

**数人云 DM/OS 快速**

**上手指南**

*数人云版本 3.5.01*

*2017年6月*

1. 上手流程 2

2. 创建用户组 4

3. 创建用户 5

4. 创建集群 7

5. 发布应用 9

5.1. 通过表单发布应用 9

5.2. 7层服务发现 14

5.3. 通过模板发布应用（应用编排） 14

5.4. 脚本（marathon 编排脚本）发布 19

6. 高级应用发布功能 19

6.1. 高级应用发布选项 19

6.2. 应用的灰度更新 22

7. 监控和日志 25

7.1. 容器实例的监控和日志 25

7.2. 数人云组件的运行状态监控 27

8. 报警规则 28

8.1. 创建基于监控数据的报警规则 28

9. 容器网络（Calico） 30

10. 镜像仓库 34

10.1. 镜像管理 34

10.2. 添加第三方镜像仓库 35

10.3. 添加镜像 36

10.4. 清理主机上的历史镜像 37

11. 审计和计量 38

12. 命令行工具（CLI） 38

13. 使用中发生的错误排查 39

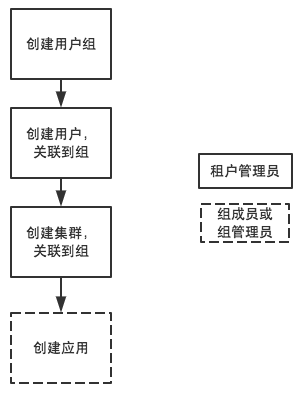
13.1. 应用发布失败 39

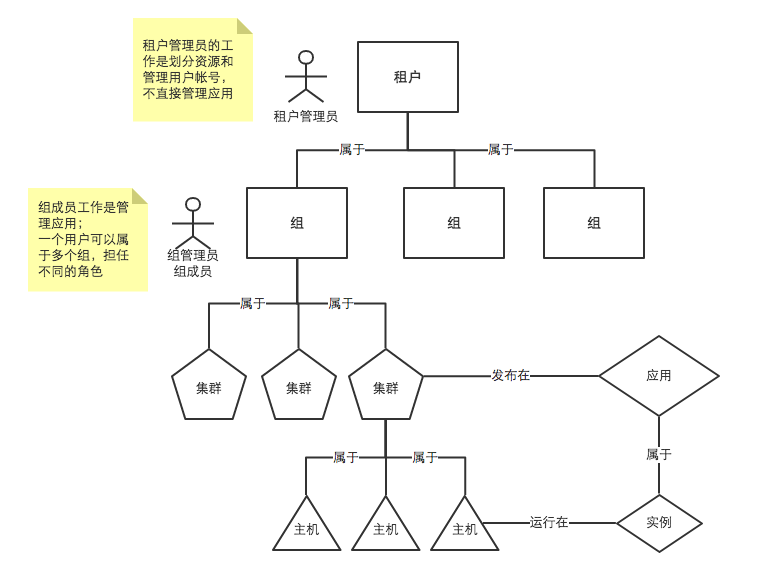
14. 后续阅读 40

**【浏览器兼容性】请确认一下您的屏幕水平分辨率至少有1280，避免部分顶部导航栏项被隐藏。如果您没法找到“系统”菜单项，应该是浏览器不够宽造成的。基于兼容性考虑，目前建议浏览器使用谷歌的 chrome。**

### 上手流程

本上手指南将引导租户管理员创建一个新的 Demo 环境，并且发布一个2048演示应用。具体流程如下：



首先解释一下数人的多租户架构。数人云每个租户内权限分为三级：租户管理员，组管理员，组成员。其中租户管理员负责管理资源划分和用户帐号，而日常发布应用等应用运维有关工作，由组内用户（组成员和组管理员）

同一个用户可以分属不同用户组，组内角色决定了用户组内的权限。一个用户组可以有多个集群。一个集群仅可以归属到一个用户组，不能被多组共享。

租户管理员：

* 可以查看所有集群的应用状态，并可以对任意应用做增、删、改等操作。
* 可以创建新用户、调整用户所属用户组及组内权限。
* 可以创建、删除集群，添加、删除集群内主机。
* 可以连接到平台内任一实例内部。
* 可以管理其他用户的有效性，包括其他超级管理员的权限。

组管理员：

* 可以查看所属集群内的所有应用状态，
* 可以对组内集群中的所有应用做增、删、改等操作。
* 可以连接到组内任一实例内部。
* 不可以对集群主机做操作。
* 不可以创建用户

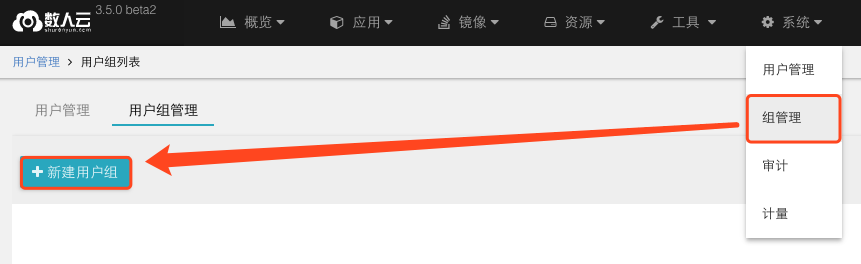
组成员：

* 可以查看本组内的应用状态，但是只能对自己发布的应用做增、删、改等操作。
* 可以查看本组集群内主机、不可以对主机做任何操作。
* 可以连接到自己发布的实例内部。

一个用户可以加入多个用户组，用户在用户组中的角色决定了这个用户对用户组中应用的操作权限。

### 创建用户组

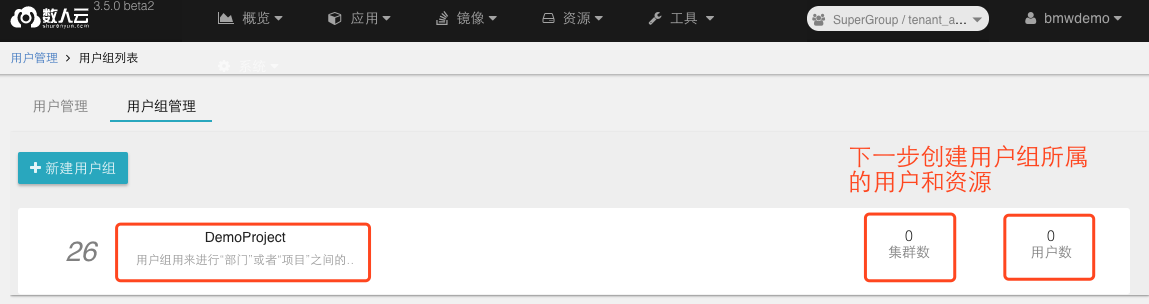
请使用“租户管理员”帐号登录数人云，在导航栏“系统”下拉菜单中选择“组管理”。用户组管理页面列出了当前租户下的所有用户组，并可以对其进行管理操作。

****

点击“新建用户组”，填写相关描述信息，创建一个或者多个用户组。

****

新建的用户组既不包括任何组员（用户），也没有划分任何计算资源（集群）。

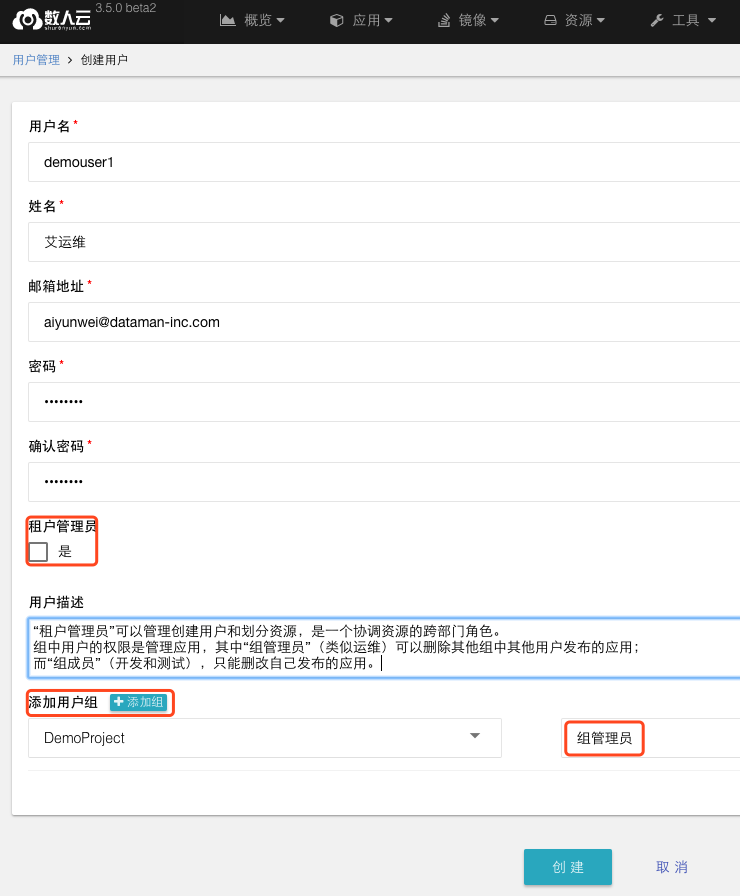
****

### 创建用户

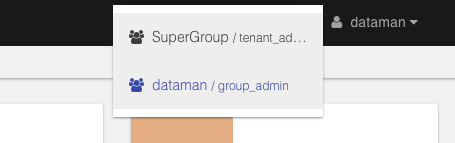
在导航栏“系统”下拉菜单中选择“用户管理”。用户管理页面列出了当前租户下的所有用户，并可以对其进行管理操作。目前只有租户管理员一个用户，我们创建一个组管理员角色的用户（管理应用），并且关联到各个创建的用户组。

****

点击左上角的“新建用户”按钮，创建一个新的用户。除了填写说明性信息（请注意密码要包括大写字母和数字），并且要为用户指定所属的“用户组”和角色。这里我们将新建用户添加到刚刚创建的用户组“DemoProject”，并且将他赋予“组管理员”的角色。“组管理员”可以删除组内其他用户发布的应用，类似运维的角色。

