

2019 年全国职业院校技能大赛

GZ-2019030 云计算技术与应用（高职组） 赛题库

第一部分：IaaS 云计算基础架构平台

任务一、IaaS 云平台搭建

基础环境：

- 1.使用命令行方式设置主机名，防火墙以及 SELinux 设置如下：
 - (1) 设置控制节点主机名 `controller`；计算节点主机名：`compute`。
 - (2) 各个节点关闭防火墙，设置开机不启动。
 - (3) 设置各个节点 `selinux` 状态为 `permissive`。
- 2.使用命令查询控制/计算节点的主机名。
- 3.使用命令查询控制/计算节点 `selinux` 的状态。
- 4.在控制节点上通过 SecureFX 上传两个镜像文件 `CentOS-7-x86_64-DVD-1511.iso`，`XianDian-IaaS-v2.2.iso` 到 `opt` 下，使用命令创建 `opt` 下两个目录，并将以上镜像文件分别挂载到上述两个目录下，并使用命令查看挂载的情况（需显示挂载的文件系统类型和具体的大小）。
- 5.在控制节点上通过 SecureFX 上传两个镜像文件 `CentOS-7-x86_64-DVD-1511.iso`，`XianDian-IaaS-v2.2.iso` 到 `opt` 下，通过命令行创建两个目录，并将以上镜像文件分别挂载到上述两个目录下。
- 6.配置控制节点本地 `yum` 源文件 `local.repo`，搭建 `ftp` 服务器指向存放 `yum` 源路径；配置计算节点 `yum` 源文件 `ftp.repo` 使用之前配置的控制节点 `ftp` 作为 `yum` 源，其中的两个节点的地址使用主机名表示。使用 `cat` 命令查看上述控制/计算节点的 `yum` 源全路径配置文件。
- 7.在控制节点和计算节点分别安装 `iaas-xiandian` 软件包，完成配置文件中基本变量的配置，并根据提供的参数完成指定变量的配置。

Mysql 搭建:

- 1.根据平台安装步骤安装至数据库服务，使用一条命令安装提供的 `iaas-install-mysql.sh` 脚本并查看脚本运行的时间。
- 2.使用 `root` 用户登录数据库，查询数据库列表信息。
- 3.使用 `root` 用户登录数据库，使用 `mysql` 数据库，查询所有表的信息。
- 4.使用 `root` 用户登录数据库，使用 `mysql` 数据库，查询所有表的信息，并查询表 `user` 中的特定的信息。

Keystone 搭建:

- 1.按要求安装完 `keystone` 脚本后，在数据库中查询 `keystone` 用户的远程访问权限信息。
- 2.列出数据库 `keystone` 中的所有表。
- 3.使用相关命令，查询角色列表信息。
- 4.使用相关命令，查询 `admin` 项目信息。
- 5.使用相关命令，查询用户列表信息。
- 6.使用相关命令，查询 `admin` 用户详细信息。
- 7.使用相关命令，查询服务列表信息。
- 8.使用一条命令将 `keystone` 的数据库导出为当前路径下的 `keystone.sql` 文件，并使用命令查询文件 `keystone.sql` 的大小。

Glance 搭建:

- 1.根据平台安装步骤安装至镜像服务，在控制节点使用提供的脚本 `iaas-install-glance.sh` 安装 glance 组件。使用镜像文件 `CentOS_7.2_x86_64_XD.qcow2` 创建 glance 镜像名为 `CentOS7.2`，格式为 `qcow2`。
- 2.使用相关命令查询镜像列表，并查询 `CentOS7.2` 镜像的详细信息。
- 3.使用相关命令，在一条命令中查询 glance 组件中所有服务的状态信息。

Nova 搭建:

- 1.根据平台安装步骤安装至 nova 计算服务，在控制节点使用提供的脚本 `iaas-install-nova-controller.sh`、在计算节点使用提供的脚本 `iaas-install-nova-compute.sh`，安装 nova 组件。
- 2.使用相关命令查询计算节点虚拟机监控器的状态。
- 3.使用相关命令查询 nova 服务状态列表。
- 4.使用相关命令查询网络的列表信息。
- 5.使用相关命令查询 nova 资源使用情况的信息。

Neutron 搭建:

- 1.根据平台安装步骤安装至 neutron 网络服务，在控制节点和计算节点通过提供的 neutron 脚本，完成 neutron 服务在控制节点和计算节点的安装，并配置为 GRE 网络。
- 2.根据平台安装步骤安装至 neutron 网络服务，在控制节点和计算节点通过提供的 neutron 脚本，完成 neutron 服务在控制节点和计算节点的安装，并配

置为 VLAN 网络。

3.使用相关命令查询网络服务的列表信息，并以下图的形式打印出来。

```
+-----+-----+-----+
| binary                | agent_type          | alive |
+-----+-----+-----+
```

4.使用相关命令查询网络服务的列表信息中的“binary”一列。

5.使用相关命令查询网络服务 DHCP agent 的详细信息。

6.使用 ovs-vsitchd 管理工具的相关命令查询计算节点的网桥列表信息。

7.使用 ovs-vsitchd 管理工具的相关命令查询控制节点的网桥 br-ex 的端口列表信息。

8.创建云主机外部网络 ext-net，子网为 ext-subnet，云主机浮动 IP 可用网段为 192.168.200.100 ~ 192.168.200.200，网关为 192.168.200.1。创建云主机内部网络 int-net1，子网为 int-subnet1，云主机子网 IP 可用网段为 10.0.0.100 ~ 10.0.0.200，网关为 10.0.0.1；创建云主机内部网络 int-net2，子网为 int-subnet2，云主机子网 IP 可用网段为 10.0.1.100 ~ 10.0.1.200，网关为 10.0.1.1。添加名为 ext-router 的路由器，添加网关在 ext-net 网络，添加内部端口到 int-net1 网络，完成内部网络 int-net1 和外部网络的连通。

9.使用相关命令查询所创建路由器的详细信息。

10.使用相关命令查询所创建子网的列表信息，并查看内网子网的详细信息。

11.使用相关命令查询所创建网络的列表信息。

Dashboard 搭建:

1.通过脚本 iaas-install-dashboard.sh 安装 dashboard 组件，使用 curl 命令查询

网址 <http://192.168.100.10/dashboard>。

2.通过脚本 `iaas-install-dashboard.sh` 安装 `dashboard` 组件，通过 `chrome` 浏览器使用 `admin` 账号登录云平台网页，进入管理员菜单中的系统信息页面。

Heat 搭建：

- 1.在控制节点使用提供的脚本 `iaas-install-heat.sh` 安装 `heat` 组件。
- 2.使用 `heat` 相关命令，查询 `stack` 列表。
- 3.从考试系统附件下载 `server.yml` 文件，通过命令行使用 `server.yml` 文件创建栈 `mystack`，指定配置参数为镜像 `CentOS7.2`、网络 `int-net2`。
- 4.查询栈 `mystack` 状态为 `CREATE_COMPLETE` 的事件详细信息。
- 5.查询栈 `mystack` 的事件列表信息。

Trove 搭建：

- 1.在控制节点配置提供的脚本 `iaas-install-trove.sh`，使其连接的网络为 `int-net1`，安装数据库 `trove` 服务，完成 `trove` 服务在控制节点的安装。完成后查询所有的数据库实例列表。
- 2.在控制节点上传提供的 `MySQL_5.6_XD.qcow2` 到系统内，并创建 `mysql` 的数据库存储类型，使用上传的镜像更新该数据库类型的版本信息和镜像信息。
- 3.在控制节点查创建名称为 `mysql-1`，大小为 `5G`，数据库名称为 `myDB`、远程连接用户为 `user`，密码为 `r00tme`，类型为 `m1.small` 完成后查询 `trove` 列表信息，并查询 `mysql-1` 的详细信息。
- 4.在控制节点查询所有数据的版本信息，完成后查询 `mysql` 数据库的详细信息。

任务二、IaaS 云平台运维

Rabbitmq 运维:

1.按以下配置在云平台中创建云主机，完成本任务下的相关试题后关闭云主机。

云主机:

- (1) 名称: IaaS
- (2) 镜像文件: Xiandian-IaaS-All.qcow2
- (3) 云主机类型: 4cpu、8G 内存、100G 硬盘
- (4) 网络: 网络 1: int-net1, 绑定浮动 IP
网络 2: int-net2

注: 该镜像已安装 IaaS 平台所有可能使用的组件, 用于完成 IaaS 平台相关运维操作题, 必须按以上配置信息配置接入两个网络才能保证云主机运行正常。根据题目要求, 连接相应的云主机或各节点服务器, 进行以下答题。

- 2.使用 rabbitmqctl 创建用户 xiandian-admin, 密码为 admin。
- 3.使用 rabbitmqctl 命令查询所有用户列表。
- 4.使用命令对 xiandian-admin 用户进行授权, 对本机所有资源可写可读权限。
- 5.使用 rabbitmqctl 命令查询集群状态。
- 6.使用命令给 xiandian-admin 用户创建 administrator 角色, 并查询。
- 7.使用命令对用户 xiandian-admin 进行授权, 对本机所有资源可写可读权限, 然后查询 xiandian-admin 用户的授权信息。
- 8.使用 rabbitmqctl 命令, 查看队列信息, 所包含的信息包括 name, arguments, messages, memory。
- 9.通过修改配置文件的方式修改 memcache 的缓存大小, 使用 ps 相关命令查

询 memcahce 进程的信息，将修改的配置文件全路径文件名、修改的参数以及相应的参数值、查询 memcache 进程信息。

10.构建 rabbitmq 集群，并对集群进行运维操作。

Mysql 运维：

1.使用数据库的相关命令查询数据库的编码方式。

2.通过 mysql 相关命令查询当前系统时间。

3.通过 mysql 相关命令，查看当前是什么用户。

4.通过 mysql 相关命令，查看 mysql 的默认存储引擎信息，并查看 mysql 支持的存储引擎有哪些。

5.进入数据库 keystone，通过 user 表和 local_user 表做联合更新，u 用来做 user 表别名，lu 用来做 local_user 表别名，sql 语句更新 neutron 用户的 enabled 状态为 0，更新语句中 user 表在 local_user 前面。

6.进入数据库 keystone，通过 user 表和 local_user 表做联合查询，u 用来做 user 表别名，lu 用来做 local_user 表别名，两表联合查询 nova 用户的 enabled 状态，查询语句中 user 表在 local_user 前面。

