中国科学技术大学
University of Science and Technology of China

容器化部署CARSI IdP

中国科学技术大学 ·网络信息中心

目录

Table of Contents

01 虚拟机、容器与Docker
VM, Container & Docker

02 CARSI IdP部署实例
Dockerized CARSI IdP Deployment

03 调试和运维
Debug & Operate

04 优化目标
Optimize

01

虚拟机、容器与Docker

一般应用部署（虚拟机）

准备虚拟机 → 安装环境 → 配置参数 → 部署运行

- 升级维护
  - 需要定期检查，执行升级
  - 可能会需要伴随升级需要重新配置环境、参数
- 重复部署
  - 人工再做一遍
  - 将过程记录/配置，到新环境自动化执行
  - 将配置好环境打包交付，相当于定制的操作系统
- 软件版本冲突
  - 额外配置
  - 通过新的虚拟机进行隔离

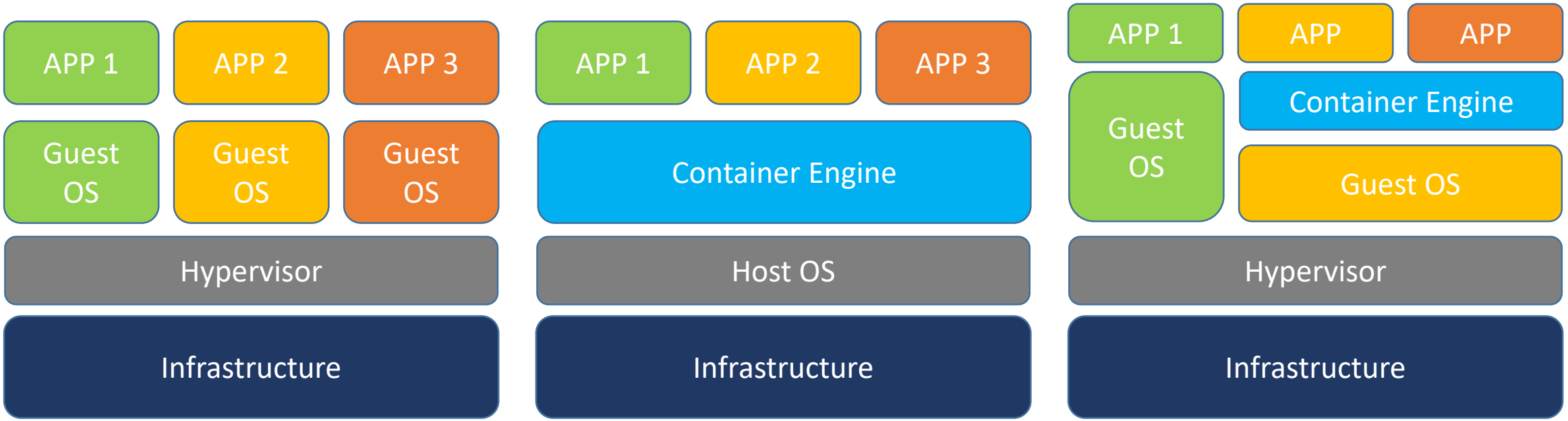
重复工作的问题

便捷性的问题

隔离性的问题

虚拟机、容器与Docker

- 虚拟机是对硬件的虚拟化，容器是对操作系统的虚拟化
- 容器技术有多种实现，Docker是目前最常用的
- Docker为一般应用部署遇到的问题提供了很好的解决方案。