****

请注意，一个用户可以属于多个用户组，登录后通过右上角的下拉菜单进行用户组的切换。



### 创建集群

“集群”代表了一个小计算资源池，由若干主机组成。下面我们会新建一个集群，并且为这个集群添加若干主机。

****

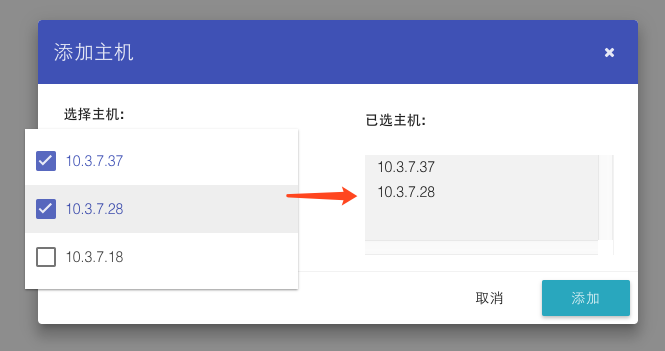
请注意，集群只能属于一个项目组。

****

新建集群并没有关联主机，所以资源为0。点击左下角的按钮来为集群添加主机。

****

下面我们从待分配主机中选择两台，添加到演示集群中。

****

添加主机需要等待一段时间进行初始化，随后可以看到集群内的资源已经不再为空，而是两台主机相应的资源总和。

****

点击集群的名称链接，可以列出该集群包含的主机列表。点击每台主机的“删除”按钮，可以将这台主机重新放回资源池供分配。请注意，运行在该主机上的容器实例会自动迁移到集群内其他主机上。

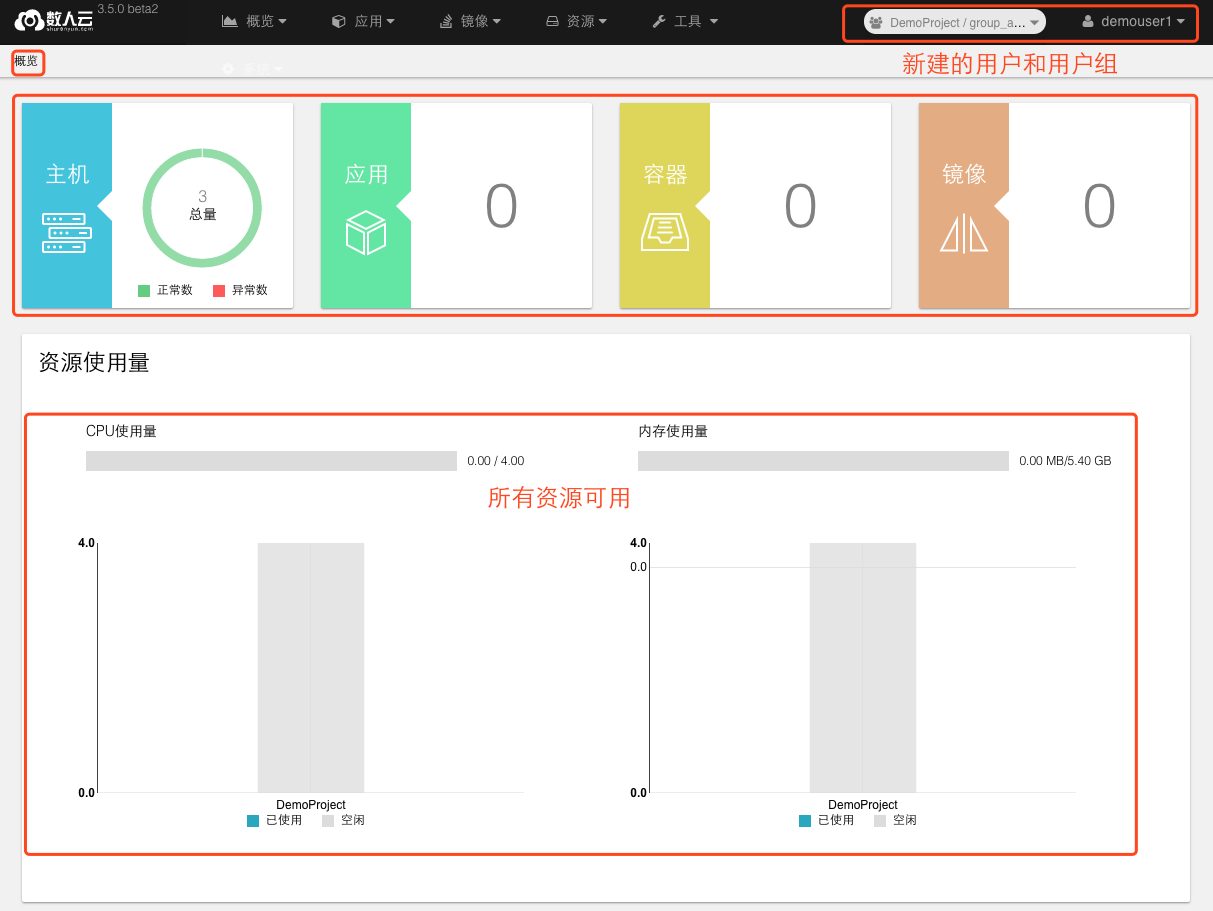
### 发布应用

### 通过表单发布应用

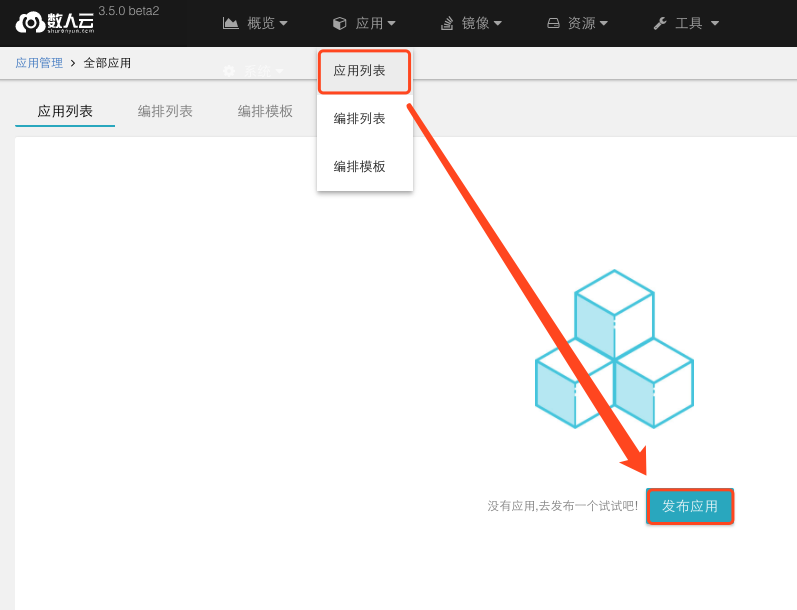
租户管理员负责分配资源，应用的运维都应该由“组成员”或者“组管理员”来进行。所以我们先退出，然后用刚刚创建的那个用户重新登录。



此时我们可以看到“概览”页面，上半页展示了本组所属资源和应用的情况，下半页展示了可用资源（除去已经分配部分）的多少。

****

从导航栏“应用”进入应用列表页，管理和发布应用。

****

这个例子中，我们采用传统的表单形式来发布一个简单的“2048”应用。

****

最重要的参数是指定容器镜像，这里填写的平台镜像仓库预置的“2048”游戏镜像，也可以填写任何公网镜像仓库的地址。

****

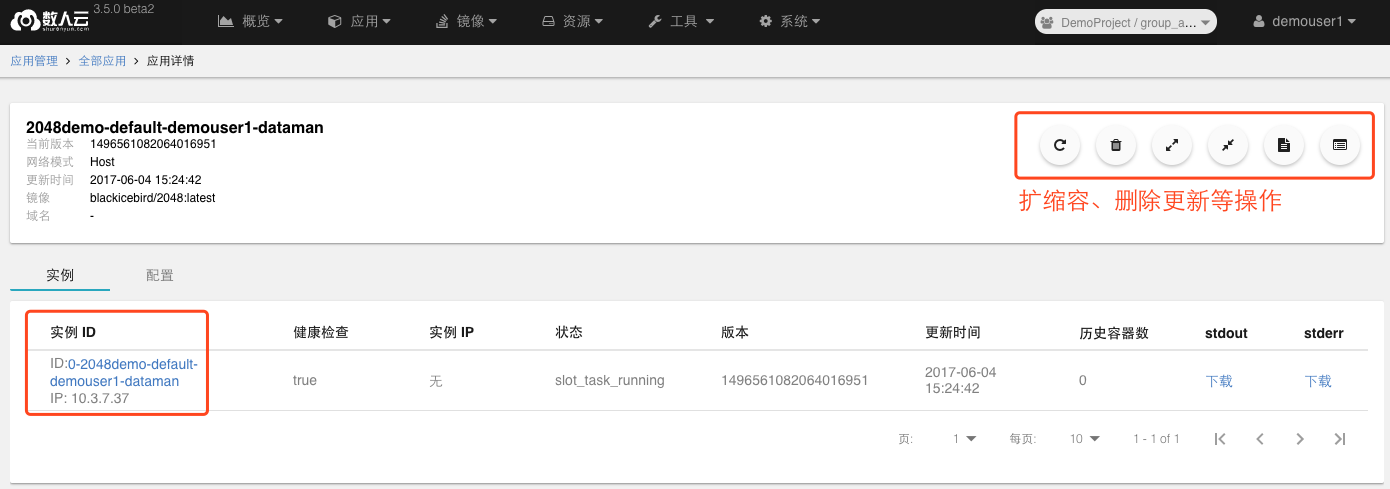
点击发布，等待一段时间之后，即可看到应用已经正常启动，包含的实例数量为1个。如果一次发布了多个应用实例，数人云会按照顺序逐次发布。也就是说，只有当一个实例发布成功，才会发布下一个实例。

****

点击应用名链接，即可看到应用所包含的实例列表，并且对应用进行扩缩容和更新等管理操作。请注意，扩容或者缩容的时候填写的是变化数量，不是最终结果。例如下图原有1·个实例，本次扩容新增了2个实例，最终应该有3个实例。



