

TiCI On Cloud

TiDB:

Use the FTS version of TiDB; no special configuration is required.

380838443567.dkr.ecr.us-west-2.amazonaws.com/tidbcloud/tidb:feature-fts-dc593c5-next-gen

TiKV:

No change.

380838443567.dkr.ecr.us-west-2.amazonaws.com/tidbcloud/tikv:v8.5.4-nextgen.202510.10

PD/TSO/Scheduling:

No change.

380838443567.dkr.ecr.us-west-2.amazonaws.com/tidbcloud/pd:v8.5.4-nextgen.202510.4

TiFlash WN:

Not need. replicas is 0.

TiFlash CN:

tiflash cn writes logs to the root directory.

1. Use the FTS version of TiFlash CN
2. Add the following configuration

代码块

```
1  [tici]
2  [tici.reader_node]
3  port = 8520
4  [tici.s3]
5  bucket = "tidbcloud-8296a34761b610d35c4bd19c0310903a6aa676b6f5cdd1ada879e"
   # Replace with the currently used S3 bucket
6  endpoint = "http://s3.us-east-1.amazonaws.com" # Replace with the currently
   used S3 endpoint
7  region = "us-east-1" # Replace with the currently used S3 region
8  prefix = "tici"
9  [tici.security]
```

```
10 ca-path = '/var/lib/tiflash-tls/ca.crt'
11 cert-allowed_cn = ['2a4bdcde']
12 cert-path = '/var/lib/tiflash-tls/tls.crt'
13 key=path = '/var/lib/tiflash-tls/tls.key'
14 [tici.frag_reader]
15 local_data_path = "/data0/frag_local_data" # Replace the data disk with the
target environment, using a different directory from the one used by
tiflash itself.
```

TiCI Meta:

1. Deployment yaml

Tici meta replicas is always 1.

代码块

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app.kubernetes.io/component: tici-meta
6    name: db-tici-meta
7  spec:
8    replicas: 1 # only 1
9    selector:
10     matchLabels:
11       app.kubernetes.io/component: tici-meta
12    template:
13     metadata:
14       annotations:
15         cluster-autoscaler.kubernetes.io/safe-to-evict: "yes"
16         prometheus.io/path: /metrics
17         prometheus.io/port: "8501"
18         prometheus.io/scrape: "true"
19       labels:
20         app.kubernetes.io/component: tici-meta
21         pingcap.com/cluster: db
22         pingcap.com/component: tici-meta
23     spec:
24       containers:
25       - command:
26         - /tici-server
27         - meta
28         - --config
```

```
29     - /etc/tici-meta/config.toml
30     - --log-file=
31     env:
32     - name: NAMESPACE
33       valueFrom:
34         fieldRef:
35           apiVersion: v1
36           fieldPath: metadata.namespace
37     - name: CLUSTER_NAME
38       value: db
39     - name: TZ
40       value: UTC
41     - name: POD_NAME
42       valueFrom:
43         fieldRef:
44           apiVersion: v1
45           fieldPath: metadata.name
46     image: 380838443567.dkr.ecr.us-west-
2.amazonaws.com/tidbcloud/tici:master
47     imagePullPolicy: Always
48     name: tici-meta
49     ports:
50     - containerPort: 8500
51       hostPort: 8500
52       name: server
53       protocol: TCP
54     - containerPort: 8501
55       hostPort: 8501
56       name: status
57       protocol: TCP
58     readinessProbe:
59       failureThreshold: 3
60       initialDelaySeconds: 10
61       periodSeconds: 5
62       successThreshold: 1
63       tcpSocket:
64         port: 8501
65       timeoutSeconds: 3
66     securityContext:
67       allowPrivilegeEscalation: false
68       capabilities:
69         drop:
70         - NET_RAW
71       privileged: false
72       readOnlyRootFilesystem: true
73     volumeMounts:
74     - mountPath: /var/lib/tici-meta-tls
```

