Getting started

First, please clone the repository with git to get the source code of the project.

|  |
| --- |
| git clone https://github.com/tidb-incubator/Serverlessdb-for-HTAP.git |

Build

Make lvmplugin, scale-operator, sldb-operator,scheduler

|  |
| --- |
| cd ./  make |

Make webhook

|  |
| --- |
| cd ./pkg/webhook/  make |

Make tidb

|  |
| --- |
| cd ./pkg/tidb/  Make server |

Docker images prepare

If the server has no access to the Internet, you need to download all Docker images used by serverlessDB for HTAP on a machine with Internet access and upload them to the server, and then use docker load to install the Docker image on the server.

The open source Docker images used by serverlessDB for HTAP are:

|  |
| --- |
| pingcap/tidb-operator:v1.2.0  pingcap/pd:v5.1.1  pingcap/tidb:v5.1.1  pingcap/tikv:v5.1.1  tnir/mysqlclient:latest  pingcap/advanced-statefulset:v0.4.0  bitnami/kubectl:latest  lvm-provider:v1  quay.io/k8scsi/csi-attacher:v3.0.0-rc1  quay.io/k8scsi/csi-node-driver-registrar:v1.3.0  quay.io/k8scsi/csi-resizer:v0.5.0  ubuntu:16.04  prometheus:v2.24.0  jimmidyson/configmap-reload:v0.5.0  busybox:1.26.2  serverlessdb/sldb-operator:v1  serverlessdb/tidb-operator:v1  serverlessdb/lvm-scheduler:v1  serverlessdb/lvm:v1  serverlessdb/webhook:v1  serverlessdb/proxy:v1  serverlessdb/scale:v1 |

## Next, download all these images using the following command:

|  |
| --- |
| docker pull pingcap/tidb-operator:v1.2.0  docker pull pingcap/pd:v5.1.1  docker pull pingcap/tidb:v5.1.1  docker pull pingcap/tikv:v5.1.1  docker pull tnir/mysqlclient:latest  docker pull pingcap/advanced-statefulset:v0.4.0  docker pull bitnami/kubectl:latest  docker pull lvm-provider:v1  docker pull quay.io/k8scsi/csi-attacher:v3.0.0-rc1  docker pull quay.io/k8scsi/csi-node-driver-registrar:v1.3.0  docker pull quay.io/k8scsi/csi-resizer:v0.5.0  docker pull ubuntu:16.04  docker pull prom/prometheus:v2.24.0  docker pull grafana/grafana:6.1.6  docker pull jimmidyson/configmap-reload:v0.5.0  docker pull busybox:1.26.2  docker pull serverlessdb/sldb-operator:v1  docker pull serverlessdb/tidb-operator:v1  docker pull serverlessdb/lvm-scheduler:v1  docker pull serverlessdb/lvm:v1  docker pull serverlessdb/webhook:v1  docker pull serverlessdb/proxy:v1  docker pull serverlessdb/scale:v1 |

Deploy

###### Prerequisites

1. Kubernetes
2. Helm 3
3. Already create namespace csi-hostpath, tidb-admin, sldb-admin, monitoring-system

## Local disk Management

Configure before deploy

1. Already created volume group name starting with "lvm" for local disk nodes(E.g. lvmvg1)
2. Set udev\_sync=0 and udev\_rules=0 in /etc/lvm/lvm.conf
3. Modify vgName in csi-storageclass.yaml to the volume group name previously set

###### Deploy Local disk

1. Modify values.yaml for csi-hostpath：

|  |
| --- |
| $ vim csi-hostpath/values.yaml  csiAttacher:  attacherImage: "quay.io/k8scsi/csi-attacher:v3.0.0-rc1"  csiPlugin:  driverImage: "quay.io/k8scsi/csi-node-driver-registrar:v1.3.0"  lvmImage: "serverlessdb/lvm:v1"  csiProvisioner:  providerImage: "serverlessdb/lvm-provider:v1"  csiResizer:  resizerImage: "quay.io/k8scsi/csi-resizer:v0.5.0"  csiCommon:  csiHostpath: "/apps/data/kubelet/plugins/csi-hostpath"  mountpointDir: "/apps/data/kubelet/pods"  pluginsDir: "/apps/data/kubelet/plugins"  registryDir: "/apps/data/kubelet/plugins\_registry"  datadir: "/apps/data/csi-hostpath-data/"  ## affinity defines pod scheduling rules,affinity default settings is empty.  ## please read the affinity document before set your scheduling rule:  ## ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-affinity  affinity: {}  ## nodeSelector ensure pods only assigning to nodes which have each of the indicated key-value pairs as labels  ## ref:https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#nodeselector  nodeSelector: {}  ## Tolerations are applied to pods, and allow pods to schedule onto nodes with matching taints.  ## refer to https://kubernetes.io/docs/concepts/configuration/taint-and-toleration  tolerations: []  # - key: node-role  # operator: Equal  # value: sldb-operator-operator  # effect: "NoSchedule" |

1. Deploy csi-hostpath：

|  |
| --- |
| helm install csi-hostpath ./csi-hostpath --namespace=csi-hostpath |

If you need to upgrade the csi-hostpath service , execute the following command to upgrade：

|  |
| --- |
| helm upgrade csi-hostpath ./csi-hostpath --namespace=csi-hostpath |

1. Create storageclass：

|  |
| --- |
| kubectl create -f csi-storageclass.yaml |

1. Verify that the service is started：

|  |
| --- |
| helm list --namespace=csi-hostpath  kubectl get pods --namespace=csi-hostpath |

## Create CRD

TiDB and sldb Operator use Custom Resource Definition (CRD) to extend Kubernetes. Therefore, to use Operator, you must first create all CRD, which is a one-time job in your Kubernetes cluster.

