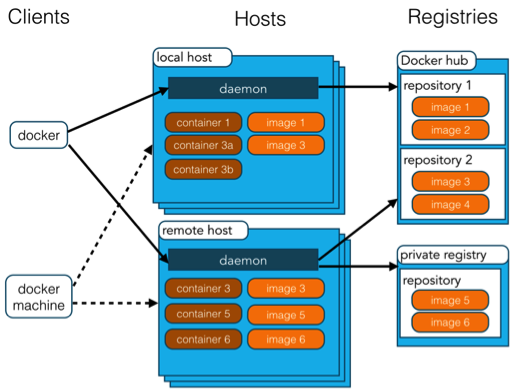
Docker Registry学习

**UserStory:作为参与了上一个Sprint的成员，我希望进一步了解Docker Registry的相关技术概念和常规操作，以便于对Docker技术有更深刻的了解**

写在最顶端的precondition：



|  |  |
| --- | --- |
| Docker 镜像(Images) | Docker 镜像是用于创建 Docker 容器的模板。 |
| Docker 容器(Container) | 容器是独立运行的一个或一组应用。 |
| Docker 客户端(Client) | Docker 客户端通过命令行或者其他工具使用 Docker API (<https://docs.docker.com/reference/api/docker_remote_api>) 与 Docker 的守护进程通信。 |
| Docker 主机(Host) | 一个物理或者虚拟的机器用于执行 Docker 守护进程和容器。 |
| Docker 仓库(Registry) | Docker 仓库用来保存镜像，可以理解为代码控制中的代码仓库。  Docker Hub([https://hub.docker.com](https://hub.docker.com/)) 提供了庞大的镜像集合供使用。 |
| Docker Machine | Docker Machine是一个简化Docker安装的命令行工具，通过一个简单的命令行即可在相应的平台上安装Docker，比如VirtualBox、 Digital Ocean、Microsoft Azure。 |

**Action1：Registry技术概念概览学习，理解DTR的产品作用与定位**

**Time：0.5h**

**定义**：商业产品、完全的镜像管理工作流、LDAP集成、镜像签名、安全扫描、UCP集成。

作为一款无状态高可度量的服务器端应用程序，Registry可用来**存储和分配**DOCKER镜像.

