# 基于容器标签token的网络访问控制

## 什么是docker

1. docker介绍

Docker是一个开源的应用容器引擎，它基于go语言开发，并遵从Apache2.0开源协议。使用Docker可以让开发者封装他们的应用以及依赖包到一个可移植的容器中，然后发布到任意的 Linux 机器上，也可以实现虚拟化。Docker 容器完全使用沙箱机制，相互之间不会有任何接口，这保证了容器之间的安全性。

1. 容器的实现原理
2. Rootfs

简单介绍一下，rootfs可以先理解为在Linux操作系统中另外创建一个独立的文件系统

1. Cgroup

cgroups，其名称源自控制组群（control groups）的简写，是Linux内核的一个功能，用来限制、控制与分离一个进程组的资源（如CPU、内存、磁盘输入输出等）

1. Namespace

Linux的命名空间机制提供了一种资源隔离的解决方案。PID,IPC,Network等系统资源不再是全局性的，而是属于特定的Namespace。Linux Namespace机制为实现基于容器的虚拟化技术提供了很好的基础，LXC（Linux containers）就是利用这一特性实现了资源的隔离。不同Container内的进程属于不同的Namespace，彼此透明，互不干扰。

Namespace是对全局系统资源的一种封装隔离，使得处于不同namespace的进程拥有独立的全局系统资源，改变一个namespace中的系统资源只会影响当前namespace里的进程，对其他namespace中的进程没有影响。

1. 容器引擎

容器创建销毁则都是通过容器引擎调用Linux内核的API接口来申请不同的命名空间和进程的cpu以及内存，可以立即对以上资源的管理系统

1. 容器和虚拟机有什么区别

最重要的区别就是容器共用宿主主机的操作系统和内核，而虚拟机有自己的操作系统和虚拟内核。

1. 操作系统、操作系统发行版、操作系统内核

## 为什么要做容器的访问控制

随着云计算的发展 docker 技术在项目部署中扮演者重要角色，因为docker容器的便利性和强大的移植性，所以目前大部分的web和数据库应用都部署在docker容器中，在一些比较大的互联网科技公司中容器数量更是达到成千上万个；那么这种情况下容器安全就愈发显得格外重要。而因为每一个容器都有自己独立的IP地址，而且所有容器IP都属于docker引擎虚拟出来的子网范围，如果一个容器被入侵那么其他容器就也有被入侵的风险从而导致安全事件，容器间的网络访问控制则有效的预防了容器被入侵的风险！

## 基于标签的访问控制的特点是什么

传统做网络访问控制的技术都是基于五元组（目的地IP、目的地端口、协议、通信协议、源IP、源端口），而由于容器内部IP的不确定性，容器IP地址随着容器销毁或者重建IP随机生成，以至于传统的访问控制技术不再适用于容器；基于标签token的访问控制不会有因为IP的变化而导致策略失效的问题。

## 实现思路

# 联邦学习在网络评论情感分析上的应用

## 什么是联邦学习

## 联邦学习的优点是什么

## 解决了什么问题

## 实现思路