Setup Cronjob For Auto Start and Stop K8s Services

There are 2 ways to setup cron job for autoscale at particular time

1. Keda Autoscaler

2. K8s cronjob

Keda Auto Scaler

1. Install Keda Autoscaler in K8s cluster.

2. Use below config file to deploy cronjob for autoscaling.

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: cron-scaledobject
  namespace: <NameSpace>
spec:
  scaleTargetRef:
    name: <Deployment Name>
  triggers:
  - type: cron
    metadata:
      timezone: Asia/Kolkata
      start: 55 18 * * *
      end: 59 18 * * *
      desiredReplicas: "4"
```

3. Update namespace and deployment name. Add multiple triggers in triggers section of config file and add timezone and desiredReplicas.

4. Deploy this in same namespace where services are deployed.

5. Reference Link - https://keda.sh/docs/2.11/scalers/cron/

K8s Cronjob

1. To provide permission to cronjob to perform scale job we need to create service account, rule and bind service account with rule.
2. Use below config to create cron job and provide permission to scale deployment.

```
kind: ServiceAccount
apiVersion: v1
metadata:
  name: start-stop-prod
  namespace: <NameSpace>
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: control-deployment
  namespace: <NameSpace>
rules:
  - apiGroups: ["apps", "extensions"]
    resources: ["deployments"]
    verbs: ["get", "patch", "list", "watch", "scale"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: control-deployment
  namespace: <NameSpace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: control-deployment
subjects:
  - kind: ServiceAccount
    name: start-stop-prod
    namespace: <NameSpace>
---
```

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: start-prod-deployment
  namespace: <NameSpace>
spec:
  successfulJobsHistoryLimit: 1
  failedJobsHistoryLimit: 2
  concurrencyPolicy: Forbid
  schedule: '15 5 * * *'
  jobTemplate:
    spec:
      backoffLimit: 2
      activeDeadlineSeconds: 600
      template:
        spec:
          serviceAccountName: start-stop-prod
          restartPolicy: OnFailure
          containers:
          - name: kubectl
            image: bitnami/kubectl
            command:
            - bash
            - -c
            - |
              kubectl -n <NameSpace> patch deployment <Deploy
---
apiVersion: batch/v1
kind: CronJob
metadata:
  name: stop-prod-deployment
  namespace: <NameSpace>
spec:
  successfulJobsHistoryLimit: 1
  failedJobsHistoryLimit: 2
  concurrencyPolicy: Forbid
  schedule: '30 9 * * *'
  jobTemplate:
    spec:
      backoffLimit: 2
      activeDeadlineSeconds: 600
      template:
        spec:
          serviceAccountName: start-stop-prod
          restartPolicy: OnFailure
          containers:
          - name: kubectl
            image: bitnami/kubectl
            command:
            - bash
            - -c
            - |
              kubectl -n <NameSpace> patch deployment <Deploy
```

3. Update namespace and deployment name