Hypervisor-based Virtualization　　Container Virtualization　　VM + Container

虚拟机、容器与Docker

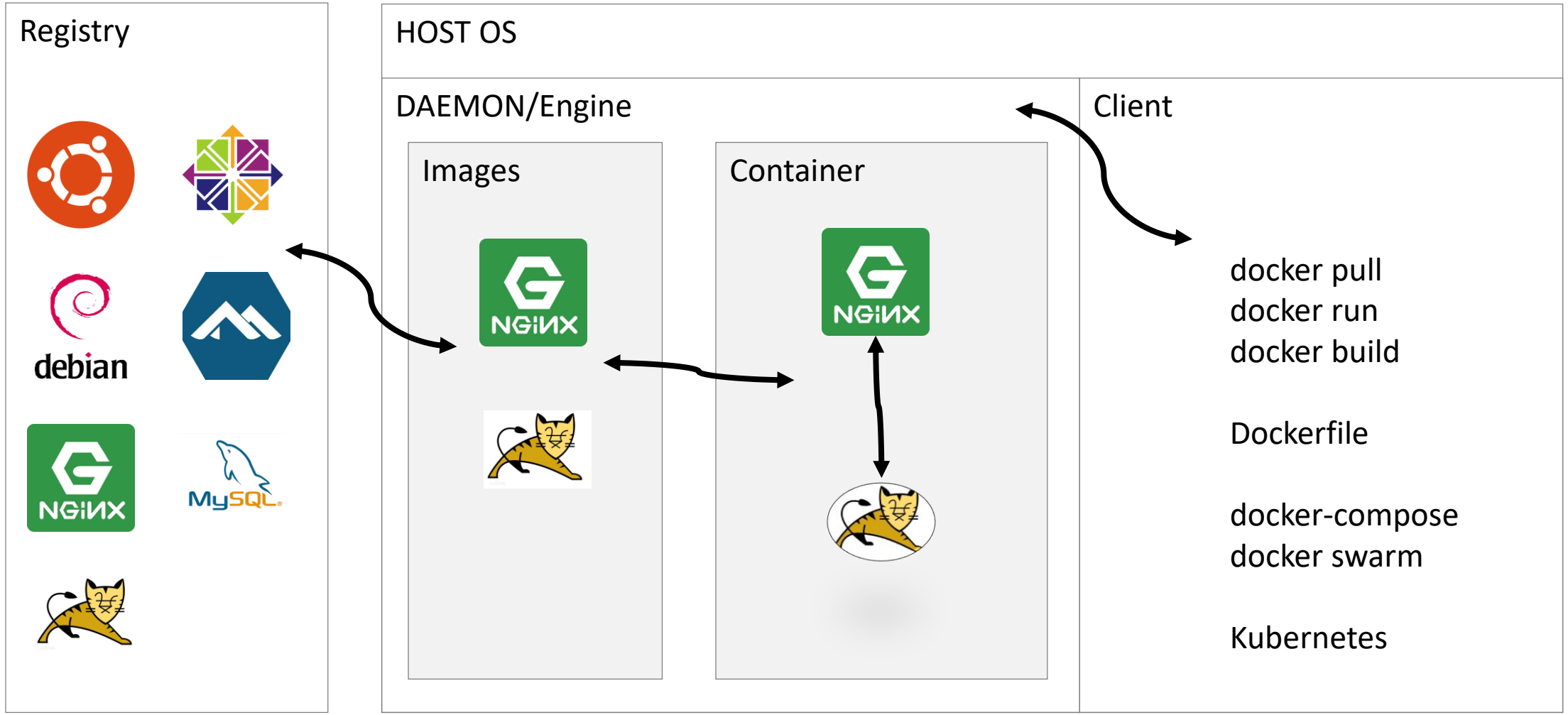
- 自2013年底开源以来，Docker迎来了爆发式的增长
  - 2014年10月，微软在Windows Server集成了Docker引擎
  - 2014年11月，Amazon在EC2中支持Docker
  - 2014年12月，IBM与Docker达成战略合作以促进Docker在IBM Cloud的集成
  - 2016年5月起，统计显示Cisco, Google, Huawei, IBM, Microsoft & Red Hat等科技公司都成为Docker社区的主要贡献者
  - 2019年5月，Docker开始WSL的支持工作

https://en.wikipedia.org/wiki/Docker_(software)

- 数据显示(Apr. 2018)
  - 23.4%的软件行业使用Docker部署应用，这一数据从2015年开始每年以3-5%在增长
  - Docker部署的主机份额约21%，这一数据自2015年以来以每年5%的速度增长
  - 在拥有超过1000+主机的组织中，超过47%采用Docker，另有超过30%已经开始行动
  - 使用docker部署的应用规模过去一年增长了75%
  - 超过50%的Docker应用是集群部署

https://www.datadoghq.com/docker-adoption/

Docker简介


Registry
debian
NGINX
MySQL
HOST OS
DAEMON/Engine
Images
NGINX
Container
NGINX
Client
docker pull
docker run
docker build
Dockerfile
docker-compose
docker swarm
Kubernetes