**为什么用**：

1. 紧密控制已安装的镜像
2. 完全拥有未镜像分配pipeline的权限
3. 将镜像存储和分配紧密集成在内部开发工作流中

**所需条件**：Docker engine version 1.6.0 or higher

**基本命令**：

启动registry

sudo docker run -d -p 5000:5000 --name registry registry:2

从hub端pull（或build）镜像

sudo docker pull ubuntu

为镜像打tag

sudo docker tag ubuntu localhost:5000/myfirstimage

push镜像myfirstimage

sudo docker push localhost:5000/myfirstimage

Pull镜像myfirstimage

sudo docker pull localhost:5000/myfirstimage

停止registry并remove所有数据

sudo docker stop registry && docker rm -v registry

**理解拉动镜像命令的命名含义**：

sudo docker pull ubuntu

该命令表示从Docker Hub上pull一个名为*ubuntu*的镜像，完整的命令如下：

sudo docker pull docker.io/library/ubuntu

如下命令的含义为：让Docker去位于myregistrydomain:port的registry中pull名为foo/bar的镜像

sudo docker pull myregistrydomain:port/foo/bar

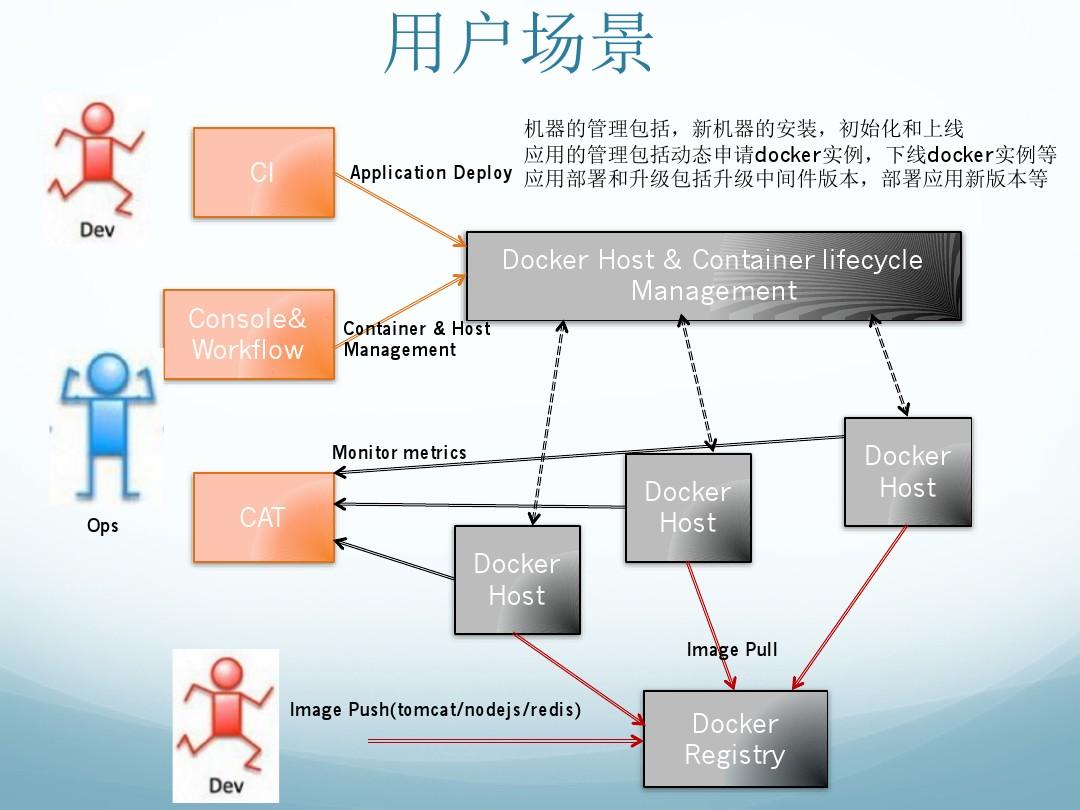
**Action2：阅读官方文档，理解Registry使用场景**

**Time：0.5h**

**使用场景**：

假设此时我们有一个Registry已经集成在完整的CI/CD系统中。在一个典型的工作流中，会有一个提交到源码控制系统的变更请求会触发CI系统的一次build。如果build成功后，会push一个新的镜像到Registry中。随后Registry将会推送一条通知消息给部署系统，告知新镜像状态为可用。

