

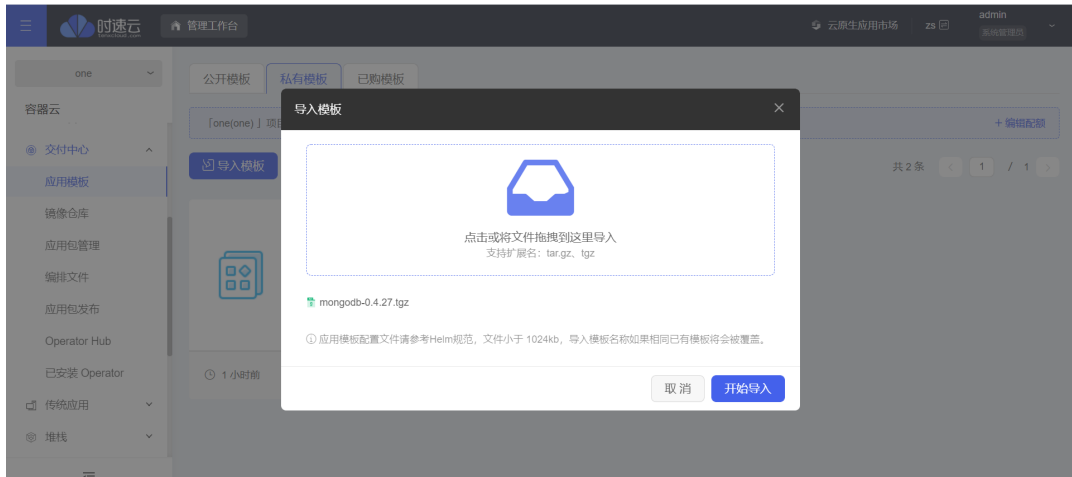
Mongodb Helm资产

一、资产基本介绍

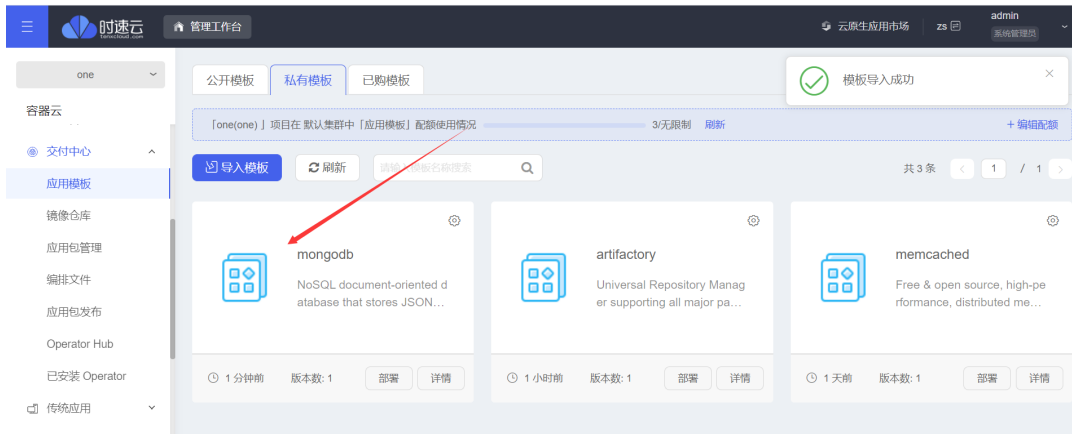
三、资产注册流程

- 导入私有模板资产

- 导入



- 成功



- 登录到平台管理后台

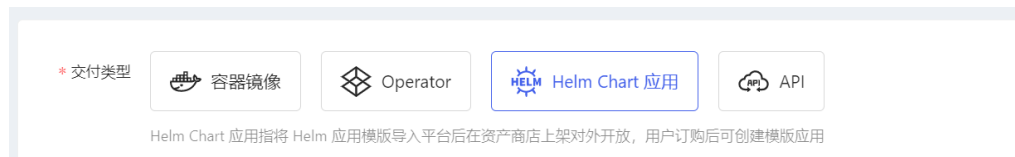
- 在左侧菜单栏找到 **资产管理**

- 点击添加资产



o 添加产基础信息

- 交付类型选择有四种，选择Helm Chart应用



- helm描述与icon信息

* 资产分类 基础应用 网站建设 企业应用 数据库中间件集群 API 服务 物联网 人工智能

* 资产名称

* 概述

* 描述



资产图标尺寸为 96×96 px，格式支持 JPG 和 PNG

- 设置规格，定价

定价支持按n年, n月计算

规格与定价

* 规格与定价	商品规格名称	价格	有效时长	操作
	<input type="text" value="1"/>	<input type="text" value="1"/> T	30 日	<input type="button" value="删除"/>

[+ 添加规格](#)