7.进入数据库，创建本地用户 examuser，密码为 000000，然后查询 mysql 数据库中的 user 表的特定字段。最后赋予这个用户所有数据库的“查询”“删除”“更新”“创建”的本地权限。

8.登录 iaas 云主机，登录 mysql 数据库，使用 keystone 数据库，查询本地用户表中的所有信息，并按照 id 的降序排序。（关于数据库的命令均使用小写）

MongoDB 运维

- 1.登录 iaas 云主机，登录 MongoDB 数据库，查看数据库，使用 ceilometer 数据库，查看此数据库下的集合，并查询此数据库用户，最后创建一个数据库并查询。
- 2.登录 iaas 云主机，登录 MongoDB 数据库，新建一个数据库，使用这个数据库，向集合中插入数据，最后查询特定的一类数据。
- 3.登录 iaas 云主机，登录 MongoDB 数据库，新建一个数据库，使用这个数据库，向集合中插入数据，插入完毕后，数据进行修改，修改完后，查询修改完的数据。
- 4.登录 iaas 云主机，登录 MongoDB 数据库，新建一个数据库，使用这个数据库，向集合中插入数据（其中某一条数据插入两遍），插入数据完毕后，发现某条数据多插入了一遍需要删除，请使用命令删除多余的一行数据，最后将数据库删除。
- 5.登录 iaas 云主机，登录 MongoDB 数据库，新建一个数据库，使用这个数据库，向集合中插入数据，插入完毕后，查询集合中的数据并按照某关键字的升序排序。
- 6.登录 iaas 云主机，登录 MongoDB 数据库，新建一个数据库，使用这个数据库，向集合中批量插入多条数据，使用 for 循环，定义变量 i=1，插入"_id": i, "name": "xiaoming", "age": "21"。插入数据完毕后，统计集合中的数据条数，然后查询集合中满足特定条件的结果。
- 7.登录 iaas 云主机，使用 mongoimport 命令，将给定的文件，导入至 MongoDB 下的相应数据库中的指定集合中。导入后登录 MongoDB 数据库。查询集合中满足特定条件的结果。注：PPG--场均得分；PTS--总得分；FG%--投篮命中率；3P%--三分命中率；MPG--平均上场时间

Keystone 运维:

- 1.在 keystone 中创建用户 testuser，密码为 password。
- 2.将 testuser 用户分配给 admin 项目，赋予用户 user 的权限。
- 3.以管理员身份将 testuser 用户的密码修改为 000000。
- 4.通过 openstack 相关命令获取 token 值。
- 5.使用命令查询认证服务的查询端点信息。
- 6.使用命令列出认证服务目录。
- 7.在 keystone 中创建用户 testuser，密码为 password，创建好之后，使用命令修改 testuser 密码为 000000，并查看 testuser 的详细信息。
- 8.在 keystone 中创建用户 testuser，密码为 password，创建好之后，使用命令修改 testuser 的状态为 down，并查看 testuser 的详细信息。
- 9.完成 keystone 证书加密的 HTTPS 服务提升。

Glance 运维:

- 1.使用 glance 相关命令上传 CentOS_6.5_x86_64_XD.qcow2 镜像到云主机中，镜像名为 testone，然后使用 openstack 相关命令，并查看镜像的详细信息。
- 2.使用 glance 相关命令上传两个镜像，一个名字为 testone，一个名字叫 testtwo，使用相同的镜像源 CentOS_6.5_x86_64_XD.qcow2，然后使用 openstack 命令查看镜像列表，接着检查这两个镜像的 checksum 值是否相同。
- 3.登录 iaas 云主机，使用 glance 相关命令，上传镜像，源使用 CentOS_6.5_x86_64_XD.qcow2，名字为 testone，然后使用 openstack 命令修改这个镜像名改为 examimage，改完后使用 openstack 命令查看镜像列表。

4 使用 glance 相关命令，上传镜像，源使用 CentOS_6.5_x86_64_XD.qcow2，名字为 examimage，然后使用 openstack 命令查看镜像列表，然后给这个镜像打一个标签，标签名字为 lastone，接着查询修改的结果。

5.通过一条组合命令获取镜像列表信息，该组合命令包含：

- (1) 使用 curl 命令获取镜像列表信息；
- (2) 使用 openstack 相关命令获取的 token 值；
- (3) 仅使用 awk 命令且用“|”作为分隔符获取 token 具体参数值。

6.通过一条组合命令获取该镜像详细信息，该组合命令要求：

- (1) 不能使用任何 ID 作为参数；
- (2) 使用 openstack 相关命令获取详细信息；
- (3) 使用 glance 相关命令获取镜像对应关系；
- (4) 仅使用 awk 命令且用“|”作为分隔符获取 ID 值。

7.查看 glance 配置文件，找到默认的镜像存储目录，进入到存储目录中，使用 qemu 命令查看任意的一个镜像信息。

Nova 运维：

1.修改云平台中默认每个 tenant 的实例注入文件配额大小，并修改。

2.通过 nova 的相关命令创建云主机类型，并查询该云主机的详细信息。

3.使用 nova 相关命令，查询 nova 所有服务状态。

4.修改云平台中默认每个 tenant 的实例配额个数并查询。

5.使用 nova 相关命令，查询 nova 所有的监控列表，并查看监控主机的详细信息。

6.使用 grep 命令配合 -v 参数控制节点/etc/nova/nova.conf 文件中有效的命令

行覆盖输出到/etc/novaback.conf 文件。

7.此题可使用物理 iaas 环境，使用 nova 相关命令，启动一个云主机，云主机类型使用 m1.small，镜像使用 CentOS_6.5_x86_64_XD.qcow2，云主机名称为 examtest。

8.此题可使用物理 iaas 环境，使用 openstack 相关命令，启动一个云主机，云主机类型使用 m1.small，镜像使用 centos6.5，云主机名称为 xxxtest，并使用 openstack 命令查看此云主机的详细信息。

9.此题可使用物理环境，登录 dashboard 界面，创建一台虚拟机，将该虚拟机使用手动迁移的方式，迁移至另一个计算节点并查看。（controller 既是控制也是计算）

10.登录 iaas-all 云主机，修改 nova 后端默认存储位置。

11.修改相应的配置文件，使得 openstack 云主机的工作负载实现所要求的性能、可靠性和安全性。

12.配置 NFS 网络存储作为 nova 的后端存储。

Cinder 运维:

1.使用分区工具，对/dev/vda 进行分区，创建一个分区，使用命令将刚创建的分区创建为物理卷，然后使用命令查看物理卷信息。

2.使用命令查看当前卷组信息，使用命令创建逻辑卷，查询该逻辑卷详细信息。

3.创建一个卷类型，然后创建一块带这个卷类型标识的云硬盘，查询该云硬盘的详细信息。

4.通过命令行创建云硬盘，将其设置为只读，查询该云硬盘的详细信息。

- 5.通过命令行创建云硬盘，查询该云硬盘的详细信息。
- 6.使用命令，对/dev/vda 分区，并把这个分区创建成物理卷，然后再把这个物理卷加入到 cinder-volumes 卷组中，查看卷组详情。
- 7.使用命令创建一个云硬盘，然后通过 lvm 相关命令查看该云硬盘的详细信息，最后通过 cinder 命令对这块云硬盘进行扩容操作，并查看详细信息。
- 8.登录 iaas 云主机，使用命令对硬盘/dev/vda 进行分区，将这个分区创建为物理卷并使用 pvs 查看，然后将这个物理卷添加到 cinder-volumes 卷组中并使用 vgs 查看。
- 9.登录 controller 节点，创建云主机，镜像使用 centos6.5，flavor 使用 m1.medium，配置好网络。然后给云主机 iaas 挂载一个云硬盘，使用这块云硬盘，把云主机 iaas 的根目录扩容，最后在 iaas 云主机上用 df -h 命令查看。
- 10.登录“iaas-all”云主机，使用命令对磁盘/dev/vda 进行分区，然后使用命令，创建 raid 磁盘阵列，最后将 md0 格式化为 ext4 格式并查看该磁盘阵列的 UUID。
- 12.登录“iaas-all”云主机，查看 cinder 后端存储空间大小，将 cinder 存储空间扩容 10 个 G 大小，最后查看 cinder 后端存储空间大小。
- 13.修改相应的配置文件，增加 cinder backup 后端备份。
- 14.配置 NFS 网络存储作为 cinder 的后端存储。

Swift 运维:

- 1.使用命令查看 swift 服务状态，然后创建一个容器，并使用命令查看容器列表。
- 2.使用 swift 相关命令，创建一个容器，然后使用命令查看该容器的状态。

- 3.使用 `swift` 相关命令，查询 `swift` 对象存储服务可以存储的单个文件大小的最大值。
- 4.使用 `swift` 相关命令，创建一个容器，然后往这个容器中上传一个文件（文件可以自行创建），上传完毕后，使用命令查看容器。
- 5.登录 `iaas` 云主机，使用 `openstack` 命令，创建一个容器，并查询，上传一个文件（可自行创建）到这个容器中，并查询。
- 6.登录 `IaaS` 云主机，创建 `swifter` 用户，并创建 `swift` 租户，将 `swifter` 用户规划到 `swift` 租户下，赋予 `swifter` 用户使用 `swift` 服务的权限，并通过 `url` 的方式使用该用户在 `swift` 中创建容器。
- 7.使用 `url` 的方式，用 `admin` 账号在 `swift` 中创建容器，创建完之后用 `url` 的方式查看容器列表。
- 8.配置 `swift` 对象存储为 `glance` 的后端存储，并查看。

KVM 运维：

- 1.在物理云平台查询云主机 `IaaS` 在 `KVM` 中的真实实例名，在计算节点使用 `virsh` 命令找到该实例名对应的 `domain-id`，使用该 `domain-id` 关闭云主机 `IaaS`。
- 2.在物理云平台查询云主机 `IaaS` 在 `KVM` 中的真实实例名，在计算节点使用 `virsh` 命令找到该实例名对应的 `domain-id`，使用该 `domain-id` 重启云主机 `IaaS`。
- 3.此题使用物理 `iaas` 平台。登录 `compute` 节点，使用命令将 `KVM` 进程绑定到特定的 `cpu` 上。
- 4.此题使用物理平台。登录 `controller` 节点，调优 `kvm` 的 `I/O` 调度算法，`centos7` 默认的是 `deadline`，使用命令将参数改为 `noop` 并查询。
- 5.此题使用物理 `iaas` 平台。登录 `controller` 节点，使用 `cat` 命令，只查看当前系统有多少大页，然后设置大页数量并查看，接着使用命令使配置永久生效，