下图源为网络上查找，远比上述使用场景全面和复杂，但对帮助理解Registry有管中窥豹之用

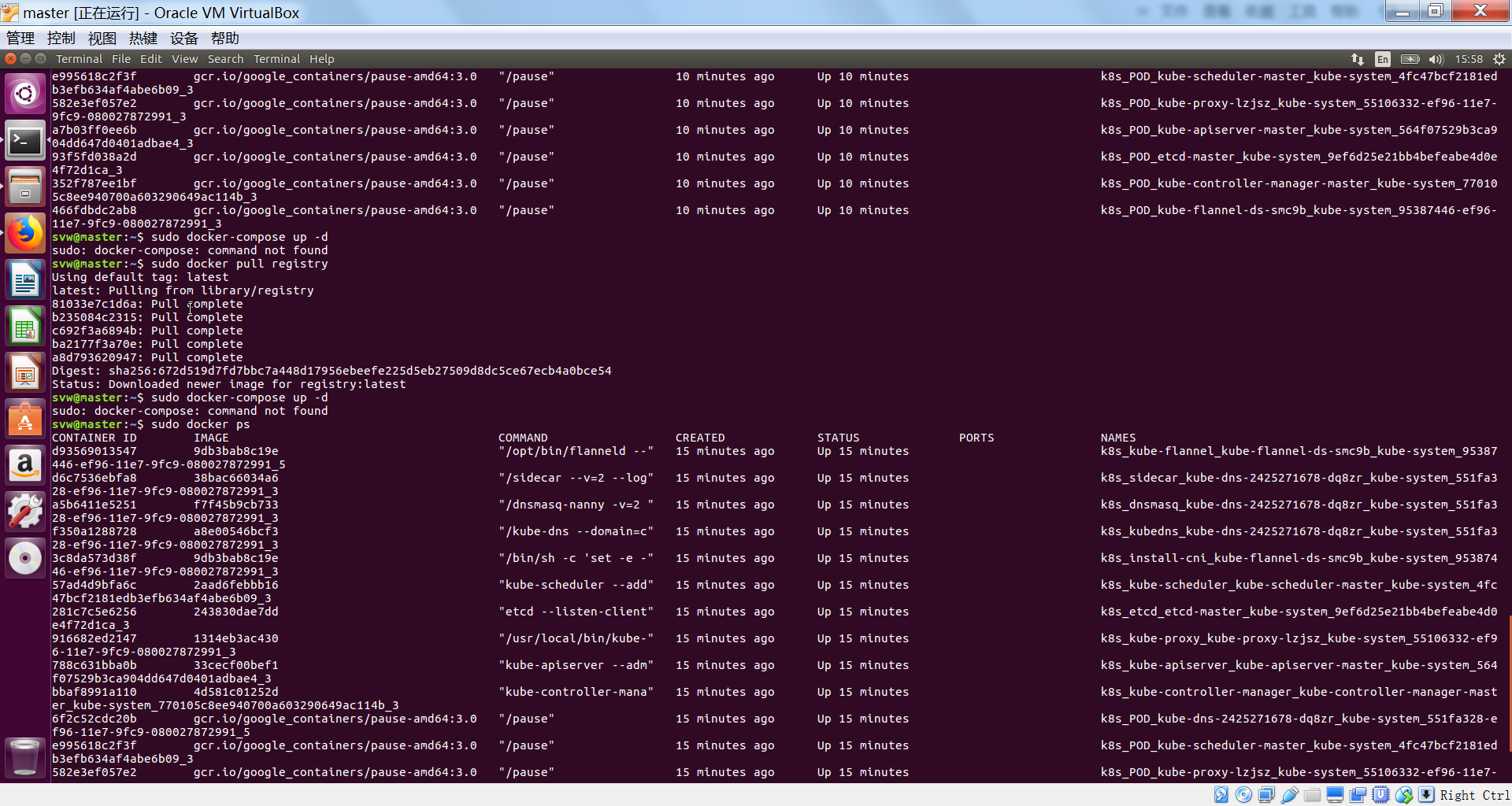


**Action3：部署一个Registry Server，启动一个本地registry**

**Time：0.5h**

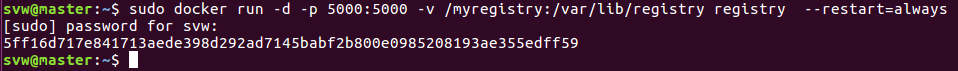
1. 安装或更新Registry：

sudo docker pull registry



1. 创建本地Registry

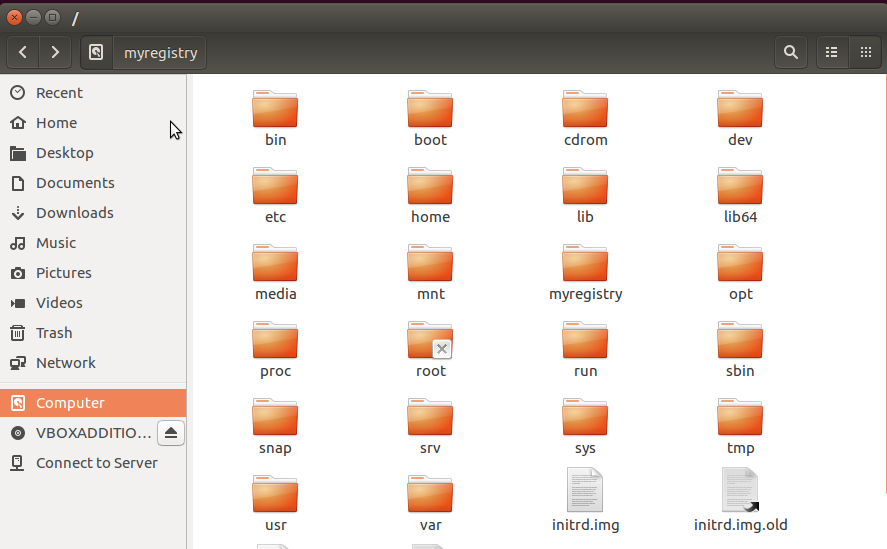
docker run -d -p 5000:5000 -v /myregistry:/var/lib/registry registry --restart=always