- 其他信息

其他信息

服务商网站

使用帮助
支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

使用条款
支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

客户案例
支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

* 资产截图  
建议图片 264 * 146px, 格式支持 JPG 和 PNG, 最多上传 2 张

平台支持范围
支持 Markdown 编辑模式和外部链接模式 (直接输入链接地址)

• 资产添加Helm

- 在资产管理列表位置，找到新增资产

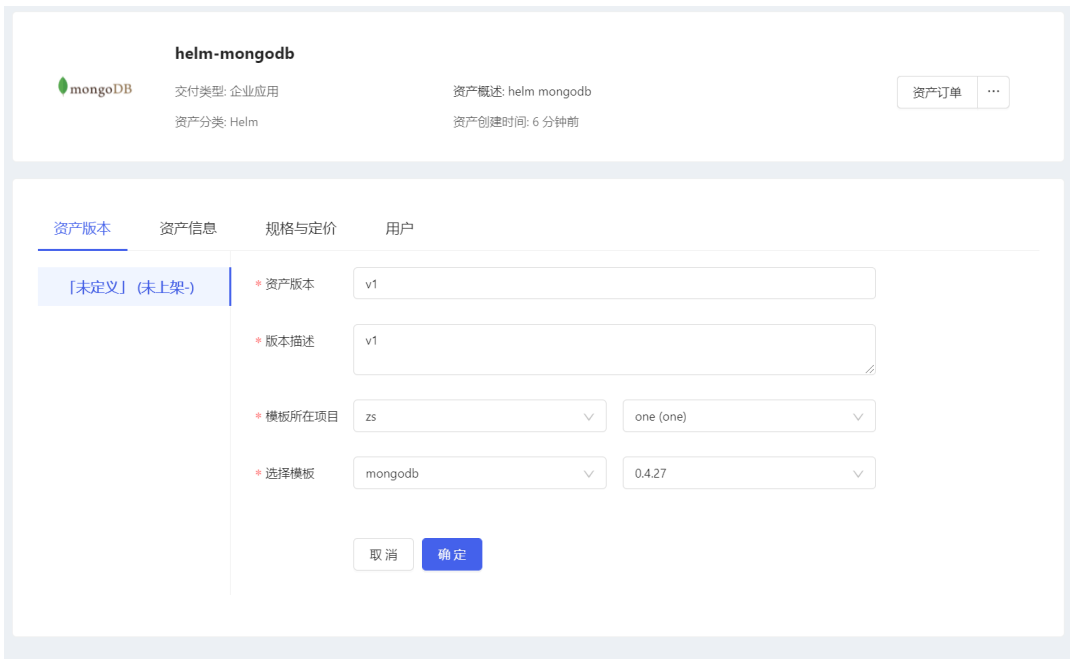


资产列表: 支持您接入容器镜像、Helm模版、Operator、API等介质作为资产, 可上架至市场中供购买者订购

+ 添加资产 请输入资产名称搜索
共计 4 条 < 1 / 1 >

资产名称	资产编码	资产版本/API	交付类型	资产分类	资产版本状态	审批状态	上架时间	操作
helm-mongodb	ASID-...	0 个	Helm	企业应用	-	-	-	创建版本 ...

