# Rakuten - Yugabyte Deployment Best Practices

## Prerequisites

Below are the prerequisites for the Yugabyte setup and deployment

* **Namespace creation:** Created ricc-db-platform and ricc-db-postgres namespaces for Yugabyte Anywhere and Yugabyte Database
* **Roles and rolebinding**

yugabyte-helm-operations role has a set of permissions required to install, upgrade, delete the yugabyte chart.

Below files are needed to execute by Kiba/Robin team having admin access to run the below yaml files

1. yugabyte-platform-universe-management-sa.yaml
2. platform\_rbac.yaml
3. ns-sy-ros-yb\_rbac.yaml
4. yugabyte-platform-universe-management-sa.yaml

Service account name: yugabyte-platform-universe-management

| apiVersion: v1 kind: ServiceAccount metadata:  name: yugabyte-platform-universe-management  namespace: ricc-db-platform |
| --- |

1. platform\_rbac.yaml

| # yugabyte-helm-operations role has set of permissions required to # install, upgrade, delete the yugabyte chart. apiVersion: rbac.authorization.k8s.io/v1 kind: Role metadata:  name: yugabyte-helm-operations  namespace: ricc-db-postgres rules: - apiGroups:  - "policy"  resources:  - "poddisruptionbudgets"  verbs:  - "get"  - "list"  - "create"  - "delete"  - "patch" - apiGroups:  - ""  resources:  - "services"  verbs:  - "get"  - "delete"  - "create"  - "patch" - apiGroups:  - "apps"  resources:  - "statefulsets"  verbs:  - "get"  - "delete"  - "create"  - "patch" - apiGroups:  - ""  resources:  - "secrets"  verbs:  - "create"  - "list" # needed by Helm  - "get"  - "delete"  - "update" # needed by Helm  - "patch" ## We don't use ServiceMonitor in platform # - apiGroups: # - "monitoring.coreos.com" # resources: # - "servicemonitors" # verbs: # - "get" # - "delete" # - "create" # - "patch" ---  # yugabyte-management role has set of permissions required by # platform software to manage YugabyteDB universes. apiVersion: rbac.authorization.k8s.io/v1 kind: Role metadata:  name: yugabyte-management  namespace: ricc-db-postgres rules: # - apiGroups: # - "" # resources: # - "namespaces" # verbs: # - "delete" # - "create" - apiGroups:  - ""  resources:  - "pods"  verbs:  - "get" # required for single pod  - "list" # required for get pods - apiGroups:  - ""  resources:  - "services"  verbs:  - "get"  - "list" # required for label selectors - apiGroups:  - ""  resources:  - "persistentvolumeclaims"  verbs:  - "list" # required for label selectors  - "delete" - apiGroups:  - ""  resources:  - "pods/exec"  verbs:  - "create" # required for exec and cp - apiGroups:  - "apps"  resources:  - "statefulsets/scale"  verbs:  - "patch" # required for scale command ---  apiVersion: rbac.authorization.k8s.io/v1 kind: RoleBinding metadata:  name: yugabyte-helm-operations  namespace: ricc-db-postgres roleRef:  apiGroup: rbac.authorization.k8s.io  kind: Role  name: yugabyte-helm-operations subjects:  - kind: ServiceAccount  name: yugabyte-platform-universe-management  namespace: ricc-db-platform ---  apiVersion: rbac.authorization.k8s.io/v1 kind: RoleBinding metadata:  name: yugabyte-management  namespace: ricc-db-postgres roleRef:  apiGroup: rbac.authorization.k8s.io  kind: Role  name: yugabyte-management subjects:  - kind: ServiceAccount  name: yugabyte-platform-universe-management  namespace: ricc-db-platform  --- |
| --- |

1. ns-sy-ros-yb\_rbac.yaml

| --- # Source: yugaware/templates/rbac.yaml kind: Role apiVersion: rbac.authorization.k8s.io/v1 metadata:  name: yb-platform  namespace: ricc-db-platform  labels:  k8s-app: yugaware  kubernetes.io/cluster-service: "true"  addonmanager.kubernetes.io/mode: Reconcile rules: - apiGroups: [""]  resources:  - nodes  - nodes/proxy  - services  - endpoints  - pods  - pods/exec  verbs: ["get", "list", "watch", "create"] - apiGroups:  - extensions  resources:  - ingresses  verbs: ["get", "list", "watch"] #- nonResourceURLs: ["/metrics"]  #verbs: ["get"] - apiGroups: [""]  resources:  - namespaces  - secrets  - pods/portforward  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"] - apiGroups: ["", "extensions"]  resources:  - deployments  - services  verbs: ["create", "get", "list", "watch", "update", "delete"]  ---  # Source: yugaware/templates/rbac.yaml kind: RoleBinding apiVersion: rbac.authorization.k8s.io/v1 metadata:  name: yb-platform  namespace: ricc-db-platform  labels:  k8s-app: yugaware  kubernetes.io/cluster-service: "true"  addonmanager.kubernetes.io/mode: Reconcile subjects: - kind: ServiceAccount  name: yb-platform  namespace: ricc-db-platform roleRef:  kind: Role  name: yb-platform  apiGroup: rbac.authorization.k8s.io  --- |
| --- |

