

Switching to Mysql from Sqlite

I constantly had issues of sqlite DB locks just "occurring" even when no traffic to the app was happening. This happened on NFS, Root NFS, and iSCSI block stores alike.

In an attempt to mitigate, I added configurations to change Ombi to MariaDB instead.

I simply added a mariadb sidecar, and added a manually created database.json to the /config directory of the ombi container like so:

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  labels:
    run: ombi
  name: ombi
  namespace: media
spec:
  podManagementPolicy: OrderedReady
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      run: ombi
  serviceName: ombi
  template:
    metadata:
      creationTimestamp: null
      labels:
        run: ombi
    spec:
      containers:
        - name: db
          image: linuxserver/mariadb
          env:
```

- name: TZ
value: America/Los_Angeles
- name: PUID
value: "1000"
- name: PGID
value: "1000"
- name: MYSQL_DATABASE
value: "ombi"
- name: MYSQL_USER
value: "ombi"
- name: MYSQL_PASSWORD
value: "xxxxxx"
- name: MYSQL_ROOT_PASSWORD
value: "xxxxxx"

volumeMounts:

- mountPath: /config
name: dbconfig

- env:

- name: TZ
value: America/Los_Angeles
- name: PUID
value: "1000"
- name: PGID
value: "1000"

image: linuxserver/ombi

imagePullPolicy: IfNotPresent

name: ombi

ports:

- containerPort: 3579
name: http
protocol: TCP

resources: {}

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

volumeMounts:

- mountPath: /config
name: config

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

```
    securityContext: {}
    terminationGracePeriodSeconds: 30
updateStrategy:
  rollingUpdate:
    partition: 0
    type: RollingUpdate
volumeClaimTemplates:
- apiVersion: v1
  kind: PersistentVolumeClaim
  metadata:
    creationTimestamp: null
    name: config
  spec:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: 2Gi
    storageClassName: gp2
    volumeMode: Filesystem
- apiVersion: v1
  kind: PersistentVolumeClaim
  metadata:
    creationTimestamp: null
    name: dbconfig
  spec:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: 20Gi
    storageClassName: gp2
    volumeMode: Filesystem
---
apiVersion: v1
kind: Service
metadata:
  labels:
    run: ombi
  name: ombi
```

```
namespace: media
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 3579
  selector:
    run: ombi
  type: ClusterIP
---
apiVersion: getambassador.io/v2
kind: Mapping
metadata:
  name: ombi
  namespace: media
spec:
  connect_timeout_ms: 10000
  host: ombi.almueti.com
  idle_timeout_ms: 5000000
  prefix: /
  service: ombi.media:80
  timeout_ms: 10000
  use_websocket: true
```

```
{
  "OmbiDatabase": {
    "Type": "MySQL",
    "ConnectionString": "Server=127.0.0.1;Database=ombi;User=ombi;Password=xxxxx;"
  },
  "SettingsDatabase": {
    "Type": "MySQL",
    "ConnectionString": "Server=127.0.0.1;Database=ombi;User=ombi;Password=xxxxx;"
  },
  "ExternalDatabase": {
    "Type": "MySQL",
    "ConnectionString": "Server=127.0.0.1;Database=ombi;User=ombi;Password=xxxxx;"
  }
}
```

Sources:

<https://github.com/tidusjar/Ombi/wiki/Using-a-different-database>

Revision #2

Created 1 August 2020 22:38:31 by Tokugero

Updated 1 August 2020 23:01:54 by Tokugero