因为本应用是采用“host” 网络模式发布，所以只能根据每个应用实例的 IP 地址和已知应用端口来访问。例如这里的第一个实例 IP 为“10.3.7.37”，2048默认的访问端口是80。大家会注意到，*10.3.7.37*并非公网 IP，通过互联网无法直接访问。

****

### 7层服务发现

数人云内置了针对7层服务的服务发现功能，特指采用 Bridge 模式发布的应用，主要是针对基于 HTTP 的 Restful 服务或者网站类应用。例如此服务的访问域名是“*ecshop.ecshopdemo.demouser1.dataman.swan.com*”，方便了各个微服务之间互相访问。

很显然，此URL 只能在数人云集群的主机上访问，因为这些主机默认配置了数人云的内网 DNS 服务。如果需要向公网暴露相关业务，需要在相应的网关（或者负载均衡）上也配置数人云内网 DNS 服务。



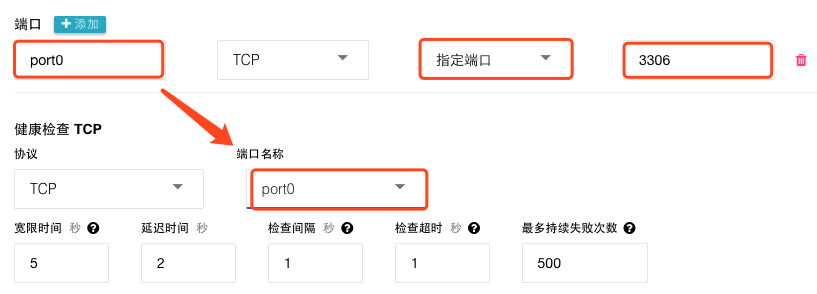
健康检查

数人云提供非常丰富的监控检查手段，以常用的两种网络模式为例。以一个 Tomcat 应用为例，以 bridge 网络模式发布。

1. 添加监控检查的端口，定义容器内部端口即可，这里是 Tomcat 标准的8080。映射协议一般选择 TCP（对于 HTTP 协议应用）。
2. 定义健康检查的协议，这里是基于 HTTP。同时可以定义 HTTP 的 URL 路径，例如是特定的监控检查页面
3. 端口名称就是步骤1定义的端口



以一个使用 Host 网络模式的 MySQL 应用为例，内部端口为3306，健康检查协议为 TCP



健康检查的几个通用指标涵义如下：

* 延迟时间：指应用启动之后，延迟若干秒开始健康检查。避免应用启动时间过长，造成第一次健康检查失败
* 检查超时：监控检查的请求最长回应时间，超过这个限制既认为失败，不管返回值是什么
* 最多连续失败次数：必须连续失败指定次数，才认为实例已经失败

### 通过模板发布应用（应用编排）

您可以使用 Docker Compose V3.0脚本来发布分布式应用。首先创建“编排模板”。

****

随后创建编排模板，模板的 YAML 脚本可以通过文本框输入，也可以从本地文件中读取。

****

这里是一个 ECShop 电商网站（LNMP 应用）的例子，包括一个 PHP 应用“ecshop” 和MySQL数据库应用“ecshopdb”。通过脚本可以定义两个应用各自的容器启动参数，以及两者之间的顺序依赖关系。

*version: "3"*

*services:*

*ecshop:*

*image:* "*sryregistry.service.consul:5002/填写租户名称/library/ecshop*"

*network\_mode: "bridge"*

*depends\_on:*

*- ecshopdb*

*ports:*

*- 80/tcp*

*ecshopdb:*

*image: "sryregistry.service.consul:5002/填写租户名称/library/ecshop-db"*

*network\_mode: "host"*

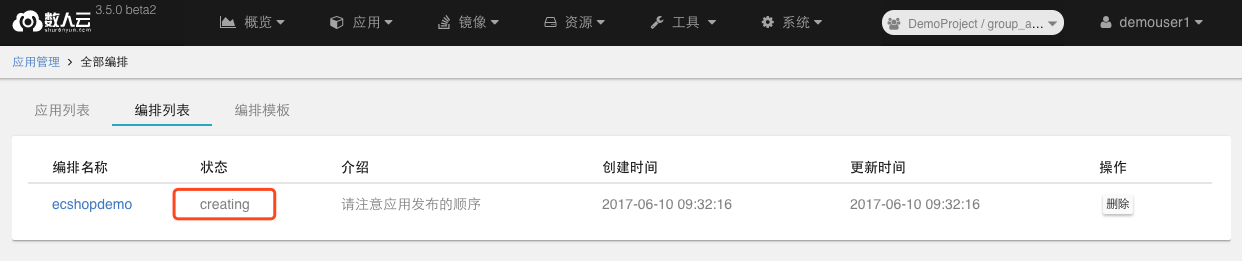
当我们要发布这个应用的时候，在新建模板的操作菜单点击“部署”。

****

应用发布的参数中，“延时”代表了两个应用发布间隔。这里当数据库应用通过健康检查之后，间隔 5秒再发布网站应用，确保数据库进入工作状态。

****

通过应用编排模式发布的应用组，通过“应用”->“编排列表”进行管理。

****

在应用列表中，可以查看每个子应用的发布和运行情况。

****

### 脚本（marathon 编排脚本）发布

数人云同上支持 Marathon Application Group 格式的脚本发布，编写规则详见：

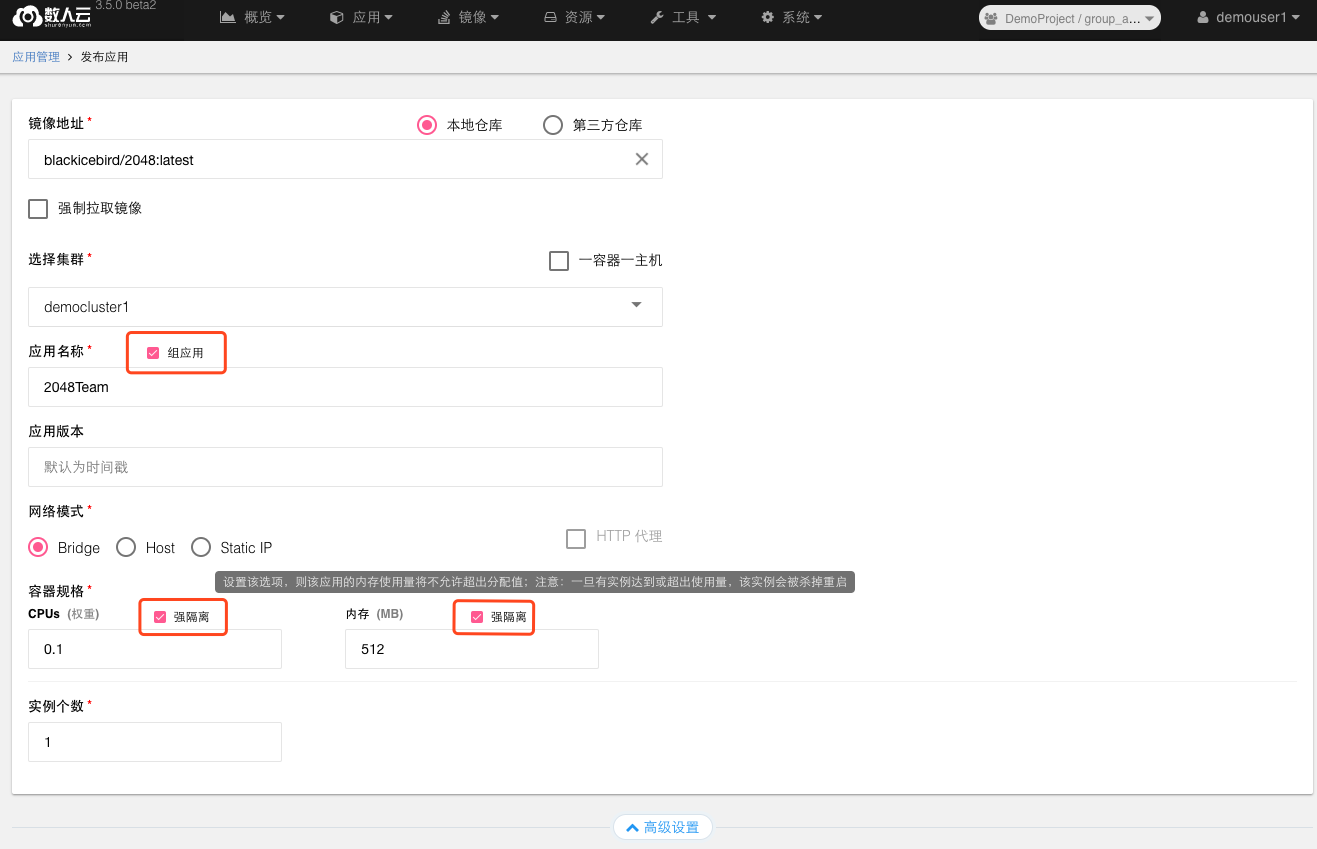
<http://mesosphere.github.io/marathon/docs/application-groups.html>

****

### 高级应用发布功能

### 高级应用发布选项

表单方式发布应用具有如下高级模式。



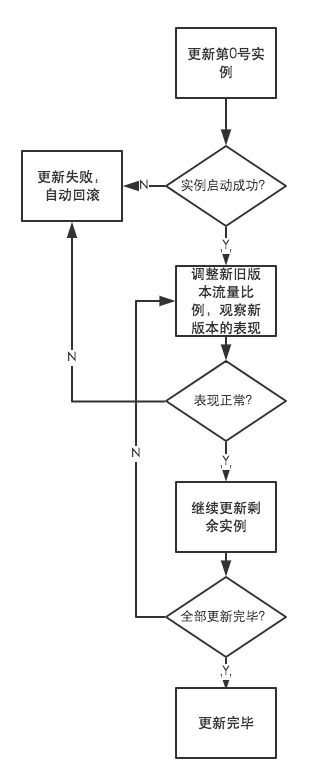
* “组应用”：用户组所有成员都可以对此应用增删改查
* CPU 强隔离：可以简单地理解为应用使用 CPU 不会超过这个绝对值
* 内存强隔离：一旦应用使用内存超过此限制，此容器实例就会被重启。所以应用应该在内部（例如 JVM 参数）限制内存使用