## Issues and Resolution

Below has the issues reported and resolution for the Yugabyte setup and deployment

* **Issue No 1:** While logging for the tenant user is getting below error

| ssh amola.bhoite@192.168.1.201 |
| --- |

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ robin login amol.bhoite-a --tenant tn-itsyscom-ricc Password: The below error has occured: ROBIN server master.robin-server.service.robin is not resolvable Please set the current context to the right server.   robin client set-current <SERVER-IP>  If listening on a different port, use  robin client set-current <SERVER-IP> --port <PORT> |
| --- |

**Resolution:** Follow the instructions mentioned in the document to get rid of the login issues

<https://drive.google.com/file/d/1QooU6u1RyMnmcrBN7sJofoIY-94PBwgC/view?usp=sharing>

* **Issue No 2:** PVC binding was having issue due to some Robin pending setup

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl describe pvc datadir0-yb-tserver-0 -n ricc-db-postgres Name: datadir0-yb-tserver-0 Namespace: ricc-db-postgres StorageClass: robin Status: Pending Volume: Labels: app=yb-tserver  chart=yugabyte  component=yugabytedb  heritage=Helm  release=yb-dev-symcarriercluster Annotations: volume.beta.kubernetes.io/storage-class: robin  volume.beta.kubernetes.io/storage-provisioner: robin  volume.kubernetes.io/selected-node: uhn7kls1arbwk002.rmnkiba.local Finalizers: [kubernetes.io/pvc-protection] Capacity: Access Modes: VolumeMode: Filesystem Mounted By: yb-tserver-0 Events:  Type Reason Age From Message  ---- ------ ---- ---- -------  Normal WaitForFirstConsumer 4m12s persistentvolume-controller waiting for first consumer to be created before binding  Normal WaitForPodScheduled 4m8s persistentvolume-controller waiting for pod yb-tserver-0 to be scheduled  Normal Provisioning 118s (x8 over 4m6s) robin\_csi-provisioner-robin-6465d7fc95-tb45g\_0db3c162-2a09-4b82-b971-07fec6037a09 External provisioner is provisioning volume for claim "ricc-db-postgres/datadir0-yb-tserver-0"  Warning ProvisioningFailed 118s (x8 over 4m6s) robin\_csi-provisioner-robin-6465d7fc95-tb45g\_0db3c162-2a09-4b82-b971-07fec6037a09 failed to provision volume with StorageClass "robin": rpc error: code = Aborted desc = Error: CreateVolume Request has already been issued  Normal ExternalProvisioning 8s (x17 over 4m6s) persistentvolume-controller waiting for a volume to be created, either by external provisioner "robin" or manually created by system administrator |
| --- |

**Resolution:**

Kiba/Robin team did changes at the storage side to fix this issue and now storage volumes are bound to all the tserver under ricc-db-postgres namespace

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl describe pvc datadir0-yb-tserver-0 -n ricc-db-postgres Name: datadir0-yb-tserver-0 Namespace: ricc-db-postgres StorageClass: robin Status: Bound Volume: pvc-142296e8-b629-41ff-bf87-76400783803d Labels: app=yb-tserver  chart=yugabyte  component=yugabytedb  heritage=Helm  release=yb-dev-symcarriercluster Annotations: pv.kubernetes.io/bind-completed: yes  pv.kubernetes.io/bound-by-controller: yes  volume.beta.kubernetes.io/storage-class: robin  volume.beta.kubernetes.io/storage-provisioner: robin  volume.kubernetes.io/selected-node: uhn7kls1arbwk002.rmnkiba.local Finalizers: [kubernetes.io/pvc-protection] Capacity: 80Gi Access Modes: RWO VolumeMode: Filesystem Mounted By: yb-tserver-0 Events: <none> |
| --- |

* **Issue No 3:** Clock skew issue between pods interconnect due to uhn7kls1arbwk001.rmnkiba.local and uhn7kls1arbwk002.rmnkiba.local host time difference

