

# 用Kubernetes搭建便携式开发环境之MongoDB

原创 KevinYan11 网管叨bi叨 7月15日

收录于话题

#Kubernetes学习笔记

27个

## 前言

这已经是我们用Kubernetes搭建便携式开发环境的第三篇文章了，前两篇文章我们分别在本地的Kubernetes集群上做了什么呢？第一篇文章我们在搭建了一个Etcd集群，由于Etcd没有什么好的管理客户端还搭建了一个Etcd的Web UI客户端。第二篇文章我们搭建了一个单点的MySQL服务和Redis服务，如果想不起来的同学可以翻看前面的两篇文章：

[用Kubernetes搭建便携式开发环境之MySQL和Redis](#)

[用Kubernetes搭建Etcd集群和WebUI](#)

那么有的同学就私信问啦，我想搭建一个 **MongoDB** 该怎么弄啊？其实如果前面搭建MySQL和Redis的文章看懂了，按照同样的思路搭建一个 **MongoDB** 环境也不是什么难事，凑巧之前有个用Spring写的服务也用了MongoDB，今天我就带大家快速搭建一个开发环境用的单点 **MongoDB** 服务。在应用过程中我们也会趁这个机会介绍一下 Kubernetes 的 Secret 应该怎么使用。

今天文章里使用的案例我已经上传到Github上我整理的Kubernetes常用YAML的仓库里了，大家点击阅读原文或者直接访问<https://github.com/kevinyan815/LearningKubernetes> 后下载使用。

## 声明MongoDB资源

### 定义Secret

我们先为MongoDB分配一个具有Root权限的账户和相应的密码，**Kubernetes专门有一种资源叫做Secret，用来解决密码、token、密钥等敏感数据的配置问题，而不需要把这些敏感数据暴露PodTemplate的Spec信息中。**Secret也分三种类型，今天我们使用的是Opaque类型的Secret，它以base64编码格式存储密码、密钥等信息。

```
# 文件名 mongo-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
type: Opaque
data:
  # echo -n 'username' | base64
  mongo-root-username: dXNlcm5hbWU=
  # echo -n 'password' | base64
  mongo-root-password: cGFzc3dvcmQ=
```

这里我把Root用户的名称和密码分别定义成了 `username` 和 `password`，有点蠢，纯属是为了好理解用。你可以自己通过在命令行里执行 `base64` 命令，得到想要的字符串的base64编码后的字符序列。比如像下面这样获得字符串root用base64编码后的结果。

```
echo -n 'root' | base64  
cm9vdA==
```

## 定义MongoDB应用

Secret和ConfigMap在使用上有点类似，也可以把配置项直接应用到Pod模板的环境变量定义里，如果说ConfigMap是Kubernetes在用管理对象的方式管理配置，那么Secret就相当于是在用管理对象的方式管理密钥之类的敏感信息。

关于ConfigMap的详细介绍，可以参考以前的文章：[ConfigMap用管理对象的方式管理配置](#)。

完整的MongoDB应用的资源定义如下：

```
# 文件名 deployment-service.yaml  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: mongodb-deployment  
  labels:  
    app: mongodb  
spec:  
  replicas: 1
```

```
selector:
  matchLabels:
    app: mongodb
template:
  metadata:
    labels:
      app: mongodb
  spec:
    containers:
      - name: mongodb
        image: mongo
        ports:
          - containerPort: 27017
        env:
          - name: MONGO_INITDB_ROOT_USERNAME
            valueFrom:
              secretKeyRef:
                name: mongodb-secret
                key: mongo-root-username
          - name: MONGO_INITDB_ROOT_PASSWORD
            valueFrom:
              secretKeyRef:
                name: mongodb-secret
                key: mongo-root-password
        volumeMounts:
          - name: mongodb-storage
            mountPath: /data/db
```

```
volumes:
  - name: mongodb-storage
    emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  type: NodePort
  selector:
    app: mongodb
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
      nodePort: 30017
```