- 点击进入详情，添加helm定义



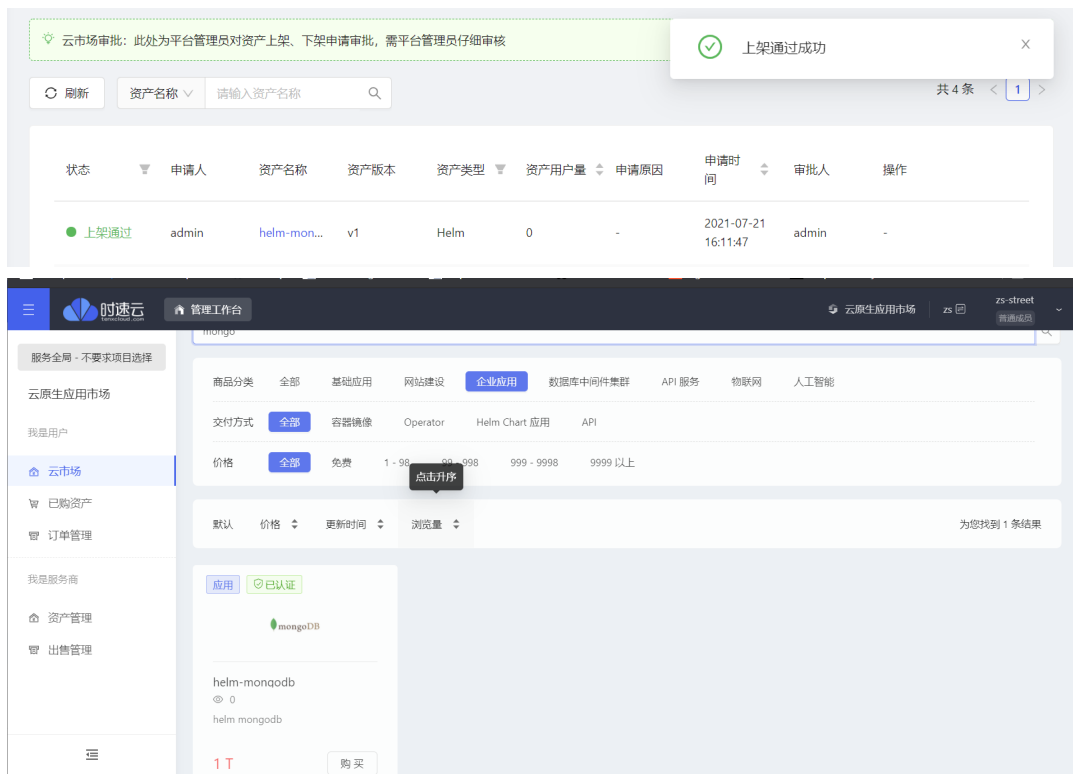
- 添加一个版本后，需要上架，可以在市场进行售卖



- 审批

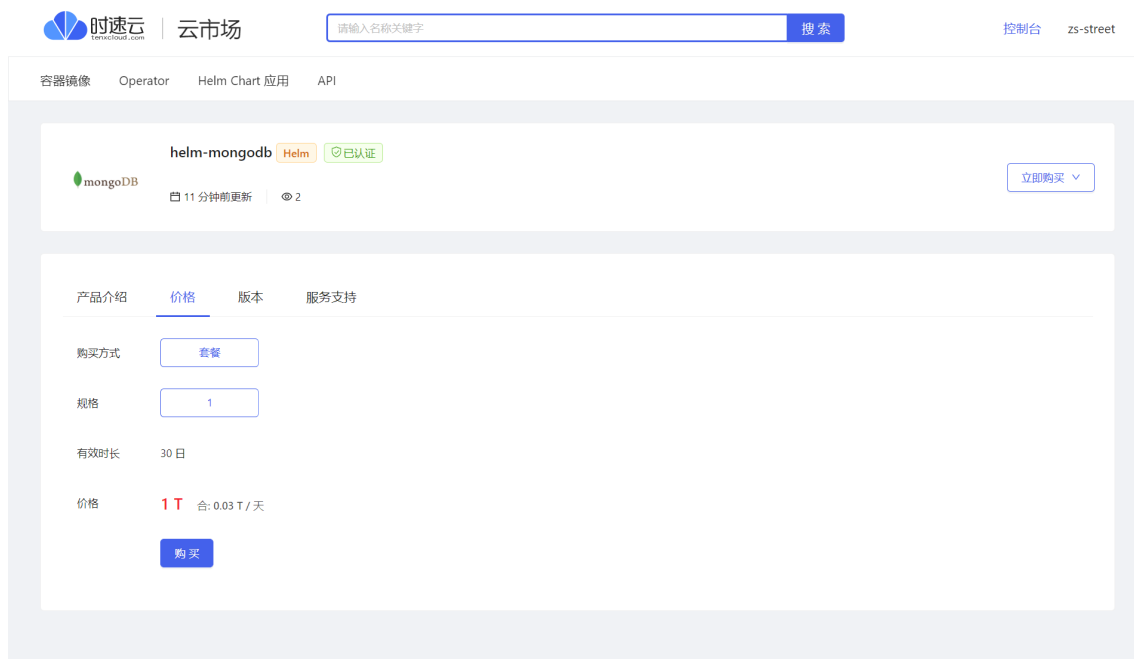
状态	申请人	资产名称	资产版本	资产类型	资产用户量	申请原因	申请时间	审批人	操作
● 上架待审核	admin	helm-mon...	v1	Helm	0	-	2021-07-21 16:11:47	-	通过 拒绝