| --------------------------------------------------------------------------- [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl get pods -n ricc-db-postgres -o wide NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES yb-master-0 2/2 Running 0 73m 240b:c0e0:205:547b:b494:2:0:4904 uhn7kls1arbwk001.rmnkiba.local <none> <none> yb-master-1 2/2 Running 0 73m 240b:c0e0:205:547b:b494:2:0:49a4 uhn7kls1arbwk001.rmnkiba.local <none> <none> yb-master-2 2/2 Running 0 73m 240b:c0e0:205:547b:b494:2:0:4c2e uhn7kls1arbwk002.rmnkiba.local <none> <none> yb-tserver-0 2/2 Running 0 8m28s 240b:c0e0:205:547b:b494:2:0:4709 uhn7kls1arbwk002.rmnkiba.local <none> <none> yb-tserver-1 2/2 Running 0 14m 240b:c0e0:205:547b:b494:2:0:4f64 uhn7kls1arbwk001.rmnkiba.local <none> <none> yb-tserver-2 2/2 Running 0 17m 240b:c0e0:205:547b:b494:2:0:41b2 uhn7kls1arbwk002.rmnkiba.local <none> <none>  --------------------------------------------------------------------------- uhn7kls1arbwk001.rmnkiba.local - yb-master-0, yb-master-1, yb-tserver-1 pods  kubectl exec -it yb-master-0 -n ricc-db-postgres date kubectl exec -it yb-master-1 -n ricc-db-postgres date kubectl exec -it yb-tserver-1 -n ricc-db-postgres date --------------------------------------------------------------------------- uhn7kls1arbwk002.rmnkiba.local - yb-master-2, yb-tserver-0,yb-tserver-2 pods  kubectl exec -it yb-master-2 -n ricc-db-postgres date kubectl exec -it yb-tserver-0 -n ricc-db-postgres date kubectl exec -it yb-tserver-2 -n ricc-db-postgres date --------------------------------------------------------------------------- Executed below commands sequentially yb-master-0 is sitting on uhn7kls1arbwk001.rmnkiba.local  yb-master-2 is sitting on uhn7kls1arbwk002.rmnkiba.local  Please find time difference in time  [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl exec -it yb-master-0 -n ricc-db-postgres date Thu Jun 9 12:09:29 UTC 2022  [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl exec -it yb-master-2 -n ricc-db-postgres date Thu Jun 9 12:09:04 UTC 2022 |
| --- |

**Resolution:** Kiba team adjusted the time on nodes to have same time, all the pods are having same time to avoid 'clock skew' issue for the interconnect communication

* **Issue No 4:** Kubectl is timing out from the prompt(after 45 seconds) due to Robin's ideal timeout setting, needing to change this default time to 30 minutes from 45 seconds. This is needed for the longer operations in the background e.g. backup

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ date Mon Jun 13 12:40:18 JST 2022  [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl exec -it yb-tserver-0 -n ricc-db-postgres -- bash Defaulting container name to yb-tserver. Use 'kubectl describe pod/yb-tserver-0 -n ricc-db-postgres' to see all of the containers in this pod. [root@yb-tserver-0 cores]#  [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ date Mon Jun 13 12:41:05 JST 2022 |
| --- |

**Resolution:** Robin team increased kubectl timeout to 30 minutes

## Yugabyte Anywhere Deployment

The YugabyteDB Anywhere UI is used in a highly-available mode, allowing you to create and manage YugabyteDB universes, or clusters, on one or more regions across public cloud and private on-premises data centers.

* Download and unzip the folder <https://drive.google.com/file/d/1Kn_D5tvrRlRFiVFZNBGey7WjjPebE4vw/view?usp=sharing>

values.yaml - Make required changes in the values.yaml file for the yugaware

| # Default values for yugaware. # This is a YAML-formatted file. # Declare variables to be passed into your templates.  image:  commonRegistry: ""  # Setting commonRegistry to say, quay.io overrides the registry settings for all images  # including the yugaware image   repository: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake/yugaware  tag: 2.12.5.0-b24  pullPolicy: IfNotPresent  pullSecret: yugabyte-k8s-pull-secret  ## Docker config JSON File name  ## If set, this file content will be used to automatically create secret named as above  # pullSecretFile:   thirdparty-deps:  registry: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake  tag: latest  name: yugabyte/thirdparty-deps   postgres:  registry: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake  tag: 11.5  name: postgres   prometheus:  registry: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake  tag: v2.27.1  name: prom/prometheus   nginx:  registry: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake  tag: 1.17.4-amd64  name: nginxinc/nginx-unprivileged  yugaware:  replicas: 1  storage: 500Gi  storageClass: robin  # storageAnnotations:  # robin.io/replication: "3"  # robin.io/faultdomain: host  multiTenant: false  serviceAccount: yugaware  serviceMonitor:  enabled: false  annotations: {}  serviceAccountAnnotations: {}  service:  annotations: {}  enabled: true  ip: ""  type: "NodePort"  pod:  annotations: {}  health:  username: ""  password: ""  email: ""  resources:  requests:  cpu: 16  memory: 64Gi  enableProxyMetricsAuth: true  ## List of additional alowed CORS origins in case of complex rev-proxy  additionAllowedCorsOrigins: []  proxyEndpointTimeoutMs: 1 minute  ## Enables features specific for cloud deployments  cloud:  enabled: false  requestIdHeader: "X-REQUEST-ID"   podDisruptionBudget:  # See https://kubernetes.io/docs/tasks/run-application/configure-pdb/  # Note that the default of 0 doesn't really make sense since a StatefulSet isn't allowed to schedule extra replicas. However it is maintained as the default while we do additional testing. This value will likely change in the future.  maxUnavailable: 0  ## Configure PostgreSQL part of the application postgres:  service:  ## Expose internal Postgres as a Service  enabled: false  ## Additional Service annotations  annotations: {}  ## Service type  type: "ClusterIP"  ## IP address for the LoadBalancer, works only if supported by the cloud provider  ip: ""   resources:  requests:  cpu: 0.5  memory: 1Gi   # If external.host is set then we will connect to an external postgres database server instead of starting our own.  external:  host: null  port: 5432  pass: ""  dbname: postgres  user: postgres   ## JDBC connection parameters including the leading `?`.  jdbcParams: ""  tls:  enabled: false  hostname: "localhost:32004"  #hostname: "localhost"  certificate: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  key: "\*\*\*\*\*\*\*\*\*\*\*\*"  sslProtocols: "" # if set, override default Nginx SSL protocols setting  ## yugaware pod Security Context ## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ securityContext:  enabled: false  fsGroup: 1001  runAsUser: 1001  helm:  timeout: 9000  package: "/opt/yugabyte/helm/yugabyte-latest.tgz"  #domainName: "uhn7kls16.local" domainName: "uhn7tt7r5.local"  helm2Legacy: false  ip\_version\_support: "v6\_only" # v4\_only, v6\_only are the only supported values at the moment  nginx:  # max size of file upload allowed by YB platform  upload\_size: 10G   resources:  requests:  cpu: 0.25  memory: 300Mi  rbac:  ## Set this to false if you don't have enough permissions to create  ## ClusterRole and Binding, for example an OpenShift cluster. When  ## set to false, some of the graphs from Container section of the  ## Metrics UI don't work.  create: false  ## In order to deploy on OpenShift Container Platform, set this to ## true. ocpCompatibility:  enabled: false  # Extra containers to add to the pod. sidecars: []  ## Don't want prometheus to scrape nodes and evaluate alert rules in some cases (for example - cloud). prometheus:  scrapeNodes: true  evaluateAlertRules: true  retentionTime: 10d   resources:  requests:  cpu: 2  memory: 4Gi  # Arbitrary key=value config entries for application.docker.conf additionalAppConf:  stringConf:  nonStringConf: |
| --- |