最后将大页挂载到/dev/hugepages/上。

6.登录 192.168.100.10/dashboard，创建一个云主机。在云主机所在的物理节点，进入 virsh 交互式界面，调整虚拟机的内存大小，最后使用命令查看该虚拟机的详情。

7.KVM 网络优化：让虚拟机访问物理网卡的层数更少，直至对物理网卡的单独占领，和物理机一样的使用物理网卡，达到和物理机一样的网络性能。

网络运维：

1.在控制节点安装配置 JDK 环境。安装完成后，查询 JDK 的版本信息。

2.在控制节点安装配置 Maven 环境。安装完成后，查询 Maven 的版本信息。

3.继续完成 OpenDaylight 的安装，完成后使用 curl 命令访问网页 <http://192.168.100.10:8181/index.html>。

4.创建网桥 br-test，把网卡 enp9s0 从原网桥迁移到 br-test，查询 openvswitch 的网桥信息和该网桥的端口信息。

5.创建命名空间 ns。

6.在网桥 br-test 中创建内部通信端口 tap。

7.在命名空间 ns 中配置端口 tap 的地址为 172.16.0.10/24。

8.在命名空间中查询端口 tap 的地址信息。

9.通过 openvswitch 手动运维 openstack 中虚拟主机的通讯信息。

Ceilometer 运维：

1.使用 ceilometer 相关命令，查询测量值的列表信息。

- 2.使用 `ceilometer` 相关命令，查询给定名称的测量值的样本列表信息。
- 3.使用 `ceilometer` 相关命令，查询事件列表信息。
- 4.使用 `ceilometer` 相关命令，查询资源列表。
- 5.按以下提供的参数及其顺序，使用 `ceilometer` 相关命令创建一个新的基于计算统计的告警。以下题目都需在这个基础上完成。
 - (1) 名字为: `cpu_hi`
 - (2) 测量值的名称为: `cpu_util`
 - (3) 阈值为: `70.0`
 - (4) 对比的方式为: 大于
 - (5) 统计的方式为: 平均值
 - (6) 周期为: `600s`
 - (7) 统计的次数为: `3`
 - (8) 转为告警状态时提交的 URL 为: `'log://'`
 - (9) 关键字: `resource_id=INSTANCE_ID`
- 6.使用 `ceilometer` 相关命令，查询用户的告警列表信息。
- 7.使用 `ceilometer` 相关命令，查询给定名称的告警的历史改变信息。
- 8.使用 `ceilometer` 相关命令，修改给定名称的告警状态为不生效。
- 9.使用 `ceilometer` 相关命令，删除给定名称的告警，并使用命令查看删除结果。
- 10.使用 `Ceilometer` 相关命令，查看某云主机有哪些样本，然后使用 `Ceilometer` 命令查看云主机的特定样本信息。

Heat 运维:

- 1.使用 heat 相关命令，查看 heat 的服务列表信息。
- 2.使用 heat 相关命令，查询给定的详细资源类型信息。
- 3.使用 heat 相关命令，查询 heat 模板的版本信息。
- 4.使用 heat 相关命令，查询 heat 最新版本模板的功能列表。
- 5.使用提供的文件 server.yml 创建名为 heat 的 stack，其中 glance 镜像使用 centos7，网络使用 int-net1。查询 stack 列表信息。
- 6.现有 server.yml 文件，请使用该 yml 文件创建堆栈 mystack，指定使用镜像 centos6.5，使用网络 int-net1，待创建完成后，查询堆栈 mystack 的状态为 CREATE_COMPLETE 的事件信息。
- 7.对提供的 server.yml 模板进行修改，添加所需参数。通过命令使用 heat 模板创建名为 test-heat 的 stack，其中 glance 镜像使用 centos7，网络使用 int-net1。查询 stack 列表信息。

数据加密:

前提：按要求配置静态 fixed_key，使 cinder 和 nova 组件可以使用加密过的 Block Storage 卷服务，配置好之后，创建一个卷类型叫 luks，并把这个类型配置为加密类型，cipher 使用“aes-xts-plain64”，key_size 使用“512”，control-location 使用“front-end”，Provider 使用“nova.volume.encryptors.luks.LuksEncryptor”。

- 1.使用命令查看卷类型列表和加密卷类型列表。
- 2.使用命令创建两个卷，前者不加密，后者使用 luks 卷类型加密。然后查看卷列表。

3.使用命令创建两个卷，前者不加密，后者使用 luks 卷类型加密。使用 nova 命令，创建一个云主机，镜像使用提供的 cirros 镜像，然后使用命令分别将创建的两块云硬盘 attach 到云主机上，最后使用 cinder list 查看。

4.使用命令创建两个卷，前者不加密，后者使用 luks 卷类型加密。使用 nova 命令，创建一个云主机，镜像使用提供的 cirros 镜像，然后使用命令分别将创建的两块云硬盘 attach 到云主机上，最后使用 strings 命令验证数据卷的加密功能。

负载均衡：

1.安装完 neutron 网络后，使用 neutron 命令查询 lbaas 服务的状态。（物理环境）

2.使用负载均衡创建 nginx 资源池，使用 http 协议，选择轮循负载均衡方式。创建完成后添加 vip: nginx-vip，使用 http 协议，端口为 80，HTTP_COOKIE 会话持久化。使用 neutron 命令查询资源池 nginx 详细信息、nginx-vip 详细信息。

3.使用负载均衡创建 nginx 资源池，使用 http 协议，选择轮循负载均衡方式。创建完成后添加 vip: nginx-vip，使用 http 协议，端口为 80，HTTP_COOKIE 会话持久化。使用命令查看所创建资源池的 haproxy 配置文件。（物理环境）

防火墙：

1.防火墙规则创建，添加名为 icmp 的规则，拒绝所有源 IP、源端口、目的 IP、目的端口的 ICMP 规则。使用 neutron 命令查询规则列表信息、详细信息。（物理环境）

2.防火墙创建，创建名为 nginx 的防火墙，添加防火墙规则 nginx-80，放行所有源 IP、源端口、目的 IP、目的端口为 80 的规则。创建防火墙策略 nginx-policy，添加 nginx-80 规则。使用 neutron 命令查询防火墙详细信息、

策略详细信息、规则详细信息。（物理环境）

Mariadb 高可用:

- 1.申请两台虚拟机，构建 mariadb 高可用数据库集群，通过命令查询集群状态。
- 2.申请两台虚拟机，构建 mariadb 主从数据库，通过命令查询数据库状态。
- 3.配置 mariadb 高可用数据库，并对集群数据库进行运维操作。

系统排错:

- 1.使用 awk 相关命令，查询表格文件中以空格为分隔符，包含关键字“key”的一行中第一个字段和最后一个字段的内容，以“，”作为间隔。
- 2.使用 sed 相关命令，显示文件中第 10 行内容。
- 3.使用 sed 相关命令，替换文件中的关键词“key”为“guanjianci”。
- 4.使用 grep 相关命令，查询显示文件中以“[”开始并以“]”结束的所有行。
- 5.使用 grep 相关命令，查询显示文件中的包含关键词“key”的行。
- 6.登录“iaas-all”云主机，使用 curl 的方式，获取 token 值，并使用该 token 值，获取用户信息。
- 7.登录“iaas-all”云主机，使用 curl 的方式，获取 token 值，并使用该 token 值，获取指定用户信息。
- 8.登录“iaas-all”云主机，获取 token 值，使用 curl 的方式，创建用户并使用命令查询验证。
- 9.在云平台后台管理的过程中出现错误导致无法获取镜像信息，找出错误原

因，并进行修复。

10.在云平台后台管理的过程中出现错误导致无法登录数据库，找出错误原因，并进行修复。

11.在云平台后台管理的过程中出现错误导致无法进行 keystone 验证，找出错误原因，并进行修复。

12.在云平台后台管理的过程中对象存储功能无法使用，找出错误原因，并进行修复。

13.在云平台后台管理的过程中块存储功能无法使用，找出错误原因，并进行修复。

14.在云平台后台管理的过程中无法上传镜像，找出错误原因，并进行修复。

15.在云平台后台管理的过程中无法将云硬盘挂载到云主机上，找出错误原因，并进行修复。

16.在云平台后台管理的过程中无法获取云主机详细信息，找出错误原因，并进行修复。

17.在云平台中创建云主机过程中出现错误无法成功创建，找出错误原因，并进行修复。

18.在云平台后台管理的过程中发生错误，错误现象为无法创建可用的云硬盘，无法获取云主机信息，也无法将云硬盘挂载到云主机上。找出错误原因，并进行修复。

19.搭建 Ceph 分布式集群存储，配置 Ceph 作为 openstack 后端的统一存储，为 glance、nova、cinder、swift 提供存储支持。

第二部分：PaaS 云计算开发服务平台

任务一、PaaS 云平台搭建

1.规划容器平台的部署架构，容器平台部署在 IaaS 平台的 2 台虚拟机上。采用分开安装的方式部署，server 部署容器平台 server 节点和 registry 节点，client 部署容器平台 client 节点。每个虚拟机配置如下：