-d后台运行

-p指定端口

-v把registry的镜像路径/var/lib/registry映射到本机的/myregistry

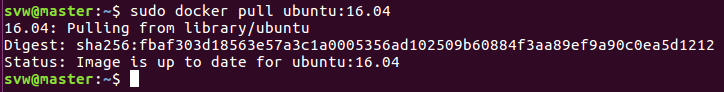


**Action4：从Docker Hub复制镜像到registry**

**Time：2h**

1. 从Docker Hub上pull镜像到本地Registry

sudo docker pull ubuntu:16.04

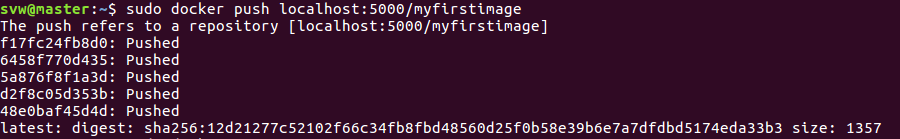


1. 将镜像tag成*localhost:5000/my-ubuntu，*此操作将为已存在的镜像创建一个tag，下次push的时候可以直接调用tag

sudo docker tag ubuntu:16.04 localhost:5000/my-ubuntu

1. Push镜像到本地registry

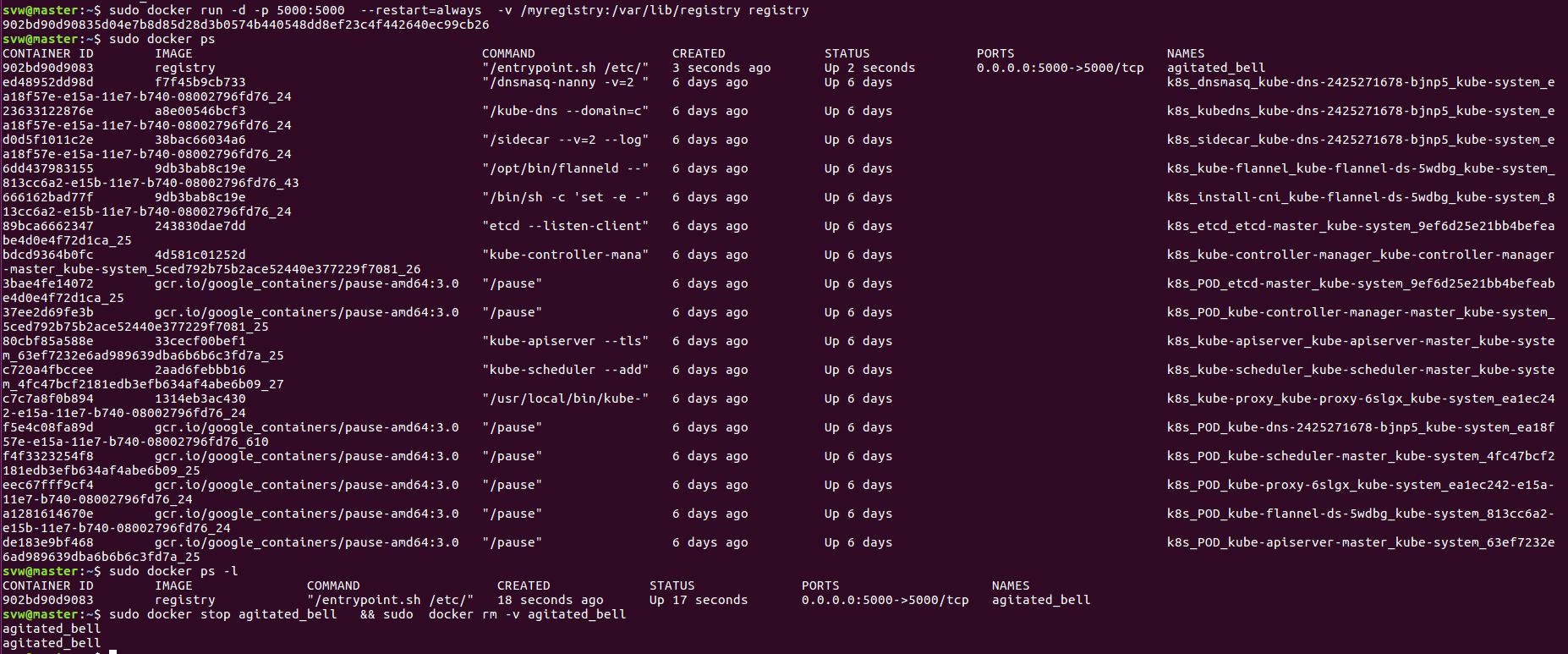
sudo docker push localhost:5000/my-ubuntu



**Action5：停止一个本地并删除registry**

**Time：0.5h**

**命令如下:**



**253\* sudo docker run -d -p 5000:5000 --restart=always -v /myregistry:/var/lib/registry registry 254 sudo docker ps**

**255 sudo docker ps -l**

**256 sudo docker stop agitated\_bell && sudo docker rm -v agitated\_bell**

**Action6：自制镜像推送到registry**

**Time：1h**

**参考文档**

****

**Action7：利用yml文件部署regitsry**

**Time：1h**

**安装 docker-compose**

**编辑** docker-compose.yml 文件

registry:

restart: always

image: registry:2

ports:

- 5001:5001

volumes:

- /path/data:/var/lib/registry

**sudo apt install docker-compose**

**sudo docker-compose up -d**