02

CARSI IdP部署实例

CARSI IdP部署实例

- CARSI采用的是Shibboleth的联盟身份认证解决方案
- Shibboleth社区已经有成员发布了Docker化的部署方案
  - https://github.com/jtgasper3/dockerized-idp-testbed
- 基于社区版本和CARSI文档，我们在中科大进行了CARSI IdP的部署
- 中科大的单点登陆采用的是CAS 3.0版本协议

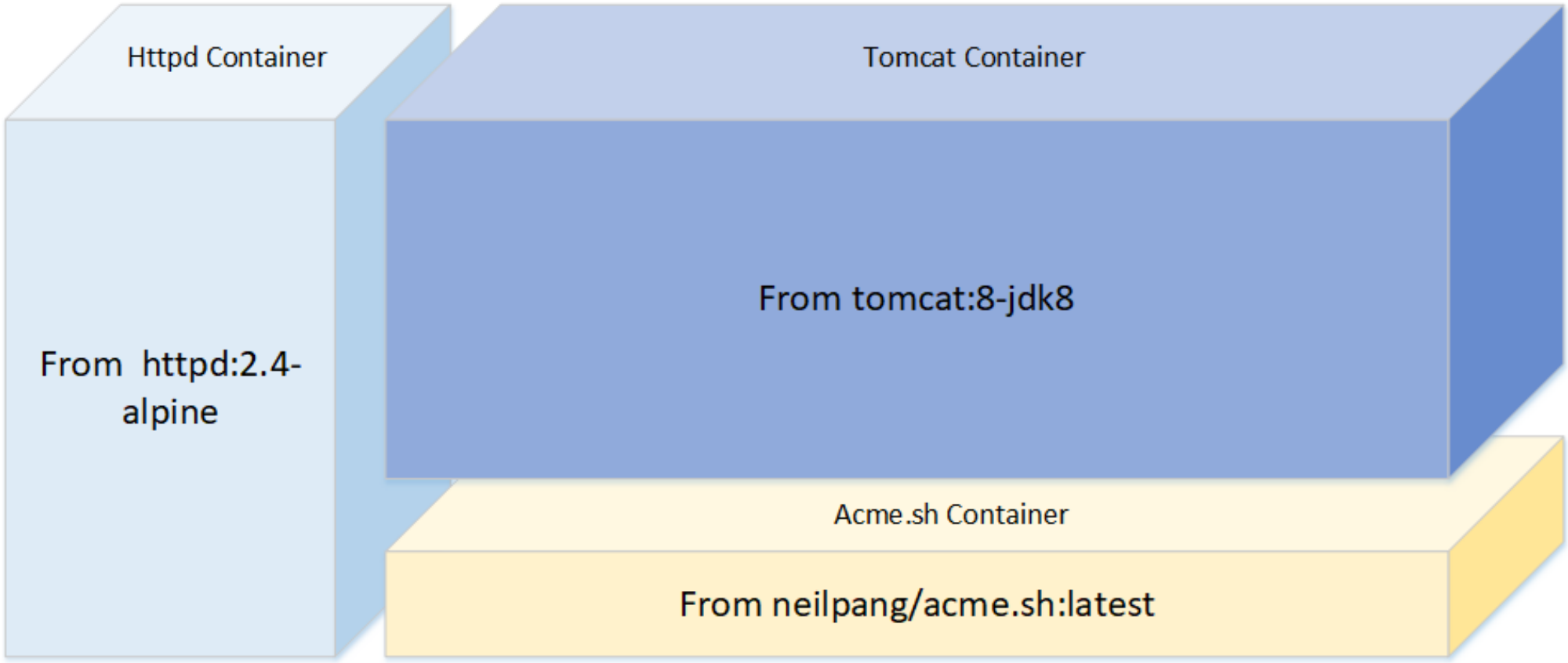
CARSI IdP部署实例


```
.
├── changelog
├── docker-compose.yml
├── Dockerfile
├── httpd
│   ├── conf
│   ├── conf.d
│   └── index.html
├── idp-config
│   ├── auditlog.sh
│   ├── conf
│   ├── credentials
│   ├── dist
│   ├── edit-webapp
│   ├── flows
│   ├── messages
│   ├── metadata
│   ├── root
│   └── views
├── secrets
│   └── idp
└── tomcat
    ├── idp.xml
    ├── server.xml
    └── tomcat.service
```