除了应用Pod的定义之外，我把像集群外暴露Mongo服务的Service也放在了同一个YAML定义文件里，我们指定了在集群外部可以通过30017 这个端口访问到Mongo服务的监听的27017端口。

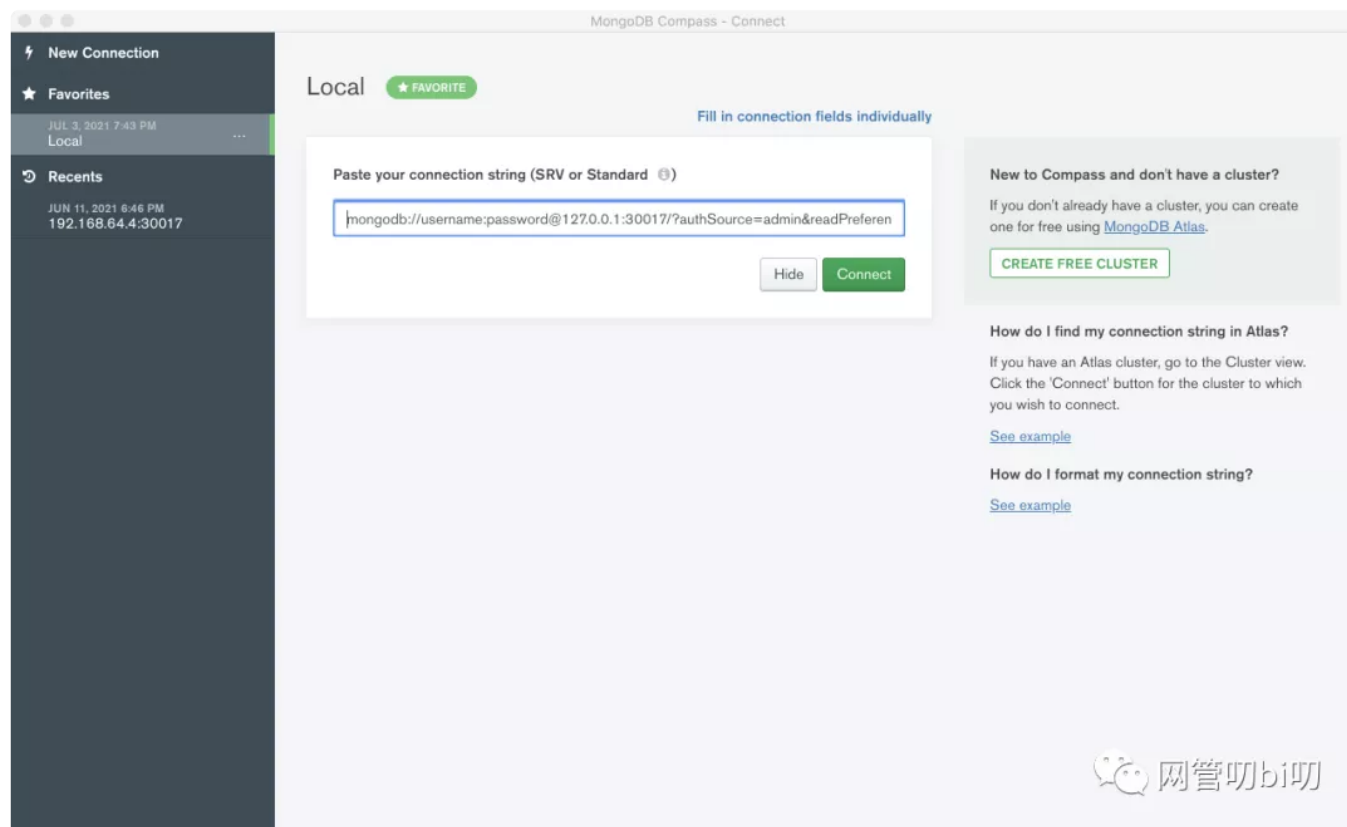
## 创建MongoDB

聊明白了MongoDB的资源定义后，创建MongoDB还是用我们一直在使用的 `kubectl apply -f` 命令，把资源定义提交给 Kubernetes 的 ApiServer，调度器就会自动帮我们创建好这些资源。

```
# 切到mongo yaml所在的目录
kubectl apply -f mongo-secret.yaml

kubectl apply -f deployment-service.yaml
```