审批通过



四、购买资产

- 购买 helm-mem 这个资产



- 支付

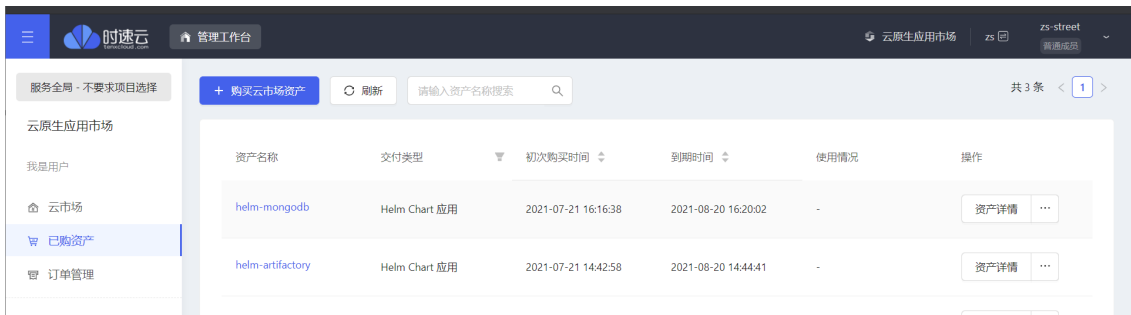
支付有两种方式，购买的时候直接付款，或者先生成订单，在一定时间范围内，继续支付。