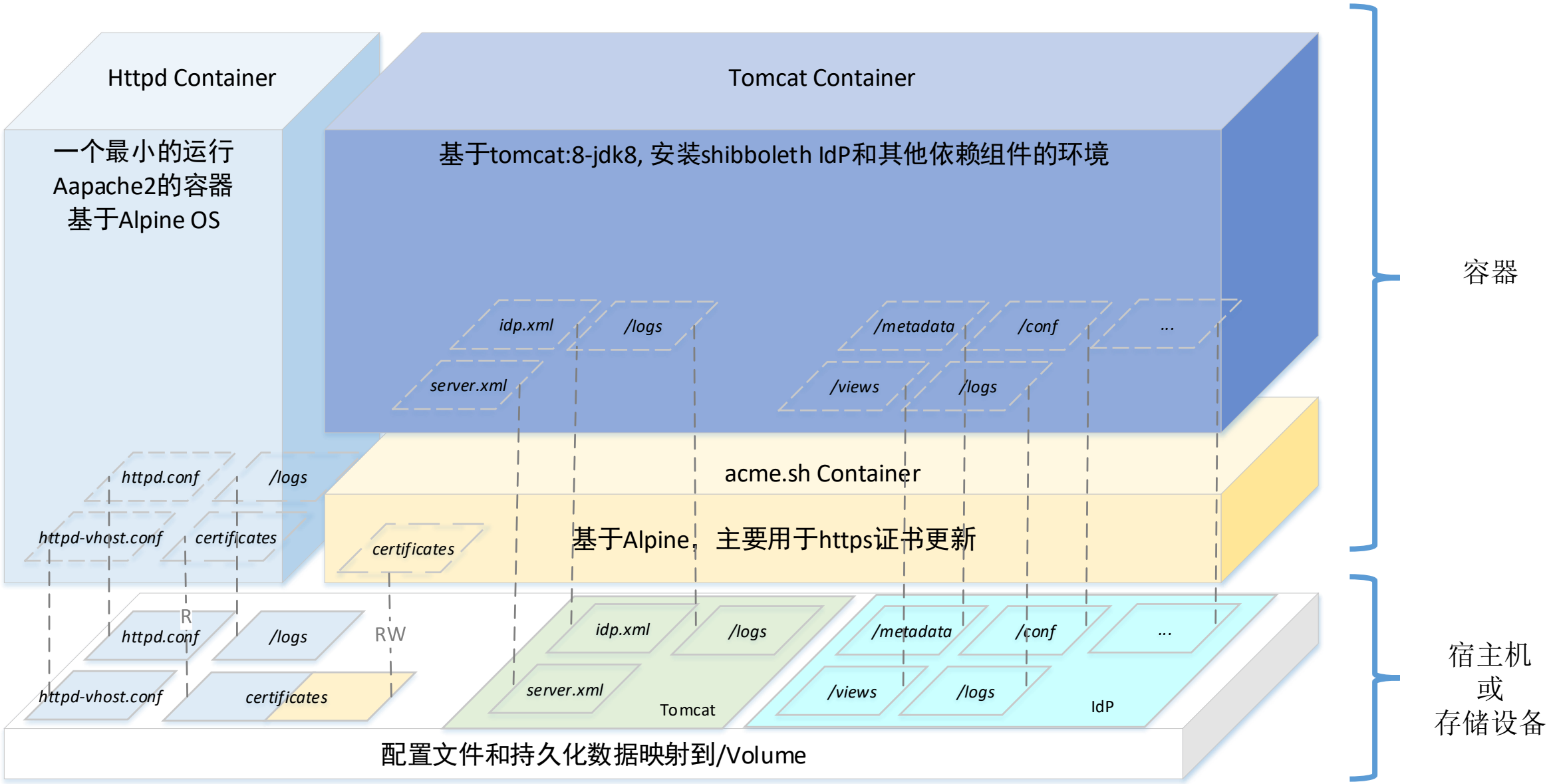
```yaml
version: "3.1"

services:
    idp-jhl:
        build:
            context: ./
            dockerfile: Dockerfile
        contain
        privile
        expose:
        ports:
            - 4
            - 8
            - 8
        network
            - b
        secrets
            - s
            - s
            - s
            - s
            - s
        volumes
            - /
            - /
            - /
            #ht
            - .
            - .
            - .
            - .
            - .
            #ht
            -
            #sh
```

```dockerfile
RUN yum -y update \
    && yum -y install wget tar which \
    && yum -y clean all \
    && yum -y install httpd \
    && yum -y install mod_ssl

ENV jdk_install_version=jdk-8u211-linux-x64 \
    jdk_version=1.8.0_211\
    tomcat_version=8.5.38

#安装JDK
RUN mkdir /usr/local/jdk
RUN wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-s
ib/$jdk_install_version.tar.gz
RUN tar -zxvf /usr/local/jdk/$jdk_install_version.tar.gz -C /usr/local/jdk
RUN rm -rf /usr/local/jdk/$jdk_install_version.tar.gz

#安装tomcat
RUN mkdir /usr/local/tomcat
RUN wget -P /usr/local/tomcat http://mirrors.hust.edu.cn/apache/tomcat/tomcat-8/v$tom
RUN tar -zxvf /usr/local/tomcat/apache-tomcat-$tomcat_version.tar.gz -C /usr/local/to
RUN rm -rf /usr/local/tomcat/apache-tomcat-$tomcat_version.tar.gz

#设置环境变量
ENV JAVA_HOME /usr/local/jdk/jdk$jdk_version
ENV CLASSPATH $JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
ENV CATALINA_HOME /usr/local/tomcat/apache-tomcat-$tomcat_version
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/lib:$CATALINA_HOME/bin
```