```

75         name: tici-meta-tls
76         readOnly: true
77     - mountPath: /etc/tici-meta
78       name: config
79       readOnly: true
80     - mountPath: /tmp
81       name: tmp-volume
82     - mountPath: /var/lib/client-ssl
83       name: client-ssl
84   dnsPolicy: ClusterFirstWithHostNet
85   hostNetwork: true
86   restartPolicy: Always
87   schedulerName: default-scheduler
88   securityContext:
89     fsGroup: 2000
90     runAsGroup: 2000
91     runAsNonRoot: true
92     runAsUser: 1000
93   terminationGracePeriodSeconds: 30
94   volumes:
95     - name: tici-meta-tls
96       secret:
97         defaultMode: 420
98         secretName: db-tici-meta-cluster-secret
99     - configMap:
100       defaultMode: 420
101       name: db-tici-meta
102       name: config
103     - emptyDir: {}
104       name: tmp-volume

```

2. Config Map

代码块

```

1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: db-tici-mate
5    labels:
6      app.kubernetes.io/component: tici-meta
7  data:
8    config.toml: |
9      [security]
10     ca-path = "/var/lib/tici-meta-tls/ca.crt"
11     cert-path = "/var/lib/tici-meta-tls/tls.crt"

```

```

12     key-path = "/var/lib/tici-meta-tls/tls.key"
13
14     [tidb_server]
15     dsns = ["mysql://root@{tidb-svc}:4000"]
16     cert_path = "/var/lib/client-ssl/ca.crt"
17
18     [server]
19     addr = "0.0.0.0:8500"
20     advertise_addr = "tici-meta-peer.svc:8500"
21     status_addr = "0.0.0.0:8501"
22     advertise_status_addr = "tici-meta-peer-peer.svc:8501"
23     pd_addr = "db-pd:2379"
24     status_server_store_path = "/tmp/worker_status_server"
25
26     [s3]
27     endpoint = "{S3Endpoint}"
28     region = "{S3Region}"
29     bucket = "{S3Bucket}"
30     prefix = "{S3Prefix}"
31     use_path_style = true
32
33     [shard]
34     max_size = "4096MB"
35     split_threshold = 0.75
36
37     [import]
38     max_concurrency = 256

```

3. Service

代码块

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app.kubernetes.io/component: tici-meta
6      name: db-tici-meta-peer
7  spec:
8    ports:
9      - name: server
10        port: 8500
11        protocol: TCP
12        targetPort: server
13      - name: status
14        port: 8501

```

```
15     protocol: TCP
16     targetPort: status
17     selector:
18       app.kubernetes.io/component: tici-meta
19     sessionAffinity: None
20     type: ClusterIP
```

TiCI Worker:

1. StatefulSet

代码块

```
1  apiVersion: apps/v1
2  kind: StatefulSet
3  metadata:
4    annotations:
5    labels:
6      app.kubernetes.io/component: tici-worker
7    name: db-tici-worker
8  spec:
9    persistentVolumeClaimRetentionPolicy:
10      whenDeleted: Delete
11      whenScaled: Delete
12    replicas: 2
13    selector:
14      matchLabels:
15        app.kubernetes.io/component: tici-worker
16    serviceName: db-worker-peer
17    template:
18      metadata:
19        annotations:
20          cluster-autoscaler.kubernetes.io/safe-to-evict: "yes"
21          prometheus.io/path: /metrics
22          prometheus.io/port: "8511"
23          prometheus.io/scrape: "true"
24        labels:
25          app.kubernetes.io/component: tici-worker
26      spec:
27        containers:
28          - command:
29              - /bin/sh
30              - -c
31              - |
32                set -ex
```

```
33     sed "s/POD_NAME/${POD_NAME}/g" /etc/tici-worker/config.toml >
/tmp/config.toml
34     cat /tmp/config.toml
35     exec /tici-server worker --config /tmp/config.toml --log-file=
36     env:
37     - name: NAMESPACE
38       valueFrom:
39         fieldRef:
40           apiVersion: v1
41           fieldPath: metadata.namespace
42     - name: CLUSTER_NAME
43       value: db
44     - name: TZ
45       value: UTC
46     - name: POD_NAME
47       valueFrom:
48         fieldRef:
49           apiVersion: v1
50           fieldPath: metadata.name
51     image: 380838443567.dkr.ecr.us-west-
2.amazonaws.com/tidbcloud/tici:master
52     imagePullPolicy: Always
53     name: tici-worker
54     ports:
55     - containerPort: 8510
56       hostPort: 8510
57       name: server
58       protocol: TCP
59     - containerPort: 8511
60       hostPort: 8511
61       name: status
62       protocol: TCP
63     readinessProbe:
64       failureThreshold: 3
65       initialDelaySeconds: 10
66       periodSeconds: 5
67       successThreshold: 1
68       tcpSocket:
69         port: 8511
70       timeoutSeconds: 3
71     securityContext:
72       allowPrivilegeEscalation: true
73       capabilities:
74         drop:
75         - NET_RAW
76       privileged: true
77       readOnlyRootFilesystem: false
```

```
78     terminationMessagePath: /dev/termination-log
79     terminationMessagePolicy: File
80     volumeMounts:
81     - mountPath: /var/lib/tici-worker-tls
82       name: tici-worker-tls
83       readOnly: true
84     - mountPath: /etc/tici-worker
85       name: config
86       readOnly: true
87     - mountPath: /tmp
88       name: tmp-volume
89     - mountPath: /var/lib/client-ssl
90       name: client-ssl
91     - mountPath: /var/lib/tici-worker
92       name: data
93     dnsPolicy: ClusterFirstWithHostNet
94     hostNetwork: true
95     restartPolicy: Always
96     securityContext: {}
97     terminationGracePeriodSeconds: 30
98     volumes:
99     - name: tici-worker-tls
100       secret:
101         defaultMode: 420
102         secretName: db-worker-cluster-secret
103     - configMap:
104         defaultMode: 420
105         name: db-tici-worker
106       name: config
107     - emptyDir: {}
108       name: tmp-volume
109     - name: client-ssl
110       secret:
111         defaultMode: 420
112         secretName: db-cluster-client-secret
113     updateStrategy:
114       rollingUpdate:
115         partition: 0
116       type: RollingUpdate
117     volumeClaimTemplates:
118     - apiVersion: v1
119       kind: PersistentVolumeClaim
120       metadata:
121         name: data
122       spec:
123         accessModes:
124         - ReadWriteOnce
```

```
125     resources:
126         requests:
127             storage: 500Gi
128     storageClassName: {sc_name}
129     volumeMode: Filesystem
```

2. Config Map

代码块

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4      name: db-tici-worker
5      labels:
6          app.kubernetes.io/component: tici-worker
7  data:
8      config.toml: |
9          [security]
10             ca-path = "/var/lib/tici-worker-tls/ca.crt"
11             cert-path = "/var/lib/tici-worker-tls/tls.crt"
12             key-path = "/var/lib/tici-worker-tls/tls.key"
13
14             [server]
15             addr = "0.0.0.0:8510"
16             advertise_addr = "POD_NAME.db-tici-worker-peer.tidb.svc:8510"
17             status_addr = "0.0.0.0:8511"
18             advertise_status_addr = "POD_NAME.db-tici-worker-peer.tidb.svc:8511"
19             pd_addr = "db-pd:2379"
20             status_server_store_path = "/tmp/worker_status_server"
21
22             [s3]
23             endpoint = "{S3Endpoint}"
24             region = "{S3Region}"
25             use_path_style = true
26             bucket = "{S3Bucket}"
27             prefix = "{S3Prefix}"
28
29             [frag_writer]
30             local_data_path = "/var/lib/tici-worker/data"
31
```

3. Service

代码块

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app.kubernetes.io/component: tici-worker
6      name: db-tici-worker-peer
7  spec:
8    ports:
9      - name: server
10       port: 8510
11       protocol: TCP
12       targetPort: server
13      - name: status
14       port: 8511
15       protocol: TCP
16       targetPort: status
17    selector:
18      app.kubernetes.io/component: tici-worker
19    sessionAffinity: None
20    type: ClusterIP

```

TiCDC Meta:

1. Create Changefeed

After the FTS index is created, synchronize incremental data updates to the index for use.

代码块

```

1  /cdc cli changefeed create --ca /var/lib/ticdc-tls/ca.crt --key
   /var/lib/ticdc-tls/tls.key --cert /var/lib/ticdc-tls/tls.crt -k {keyspace} --
   sink-uri='s3://{bucket}/tici/cdc?protocol=canal-json&enable-tidb-
extension=true&output-row-key=true' # The suffix cdc must be used

```