Check and adjust below important settings

**repository: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake/yugaware**

Image version downloaded from https://docs.yugabyte.com/preview/releases/release-notes/v2.12/

docker pull yugabytedb/yugabyte:2.12.5.0-b24

**tag: 2.12.5.0-b24**

This is build name downloaded from docker pull command

**registry: kldev-docker.slb-wartifactory-v.stg.rmn.local/rakuten/data-lake**

Rakuten internal image location destination

**storage: 500Gi**

Storage for Yugaware which can be customized as needed

**storageClass: robin**

Storage class for Robin cluster

* Run helm install command for the ricc-db-platform namespace, yugaware will install inside ricc-db-platform namespace

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s active\_cluster]$ helm install yb-platform . -f values.yaml -n ricc-db-platform WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /home/amola.bhoite/.kube/config WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /home/amola.bhoite/.kube/config NAME: yb-platform LAST DEPLOYED: Wed Jun 8 14:55:56 2022 NAMESPACE: ricc-db-platform STATUS: deployed REVISION: 1 TEST SUITE: None |
| --- |

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl get pvc -n ricc-db-platform NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE yb-platform-yugaware-storage Bound pvc-570d6250-2a5b-4305-a96f-4c3a37a7f214 500Gi RWO robin 5d3h |
| --- |

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl get pods -n ricc-db-platform NAME READY STATUS RESTARTS AGE yb-platform-yugaware-0 4/4 Running 0 5d3h |
| --- |

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl get svc -n ricc-db-platform NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE yb-platform-yugaware-ui NodePort 240b:c0e0:205:547b:b494:2:0:89aa <none> 80:31336/TCP,9090:32284/TCP 5d4h |
| --- |

* Configure the Kubernetes cloud provider: Download generate\_kubeconfig.py file

Run the following wget command to get the Python script for generating the kubeconfig file

wget <https://raw.githubusercontent.com/YugaByte/charts/master/stable/yugabyte/generate_kubeconfig.py>

kubeconfig file: You can create a kubeconfig file for the previously created yugabyte-platform-universe-management service account as follows

Run the following command to generate the kubeconfig file:

| python generate\_kubeconfig.py -s yugabyte-platform-universe-management -n ricc-db-platform  Generated the kubeconfig file: /tmp/yugabyte-platform-universe-management.conf |
| --- |

Use this generated kubeconfig file as the kubeconfig in the YugabyteDB Anywhere Kubernetes provider configuration.

Check service account name created as yugabyte-platform-universe-management service

| [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl get sa -n ricc-db-platform NAME SECRETS AGE default 1 5d23h yb-platform 1 5d4h yugabyte-platform-universe-management 1 5d20h |
| --- |

## Yugabyte Database deployment

* Copy /opt/yugabyte/helm/yugabyte-2.12.5.0-b24-helm.tar.gz helm chart to

| /opt/yugabyte/releases/2.12.5.0-b24/ location, so that this helm chart will be used to create Yugabyte database universe creation  [amola.bhoite@lab.rmn.local@uhn5klbjhdmz-s ~]$ kubectl exec -it yb-platform-yugaware-0 -c yugaware -n ricc-db-platform -- bash  [root@yb-platform-yugaware-0 2.12.5.0-b24]# cp /opt/yugabyte/helm/yugabyte-2.12.5.0-b24-helm.tar.gz heml /opt/yugabyte/releases/2.12.5.0-b24/ |
| --- |