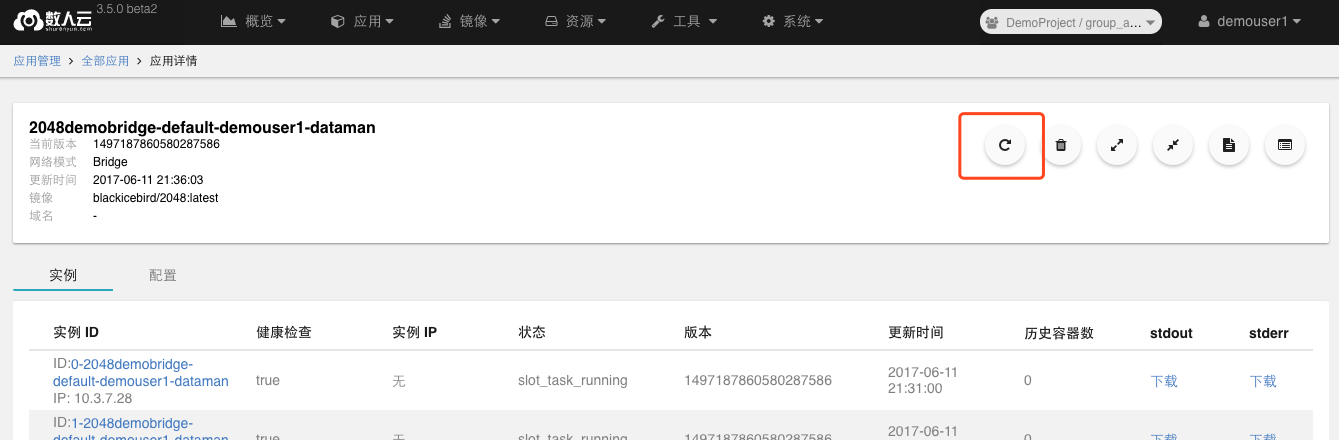
* 环境变量：此处定义的环境变量，应用可以在容器实例内部读到
* 挂载目录：相当于 docker 的-v 命令，挂载宿主机目录
* CMD：相当于 dockerfile 里面的 CMD 参数
* Docker 参数：相当于 docker 命令的各种参数。请注意，平台默认添加的 docker 参数，请不要修改或删除
* 文件日志采集：凡是要对宿主机的日志文件采集时，key值为固定关键字*label*，value值格式如logpath=/var/log/test1.log,/var/log/test2.log并用逗号分隔。可以先将容器内的业务文件日志映射到宿主机目录下，然后通过这种方法收集。

### 应用的灰度更新

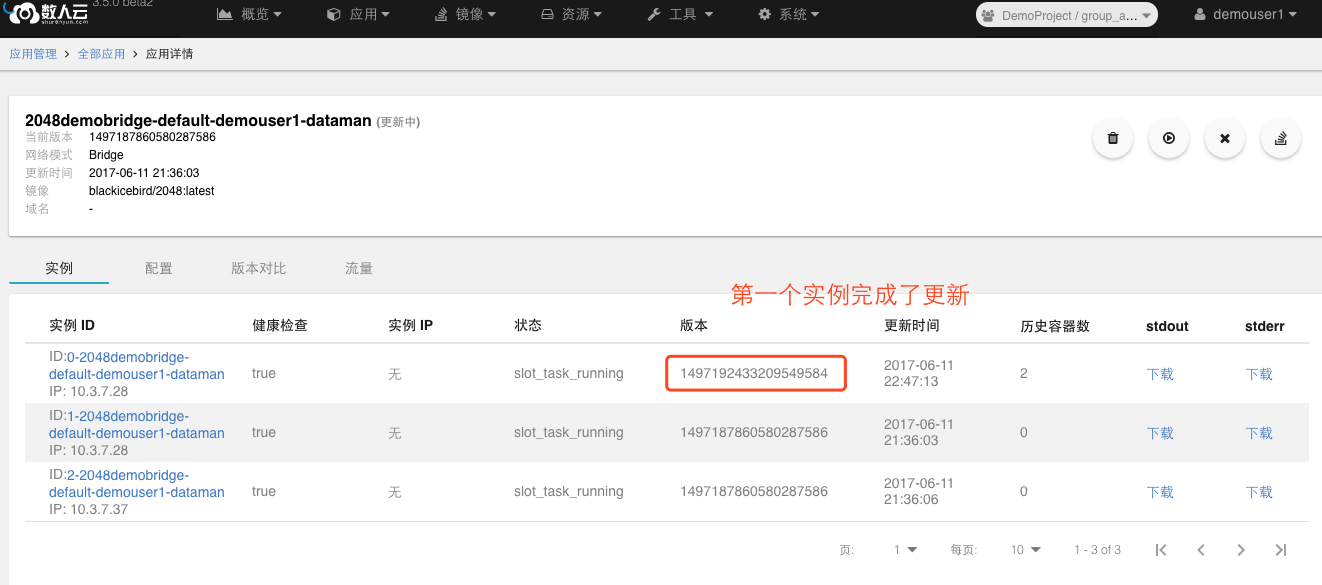
数人云的灰度更新流程完全模仿谷歌Borg，其特点为分步更新所有实例，每一步都可以调整新旧版本的流量比例，实现灰度发布。



1. 开始更新。请注意，目前只有通过表单发布的应用可以进行更新。



1. 在表单页修改版本或者参数之后，点击“更新”按钮。首先会更新应用的第一个实例（编号为0），如下图



1. 当第一个实例成功更新之后，可以调整新旧版本之间的流量。例如下图将新版本的流量从0调整到18%。

首先更新第一个实例是为了确保容器内应用没有逻辑错误，能够正常启动运行。





1. 观察相应的应用层监控和日志数据（数人云 scope 之外），如果新版本应用运行符合预期，可以继续更新剩余的1个或者多个实例。



1. 如果分为多次逐步完成更新，每一步都可以自行定义新旧版本之间的流量比例，实现灰度发布。

请注意这个灰度过程仅仅限于更新，应用的首次发布无法灰度进行。

### 监控和日志

### 容器实例的监控和日志

数人云支持针对单个容器实例的监控和日志信息展示和搜索，目前不支持应用范围内跨实例的搜索。

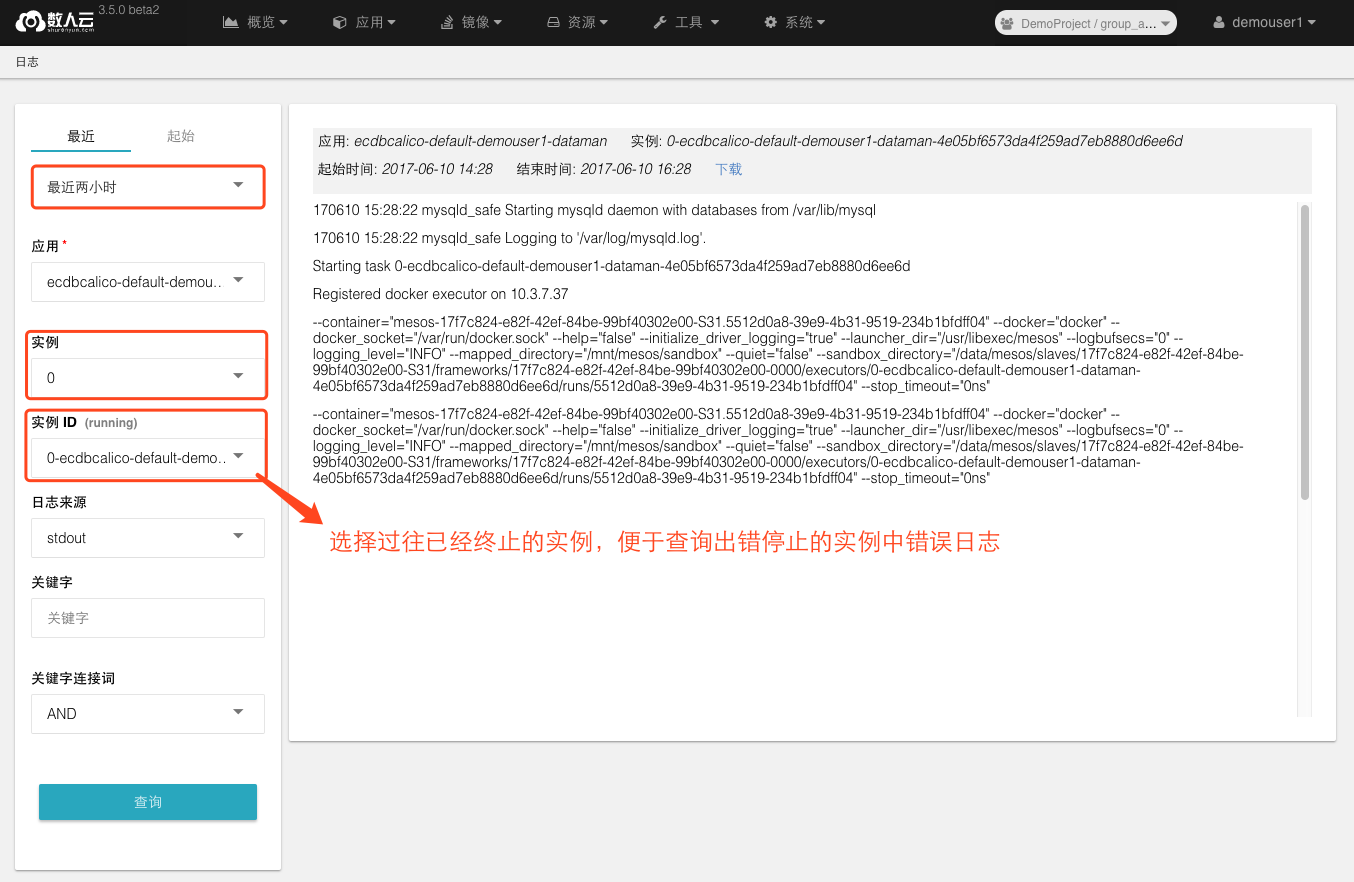
就监控而言，需要指定实例。数人云的实例安装0，1，2顺序编号，其中第0号是应用的第一个实例。监控指标为 Docker stat 的四个标准指标。如果配置了 JMX，还可以读取类似 Tomcat 线程池等数据。