(1) 系统配置：

Server 节点：2CPU，2G 内存，60G 硬盘

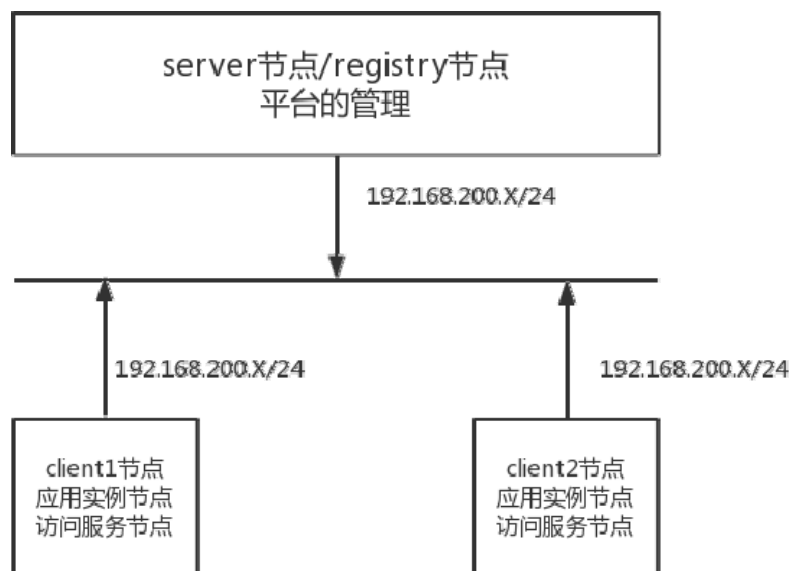
Client 节点：2CPU，4G 内存，60G 硬盘

(2) 操作系统：centos_7-x86_64

(3) IP：Server 和 Client 节点 ip 动态分配

(4) 主机名配置：Server 节点的主机名为：Server； Client 节点的主机名为：Client。

根据配置要求，完成配置文件的自定义与修改，搭建 PaaS 平台。



2.根据提供的软件包，搭建 rancher 平台。通过 curl 命令查询 Rancher 管理平台首页。

- 3.根据提供的软件包，通过应用商店部署 Gogs，修改网页访问端口为 9093，通过 curl 命令访问用户列表。
- 4.根据提供的软件包，通过应用商店部署 Elasticsearch 2.x，修改网页访问端口为 9094，通过 curl 命令访问首页。
- 5.根据提供的软件包，通过应用商店部署 Grafana，修改网页访问端口为 9090，通过 curl 命令访问首页。
- 6.根据提供的软件包，通过应用商店部署 Grafana，访问 3000 端口，使用 curl 命令访问 Grafana 服务的 3000 端口。
- 7.在 server 节点，修改配置文件，使仓库指向我们自己创建的 registry 节点，使用 docker info 命令查看修改后 docker 的仓库指向。
- 8.搭建 rancher 平台，打开系统的内核转发功能。
- 9.当要使用 docker api 查询信息的时候，我们需要修改 docker 的配置文件，添加一条 OPTIONS 来使得 api 功能可以使用，请把添加的 OPTIONS 参数以文本形式提交到答题框。
- 10.配置 docker 容器实现 nginx 的负载均衡，需要修改 nginx 的配置文件，请把定义 tomcat 负载均衡的参数以文本形式提交到答题框。
- 11.根据提供的模板文件与资源包，在 Rancher 应用商店自定义应用，并部署、查询。

任务二、PaaS 云平台运维

- 1.在 server 节点使用 netstat 命令查询仓库监听端口号，查询完毕后通过 lsof 命令（如命令不存在则手工安装）查询使用此端口号的进程。
- 2.在 server 节点通过 netstat 命令（如命令不存在则手工安装）查询 docker 镜像仓库 PID，使用 top 命令查询上一步查询到的 PID 的资源使用情况。

- 3.在 server 节点通过 docker 命令查询 docker registry 容器最后几条日志。
- 4.在 server 节点，查询 rancher/server 容器的进程号，建立命名空间 /var/run/netns 并与 rancher/server 容器进行连接，通过 ip netns 相关命令查询该容器的 ip。
- 5.在 server 节点查询当前 cgroup 的挂载情况。
- 6.在 server 节点创建 memory 控制的 cgroup，名称为：xiandian，创建完成后将当前进程移动到这个 cgroup 中，通过 cat 相关命令查询 cgroup 中的进程 ID。
- 7.在 server 节点创建 cpu 控制的 cgroup，名称为：xiandian。假设存在进程号为 8888 的进程一直占用 cpu，并且达到 100%，严重影响系统的正常运行。使用 cgroup 相关知识在创建的 cgroup 中将此进程操作 cpu 配额调整为 30%。
- 8.在 server 节点使用 nginx 镜像创建一个容器，只能使用特定的内核，镜像使用 nginx: latest，并通过查看 cgroup 相关文件查看内核使用情况。
- 9.在 server 节点创建目录，创建完成后启动镜像为 nginx:latest 的容器，并指定此目录为容器启动的数据卷，创建完成后通过 inspect 命令指定查看数据卷的情况。
- 10.在 server 节点创建目录，创建完成后启动镜像为 nginx:latest 的容器，并指定此目录为容器数据卷/opt 的挂载目录，设置该数据卷为只读模式，创建完成后通过 inspect 命令指定查看 HostConfig 内的 Binds 情况。
- 11.在 server 节点使用 docker 相关命令使用 mysql:8.0 镜像创建名为 mysqlpdb 的容器，使用镜像 nginx:latest 创建名为 nginxweb 容器，容器连接 mysqlpdb 容器内数据库，操作完成后使用 inspect 查看有关链接内容的字段。
- 12.在 server 节点通过 bridge 命令（如果不存在则安装该命令 bridge-utils）查看网桥列表。

- 13.在 server 节点创建 xd_br 网桥，设立网络的地址和掩码为 192.168.2.1/24，创建完成后启动该网桥，完成后查看 xd_br 网卡和网桥详细信息。
- 14.在 server 节点利用 nginx:latest 镜像运行一个无网络环境的容器，使用 inspect 命令只查看该容器的 networks 信息。
- 15.在 client 节点拉取 mysql:8.0 镜像，拉取完成后查询 docker 镜像列表目录。
- 16.在 server 节点运行 mysql:8.0 镜像，设置数据库密码为 xd_root，将宿主机 13306 端口作为容器 3306 端口映射，进入容器后创建数据库 xd_db，创建用户 xiandian，密码为 xd_pass，将此用户对 xd_db 拥有所有权限和允许此用户远程访问，完成后使用 xiandian 用户远程登录数据库查询数据库内的数据库列表。
- 17.在 server 节点将 mysql 镜像导出，导出名称为 mysql_images.tar，放在/media 目录下，导出后，查看目录。
- 18.在 server 节点，运行数据库容器，设置数据库密码，使用镜像为 mysql:8.0，运行之后，使用命令将容器导出，导出名称为 mysql_container.tar，放在/media 目录下，导出后，查看目录。
- 19.在 server 节点将 tomcat_latest.tar 镜像导入，并打标签，上传至仓库中。
- 20.在 server 节点运行 mysql 容器，使用镜像为 mysql:8.0 指定 mysql 密码，容器运行在后台，使用随机映射端口，容器运行完成后查询容器列表。
- 21.在 server 节点运行 mysql 容器，使用镜像为 mysql:8.0 指定 mysql 密码，容器运行在后台，使用随机映射端口，容器运行完成后查询容器列表，然后将运行的 mysql 容器停止，完成后查询容器状态。
- 22.在 server 节点，将上题停止的容器启动运行，完成后查询容器状态。

- 23.在 server 节点，将运行的 mysql 容器重启。
- 24.在 server 节点，执行一条命令使用 exec 获取 rancher/server 容器正在运行的网络套接字连接情况。
- 25.在 server 节点，使用 inspect 只查询 rancher/server 容器的 NetworkSettings 内 Networks 网桥信息。
- 26.在 server 节点，使用 inspect 只查询 rancher/server 容器的 PortBindings 信息。
- 27.在 server 节点，使用 inspect 只查询 rancher/server 容器的 NetworkSettings 内 Ports 信息。
- 28.在 server 节点，使用 inspect 只查询 rancher/server 镜像的 Volumes 卷组信息。
- 29.在 server 节点，使用 inspect 只查询 rancher/server 镜像的 Entrypoint 信息。
- 30.在 server 节点，使用 docker 命令查询 rancher/server 容器的进程。
- 31.在 server 节点，使用 docker 命令查列出 rancher/server 容器内发生变化的文件和目录。
- 32.在 server 节点，使用 docker 命令查看最后退出的容器的 ID。
- 33.在 server 节点，将运行的 mysql 容器创建为镜像，完成后查询该镜像。
- 34.在 server 节点查询 registry 容器的 CPU、内存等统计信息。
- 35.在 server 节点修改运行的 rancher/server 容器的名称，修改名称为 xiandian_server，完成后查询容器列表。
- 36.在 server 节点，使用 docker 命令列举所有的网络。

- 37.在 server 节点，使用 docker 命令查询 bridge 网络的所有详情。
- 38.在 server 节点，使用 docker 命令创建名为 xd_net 的网络，网络网段为 192.168.3.0/24，网关为 192.168.3.1，创建完成后查询网络列表。
- 39.在 server 节点，使用 docker 命令创建名为 xd_net 的网络，网络网段为 192.168.3.0/24，网关为 192.168.3.1，创建完成后查询此网络的详细信息。
- 40.在 server 节点，使用 docker 命令创建名为 xd_net 的网络，网络网段为 192.168.3.0/24，网关为 192.168.3.1，创建镜像为 centos:latest，容器名称为 centos，使用 docker 网络为 xd_net，创建完成后查询容器使用的网络名称和查询该容器的运行状态。
- 41.在 server 节点，使用 docker 命令创建名为 xd_net 的网络，网络网段为 192.168.3.0/24，网关为 192.168.3.1，创建镜像为 centos:latest，容器名称为 centos，使用 docker 网络为 xd_net，创建完成后查询容器 IP 地址。
- 42.在 server 节点，使用 docker 命令创建名为 xd_net 的网络，网络网段为 192.168.3.0/24，网关为 192.168.3.1，创建完成后，查询网络列表，接着删除 docker 网络 xd_net，完成后查询 docker 网络列表。
- 43.在 server 节点，使用 docker 命令只列举 rancher/server 容器的端口映射状态。
- 44.在 server 节点，使用 docker 命令打印 rancher/server 镜像的大小。
- 45.在 server 节点，使用 docker 命令运行 centos 镜像，运行输出打印“Hello world”。
- 46.在 server 节点，使用 docker 命令运行 centos 镜像，运行输出打印“Hello world”，要求启动命令包含打印完成后自动删除此容器及产生的数据。
- 47.在 server 节点，使用 docker 命令将 rancher/server 容器内的/opt/目录拷贝

到宿主机的/media/目录下。

48.在 server 节点，使用 docker 命令将当前操作系统的 yum 源的 local.repo 文件拷贝到 rancher/server 容器内的/opt/目录下。完成后使用 exec 命令查询容器的/opt 目录下的所有文件列表。

49.在 server 节点，使用 docker 查询当前系统使用的卷组信息。

50.在 server 节点，使用 centos:latest 的镜像创建容器，容器挂载使用创建的 xd_volume 卷组挂载到 root 分区，完成后通过 inspect 指定查看容器的挂载情况。