* Update chartPath value

| [amola.bhoite@lab.rmn.local@uhn5klbjhmgmt-p ~]$ kubectl exec -it yb-platform-yugaware-0 -n yb-platform -c postgres -- bash root@yb-platform-yugaware-0:/# su postgres postgres@yb-platform-yugaware-0:/$ psql psql (11.5 (Debian 11.5-3.pgdg90+1)) Type "help" for help.   postgres=# \c yugaware   update yugaware\_property set value='{"2.12.5.0-b24":{"state":"ACTIVE","notes":[],"filePath":"/opt/yugabyte/releases/2.12.5.0-b24/yugabyte-2.12.5.0-b24-centos-x86\_64.tar.gz","chartPath":"/opt/yugabyte/releases/2.12.5.0-b24/yugabyte-2.12.5.0-b24-helm.tar.gz","imageTag":"2.12.5.0-b24"}}' where name='SoftwareReleases'; |
| --- |

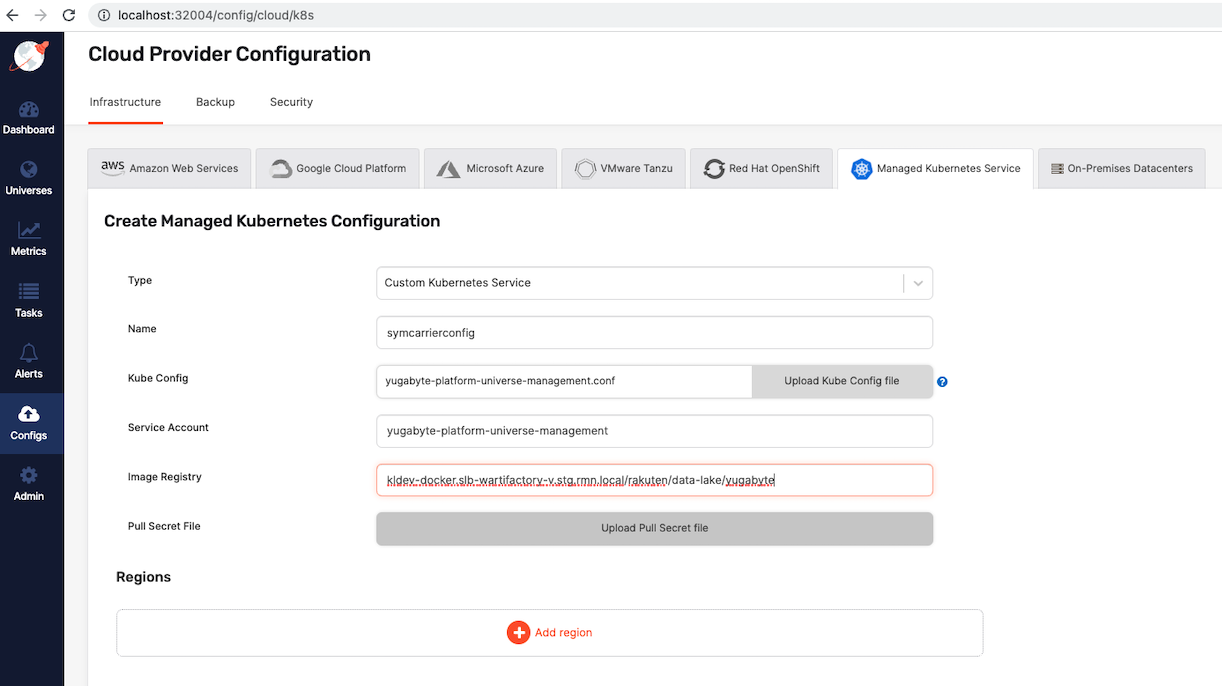
* Redirect Yugaware UI to localhost 32004 port

| ssh -L 32004:[240b:c0e0:104:5400:b4a2:2:1:0]:31354 amola.bhoite@192.168.1.203 |
| --- |

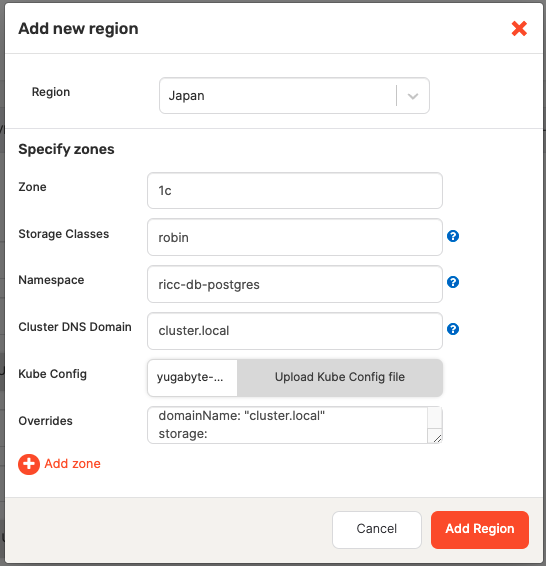
* Create an admin account via <http://localhost:32004/> The following illustration shows the admin console



* Select the Kubernetes service - In the YugabyteDB Anywhere UI, navigate to Configs > Cloud Provider Configuration > Managed Kubernetes Service and select one of the Kubernetes service providers using the Type field, as per the following illustration:



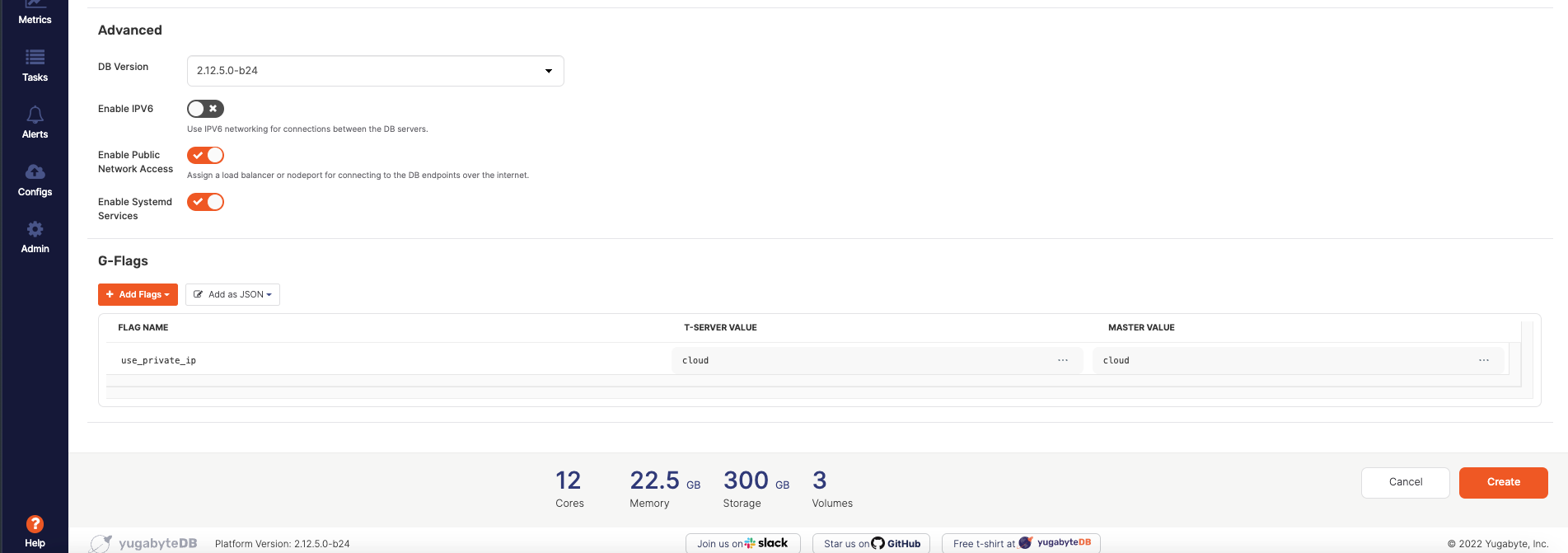
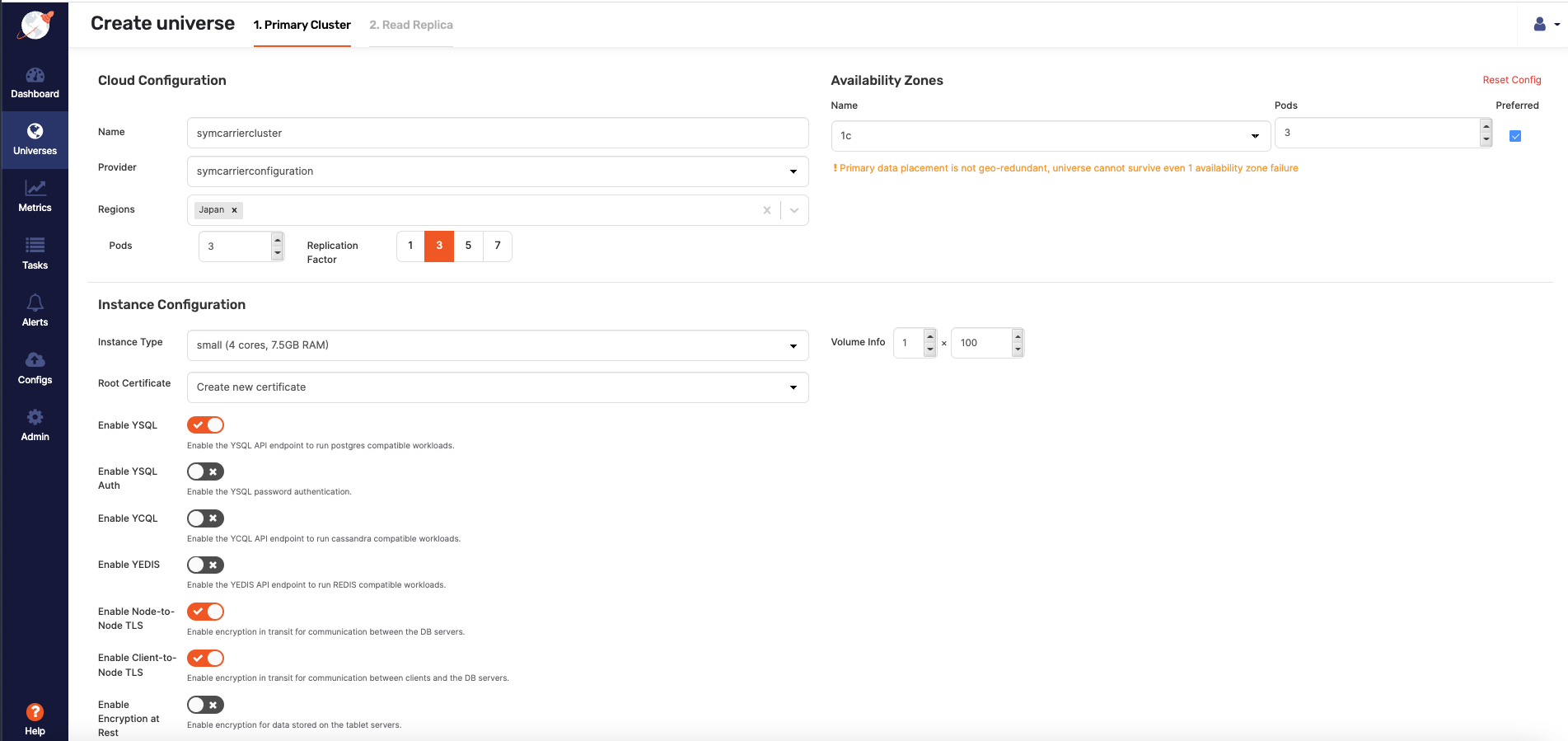
* Click on ‘Add Region’ to add the region details. Provide below details and upload yugabyte-platform-universe-management.conf as a Kube Config.