如果用户对于 Prometheus 查询语法很熟悉，可以直接在“高级查询”内输入查询语句。

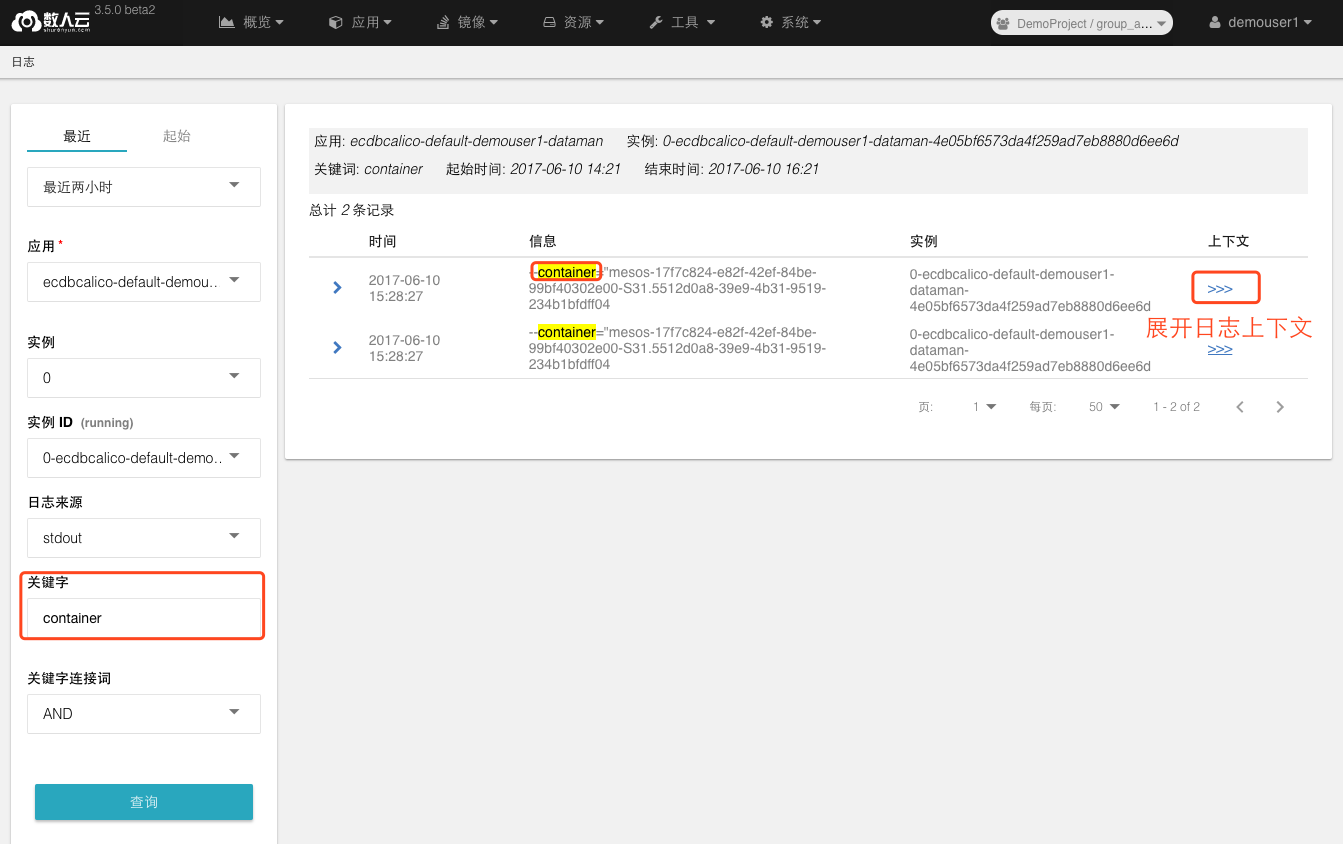
****

就日志而言，数人云提供了按指定时间段顺序读取日志和按关键词搜索的功能，两者在同一个页面实现。

首先请选定时间段和日志实例。日志来源既日志内容，目前包括标准输出和错误。特别要注意“实例 ID”，容器实例可能历史上意外停止重启，通过选择历史版本的 ID，便于查询错误退出的实例中的历史日志。

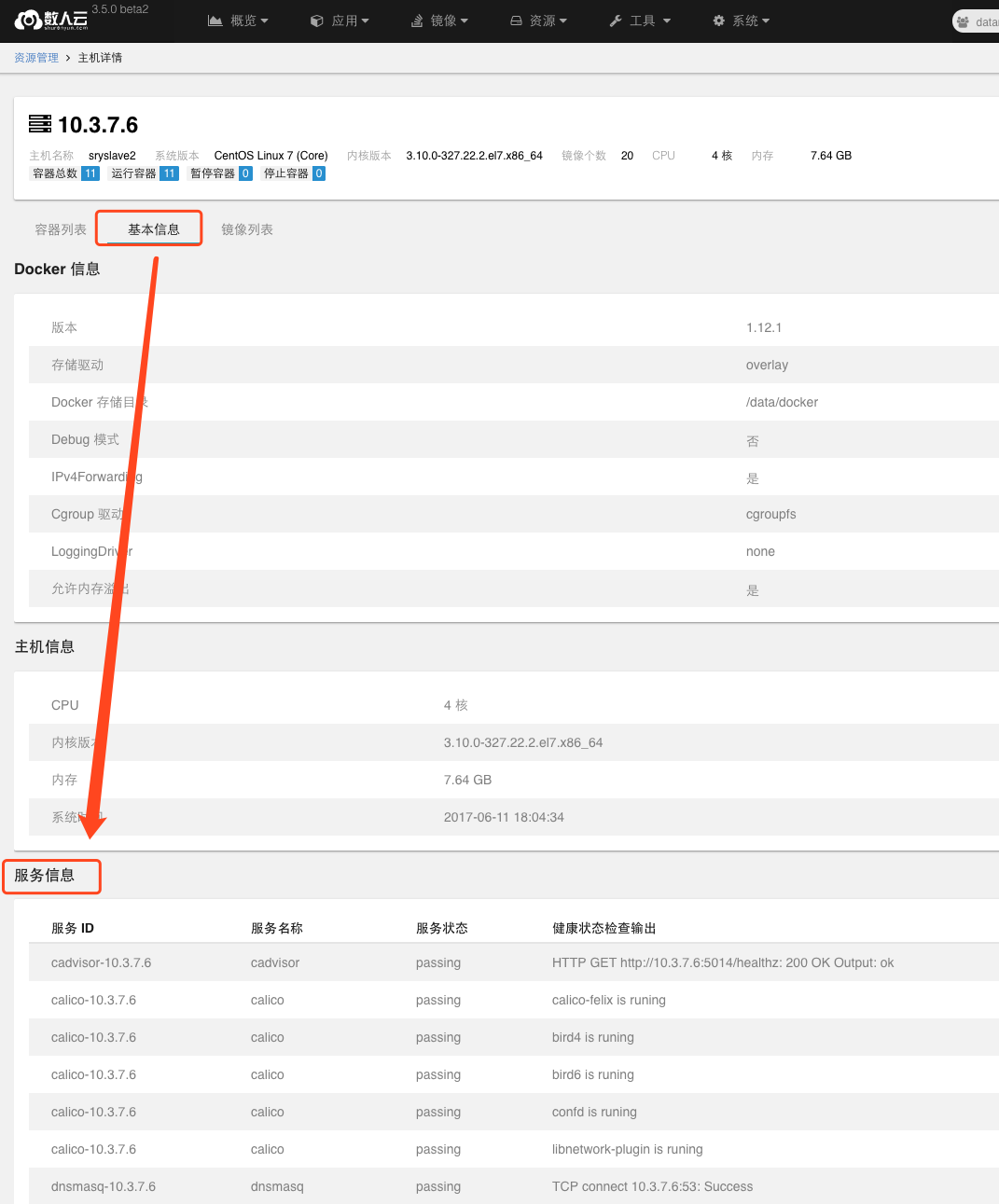
****

用户可以增加关键词过滤条件，例如“404”等错误代码。点击查询结果的“上下文”链接，可以看见日志产生的上下文

****

### 数人云组件的运行状态监控

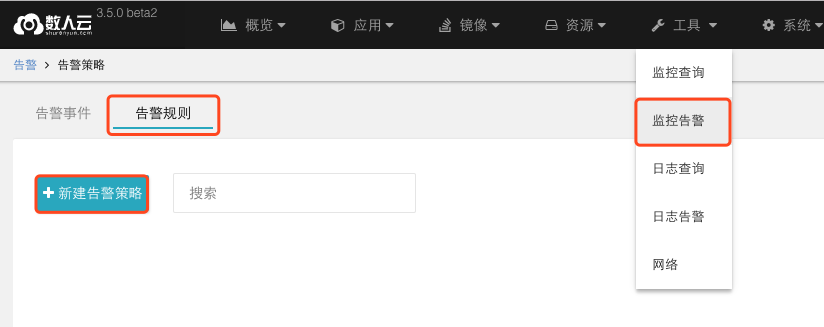
在数人云的集群详情页面，列出了集群所属所有的主机。点击主机 IP 的链接，可以访问主机的详情，还包括该主机上数人云系统组件的运行状态：