51.使用 supermin5 命令（若命令不存在则自己安装）构建 centos7 系统的 docker 镜像，镜像名称为 centos-7，镜像预装 yum、net-tools、initscripts 和 vi 命令，构建完成后提交镜像仓库上传操作，并查看此镜像。

52.编写以上题构建的 centos-7 镜像为基础镜像，构建 http 服务，Dockerfile 要求删除镜像的 yum 源，使用当前系统的 yum 源文件，完成后安装 http 服务，此镜像要求暴露 80 端口。构建的镜像名字叫 http:v1.0。完成后查看 Dockerfile 文件并查看镜像列表。

53.编写以上题构建的 centos-7 镜像为基础镜像，构建数据库服务，Dockerfile 要求删除镜像的 yum 源，使用当前系统的 yum 源文件，完成后安装 mariadb 服务，使用 mysql 用户初始化数据库，添加 MYSQL_USER=xiandian、MYSQL_PASS=xiandian 环境变量，要求数据库支持中文，暴露端口 3306，容器开机运行 mysld_safe 命令，完成后启动创建的镜像并查询 Dockerfile 文件，进入容器查看容器的数据库列表。

54.编写以上题构建的 centos-7 镜像为基础镜像，构建 Tomcat 服务，Dockerfile 要求删除镜像的 yum 源，使用当前系统的 yum 源文件，安装 java 和 unzip 服务，将提供的 apache-tomcat.zip 文件添加到/root/目录下，暴露端口 8080，将提供的 index.html 文件添加到 tomcat 的网页运行的目录下，容器开机运行

catalina.sh 脚本，完成后查询 Dockerfile 文件，查询镜像列表。

55.在 server 节点通过 docker api 查询 docker 的系统信息。

56.在 server 节点通过 docker api 查询 docker 的版本。

57.在 server 节点通过 docker api 查询 docker 内所有容器。

58.在 server 节点通过 docker api 查询 docker 内所有镜像。

59.在 server 节点通过 docker api 相关命令查询 rancher/server 镜像的具体信息。

60.根据提供的 tomcat 镜像，创建容器，使用该镜像，并创建 /root/www1 目录，在 www1 目录下编写 index.jsp 文件，容器的默认项目地址连接到创建的 www1 目录，要求访问 tomcat 的时候输出一句话为 this is Tomcat1，最后启动容器，并启动 tomcat 服务，使用 curl 命令查询 tomcat 的首页。

61.在 server 节点，使用 docker 命令查看最近创建的 2 个容器的 id。

62.在 server 节点，创建容器，然后将容器的卷空间值扩容（不要求扩容文件系统），最后查看容器的卷空间值。

63.在 server 节点，创建容器。创建完之后进入容器，通过修改相应的文件来限制写磁盘的速度，最后验证。

64.在 server 节点，查询私有仓库 redistry 中有哪些镜像。

65.在 server 节点，创建两个容器为 test1 和 test2，若只有这两个容器，该怎么设置容器的权重，才能使得 test1 和 test2 的 CPU 资源占比为 33.3%和 66.7%。

66.在 server 节点，创建容器，并明确限制容器对 CPU 资源的使用上限。

67.在容器使用的过程中，出现服务无法使用的情况，请排查容器中的错误，并修复运行。

68.使用 dockerfile 结合 docker compose 编排一组容器，形成容器组。

69.修改 docker 使用的文件系统引擎从 devicemapper 改为 overlayfs。

70.优化 docker 后端存储。

第三部分：大数据平台

任务一、大数据平台搭建

1.配置 master Node 的主机名为：master；slaver1 Node 的主机名为：slaver1。

2.修改 2 个节点的 hosts 文件，使用 FQDN 的方式，配置 IP 地址与主机名之间的映射关系。查询 hosts 文件的信息。

3.配置 2 个节点使用 Apache Ambari 和 iaas 中的 centos7 的 yum 源。其中 Ambari yum 源在 XianDian-BigData-v2.2.iso 软件包中。

4.在 master 节点安装 ntp 时钟服务，使用文件/etc/ntp.conf 配置 ntp 服务；在 slaver 节点安装 ntpdate 软件包，将 slaver1 节点时钟同步到 master 节点。

5.检查 2 个节点是否可以通过无密钥相互访问，如果未配置，则进行 SSH 无密码公钥认证配置。

6.安装 2 个节点的 JDK 环境，其中 jdk-8u77-linux-x64.tar.gz 在 XianDian-BigData-v2.2.iso 软件包中，使用命令查看 JAVA 版本信息。

7.在 master 节点安装配置 HTTP 服务，将软件包 XianDian-BigData-v2.2.iso 中的 HDP-2.6.1.0 和 HDP-UTILS-1.1.0.21 拷贝到/var/www/html 目录中，并启动 HTTP 服务使用命令只查看 http 服务的状态。

8.查询 2 个节点的 yum 源配置文件、JDK 版本信息、slaver1 节点同步 master 节点的命令及结果和 HTTP 服务的运行状态信息。

- 9.在 master 节点上安装 ambari-server 服务和 MariaDB 数据库服务，创建 ambari 数据库和 ambari 用户，用户密码为 bigdata。赋予 ambari 用户访问 ambari 数据库的权限，并导入 /var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql 文件至 ambari 数据库中。配置完成后安装 mysql-connector-java 软件包。查询 master 节点中 ambari 数据库中的所有表的列表信息。
- 10.在 master 节点上安装 ambari-server 服务和 MariaDB 数据库服务，创建 ambari 数据库和 ambari 用户，用户密码为 bigdata。赋予 ambari 用户访问 ambari 数据库的权限，并导入 /var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql 文件至 ambari 数据库中。操作完成后进入 MariaDB 数据库，查询 mysql 数据库中 user 表中的文件内容。
- 11.在 master 节点对 ambari-server 进行设置 (ambari-server setup)，指定 JDK 安装路径和数据库的主机、端口、用户、密码等参数，并启动 ambari-server 服务。配置完成后，通过 curl 命令在 Linux Shell 中查询 http://master:8080 界面内容。
- 12.在 master 节点对 ambari-server 进行设置 (ambari-server setup)，指定 JDK 安装路径和数据库的主机、端口、用户、密码等参数，并启动 ambari-server 服务。配置完成后，查询 ambari-server 的运行状态信息。
- 13.在 2 台节点中安装 ambari-agent 服务，修改 /etc/ambari-agent/conf/ambari-agent.ini 文件 server 端主机位 master 节点，启动 ambari-agent 服务，查看 agent 端/var/log/ambari-agent/ambari-agent.log 日志文件。
- 14.在先电大数据平台中创建 Hadoop 集群“XIANDIAN HDP”，选择安装栈为 HDP 2.6, 安装服务为 HDFS、YARN+MapReduce2、Zookeeper、Ambari Metrics。安装完成后，在 master 节点和 slaver 节点的 Linux Shell 中查看 Hadoop 集群

的服务进程信息。

15.在先电大数据平台中创建 Hadoop 集群“XIANDIAN HDP”，选择安装栈为 HDP 2.6, 安装服务为 HDFS、YARN+MapReduce2、Zookeeper、Ambari Metrics。安装完成后，在 Linux Shell 中查看 Hadoop 集群的基本统计信息。

16.在 master 节点的数据库中创建 hive 数据库，并赋予 hive 数据库远程访问的权限。

17.禁用 Transparent Huge Pages。

18.使用提供的模板文件和资源包，在 ambari 界面自定义添加服务，并部署验证。

任务二、大数据平台运维

HDFS 题：

1.在 HDFS 文件系统的根目录下创建递归目录“1daoyun/file”，将附件中的 BigDataSkills.txt 文件，上传到 1daoyun/file 目录中，使用相关命令查看文件系统中 1daoyun/file 目录的文件列表信息。

2.在 HDFS 文件系统的根目录下创建递归目录“1daoyun/file”，将附件中的 BigDataSkills.txt 文件，上传到 1daoyun/file 目录中，并使用 HDFS 文件系统检查工具检查文件是否受损。

3.在 HDFS 文件系统的根目录下创建递归目录“1daoyun/file”，将附件中的 BigDataSkills.txt 文件，上传到 1daoyun/file 目录中，上传过程指定 BigDataSkills.txt 文件在 HDFS 文件系统中的复制因子为 2，并使用 fsck 工具检查存储块的副本数。

4.HDFS 文件系统的根目录下存在一个/apps 的文件目录，要求开启该目录的可创建快照功能，并为该目录文件创建快照，快照名称为 apps_1daoyun，使

用相关命令查看该快照文件的列表信息。

5.当 Hadoop 集群启动的时候，会首先进入到安全模式的状态，该模式默认 30 秒后退出。当系统处于安全模式时，只能对 HDFS 文件系统进行读取，无法进行写入修改删除等的操作。现假设需要对 Hadoop 集群进行维护，需要使集群进入安全模式的状态，并检查其状态。

6.为了防止操作人员误删文件，HDFS 文件系统提供了回收站的功能，但过多的垃圾文件会占用大量的存储空间。要求在先电大数据平台的 WEB 界面将 HDFS 文件系统回收站中的文件彻底删除的时间间隔为 7 天。

7.为了防止操作人员误删文件，HDFS 文件系统提供了回收站的功能，但过多的垃圾文件会占用大量的存储空间。要求在 Linux Shell 中使用“vi”命令修改相应的配置文件以及参数信息，关闭回收站功能。完成后，重启相应的服务。

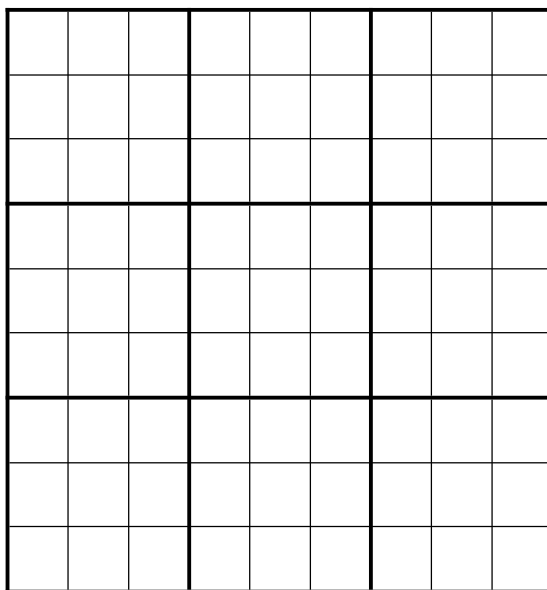
8.Hadoop 集群中的主机在某些情况下会出现宕机或者系统损坏的问题，一旦遇到这些问题，HDFS 文件系统中的数据文件难免会产生损坏或者丢失，为了保证 HDFS 文件系统的可靠性，现需要在先电大数据平台的 WEB 界面将集群的冗余复制因子修改为 5。

