

Tungsten Operator Master Class

Intermediate: Advanced Configurations

Presented by Chris Parker

VP of Customer Success, EMEA & APAC

Topics

In this short course, we will:

- Review YAML customizations
- Learn how to configure backups
- Learn how to initiate restores
- Learn how to import data

YAML Customizations

- Enable/disable monitoring
- Adjust topology type
- Configure connectors
- Adjust pod configuration
- Adjust storage
- Custom tungsten.ini settings

Enable/Disable Monitoring

Enables Prometheus exporters

```
. . .  
monitoring:  
  enabled: false  
. . .
```

```
. . .  
monitoring:  
  enabled: true  
. . .
```

Topology Types

Available Topologies

- clustered
- composite-active-passive
- composite-active-active

```
. . .  
dataServices:  
  - name: alpha  
    topology: clustered  
    size: 3  
. . .
```

```
. . .  
dataServices:  
  - name: global  
    topology: composite-active-passive  
    sources:  
      - east  
      - west  
  - name: east  
    topology: clustered  
    size: 3  
  - name: west  
    topology: clustered  
    sources:  
      - east  
    size: 3  
. . .
```

Connector

- Define number of connectors
 - For CAP/CAA:
 $\text{replicas} \times \text{number of clusters} = \text{number of connectors}$
- Define resources

```
. . .  
connector:  
  replicas: 2  
  podTemplate:  
    spec:  
      containers:  
        - name: connector  
          resources:  
            requests:  
              cpu: 1  
              memory: 512Mi  
            limits:  
              memory: 512Mi  
. . .
```

Pod Configuration

- Configuration of individual pods for
 - MySQL
 - Tungsten Manager
 - Tungsten Replicator
- name values should not be changed

```
. . .
podTemplate:
  spec:
    containers:
      - name: mysql
        resources:
          requests:
            cpu: 1
            memory: 2Gi
          limits:
            memory: 2Gi
      - name: manager
        resources:
          requests:
            cpu: 100m
            memory: 512Mi
          limits:
            memory: 512Mi
      - name: replicator
        resources:
          requests:
            cpu: 1
            memory: 2Gi
          limits:
            memory: 2Gi
. . .
```

Storage

- Configuration of storage
 - ONLY adjust storage values

```
. . .
volumeClaimTemplates:
- metadata:
  name: mysql-data
  spec:
    accessModes: [ "ReadWriteOnce" ]
    resources:
      requests:
        storage: 40Gi
- metadata:
  name: th1
  spec:
    accessModes: [ "ReadWriteOnce" ]
    resources:
      requests:
        storage: 20Gi
- metadata:
  name: state
  spec:
    accessModes: [ "ReadWriteOnce" ]
    resources:
      requests:
        storage: 100Mi
. . .
```

Custom tungsten.ini Settings

- Equivalent to setting custom properties in tungsten.ini
- Name/value key pairs within the YAML file
- Include section
customTungstenIniProperties
- configuration = ini [defaults] stanza
- dataServices = ini [serviceName] stanza
- See Tungsten Cluster docs for all ini properties

```
...
spec:
  configuration:
    customPasswordSecret: custom-tungsten-passwords-sample
    customTungstenIniProperties:
      - name: connector-bridge-mode
        value: "true"
      - name: repl-buffer-size
        value: "10"
  dataServices:
    - name: composite
      topology: composite-active-passive
      sources:
        - east
        - west
    - name: east
      topology: clustered
      size: 3
      customTungstenIniProperties:
        - name: connector-disconnect-timeout
          value: "10"
        - name: property=manager.prometheus.exporter.enabled
          value: "true"
  ...
```