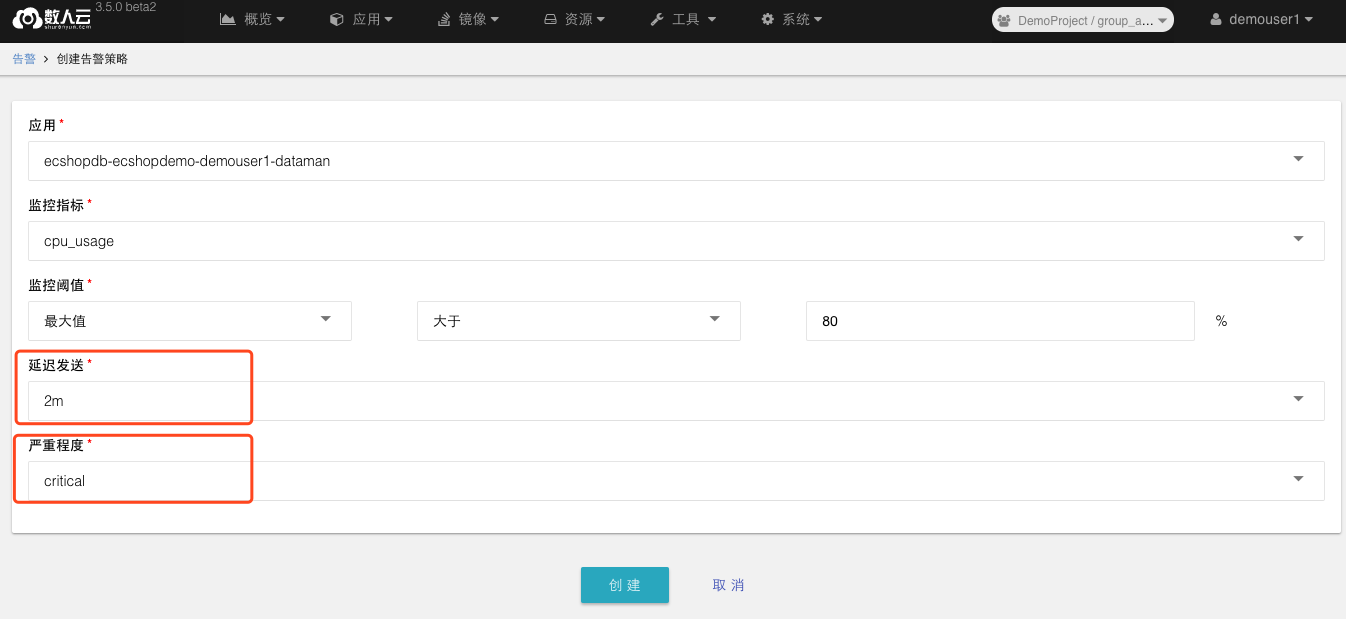
### 报警规则

### 创建基于监控数据的报警规则

创建报警规则

****

“延迟发送”可以用来调节报警的敏感程度，报警事件必须持续超过这个时间，报警才会触发。同时可以定义“严重程度”，以便有选择地通过邮件等手段发送关键报警信息。

****

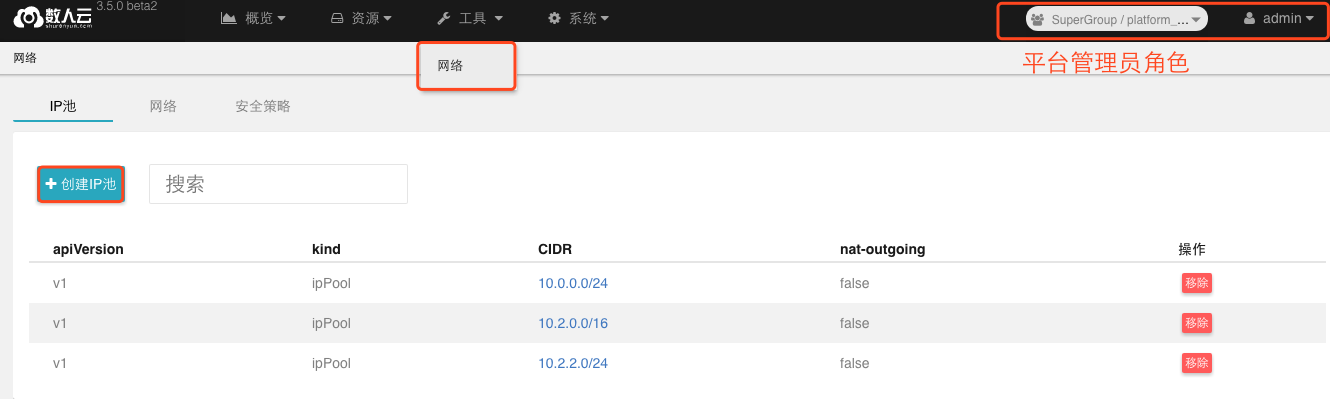
****

### 容器网络（Calico）

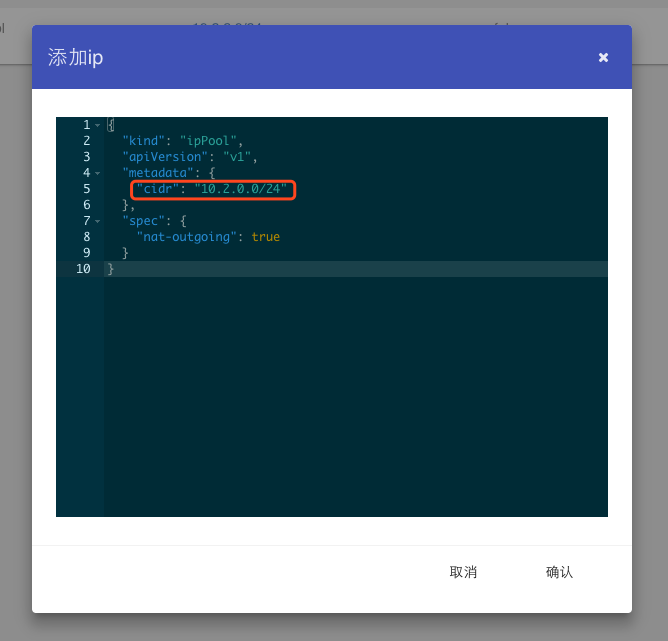
数人云支持基于 calico 的容器网络方案（https://www.projectcalico.org）。容器实例可以指定 IP 地址，而且当实例重启时，保持 IP 地址不变。这样方便了基于 TCP 向外提供服务的传统应用（例如 weblogic MS，MySQL） 容器化上云。

