubernetes.default.svc/apis/batch/v1/namespaces/spark-operator/jobs/my-release-
  ---
# Source: spark-operator/templates/webhook-init-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: my-release-spark-operator-webhook-init
  annotations:
    helm.sh/hook: pre-install, pre-upgrade
    helm.sh/hook-weight: "50"
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
spec:
  template:
    metadata:
      name: my-release-spark-operator-webhook-init
    spec:
      serviceAccountName: my-release-spark-operator
      restartPolicy: OnFailure
      containers:
      - name: main
        image: ghcr.io/googlecloudplatform/spark-operator:v1beta2-1.3.3-3.1.1
        imagePullPolicy: IfNotPresent
        securityContext:
          {}

```

```

        command: [
            "/usr/bin/gencerts.sh",
            "-n", "spark-operator",
            "-s", "my-release-spark-operator-webhook",
            "-r", "my-release-spark-operator-webhook-certs",
            "-p"
        ]
MANIFEST:
---
# Source: spark-operator/templates/spark-serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-release-spark
  namespace: spark-apps
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
---
# Source: spark-operator/templates/spark-rbac.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: spark-role
  namespace: spark-apps
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
rules:
- apiGroups:
  - ""
  resources:
  - pods
  verbs:
  - "*"
- apiGroups:
  - ""
  resources:
  - services
  verbs:

```



```

- "*"
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - "*"
- apiGroups:
  - ""
  resources:
  - persistentvolumeclaims
  verbs:
  - "*"
---
# Source: spark-operator/templates/spark-rbac.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: spark
  namespace: spark-apps
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
subjects:
- kind: ServiceAccount
  name: my-release-spark
  namespace: spark-apps
roleRef:
  kind: Role
  name: spark-role
  apiGroup: rbac.authorization.k8s.io
---
# Source: spark-operator/templates/webhook-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: my-release-spark-operator-webhook
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm

```

```

spec:
  ports:
    - port: 443
      targetPort: 8080
      name: webhook
  selector:
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
---
# Source: spark-operator/templates/deployment.yaml
# If the admission webhook is enabled, then a post-install step is required
# to generate and install the secret in the operator namespace.

# In the post-install hook, the token corresponding to the operator service account
# is used to authenticate with the Kubernetes API server to install the secret bundle.

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-release-spark-operator
  labels:
    helm.sh/chart: spark-operator-1.1.19
    app.kubernetes.io/name: spark-operator
    app.kubernetes.io/instance: my-release
    app.kubernetes.io/version: "v1beta2-1.3.3-3.1.1"
    app.kubernetes.io/managed-by: Helm
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: spark-operator
      app.kubernetes.io/instance: my-release
  strategy:
    type: Recreate
  template:
    metadata:
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/port: "10254"
        prometheus.io/path: /metrics
      labels:
        app.kubernetes.io/name: spark-operator
        app.kubernetes.io/instance: my-release
    spec:
      serviceAccountName: my-release-spark-operator
      securityContext:

```

```

    {}
  containers:
    - name: spark-operator
      image: ghcr.io/googlecloudplatform/spark-operator:v1beta2-1.3.3-3.1.1
      imagePullPolicy: IfNotPresent
      securityContext:
        {}
      ports:
        - name: "metrics"
          containerPort: 10254

      args:
        - -v=2
        - -logtostderr
        - -namespace=spark-apps
        - -enable-ui-service=true
        - -ingress-url-format=
        - -controller-threads=10
        - -resync-interval=30
        - -enable-batch-scheduler=false
        - -label-selector-filter=
        - -enable-metrics=true
        - -metrics-labels=app_type
        - -metrics-port=10254
        - -metrics-endpoint=/metrics
        - -metrics-prefix=
        - -enable-webhook=true
        - -webhook-svc-namespace=spark-operator
        - -webhook-port=8080
        - -webhook-timeout=30
        - -webhook-svc-name=my-release-spark-operator-webhook
        - -webhook-config-name=my-release-spark-operator-webhook-config
        - -webhook-namespace-selector=
        - -enable-resource-quota-enforcement=false
      resources:
        {}
      volumeMounts:
        - name: webhook-certs
          mountPath: /etc/webhook-certs
  volumes:
    - name: webhook-certs
      secret:
        secretName: my-release-spark-operator-webhook-certs

```