等创建完成后，我们可以使用MongoDB Compass 这个客户端，尝试连接一下。



图片

连接信息设置成下面这个，就能连接上我们刚刚创建的MongoDB。

```
mongodb://username:password@127.0.0.1:30017/?authSource=admin&readPreference=primary&appName=MongoDB%20Compass&ssl=false
```

## 在程序里连接MongoDB

在这里，我还想跟大家拓展一下怎么在MongoDB里创建业务用的DB和响应的用户名密码，以及怎么在Java和Go的项目里连接上MongoDB。

### 创建业务DB

当我们连接上创建的MongoDB时，它只有默认的三个系统自带的db，一般我们的项目程序在用到MongoDB存储数据时会跟 MySQL一样创建一个db。

```
> use my-database

> db.createUser(
  {
    user: "my-user",
    pwd: "passw0rd",
```

```
roles: [ { role: "readWrite", db: "my-database" } ]  
}  
)
```

通过上面这两个命令我就在MongoDB里创建了一个名为 `my-database` 的 db，为这个db分配了一个可以读写的用户 `my-user`，密码是 `passwd`。

## 在SpringBoot项目里连接MongoDB

如果你使用的是用SpringBoot做自动配置的Java项目的话，要连接MongoDB只需要在POM文件里引入 `spring-boot-starter-data-mongodb` 这个依赖

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-mongodb</artifactId>  
</dependency>
```

在application.properties 文件里加上

```
spring.data.mongodb.uri=mongodb://my-user:passwd@127.0.0.1:30017/my-database
```

SpringBoot在项目启动时就会自动帮我们连接上MongoDB。



## 在Go项目里连接MongoDB

而如果你使用的是Golang开发的项目的话，则需要引入 [go.mongodb.org](https://go.mongodb.org/mongo-driver/) 下的几个包

```
import (  
    "time"  
    "context"  
    "fmt"  
  
    "go.mongodb.org/mongo-driver/mongo"  
    "go.mongodb.org/mongo-driver/mongo/options"  
)  
  
var (  
    MongoClient *mongo.Client  
    mongoURI = "mongodb://my-user:passwd@127.0.0.1:30017/my-database"  
)  
  
func init() {  
    ctx, _ := context.WithTimeout(context.Background(), 5*time.Second)  
    client, err := mongo.Connect(ctx,  
        options.Client().ApplyURI(mongoURI),  
    )  
    MongoClient = client  
}
```

当然SpringBoot和Go连接MongoDB时都还有不少细分的连接参数，这个已经超出了我们这篇文章内容的范围了，就不展开往下说了。

## 总结

今天通过实践在 Kubernetes 上安装一个单点的 MongoDB 服务，我们穿插的介绍了一下 Kubernetes 是怎么通过 Secret 管理密钥之类的敏感配置项的，通过这种实践中学习的方式能让大家更快地接受新知识。捎带着我们还扩展了一下在使用 SpringBoot 或者 Golang 的项目里怎么去连接 MongoDB，希望大家能喜欢今天的文章。



**网管叨bi叨**

分享软件开发和系统架构设计基础、Go 语言和Kubernetes。

174篇原创内容

公众号

收录于话题 #Kubernetes学习笔记·27个

上一篇

你知道K8S暴露服务的方式有哪些吗？

下一篇

试试Docker桌面应用自带的K8s集群，一个字“简单”

阅读原文

喜欢此内容的人还喜欢

分享一个Go项目的整洁架构模板

网管叨bi叨

---

【方向盘】 版本历史&代码示例之： JavaMail、JDBC

Java方向盘