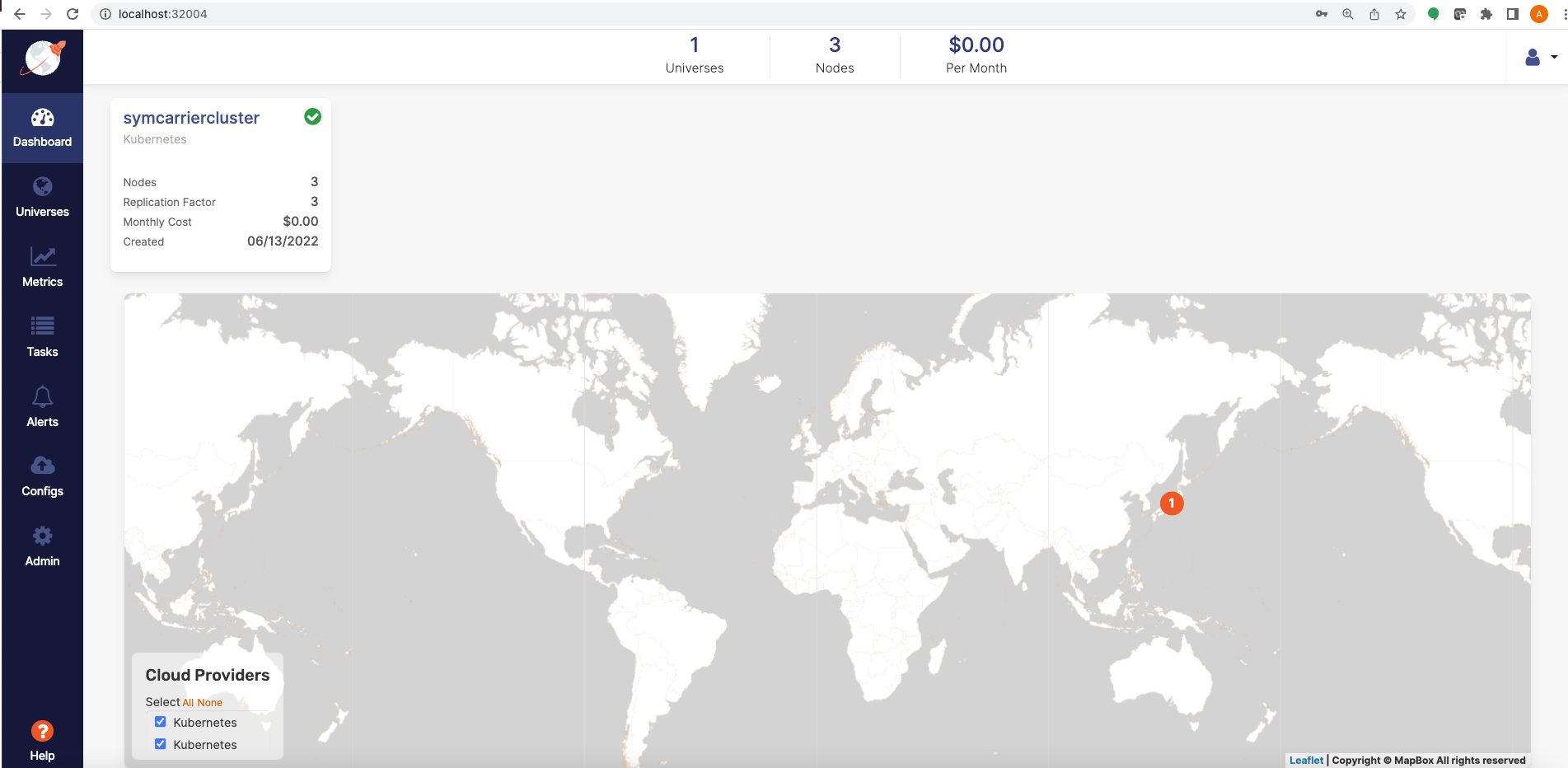
* Add below code in the Overrides section. This is basically a master pods configuration.