Configure Backups

- Two custom resources:
 - TungstenBackup
 - Represents single backup
 - Source cluster
 - Status & size
 - BackupConfiguration
 - Backup policy
 - Storage destination
 - Backend configuration
 - rclone only at present

```
apiVersion: tungsten.continuent.com/v1alpha1
kind: BackupConfiguration
metadata:
  labels:
    app.kubernetes.io/name: backupconfiguration
    app.kubernetes.io/instance: backupconfiguration-sample
    app.kubernetes.io/part-of: tungsten-operator
    app.kubernetes.io/managed-by: kustomize
    app.kubernetes.io/created-by: tungsten-operator
    name: backupconfiguration-sample
spec:
  backend: sample-rclone-backend
  backendConfigurations:
    - name: sample-rclone-backend
      type: rclone
      rclone:
        type: mysqldump
        destination:
          bucket: "backups"
          directory: "test"
        flags:
          purgeRemoteBackupOnDeletion: true
          overwriteExistingPhysicalResource: false
      env:
        - name: RCLONE_CONFIG_REMOTE_TYPE
          value: s3
        - name: RCLONE_CONFIG_REMOTE_ENDPOINT
          value: "http://minio.default.svc.cluster.local"
        - name: RCLONE_CONFIG_REMOTE_ACCESS_KEY_ID
          valueFrom:
            secretKeyRef:
              name: backups-user-1
              key: CONSOLE_ACCESS_KEY
        - name: RCLONE_CONFIG_REMOTE_SECRET_ACCESS_KEY
          valueFrom:
            secretKeyRef:
              name: backups-user-1
              key: CONSOLE_SECRET_KEY
```

Scheduling Backups

- Defined via the MySQLCluster resource
- Cron format
- Retention can also be specified
- View TungstenBackup resource status field
 - Pending
 - Processing
 - Succeeded
 - Failed

```
...
spec:
  configuration:
    customPasswordSecret: tungsten-passwords-sample-secret
  backup:
    configurationRef:
      name: tungstenbackupconfiguration-sample
    schedule: "* * * * *"
    keep: 5
...
```

Restoring Backups

- Two Approaches:
 - New cluster and initialize with backup
 - Or use kubectl

Restoring Backups

New Cluster and Initialise With Backup

```
shell> kubectl tungsten backup list
```

```
...  
  name: sample-from-backup  
spec:  
  configuration:  
    customPasswordSecret: tungsten-passwords-sample-secret  
  backup:  
    configurationRef:  
      name: backupconfiguration-sample  
  init:  
    fromBackup:  
      name: tungstenbackup-sample  
  dataServices:  
  - name: alpha  
...
```

```
...  
  name: backupconfiguration-sample  
spec:  
  destination:  
    bucket: "backups"  
    directory: "test"  
    backend: sample-rclone-backend  
  policy:  
    purgeRemoteBackupOnDeletion: true  
    overwriteExistingPhysicalResource: false  
...
```

Restoring Backups

Use kubectl

- Reset using latest

```
shell> kubectl tungsten cluster reset ClusterName DataSourceName
```

- or, obtain list of backups and reset with specific backup

```
shell> kubectl tungsten backup list
```

```
kubectl tungsten cluster reset ClusterName DataSourceName --backup MyBackupName
```

Import Data

- Import SQL or use a URL
- Configure init script (mysql-init-script.yaml as example)
- Make note of the name value

```
...
name: sample-with-init
spec:
  configuration:
    customPasswordSecret: tungsten-passwords-sample-secret
  init:
    fromVolume:
      configMap:
        name: mysql-init-script
  dataServices:
  - name: alpha
...
```

```
shell> kubectl apply -f mysql-init-script.yaml
shell> kubectl apply -f tungsten_v1alpha1_mysqlcluster_with_init.yaml
```

SQL Example

```
...
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-init-script
data:
  import.sql: |
    create database sample;
    use sample;
    CREATE TABLE test(data VARCHAR(255));
    insert into test values("testing");
...
```

URL Example

```
...
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-init-script
spec:
  init:
    fromExternalArtifact:
      url: https://test.com/sample.sql.gz
...
```

Summary

In this short course, we've:

- Reviewed YAML customizations
- Learnt how to configure backups
- Learnt how to initiate restores
- Learnt how to import data

Thank you for listening

continuent.com

Presented by Chris Parker

VP of Customer Success, EMEA