1. 创建容器网络

整体规划容器网络是平台层面的工作，请退出后以“平台管理员”的角色登录。在网络菜单下可以管理当前全平台已有的容器网络。



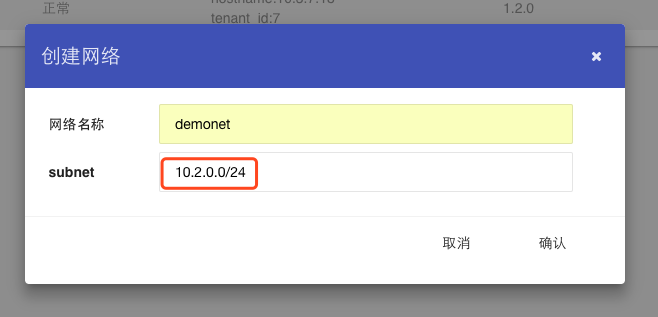
指定IP 池地址段（CIDR），目前数人云还没有 IPAM 功能。



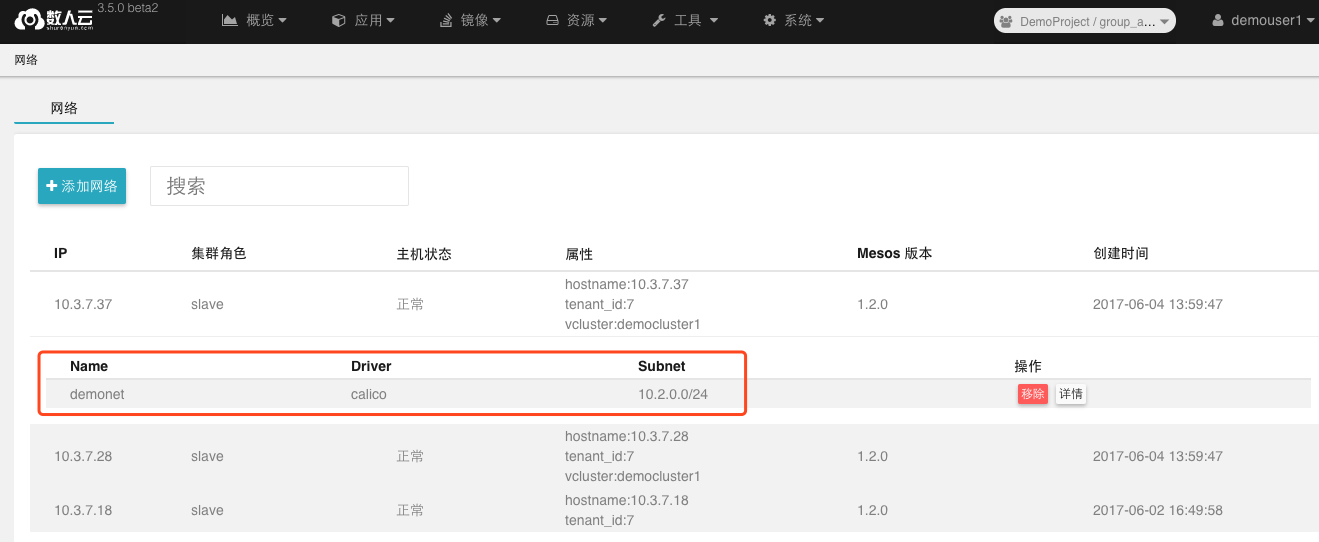
1. 初始化容器网络

退出平台管理员，以普通租户用户角色登录，然后初始化刚刚创建的容器网络（10.2.0.0/24）。





成功创建网络之后，点击列表中的主机 IP，就可以看见新建的网络。



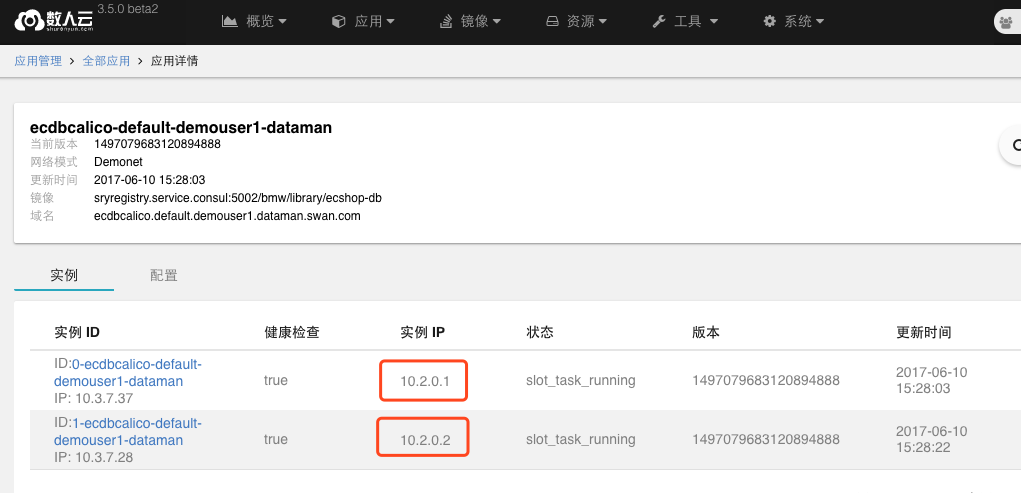
1. 创建一容器一 IP 的应用

在应用发布页面。请注意首先要选择集群，然后点击“Static IP”，才能在下拉菜单中看见刚刚创建的 demonet。





我们可以在实例列表内看见刚才指定的 IP。即使容器实例意外重启，IP 保持不变，几乎可以认为是个轻量的虚拟机。



### 镜像仓库

### 镜像管理

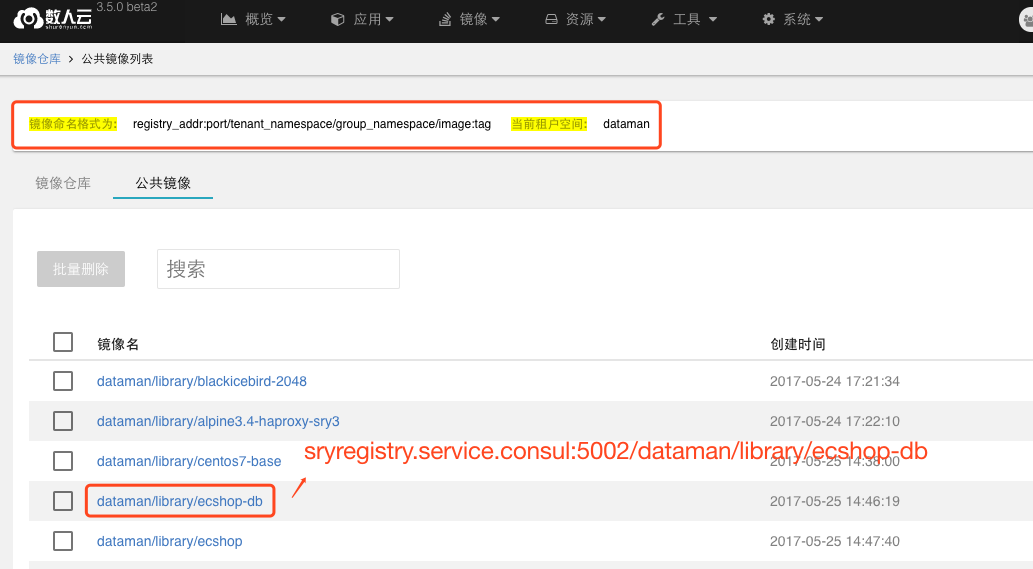
数人云的内置镜像仓库以租户为单位进行镜像的隔离。每个租户下所有用户共享本租户的“公共镜像”，每个“用户组”内用户共享组内私有镜像（通过“镜像仓库”菜单项查看）。

大家可以注意到，镜像名称实际上使用了租户名来划分 namespace。镜像 URL 的命名规则为：

*仓库地址/镜像名*

下图显示了 dataman 租户下所有用户共享的公共镜像（library之下）。此环境的默认镜像仓库地址 sryregistry.service.consul:5002，租户名称为 dataman，所以 ecshop-db 的 URL 为：

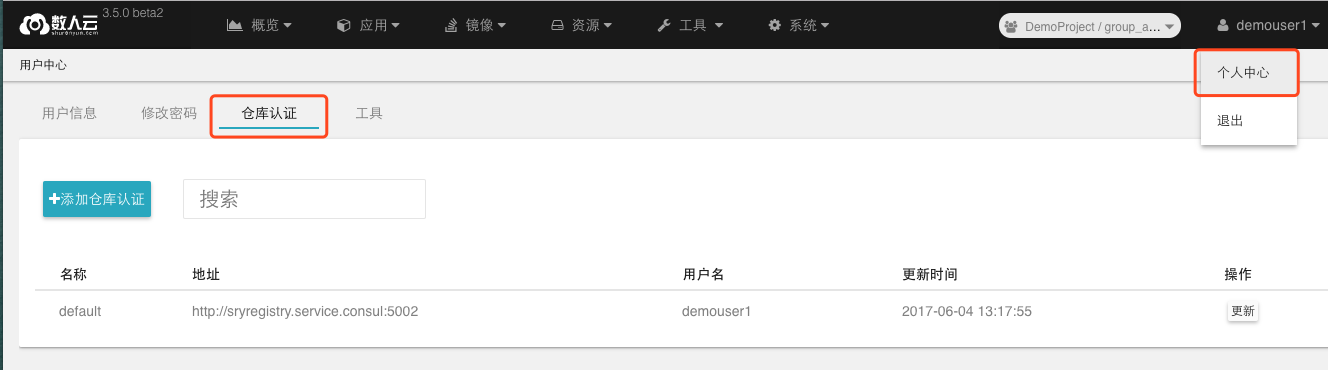
*sryregistry.service.consul:5002/bmw/library/ecshop-db*



目前数人云 UI 不支持页面上传镜像，推送镜像需要采用Docker的命令行（见 添加镜像），或者通过 Jenkins 等 CI/CD平台。

### 添加第三方镜像仓库

当用户希望通过第三方的镜像仓库发布应用的时候，首先需要在自己的“个人中心”中添加第三方镜像仓库。请注意，目前第三方仓库是和用户绑定，不是和用户组或者平台绑定。意味着除了平台默认的镜像仓库，每个用户必须分别添加相应的第三方镜像仓库。

****

目前只支持基于用户名和密码的认证方式。连接第三方镜像仓库之后，即可在应用发布表单内看到。



### 添加镜像

在任一可以连通仓库主机上，修改本地 docker的参数，加上“--insecure-registry ip:5000“，其中IP为镜像仓库的地址。

* ubuntu 14.04, 需要配置 /etc/default/docker , 将 DOCKER\_OPTS 加上 --insecure-registry ip:5000 , 然后重启 docker daemon
* ubuntu 15 或者 centos， 需要配置 /lib/systemd/system/docker.service ， 加上 --insecure-registry ip:5000 , ，考虑到centos下overlay的性能会好，可以加入－s overlay选项（低版本的系统需要升级内核），
* 然后运行:systemctl daemon-reload && service docker restart、systemtl enable docker设置docker开机启动。
* 将待推送的镜像改名，命令：注意修改相应IP，namespace（按照前文所述的租户命名方式）及镜像名称，执行命令如下：