• 完成支付



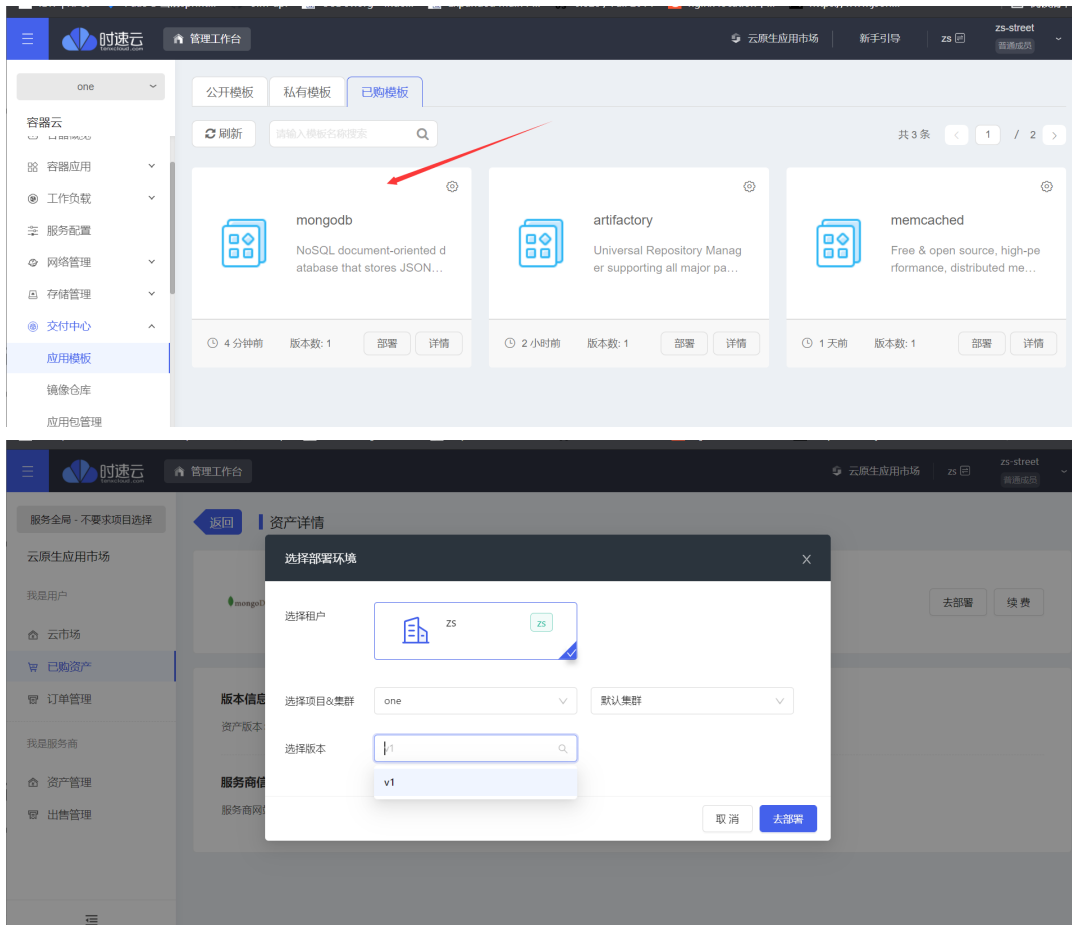
• 查看已购资产



五、部署

- 从已购资产位置部署

- 找到已购资产，点击部署，或者从资产商店 已购资产详情 点击部署



- 填写部署名称

返回 | 部署模板

模板名称:

模板版本:

* 集群:

发布名称:

描述:

配置文件:

修改 `values.yaml`，给mongodb配置存储,也可以不做任何修改，直接部署。

```
## Bitnami MongoDB image version
## ref: https://hub.docker.com/r/bitnami/mongodb/tags/
##
image: bitnami/mongodb:3.7.1-r0

## Specify a imagePullPolicy
## 'Always' if imageTag is 'latest', else set to 'IfNotPresent'
## ref: http://kubernetes.io/docs/user-guide/images/#pre-pulling-images
##
# imagePullPolicy:
```

```

## MongoDB admin password
## ref: https://github.com/bitnami/bitnami-docker-
mongodb/blob/master/README.md#setting-the-root-password-on-first-run
##
# mongodbRootPassword:

## MongoDB custom user and database
## ref: https://github.com/bitnami/bitnami-docker-
mongodb/blob/master/README.md#creating-a-user-and-database-on-first-run
##
# mongodbUsername:
# mongodbPassword:
# mongodbDatabase:

## Kubernetes service type
serviceType: ClusterIP

## Enable persistence using Persistent Volume Claims
## ref: http://kubernetes.io/docs/user-guide/persistent-volumes/
#####
#####
## 注意, 如果想要mongodb使用存储, persistence.enabled需要设置为true, 并且配置
## persistence.storageClass字段。
persistence:
  enabled: false
  ## A manually managed Persistent volume and Claim
  ## Requires persistence.enabled: true
  ## If defined, PVC must be created manually before volume will be
bound
  # existingClaim:

  ## mongodb data Persistent Volume Storage Class
  ## If defined, storageClassName: <storageClass>
  ## If set to "-", storageClassName: "", which disables dynamic
provisioning
  ## If undefined (the default) or set to null, no storageClassName spec
is
  ## set, choosing the default provisioner. (gp2 on AWS, standard on
  ## GKE, AWS & OpenStack)
  ##
  storageClass: "-"
  accessMode: ReadWriteOnce
  size: 8Gi

## Configure resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  requests:
    memory: 256Mi
    cpu: 100m

## Node selector
## ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-
node/#nodeselector
nodeSelector: {}