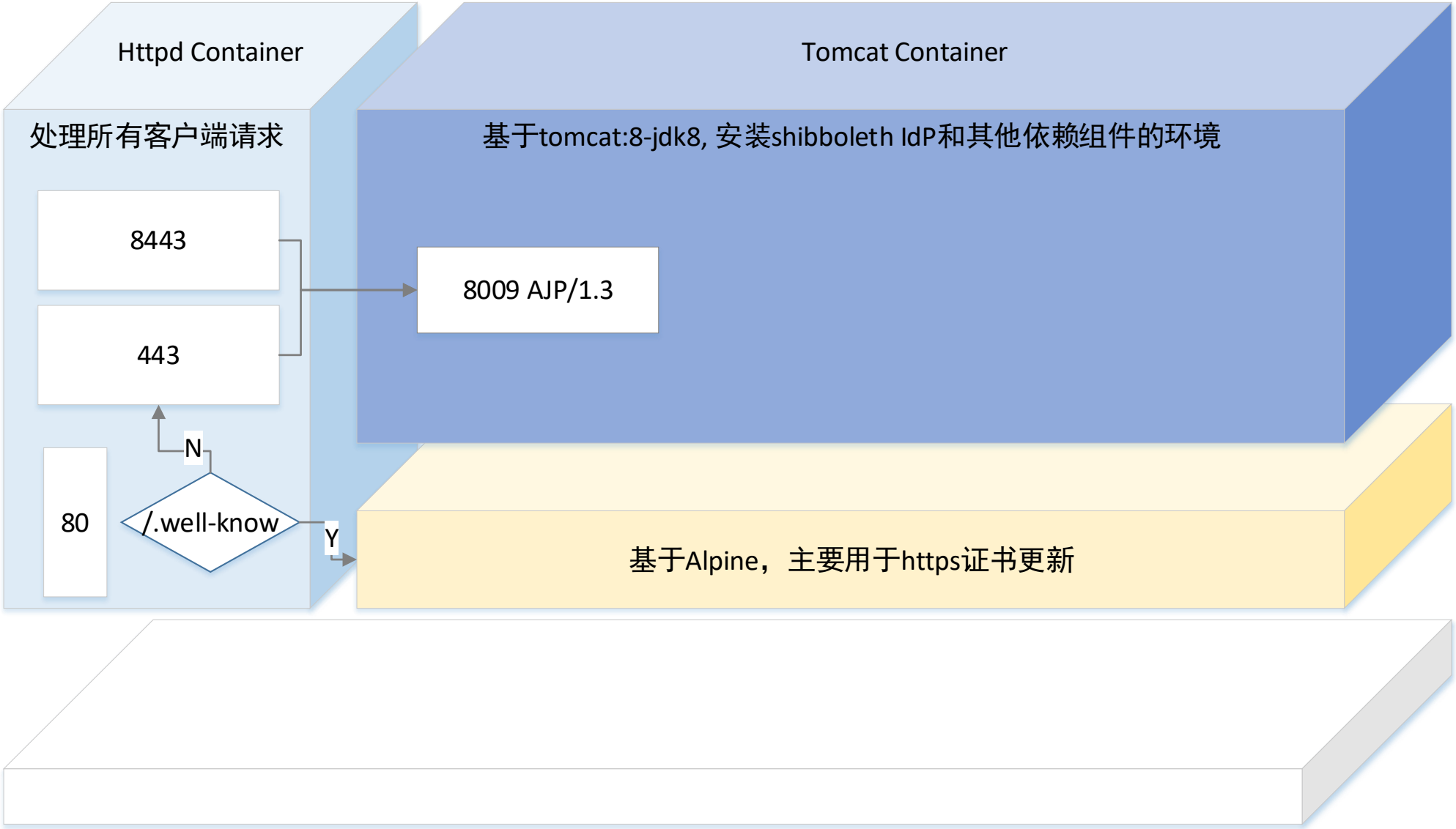
Image ｜ Dockerfile 和 Container

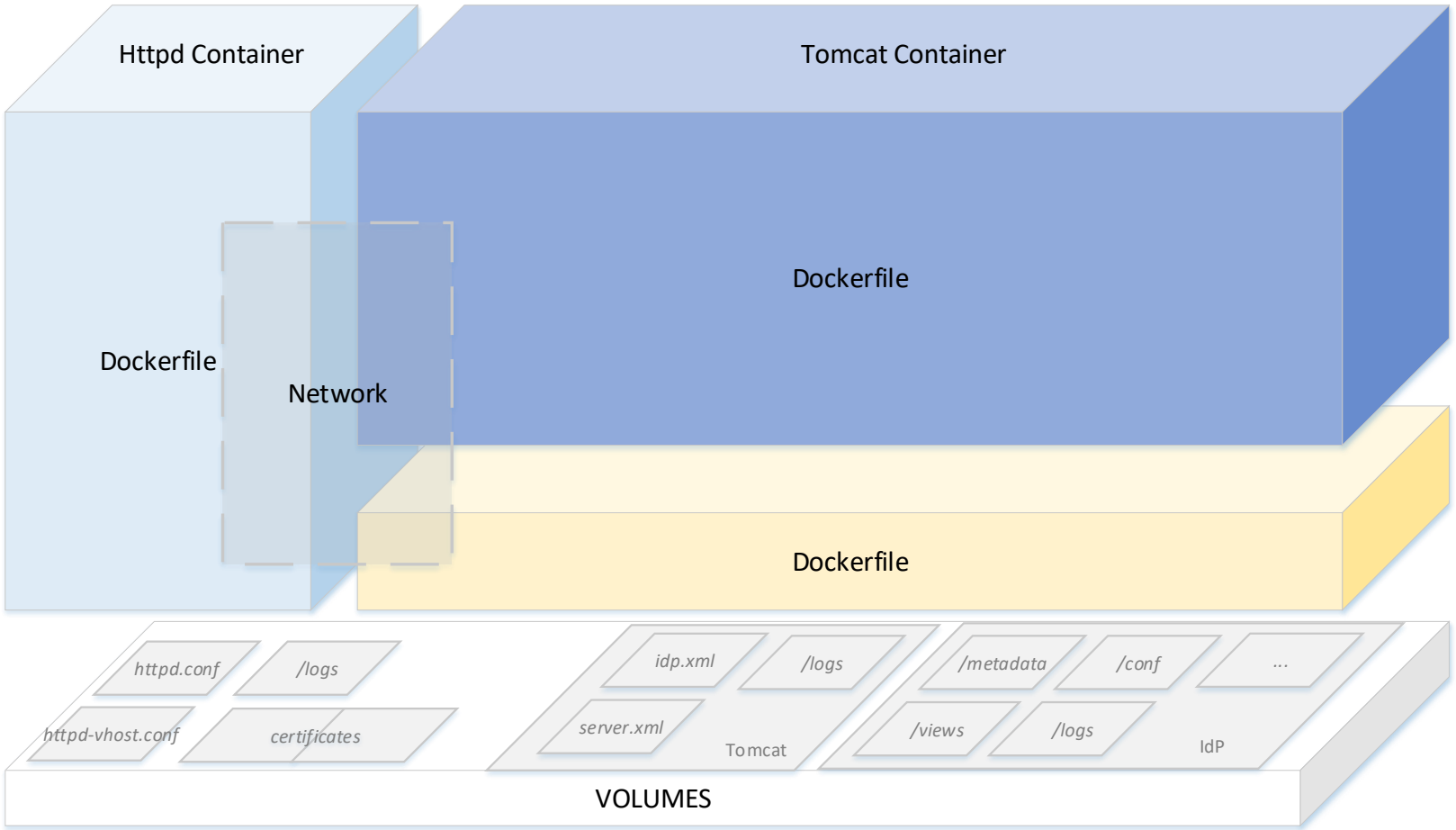


配置和持久化


Httpd Container
一个最小的运行
Aapache2的容器
基于Alpine OS
Tomcat Container
基于tomcat:8-jdk8, 安装shibboleth IdP和其他依赖组件的环境
idp.xml
/logs
/metadata
/conf
...
server.xml
/views
/logs
acme.sh Container
基于Alpine，主要用于https证书更新
certificates
httpd.conf
/logs
httpd-vhost.conf
certificates
R
RW
idp.xml
/logs
/metadata
/conf
...
server.xml
Tomcat
/views
/logs
IdP
配置文件和持久化数据映射到/Volume
容器
宿主机
或
存储设备

Network


Httpd Container
Tomcat Container
处理所有客户端请求
基于tomcat:8-jdk8, 安装shibboleth IdP和其他依赖组件的环境
8443
443
8009 AJP/1.3
N
80
/.well-know
Y