*docker tag yourimages:tag ip:5000/租户名称/组名称/images:tag*

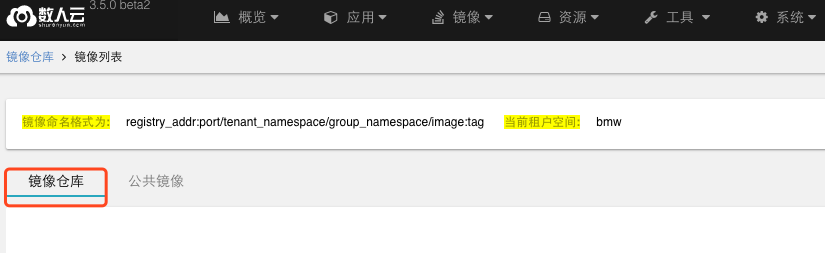
*docker login ip:5000*

*用户名： 平台id*

*密码： 平台密码*

*docker push ip:5000/租户名称/组名称/images:tag*

推送成功的镜像的镜像可以在镜像仓库界面查询



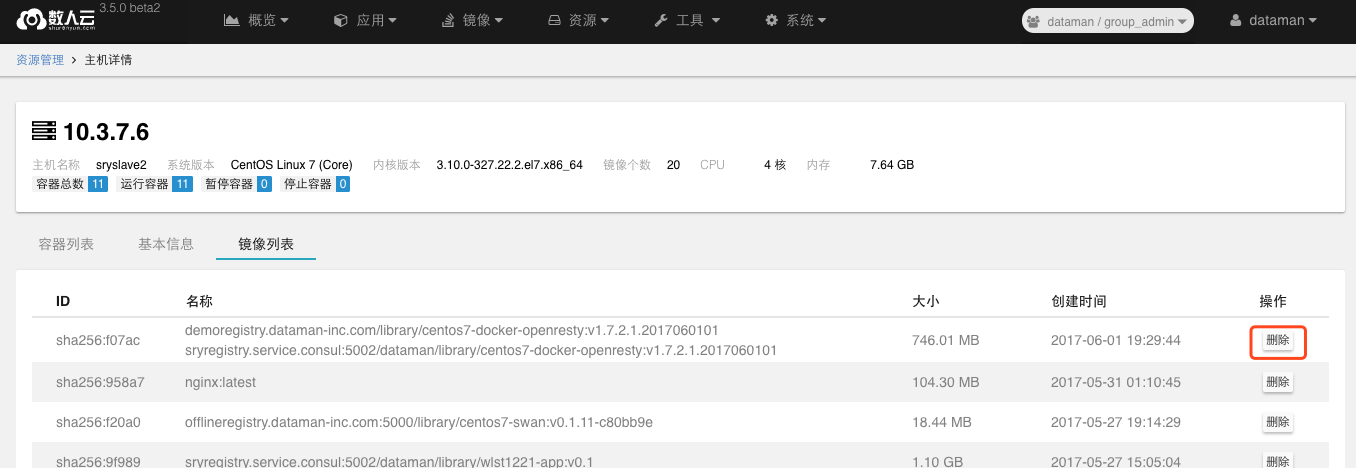
* 如果需要推送到公共仓库，执行命令如下：

*docker tag yourimages:tag ip:5000/租户名称/library/images:tag*

*docker push ip:5000/租户名称/library/images:tag*

### 清理主机上的历史镜像

在数人云的集群详情页面，列出了集群所属所有的主机。点击主机 IP 的链接，可以访问主机的详情。用户可以在详情页的“镜像列表”内清理该主机上的历史镜像，释放空间。



如果正在运行的容器实例依赖某个镜像（或者存在多层依赖），该删除操作会报错。

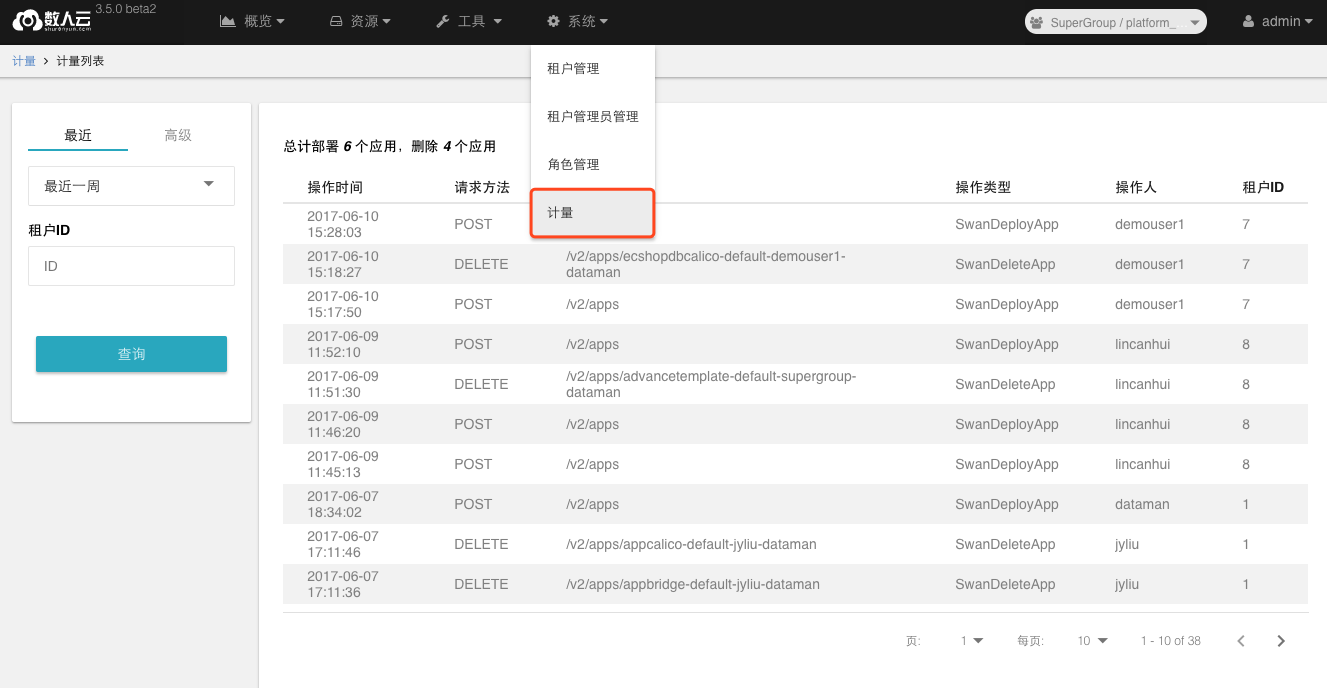
### 审计和计量

租户管理员可以使用审计功能，查询租户用户的操作记录。输入项包括审计对象、操作类型、操作人、关键词、查询给定时间范围内的操作记录。除时间范围外，均为非必选项目，不选择的项目，默认查询所有记录。默认按时间倒序排列。

平台管理员可以访问平台的计量功能，计量功能目前是基于资源有关的审计信息。可以根据时间显示某租户所有资源有关的操作，包括：

* 主机的划分和移除
* 应用的发布和删除

基于这些信息，可以汇总出该租户的话单，作为计费系统的输入。



### 命令行工具（CLI）

数人云平台提供了CLI管理工具，该工具以容器形式运行。

运行环境要求：

1. 预先安装docker， 版本：建议1.9.1以上；

2. 主机可以连通线下云平台；

3. 修改/etc/hosts，加入部署云平台仓库的ip地址

*镜像仓库ip offlineregistry.dataman-inc.com；*

使用方式：

*docker run -it offlineregistry.dataman-inc.com:5000/shurenyun/centos7-python-borgcli bash*

进入容器内部;

命令的格式参考：

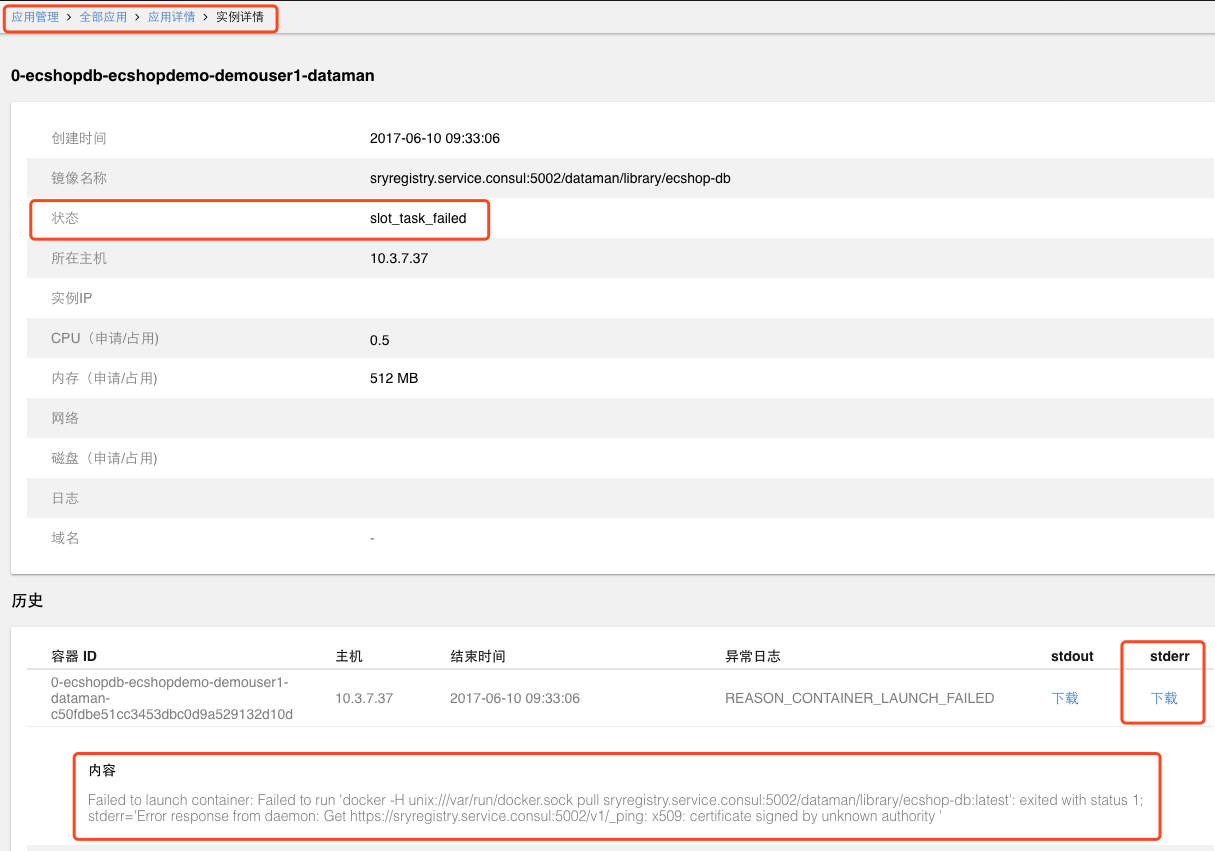
*python3 borgapi\_cli.py -h*

### 使用中发生的错误排查

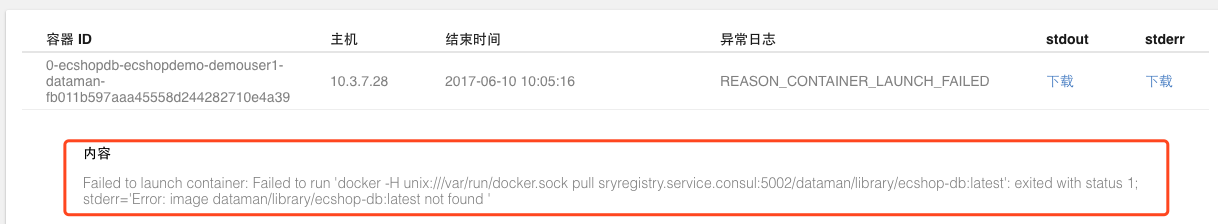
### 应用发布失败

应用发布失败一般有两种原因。选取应用的第一个实例，在其详情页的历史记录中，可以查看相关错误信息。

1. 容器镜像拉取失败。点击容器 ID，能够看到拉取和启动镜像的相关信息。例如下图说明 slave 节点的证书认证失败，无法从镜像仓库拉取镜像。
2. 容器内部应用启动失败。通过下载容器内的标准错误输出文件，可以了解应用启动失败的原因。



例如应用镜像不存在（或者 URL 输入错误）的信息如下：



### 后续阅读

进入产品实施阶段之后，您可以如下延伸阅读：

* 《数人云产品使用手册》：容器云平台的操作和使用
* 《数人云部署手册》：私有部署的流程
* 《数人云运维和巡检手册》：数人云平台日常运维和巡检
* 《数人云备份和灾难恢复手册》：日常的数据备份和灾难恢复
* 《数人云系统集成手册》：同既有周边系统的集成，例如负载均衡、监控日志