9.Hadoop 集群中的主机在某些情况下会出现宕机或者系统损坏的问题，一旦遇到这些问题，HDFS 文件系统中的数据文件难免会产生损坏或者丢失，为了保证 HDFS 文件系统的可靠性，需要将集群的冗余复制因子修改为 5，在 Linux Shell 中使用“vi”命令修改相应的配置文件以及参数信息，完成后，重启相应的服务。

10.使用命令查看 hdfs 文件系统中/tmp 目录下的目录个数，文件个数和文件总大小。

MapReduce 题:

- 1.在集群节点中/usr/hdp/2.4.3.0-227/hadoop-mapreduce/目录下,存在一个案例 JAR 包 hadoop-mapreduce-examples.jar。运行 JAR 包中的 PI 程序来进行计算圆周率 π 的近似值,要求运行 5 次 Map 任务,每个 Map 任务的投掷次数为 5。
- 2.在集群节点中/usr/hdp/2.4.3.0-227/hadoop-mapreduce/目录下,存在一个案例 JAR 包 hadoop-mapreduce-examples.jar。运行 JAR 包中的 wordcount 程序来对/1daoyun/file/BigDataSkills.txt 文件进行单词计数,将运算结果输出到 /1daoyun/output 目录中,使用相关命令查询单词计数结果。
- 3.在集群节点中/usr/hdp/2.4.3.0-227/hadoop-mapreduce/目录下,存在一个案例 JAR 包 hadoop-mapreduce-examples.jar。运行 JAR 包中的 sudoku 程序来计算下表中数独运算题的结果。。



- 4.在集群节点中/usr/hdp/2.4.3.0-227/hadoop-mapreduce/目录下,存在一个案例 JAR 包 hadoop-mapreduce-examples.jar。运行 JAR 包中的 grep 程序来统计文件系统中/1daoyun/file/BigDataSkills.txt 文件中“Hadoop”出现的次数,统计完成后,查询统计结果信息。

HBase 题:

- 1.启动先电大数据平台的 Hbase 数据库，其中要求使用 master 节点的 RegionServer。在 Linux Shell 中启动 Hbase shell，查看 HBase 的版本信息。
(相关数据库命令语言请全部使用小写格式)
- 2.启动先电大数据平台的 Hbase 数据库，其中要求使用 master 节点的 RegionServer。在 Linux Shell 中启动 Hbase shell，查看 HBase 的状态信息。
(相关数据库命令语言请全部使用小写格式)
- 3.启动先电大数据平台的 Hbase 数据库，其中要求使用 master 节点的 RegionServer。在 Linux Shell 中启动 Hbase shell，查看进入 HBase shell 的当前系统用户。(相关数据库命令语言请全部使用小写格式)
- 4.在 HBase 数据库中创建表 xiandian_user，列族为 info，创建完成后查看 xiandian_user 表的描述信息。(相关数据库命令语言请全部使用小写格式)
- 5.开启 HBase 的安全认证功能，在 HBase Shell 中设置 root 用户拥有表 xiandian_user 的读写与执行的权限，设置完成后，使用相关命令查看其权限信息。(相关数据库命令语言请全部使用小写格式)
- 6.在 HBase Shell 创建表 xiandian_user，列族为 info，并 list 查询，之后删除这个表，并 list 查询。
- 7.在 Hbase Shell 中创建表 xiandian，向表 xiandian 中插入一组数据为 xiandian, row1, info: name, xiaoming，插入后查询表 xiandian 中 rowkey 为 row1 的记录。
- 8.在 Hbase Shell 中创建表 xiandian，列族为“info”然后查询表中所有的记录。
- 9.登录 hbase 数据库，使用命令创建一张表，列族为 member_id', 'address', 'info'，创建完毕后查看该表的详细信息，后来发现列族'member_id'这个列族是多余的，需要删除，使用命令将该列族删除并查看详细信息，最后查看该

表是否是 `enabled` 的。

10.登录 `hbase` 数据库，创建一张表，列族为'`address`', '`info`'，创建完之后，向该表插入数据，插入完毕后，使用命令按照要求查询所需信息。

11.登录 `hbase` 数据库，新建一张表，列族为'`address`', '`info`'，创建完之后，向该表插入数据，插入之后查询这条信息，并修改信息，改完后，查询修改前和修改后的信息。

12.登录 `hbase` 数据库，创建一张表，列族为'`address`', '`info`'，创建完之后，向该表插入数据，插入完毕后，使用 `scan` 命令查询该表指定 `startrow` 的信息。

13.在关系数据库系统中，命名空间 `namespace` 是表的逻辑分组，同一组中的表有类似的用途。登录 `hbase` 数据库，新建一个命名空间叫 `newspace` 并用 `list` 查询，然后在这个命名空间中创建表，列族为'`address`', '`info`'，创建完之后，向该表插入数据，插入完毕后，使用 `scan` 命令只查询表中特定的信息。

14.登录 `master` 节点，在本地新建一个文件叫 `hbasetest.txt` 文件，编写内容，要求新建一张表为'`test`'，列族为'`cf`'，然后向这张表批量插入数据，数据如下所示：

```
'row1', 'cf:a', 'value1'
```

```
'row2', 'cf:b', 'value2'
```

```
'row3', 'cf:c', 'value3'
```

```
'row4', 'cf:d', 'value4'
```

在插入数据完毕后用 `scan` 命令查询表内容，然后用 `get` 命令只查询 `row1` 的内容，最后退出 `hbase shell`。

Hive 题：

1.启动先电大数据平台的 `Hive` 数据仓库，启动 `Hvie` 客户端，通过 `Hive` 查

看 `hadoop` 所有文件路径（相关数据库命令语言请全部使用小写格式）。

2.使用 Hive 工具来创建数据表 `xd_phy_course`，将 `phy_course_xd.txt` 导入到该表中，其中 `xd_phy_course` 表的数据结构如下表所示。导入完成后，通过 `hive` 查询数据表 `xd_phy_course` 中数据在 HDFS 所处的文件位置列表信息。

（相关数据库命令语言请全部使用小写格式）

<code>stname(s</code>	<code>stID</code>	<code>class(st</code>	<code>opt_cour(s</code>
<code>tring)</code>	<code>(int)</code>	<code>ring)</code>	<code>tring)</code>

3.使用 Hive 工具来创建数据表 `xd_phy_course`，并定义该表为外部表，外部存储位置为 `/1daoyun/data/hive`，将 `phy_course_xd.txt` 导入到该表中，其中 `xd_phy_course` 表的数据结构如下表所示。导入完成后，在 `hive` 中查询数据表 `xd_phy_course` 的数据结构信息。（相关数据库命令语言请全部使用小写格式）

<code>stname(s</code>	<code>stID</code>	<code>class(st</code>	<code>opt_cour(s</code>
<code>tring)</code>	<code>(int)</code>	<code>ring)</code>	<code>tring)</code>

4.使用 Hive 工具来查找出 `phy_course_xd.txt` 文件中某高校 `Software_1403` 班级报名选修 `volleyball` 的成员所有信息，其中 `phy_course_xd.txt` 文件数据结构如下表所示，选修科目字段为 `opt_cour`，班级字段为 `class`。（相关数据库命令语言请全部使用小写格式）

<code>stname(s</code>	<code>stID</code>	<code>class(st</code>	<code>opt_cour(s</code>
<code>tring)</code>	<code>(int)</code>	<code>ring)</code>	<code>tring)</code>

5.使用 Hive 工具来统计 `phy_course_xd.txt` 文件中某高校报名选修各个体育科目的总人数，其中 `phy_course_xd.txt` 文件数据结构如下表所示，选修科目

字段为 `opt_cour`，将统计的结果导入到表 `phy_opt_count` 中，通过 `SELECT` 语句查询表 `phy_opt_count` 内容。（相关数据库命令语言请全部使用小写格式）

<code>stname(s tring)</code>	<code>stID (int)</code>	<code>class(st ring)</code>	<code>opt_cour(s tring)</code>
----------------------------------	-----------------------------	---------------------------------	------------------------------------

6.使用 Hive 工具来查找出 `phy_course_score_xd.txt` 文件中某高校 `Software_1403` 班级体育选修成绩在 90 分以上的成员所有信息，其中 `phy_course_score_xd.txt` 文件数据结构如下表所示，选修科目字段为 `opt_cour`，成绩字段为 `score`。（相关数据库命令语言请全部使用小写格式）

<code>stname(s tring)</code>	<code>stID (int)</code>	<code>class(st ring)</code>	<code>opt_cour(s tring)</code>	<code>score(float)</code>
----------------------------------	-----------------------------	---------------------------------	------------------------------------	--------------------------------

7.使用 Hive 工具来统计 `phy_course_score_xd.txt` 文件中某高校各个班级体育课的平均成绩，使用 `round` 函数保留两位小数。其中 `phy_course_score_xd.txt` 文件数据结构如下表所示，班级字段为 `class`，成绩字段为 `score`。（相关数据库命令语言请全部使用小写格式）

<code>stname(s tring)</code>	<code>stID (int)</code>	<code>class(st ring)</code>	<code>opt_cour(s tring)</code>	<code>score(float)</code>
----------------------------------	-----------------------------	---------------------------------	------------------------------------	--------------------------------

8.使用 Hive 工具来统计 `phy_course_score_xd.txt` 文件中某高校各个班级体育课的最高成绩。其中 `phy_course_score_xd.txt` 文件数据结构如下表所示，班级字段为 `class`，成绩字段为 `score`。（相关数据库命令语言请全部使用小写格式）

<code>stname(s</code>	<code>stID</code>	<code>class(st</code>	<code>opt_cour(s</code>	<code>score(float</code>
-----------------------	-------------------	-----------------------	-------------------------	--------------------------

tring)	(int)	ring)	tring))
--------	-------	-------	--------	---

9.在 Hive 数据仓库将网络日志 weblog_entries.txt 中分开的 request_date 和 request_time 字段进行合并，并以一个下划线“_”进行分割，如下图所示，其中 weblog_entries.txt 的数据结构如下表所示。（相关数据库命令语言请全部使用小写格式）