```



```

## Configure extra options for liveness and readiness probes
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/#configure-probes)
livenessProbe:
  initialDelaySeconds: 30
  periodSeconds: 10
  timeoutSeconds: 5
  failureThreshold: 6
  successThreshold: 1
readinessProbe:
  initialDelaySeconds: 5
  periodSeconds: 10
  timeoutSeconds: 5
  failureThreshold: 6
  successThreshold: 1

```

部署成功

The screenshot displays the 'one' management console interface. The top part shows a deployment list with the following data:

名称	状态	镜像	创建时间	操作
mongodb-mongodb	运行中	查看镜像地址	7分钟前	滚动发布 ...

The bottom part shows the application details for 'mongodb':

- 名称: mongodb
- 状态: 已部署
- 创建: 7分钟前
- 更新: 7分钟前
- 当前版本: 1
- 资源: 3
- 描述: mongodb

The '资源' (Resources) section lists the following items:

资源名称	类型	创建时间	操作
mongodb-mongodb	Secret	7分钟前	查看Yaml
mongodb-mongodb	Service	7分钟前	查看Yaml
mongodb-mongodb	Deployment	7分钟前	查看Yaml

访问

添加路由

[返回](#) | 添加路由规则

* 规则名称: ✓

规则描述:

* 选择出口:

* 选择服务:

* 端口协议:

服务端口协议	代理端口协议	操作
<input type="text" value="27017"/> <input type="text" value="TCP"/>	<input type="text" value="TCP"/>	<input type="text" value="指定端口, 缺省可系统..."/> <input type="button" value="删除"/>

○ 查看路由信息

[返回](#) | 路由规则详情

mongodb

出口类型: 服务代理 绑定服务: mongodb-mongodb

服务域名: - 代理 IP: 192.168.2.217

规则描述: mongodb

路由规则

✦ TCP 协议通过“代理 IP 加代理端口”方式访问; HTTP、HTTPS 协议通过“服务域名加代理端口”方式访问 (HTTP 的 80 和 HTTPS 的 443 默认可以不写)

共计1条 < 1 >

容器端口	代理端口
27017(TCP)	52035(TCP)

○ 访问

```

[root@centos8 ~]# docker run -it --rm mongo:3.7 mongo --host 192.168.2.217 --port 52035
MongoDB shell version v3.7.9
connecting to: mongodb://192.168.2.217:52035/
MongoDB server version: 3.7.1
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
2021-07-22T18:07:23.643+0000 I STORAGE [main] In File::open(), ::open for '/home/mongodb/mongorc.js' failed with No such file or directory
Server has startup warnings:
2021-07-21T08:39:37.795+0000 I CONTROL [initandlisten]
2021-07-21T08:39:37.795+0000 I CONTROL [initandlisten] ** NOTE: This is a development version (3.7.1) of MongoDB.
2021-07-21T08:39:37.795+0000 I CONTROL [initandlisten] ** Not recommended for production.
2021-07-21T08:39:37.795+0000 I CONTROL [initandlisten]
2021-07-21T08:39:37.795+0000 I CONTROL [initandlisten]
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** WARNING: You are running on a NUMA machine.
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** We suggest launching mongod like this to avoid performance problems:
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** numactl --interleave=all mongod [other options]
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten]
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten]
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2021-07-21T08:39:37.797+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2021-07-21T08:39:37.798+0000 I CONTROL [initandlisten]
> show dbs
admin 0.00068
config 0.00068
local 0.00068
> use test
switched to db test
> db.post.insert({"title":"title"})
writeResult({ "ninserted" : 1 })
> db.post.find()
{ "_id" : ObjectId("60f84658da157ee606bf62b"), "title" : "title" }
>

```

