# Below document applies to only the tenants, where app team doesn’t want the Spark as part of their tenant.

Once the tenant is created using UI , headover to CLI and run,

C:\Users\naveen.rajan>kubectl edit tenant nr-minion

tenant.hcp.hpe.com/nr-minion edited

Update the resource count to zero for hivemetastore, sparkhs, tenantcli as seen below, ctrl + s and close.



C:\Users\naveen.rajan>kubectl get statefulsets,deploy -n nr-minion

NAME READY AGE

statefulset.apps/hivemeta 0/1 7m4s

statefulset.apps/tenantcli 0/1 7m4s

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/kiali 0/1 0 0 7m7s

deployment.apps/livy 0/0 0 0 7m4s

deployment.apps/spark-master 0/0 0 0 7m4s

deployment.apps/spark-ui-proxy 0/0 0 0 7m4s

deployment.apps/spark-worker 0/0 0 0 7m4s

deployment.apps/sparkhs 0/1 0 0 7m4s

deployment.apps/sparkts 0/0 0 0 7m4s

Unless the app team specifically ask for spark deployment as part of their tenant, we can scale them down as follows. It is done to avoid resource crunch as the team doesn’t need spark running.

C:\Users\naveen.rajan>kubectl scale sts hivemeta --replicas=0 -n nr-minion

statefulset.apps/hivemeta scaled

C:\Users\naveen.rajan>kubectl scale sts tenantcli --replicas=0 -n nr-minion

statefulset.apps/tenantcli scaled

C:\Users\naveen.rajan>kubectl scale deploy sparkhs --replicas=0 -n nr-minion

deployment.apps/sparkhs scaled

Validate that they are scaled down to zero,

C:\Users\naveen.rajan>kubectl get statefulsets,deploy -n nr-minion

NAME READY AGE

statefulset.apps/hivemeta 0/0 3m55s

statefulset.apps/tenantcli 0/0 3m42s

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/kiali 0/1 0 0 12m

deployment.apps/livy 0/0 0 0 12m

deployment.apps/spark-master 0/0 0 0 12m

deployment.apps/spark-ui-proxy 0/0 0 0 12m

deployment.apps/spark-worker 0/0 0 0 12m

deployment.apps/sparkhs 0/0 0 0 2m11s

deployment.apps/sparkts 0/0 0 0 12m