2012-05-10_21:33:26

2012-05-10_21:13:10

md5(ST RING)	url(STR ING)	request_dat e (STRING)	request_tim e (STRING)	ip(ST RING)
-----------------	-----------------	---------------------------	---------------------------	----------------

10.在 Hive 数据仓库将网络日志 weblog_entries.txt 中的 IP 字段与 ip_to_country 中 IP 对应的国家进行简单的内链接，输出结果如下图所示，其中 weblog_entries.txt 的数据结构如下表所示。（相关数据库命令语言请全部使用小写格式）

3ab7888ffe27c2f98d48eb296449d5 /khvc.html 2012-05-10 21:25:58 111.147.83.42 China
65827078a9f7ccce59632263294782db /c.html 2012-05-10 21:34:28 137.157.65.89 Australia

md5(ST RING)	url(STR ING)	request_dat e (STRING)	request_tim e (STRING)	ip(ST RING)
-----------------	-----------------	---------------------------	---------------------------	----------------

11.使用 Hive 动态地关于网络日志 weblog_entries.txt 的查询结果创建 Hive 表。通过创建一张名为 weblog_entries_url_length 的新表来定义新的网络日志数据库的三个字段，分别是 url，request_date，request_time。此外，在表中定义一个获取 url 字符串长度名为“url_length”的新字段，其中 weblog_entries.txt 的数据结构如下表所示。完成后查询 weblog_entries_url_length 表文件内容。（相关数据库命令语言请全部使用小写格式）

md5(ST RING)	url(STR ING)	request_dat e (STRING)	request_tim e (STRING)	ip(ST RING)
-----------------	-----------------	---------------------------	---------------------------	----------------

Sqoop 题:

- 1.在 master 和 slaver 节点安装 Sqoop Clients，完成后，在 master 节点查看 Sqoop 的版本信息。
- 2.使用 Sqoop 工具列出 master 节点中 MySQL 中所有数据库。
- 3.使用 Sqoop 工具列出 master 节点中 MySQL 中 ambari 数据库中所有的数据表。
- 4.在 MySQL 中创建名为 xiandian 的数据库，在 xiandian 数据库中创建 xd_phy_course 数据表，其数据表结构如表 1 所示。使用 Hive 工具来创建数据表 xd_phy_course，将 phy_course_xd.txt 导入到该表中，其中 xd_phy_course 表的数据结构如表 2 所示。使用 Sqoop 工具将 hive 数据仓库中的 xd_phy_course 表导出到 master 节点的 MySQL 中 xiandain 数据库的 xd_phy_course 表。

表 1

stname VARCHAR(20)	stI D INT(1)	class VARCHAR(20)	opt_cour VARCHAR(20)
-----------------------	--------------------	----------------------	-------------------------

表 2

stname(s tring)	stID (int)	class(st ring)	opt_cour(s tring)
--------------------	---------------	-------------------	----------------------

- 5.在 Hive 中创建 xd_phy_course 数据表，其数据表结构如下表所示。使用 Sqoop 工具将 MySQL 中 xiandian 数据库下 xd_phy_course 表导入到 Hive 数

据仓库中的 `xd_phy_course` 表中。

stname(s tring)	stID (int)	class(st ring)	opt_cour(s tring)
--------------------	---------------	-------------------	----------------------

Pig 题:

- 1.在 master 节点安装 Pig Clients，打开 Linux Shell 以 MapReduce 模式启动它的 Grunt。
- 2.在 master 节点安装 Pig Clients，打开 Linux Shell 以 Local 模式启动它的 Grunt。
- 3.使用 Pig 工具在 Local 模式计算系统日志 `access-log.txt` 中的 IP 的点击数，要求使用 `GROUP BY` 语句按照 IP 进行分组，通过 `FOREACH` 运算符，对关系的列进行迭代，统计每个分组的总行数，最后使用 `DUMP` 语句查询统计结果。
- 4.使用 Pig 工具计算天气数据集 `temperature.txt` 中年度最高气温，要求使用 `GROUP BY` 语句按照 `year` 进行分组，通过 `FOREACH` 运算符，对关系的列进行迭代，统计每个分组的最大值，最后使用 `DUMP` 语句查询计算结果。
- 5.使用 Pig 工具统计数据集 `ip_to_country` 中每个国家的 IP 地址数。要求使用 `GROUP BY` 语句按照国家进行分组，通过 `FOREACH` 运算符，对关系的列进行迭代，统计每个分组的 IP 地址数目，最后将统计结果保存到 `/data/pig/output` 目录中，并查看数据结果。

Mahout 题:

- 1.在 master 节点安装 Mahout Client，打开 Linux Shell 运行 `mahout` 命令查看 Mahout 自带的案例程序。

- 2.使用 Mahout 工具将解压后的 20news-bydate.tar.gz 文件内容转换成序列文件，保存到/data/mahout/20news/output/20news-seq/目录中，并查看该目录的列表信息。
- 3.使用 Mahout 工具将解压后的 20news-bydate.tar.gz 文件内容转换成序列文件，保存到/data/mahout/20news/output/20news-seq/目录中，使用-text 命令查看序列文件内容（前 20 行即可）。
- 4.使用 Mahout 挖掘工具对数据集 user-item-score.txt（用户-物品-得分）进行物品推荐，要求采用基于项目的协同过滤算法，欧几里得距离公式定义，并且每位用户的推荐个数为 3，设置非布尔数据，最大偏好值为 4，最小偏好值为 1，将推荐输出结果保存到 output 目录中，通过-cat 命令查询输出结果 part-r-00000 中的内容。

Flume 题：

- 1.在 master 节点安装启动 Flume 组件，打开 Linux Shell 运行 flume-ng 的帮助命令，查看 Flume-ng 的用法信息。
- 2.根据提供的模板 log-example.conf 文件，使用 Flume NG 工具收集 master 节点的系统日志/var/log/secure，将收集的日志信息文件的名称以“xiandian-sec”为前缀，存放于 HDFS 文件系统的/1daoyun/file/flume 目录中，并且定义在 HDFS 中产生的文件的时间戳为 10 分钟。进行收集后，查询 HDFS 文件系统中/1daoyun/file/flume 的列表信息。
- 3.根据提供的模板 hdfs-example.conf 文件，使用 Flume NG 工具设置 master 节点的系统路径/opt/xiandian/为实时上传文件至 HDFS 文件系统的实时路径，设置 HDFS 文件系统的存储路径为/data/flume/，上传后的文件名保持不变，文件类型为 DataStream，然后启动 flume-ng agent。

Spark 题:

- 1.在先电大数据平台部署 Spark 服务组件，打开 Linux Shell 启动 spark-shell 终端，将启动的程序进程信息以文本形式提交到答题框中。
- 2.启动 spark-shell 后，在 scala 中加载数据“1, 2, 3, 4, 5, 6, 7, 8, 9, 10”，求这些数据的 2 倍乘积能够被 3 整除的数字，并通过 toDebugString 方法来查看 RDD 的谱系。
- 3.启动 spark-shell 后，在 scala 中加载 Key-Value 数据("A", 1), ("B", 2), ("C", 3), ("A", 4), ("B", 5), ("C", 4), ("A", 3), ("A", 9), ("B", 4), ("D", 5)，将这些数据以 Key 为基准进行升序排序，并以 Key 为基准进行分组。
- 4.启动 spark-shell 后，在 scala 中加载 Key-Value 数据("A", 1), ("B", 3), ("C", 5), ("D", 4), ("B", 7), ("C", 4), ("E", 5), ("A", 8), ("B", 4), ("D", 5)，将这些数据以 Key 为基准进行升序排序，并对相同的 Key 进行 Value 求和计算。
- 5.启动 spark-shell 后，在 scala 中加载 Key-Value 数据("A", 4), ("A", 2), ("C", 3), ("A", 4), ("B", 5), ("C", 3), ("A", 4)，以 Key 为基准进行去重操作，并通过 toDebugString 方法来查看 RDD 的谱系。
- 6.启动 spark-shell 后，在 scala 中加载两组 Key-Value 数据("A", 1), ("B", 2), ("C", 3), ("A", 4), ("B", 5), ("A", 1), ("B", 2), ("C", 3), ("A", 4), ("B", 5)，将两组数据以 Key 为基准进行 JOIN 操作。
- 7.登录 spark-shell，定义 i 值为 1，sum 值为 0，使用 while 循环，求从 1 加到 100 的值，最后使用 scala 的标准输出函数输出 sum 值。
- 8.登录 spark-shell，定义 i 值为 1，sum 值为 0，使用 for 循环，求从 1 加到 100 的值，最后使用 scala 的标准输出函数输出 sum 值。

9.登录 spark-shell，定义变量 i、sum，并赋 i 初值为 1、sum 初值为 0、步长为 3，使用 while 循环，求从 1 加到 2018 的值，最后使用 scala 的标准输出函数输出 sum 值。

10.任何一种函数式语言中，都有 map 函数与 flatMap 这两个函数：

map 函数的用法，顾名思义，将一个函数传入 map 中，然后利用传入的这个函数，将集合中的每个元素处理，并将处理后的结果返回。

而 flatMap 与 map 唯一不一样的地方就是传入的函数在处理完后返回值必须是 List，所以需要返回值是 List 才能执行 flat 这一步。

(1) 登录 spark-shell，自定义一个 list，然后利用 map 函数，对这个 list 进行元素乘 2 的操作。

(2) 登录 spark-shell，自定义一个 list，然后利用 flatMap 函数将 list 转换为单个字母并转换为大写。

11.登录大数据云主机 master 节点，在 root 目录下新建一个 abc.txt，里面的内容为：

```
hadoop  hive
solr    redis
kafka  hadoop
storm  flume
sqoop  docker
spark  spark
hadoop  spark
elasticsearch  hbase
hadoop  hive
spark  hive
hadoop  spark
```

然后登录 spark-shell，首先使用命令统计 abc.txt 的行数，接着对 abc.txt 文档中的单词进行计数，并按照单词首字母的升序进行排序，最后统计结果行数。