Extract crds.tar and apply CRD：

|  |
| --- |
| kubectl apply -f ./crds/ |

If the following message is displayed, the CRD installation is successful:

|  |
| --- |
| kubectl get crd  NAME CREATED AT  dmclusters.pingcap.com 2021-08-23T13:03:12Z  restores.pingcap.com 2021-08-23T13:03:12Z  serverlessdbs.bcrds.cmss.com 2021-08-23T12:52:48Z  tidbclusterautoscalers.pingcap.com 2021-08-23T13:03:12Z  tidbclusters.pingcap.com 2021-08-23T13:03:12Z  tidbinitializers.pingcap.com 2021-08-23T13:03:12Z  tidbmonitors.pingcap.com 2021-08-23T13:03:12Z |

## Deploy TiDB Operator

1. Configure TiDB Operator

TiDB Operator manages all TiDB clusters in the Kubernetes cluster by default. If you only need it to manage clusters in a specific namespace, you can set clusterScoped: false in values.yaml. (Note:After setting clusterScoped: false, TiDB Operator will still operate Nodes, Persistent Volumes, and Storage Classe in the Kubernetes cluster by default. If the role that deploys TiDB Operator does not have the permissions to operate these resources, you can set the corresponding permission request under controllerManager.clusterPermissions to false to disable TiDB Operator's operations on these resources.)

You can modify other items such as limits, requests, and replicas as needed.

We use Admission webhooks to ensure users would not be affected at all when auto-scaling,after all object modifications are complete, and after the incoming object is validated by the API server, validating admission webhooks are invoked and can reject requests to enforce custom policies. Besides,the advanced StatefulSet is used to specify the location to scale out or scale in,which is implemented based on the built-in StatefulSet controller, it supports freely controlling the serial number of Pods.

1. Deploy TiDB Operator

|  |
| --- |
| # 1. install  helm install tidb-operator ./tidb-operator --namespace=tidb-admin  # 2. upgrade  helm upgrade tidb-operator ./tidb-operator --namespace=tidb-admin  # 3. stop service  helm uninstall tidb-operator --namespace=tidb-admin |

## Deploy Sldb Operator

1. Configure TiDB Operator, modify operator-configmap.yaml as needed

|  |
| --- |
| $ vim ./sldb-operator/operator-configmap.yaml  apiVersion: v1  kind: ConfigMap  metadata:  name: serverlessdb-operator  namespace: tidb-admin  labels:  name: serverlessdb-operator  data:  operator-config: |  storage:  #Name storageClassName of local disk as lvm-hostpath by default  storageClassName: "lvm-hostpath"  tidbCluster:  version: v5.1.1  pd:  requests:  cpu: "2"  memory: 4Gi  storage: 10Gi  limits:  cpu: "2"  memory: 4Gi  storage: 10Gi  replicas: 3  tidb:  requests:  cpu: "1"  memory: 2Gi  limits:  cpu: "1"  memory: 2Gi  replicas: 1  tikv:  requests:  cpu: "4"  memory: 8Gi  storage: 10Gi  ...... |

1. Deploy Sldb Operator

|  |
| --- |
| kubectl apply -f ./sldb-operator/ |

1. Verify that the service is started：

|  |
| --- |
| kubectl get pods --namespace=tidb-admin |

## Deploy Serverless

1. Configure Serverless according to ./scale-operator/scale-deploy.yaml

|  |
| --- |
| vim ./scale-operator/scale-deploy.yaml  The following explains the relevant parameters  CPU\_AVERAGE\_UTILIZATION tidb scale out will be activated when CPU usage higher than this value  TIKV\_AVERAGE\_UTILIZATION tikv scale out will be activated when remaining storage capacity lower than this value  TIKV\_MAX\_REPLIAS Maximum number of replicas of the same norm tikv  MAX\_CPU\_TYPE Maximum number of types of cpu specifications  SPLIT\_REPLICAS Maximum number of replicas of the same norm tidb  TIKV\_CHECK\_INTERVAL The required interval time between each auto-scale, if interval time is lower than this value, auto-scale will be rejected |

1. deploy

kubectl create -f ./scale-operator/

1. Verify that the service is started

kubectl get pods -n sldb-admin

## Deploy Monitoring Module

Prometheus is used to collect data such as performance monitoring indicators of Tidb Instance, and persistently store the most recent data,All components of the monitoring acquisition module are installed under the monitoring-system namespace, grafana is a visual interface.

kubectl apply -f monitoring-system.yaml

kubectl apply -f tidb-grafana.yaml

# Appendix