基于Alpine，主要用于https证书更新



docker-compose.yml

- Image & Dockerfile
- Containers & Services
- Volumes
- Network

```
uscai@saisi:~/wangfeng$ tree -L 2
.
├── carsi-idp-cas-config
│   ├── auditlog.sh
│   ├── conf
│   ├── credentials
│   ├── dist
│   ├── edit-webapp
│   ├── flows
│   ├── messages
│   ├── metadata
│   ├── root
│   └── views
├── docker-compose.yml
├── Dockerfile
├── dspre
│   ├── carsifed-metadata.xml
│   └── dsmeta.pem
├── httpd
│   ├── conf
│   ├── htdocs
│   └── logs
├── init-idp.sh
├── README.md
└── tomcat
    ├── idp.xml
    └── server.xml
```

03

调试和运维

调试和运维

- 调试
  - 容器和服务启动问题通过docker logs [container name]来查看
  - 服务运行和运行异常通过查阅httpd、tomcat和shibboleth的日志来追踪
  - 应用对接报错通过邮件联系SP并寻求CARSI帮助
- 运维
  - docker-compose up|stop|rm & docker start|stop|rm [container name]管理容器
  - docker container|network|volume [COMMAND]查阅容器、网络和卷
  - docker inspect [container|image|network|volume name] 获取元数据
  - 数据备份只需要备份Dockerfile、docker-compose.yml和volume数据卷
  - 监控使用第三方监控如zabbix、 prometheus等

04

优化和目标

优化目标

- 将部署过程中，除去生成证书、metadata提交和更新、页面自定义之外的工作通过Docker标准化交付，使得那些采用LDAP、CAS和OAuth的组织加入CARSI时，只需要通过数行命令就可以完成IdP的运行，从而加速CARSI在国内的普及。

```
git clone …
docker run –it -v $(pwd):/ext-mount --rm carsi_idp:latest init.sh
…
…
…
docker-compose up –d
```

谢谢大家

王峰 · wf0229@ustc.edu.cn