**总结2020.11.5至2020.11.10大概工作：**

1. 浅显了解cs业务流程
2. 环境部署，安装vm、在vm安装centos7、在centos7上安装docker、在清华docker仓库下载nginx镜像，使用该镜像run一个nginx-container，并在centos内可访问该nginx服务。
3. 初步熟悉dockerfile build 镜像与docker-compose build and up container。

**具体记录（基于写报告时想起什么写什么，内容可能有误，希望可以指出）：**

1. 关于docker概念的个人理解

类似一个沙盒，分镜像云库、镜像、容器三部分。拉取一些image在docker内，就可以基于一个image运行多个container。Container可以是很多东西，引擎、程序、系统等等，且每个container相互隔离（docker有点像vm，image就是镜像，container就是实际安装的虚拟机。相当于超级简化版的虚拟机技术）。image就像酒店房间，里面床，空调，热水器等都已经准备好。客人暂时租用其中一间后，需要带上个人用品（添加的环境变量，增加修改的数据等）。用什么就带什么。客人走后就会清空里面所有的个人用品，仅保留基本设备（镜像的原配置）。

1. Docker的具体操作及命令

Version--查看查看docker版本

Pull--在docker云库下载image

Login--登录个人云库账号

Logout--退出个人云库账号

Push--上传个人image到云库，前提要login

Run--基于一个image运行一个container。（-i：可交互、-t生成一个虚拟的终端、-d：后台运行container、-p：宿主机端口号映射到容器端口，0.0.0.0：9999:80、-P：随机生成一个端口号映射到container端口、-v：挂载目录到container，使container内的数据持久化，从而不会发生因为退出docker导致container内数据丢失的情况。分为本地目录挂载[本地路径：容器内路径]与volume目录挂载[volume名称:容器内路径]、--name：为运行的容器起一个名字，等等）

Exe--进入container内部，不会因为退出container内部导致container停止运行。（-i：可交互、-t：虚拟终端）

Attc--进入container内部，与exec几近相同，但是会在退出container时导致container停止运行。

Ps--查看所有当前运行的container信息。（-a：可查看所有的contianer信息，不管有没有在运行。）

Image--查看所有镜像信息。

Start--启动一个或多个container。

Stop--停止一个或多个container。

Build--使用dockerfile建立一个image。

Commi--打包container成new image。

Rm--删除一个或多个container。（-f：不管container是否运行，强制删除container）

Rmi--删除一个或多个image。

Copy--复制本地文件到container内。

Kill--杀死一个正在运行的container，应该是container不能正常关闭时使用。

1. Docker-compose的使用

Dockerfil--使用该文件建立一个image，并写入一些具体操作命令。

Docker-compose.yml--该文件为配置具体的container启动参数。（version：版本、services：服务集合，内部可以包含许多个服务、web：具体的服务、build：使用dockerfile建立一个image，使用上下文路径或者绝对路径、ports：配置服务端口、depend\_on:依赖启动服务，如a服务依赖b服务，a服务就会在b服务启动后启动，但不会在b服务完全启动后再启动、volume：挂载volume目录，为数据持久化。

Docker-compose命令：up-启动container，-d，-t，-i等与docker几乎相同、down-停止container、ps-查看container信息、等等

**20-11-1了解k8s与部署：**

K8s：Kubernetes是一个轻便的和可扩展的开源平台，用于管理容器化应用和服务。通过Kubernetes能够进行应用的自动化部署和扩缩容。

根据<https://www.bilibili.com/read/cv4219112/>进行安装

Record1：ETCD是用于共享配置和服务发现的分布式，一致性的KV存储系统。使用Raft协议来维护每个节点的状态一致性。Raft遵循选主算法来维持各个节点的一致性，每个节点都必定处于leader、follower、candidate中的一个状态。

Record2：服务发现概念，服务发现有三个角色，服务提供者、服务消费者和服务中介。服务中介是联系服务提供者和服务消费者的桥梁。服务提供者将自己提供的服务地址注册到服务中介，服务消费者从服务中介那里查找自己想要的服务的地址，然后享受这个服务。服务中介提供多个服务，每个服务对应多个服务提供者。

Problem1：安装kubernetes时发现k8s的docker的版本与之前安装过的docker版本冲突，导致安装k8s失败。



Solution：卸载之前安装的docker，再重新安装。

删除docker语句--yum -y remove containerd.io.x86\_64 docker-ce.x86\_64 docker-ce-cli.x86\_64

（注：docker数据需要手动清理，数据不会因为卸载docker而被清理。重新安装的docker可以使用之前的docker的数据。）

Record3：如何查询网卡等信息，centos7废弃了ipconfig命令，改用ip addr，ip link等。

Record4：Flannel是 CoreOS 团队针对 Kubernetes 设计的一个覆盖网络（Overlay Network）工具，其目的在于帮助每一个使用 Kuberentes 的 CoreOS 主机拥有一个完整的子网。

未完成，再学习一下k8s架构再进行部署。。。。。。