12.登录 spark-shell，自定义一个 List，使用 spark 自带函数对这个 List 进行去重操作。

13.登录“spark-shell”交互界面。给定数据，使用 spark 工具，统计每个日期新增加的用户数，最后显示统计结果。

14.登录“spark-shell”交互界面。定义一个函数，函数的作用是比较传入的两个变量，返回大的那个。

第四部分：SaaS 云应用开发

任务一、大数据学情分析

1.导入开发环境

检查已安装的 MongoDB, HBase, MySQL 本地服务，解压 guosai-xueqing.rar 学情开发框架项目，搭建大数据学情开发环境。

- (1) 导入三个 MongoDB 数据库目录 employ, job_internet, question_survey;
- (2) 导入 MySQL 的 xueqing-client 项目的 sql 文件 xueqing-client.sql;
- (3) 导入 xueqing-web 项目并运行。

2.HBase 建表操作

完成 xueqing-server 之中 HBase 建数据库表的过程，建立总表和相应的行业表，运行后提交代码与 HBase 中表的查询结果。

3.爬取岗位信息

通过解析网站（xueqing-web）源代码，抓取所有岗位页面信息中的岗位名称，发布日期，薪水条件，招聘人数，学历要求，工作经验年限，岗位描述等信息，爬取的信息存入指定的 HBase 表。运行后提交代码与 HBase 中表的查询结

果。

4.对爬取的岗位信息清洗

使用 HBase 中保存的岗位信息，根据指定的数据转换规则，对指定的列族和列的地区分布、学历分布、薪资分布、企业规模、企业性质、工作经验等数据进行清洗。

5.岗位数据统计

完成 xueqing-server 之中爬取的岗位信息数据进行数据统计，将数据保存至 MongoDB 之中，实现 Web 前端代码并通过指定 EChart 的图表可视化展示岗位地区分布、学历分布、薪资分布、企业规模、企业性质、工作经验等统计可视化数据开发。

6.岗位聚类

根据给定的岗位数据源，使用指定的聚类或分类算法，进行 IT 行业的云计算、大数据、移动互联、人工智能或物联网等行业进行岗位聚类分析，将数据保存至 MongoDB 之中，实现 Web 前端代码并通过指定 EChart 的图表可视化展示。

7.聚类岗位的主要技能图谱

完成 xueqing-server 之中的聚类结果，提取每一个聚类结果中的主要技能点，将数据保存至 MongoDB 之中，通过 EChart 指定的图表展示岗位的技能图谱。

8.推荐岗位

给定一个学生的技能数据，使用机器学习推荐算法，帮助该学生推荐三个最佳的招聘岗位，并通过 EChart 指定的图表展示推荐岗位和学生技能对比。

任务二、微信小程序开发

1.商店界面开发

根据下面给出的效果图实现微信 O2O 商城商店界面的开发。

效果图：



2.预约界面开发

根据下面给出的效果图实现微信 O2O 商城预约界面的开发。

效果图：



3. 订单界面开发

根据下面给出的效果图实现微信 O2O 商城订单界面的开发。

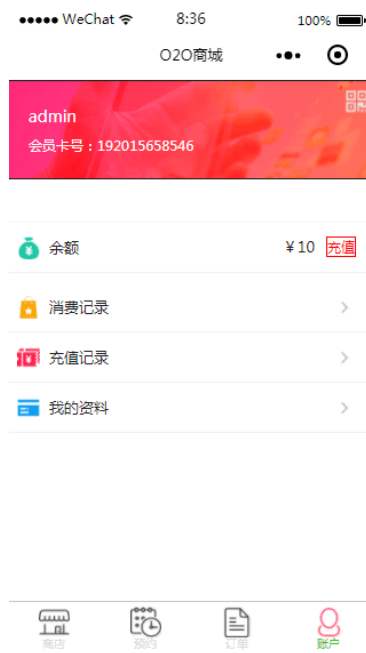
效果图：



4. 账户界面开发

根据下面给出的效果图实现微信 O2O 商城账户界面的开发。

效果图：



5.充值界面开发

根据下面给出的效果图实现微信 o2o 商城充值界面的开发。

效果图：



6.消费记录界面开发

根据下面给出的效果图实现微信 o2o 商城消费记录界面的开发。

效果图：



7.充值记录界面开发

根据下面给出的效果图实现微信 o2o 商城充值记录界面的开发。

效果图：



8.我的资料界面开发

根据下面给出的效果图实现微信 o2o 商城我的资料界面的开发。

效果图：



9.商品详情界面开发

根据下面给出的效果图实现微信 o2o 商城商品详情界面的开发。

效果图：



10.订单详情界面开发

根据下面给出的效果图实现微信 o2o 商城订单详情界面的开发。

效果图：



11. 微信后台用户服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的用户信息、用户登录或登出等接口代码，实现用户模块相关功能。

12. 微信小程序商品服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的商品查询、搜索或获取商品等接口代码，实现商品模块相关功能对接。

13. 微信小程序订单服务对接

基于给定的数据库数据，开发后台 Restful 服务，实现 O2O 商城的订单售后、订单详情或订单状态等接口代码，实现订单模块相关功能对接。

第五部分：文档及职业素养

任务一、工作总结报告

云架构设计和说明

- 1.绘制 IaaS 平台的架构组件图，组件包含本次项目实施中涉及到的 IaaS 组件服务，架构组件绘制各组件之间的关系。并对架构图进行解释说明。
- 2.绘制 Hadoop 分布式存储 HDFS 的架构图，并对架构图进行解释说明。
- 3.构建存储型、高可用的 IaaS 平台的需求，设计包含存储节点 3 台、计算节点 2 台、控制节点 3 台的高可用 IaaS 方案。包括硬件设备、网络拓扑、服务模块的架构图。
- 4.绘制云计算 SPI 模型，并对各服务组成进行说明。
- 5.绘制 IaaS 云平台新建云主机的流程图，并详细描述云主机创建的过程。要求该流程从网页进行创建，使用 keystone 作为所有组件的权限认证方式，nova 作为核心组件，neutron 提供网络，由 cinder 提供持久化的存储。
- 6.IaaS 云平台中，用户请求云主机的流程涉及认证 Keystone 服务、计算 Nova 服务、镜像 Glance 服务，网络 Neutron 服务，在服务流程中，令牌（Token）作为流程认证传递。绘制服务申请认证机制的流程并进行简要说明。
- 7.由于镜像文件都比较大，镜像从创建到成功上传到 Glance 文件系统中，是通过异步任务的方式一步步完成的，状态包括 Queued（排队）、Saving（保存中）、Active（有效）、deactivated（无效）、Killed（错误）、Deleted（被删除）和 Pending_delete（等待删除）。绘制镜像服务的状态图，并对各状态进行简要的说明。
- 8.绘制 cinder 的架构图，并进行简要的说明。

- 9.绘制 cinder Volume 的创建流程，并进行简要说明。
- 10.绘制 ceilometer 采集监控数据到持久化存储的流程图，并进行简要的说明。
- 11.swift 使用称之为“Ring”的环形数据结构，在图中绘制 swift 的哈希 Ring，并进行简要的说明。
- 12.Swift 采用层次数据模型，绘制 Swift 数据模型图，并进行简要说明。
- 13.绘制 GRE 网络中，同一个 host 上同一个子网内云主机之间的通信过程。
- 14.绘制 GRE 网络中，云主机访问外网的流程图，并简要说明数据包的流向。
- 15.绘制 Docker 的系统架构图，并进行说明。
- 16.绘制 openstack 中 flat 网络的网络模式图。并做简要说明。
- 17.绘制 openstack 中 gre 网络的网络模式图。并做简要说明。
- 18.绘制 openstack 中 vlan 网络的网络模式图。并做简要说明。
- 19.绘制 openstack nova 的服务进程图，并做简要说明。

运维及脚本解读

- 1.针对安装脚本 iaas-pre-host.sh，逐行进行解读，指出各行内容所进行的操作以及各自的作用。
- 2.针对安装脚本 iaas-install-mysql.sh，逐行进行解读，指出各行内容所进行的操作以及各自的作用。
- 3.针对安装脚本 iaas-install-keystone.sh，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

4.针对安装脚本 `iaas-install-glance.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

5.针对安装脚本 `iaas-install-nova-controller.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

6.针对安装脚本 `iaas-install-nova-compute.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

7.针对安装脚本 `iaas-install-neutron-controller.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

8.针对安装脚本 `iaas-install-neutron-controller-gre.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

9.针对安装脚本 `iaas-install-neutron-controller-vlan.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

10.针对安装脚本 `iaas-install-neutron-controller-flat.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

11.针对安装脚本 `iaas-install-swift-controller.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

12.针对安装脚本 `iaas-install-heat.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

13.针对安装脚本 `iaas-install-cinder-controller.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

14.针对安装脚本 `iaas-install-cinder-compute.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

15.针对安装脚本 `iaas-install-ceilometer-controller.sh`，逐行进行解读，指出各

行内容所进行的操作以及各自的作用。

16.针对安装脚本 `iaas-install-ceilometer-compute.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

17.针对安装脚本 `iaas-install-alarm.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

18.针对安装脚本 `iaas-install-dashboard.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

19.针对安装脚本 `iaas-install-trove.sh`，逐行进行解读，指出各行内容所进行的操作以及各自的作用。

20.linux 下给文件 `start.sh` 设置权限为自己可读可修改可执行，组内用户为可读可执行不可修改，其余用户没有任何权限，那么设置该文件权限的命令是什么？

21.简述 `raid0` `raid1` `raid5` 三种工作模式的工作原理及特点。

22.Yum 命令与 rpm 命令的区别？如何使用 rpm 安装一个有依赖的包？

23.当我们在 IAAS 平台中创建了一台云主机，硬盘大小为 20G 且全分给了根目录，随着使用的过程中，发现根目录的容量不够了，现在需要扩容这台云主机的根目录（云主机的分区是 LVM 形式的），请把操作的过程描述一下。

24.当我们搭建完 iaas 平台，打开浏览器 dashboard，进行登录的时候，显示无效的证书，在确保使用的是正确的用户名和密码后，请回答下最大的原因是什么？

25.当我们搭建完 iaas 平台，在创建好网络之后，创建云主机的时候，报错“`No valid host was found. There are not enough hosts available`”，请给出你的解决思

路。

26.默认情况下，创建的 docker 容器的空间是 10G。在实际生产环境下，对 docker 容器进行热扩容(动态扩容)是非常重要的一个需求。请简述如何将容器根目录扩容。

任务二、职业素养

（一）职业素养

1.依工作作风、安全意识、团队协作和遵守考场纪律情况由裁判现场判分。