| domainName: "cluster.local" storage:  master:  size: 40Gi resource:  master:  requests:  cpu: 8  memory: 16Gi  limits:  cpu: 8  memory: 16Gi serviceEndpoints:  - name: "yb-master-ui"  type: NodePort  app: "yb-master"  ports:  http-ui: "7000"  - name: "yb-tserver-service"  type: NodePort  externalTrafficPolicy: Cluster  app: "yb-tserver"  ports:  tcp-yql-port: "9042"  tcp-yedis-port: "6379"  tcp-ysql-port: "5433" |
| --- |

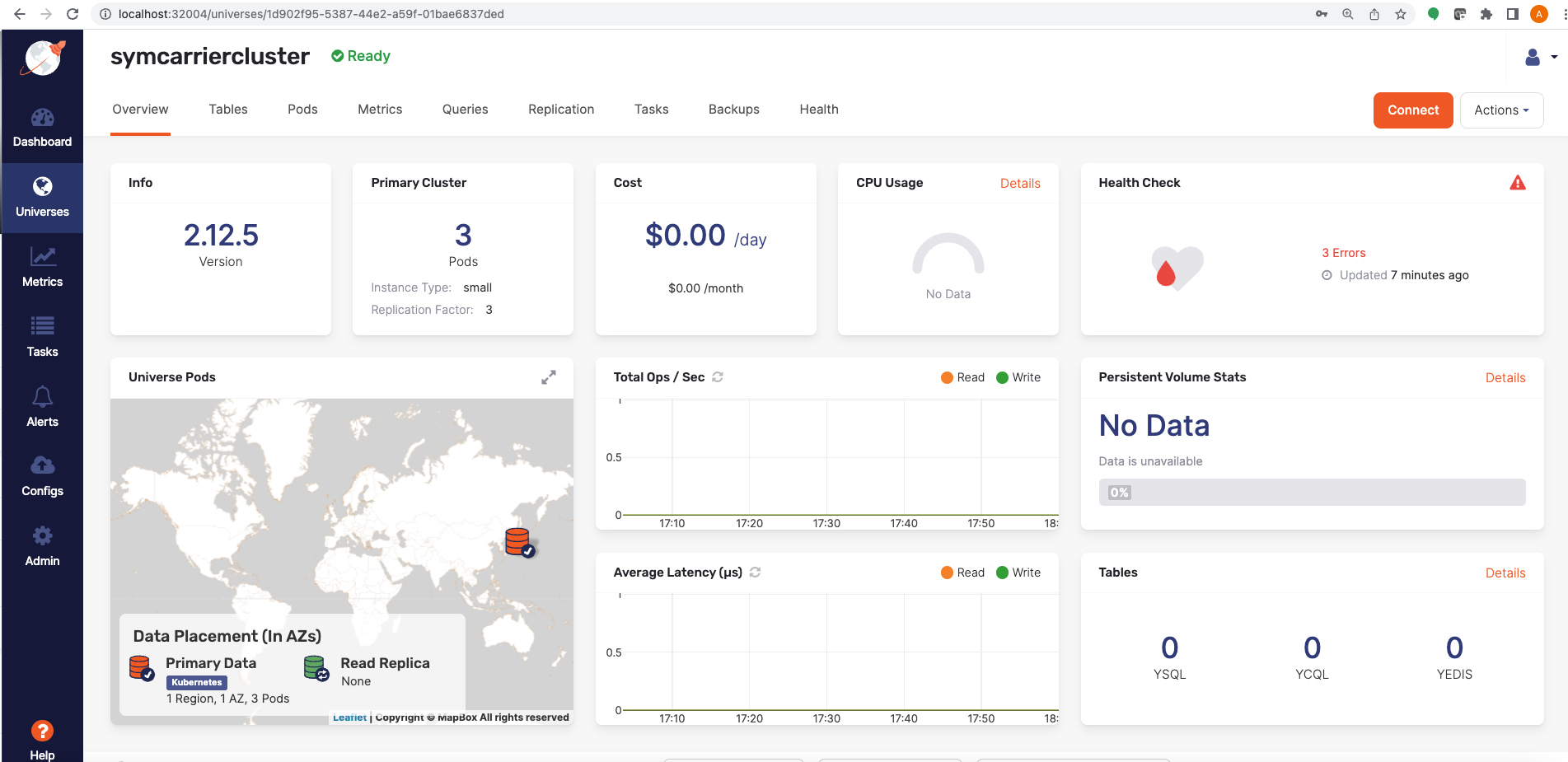
* Go in the universe and create universe with below parameters



* Created universe can be seen in the Dashboard section



* Click on the universe to see cluster details



## MinIO Backup Configuration

* Enable enablePathStyleAccess flag on http://localhost:32004/features

