

# 运维面笔试 题库

日期：2022.04.22

# 《云计算运维面笔试题库》

提示：

- 1、题目类型包括：选择题、多选题、简答题、填空题、编程题等
- 2、题目前缀【类型】【知识点】

知识点目录			
零基础入门	Linux 系统管理与 Shell 编程	Linux 服务管理	数据库管理
集群管理	安全防护	虚拟化与云计算	Docker
Kubernerres	Python 自动化		

## 1、零基础入门

### 1. 【简答题】【云计算优势和就业方向】使用云计算有哪些优点？

使用云计算有下列优点：

- 1.备份数据和存储数据
- 2.强大的服务器功能
- 3.SaaS(软件即服务)
- 4.信息技术沙盒功能
- 5.提高生产力
- 6.具有成本效益，并节省时间

### 2. 【简答题】【云计算优势和就业方向】云计算与移动计算有何区别？

移动计算使用与云计算同样的概念。借助互联网而不是借助单个设备，云计算因数据而变得活跃。它为用户提供了需要按需获取的数据。在移动计算中，应用程序在远程服务器上运行，为用户提供了访问所存储数据的权限。

### 3. 【简答题】【云计算优势和就业方向】云架构有什么样的业务好处？

云架构具有的好处包括：

- a)无需基础设施投入

- b)适时的基础设施
- c)更高效地利用资源

#### 4.【简答题】【计算机常识】请描述计算机系统的定义？

计算机系统是由 CPU，存储器和相关电子设备的组合，与其连接的有外部设备及其操作系统

#### 5.【简答题】【计算机常识】计算机系统的基本组成部分是什么？

计算机系统的基本组件是中央处理器（算术逻辑单元和控制单元），存储器（主存储器和辅助存储器）以及输入和输出设备。

#### 6.【简答题】【计算机常识】路由器，调制解调器和交换机之间的区别？

调制解调器是建立与互联网连接的设备。

路由器是网络的“流量控制器”。它将调制解调器提供的信息提供给连接的各种设备，并创建网络地址转换（NAT）内部专用 IP 地址。

交换机只是连接同一网络上多个设备的设备。与路由器不同，交换机仅用于同一网络上设备间的通信。

#### 7.【简答题】【计算机基础】什么是微处理器？

微处理器是一组集成电路，包含计算机中央处理器的所有功能。

#### 8.【简答题】【计算机基础】32 位和 64 位处理器有什么区别？

32 位系统可以访问 32 个内存地址，即 4 GB 内存。64 位系统可以访问 64 个内存地址，即 18 亿 GB 的 RAM。64 位处理器可以轻松处理大于 4 GB 的任何内存量。

#### 9.【简答题】【计算机基础】什么是主存储器和辅助存储器？

主存储器（RAM）是计算机的主存储器，可由 CPU 直接访问。主存储器存储临时信息，直到该过程完成。辅助存储器（文件存储设备）是指可用于永久存储数据或信息的外部存储设备。

#### 10.【简答题】【计算机基础】什么是芯片组？

芯片组是集成电路的集合，其被设计为作为一个单元一起工作，以在计算机系统内执行特定任务。

### 11.【简答题】【计算机基础】.数据，信息和程序有什么区别？

数据是一个原始的，无组织的内容，需要进行处理以使其有意义。信息是一组数据，根据给定的要求以有意义的方式处理。程序是计算机要执行的一组特定的有序操作。

### 12.【简答题】【计算机基础】系统软件与应用软件有什么区别？

系统软件是用于与计算机硬件进行操作和通信的软件。它提供了运行应用程序软件的平台。应用程序软件是用户用于执行特定任务、特定用途软件。

### 13.【简答题】【计算机基础】何为网桥？防火墙的端口防护是指什么？

网桥：网桥（Bridge）像一个聪明的中继器。中继器从一个网络电缆里接收信号，放大它们，将其送入下一个电缆。相比较而言，网桥对从网卡上传下来的信息更敏锐一些。网桥是一种对帧进行转发的技术，根据 MAC 分区块，可隔离碰撞。网桥将网络的多个网段在数据链路层连接起来。

网桥也叫桥接器，是连接两个局域网的一种存储/转发设备，它能将一个大的 LAN 分割为多个网段，或将两个以上的 LAN 互联为一个逻辑 LAN，使 LAN 上的所有用户都可访问服务器。

防火墙端口防护：指通过对防火墙的端口开关的设置，关闭一些非必需端口，达到一定安全防护目的行为

### 14.【简答题】【操作系统基础】什么是操作系统？列举一些流行的操作系统。

操作系统是允许计算机硬件与软件通信和操作的软件。它提供了用户和计算机系统之间的接口。如 Windows，Linux 和 OSX 是一些流行的操作系统。

### 15.【简答题】【操作系统基础】计算机网络和分布式计算机系统的区别

两者在计算机硬件连接、系统拓扑结构和通信控制等方面基本都是一样的，它们都具有通信和资源共享的功能。

区别关键在于:分布式计算机系统是在分布式计算机操作系统支持下，进行分布式数据库处理的，也就是说各互联的计算机可以互相协调工作，共同完成一项任务，多台计算机上并行运行。且具有透明性，用户不知道数据、资源的具体位置，整个网络中所有计算机就像是一台计算机一样；而计算机网络却不具备这种功能，计算机网络系统中的各计算机通常是各自独立进行工作的。

**16.【简答题】【Windows Server 服务配置】**在 Windows Server 2003 系统中，默认情况下，用户的口令设置默认必须符合复杂性要求，且不可更改此账号策略，是否正确。并简要说明理由。

这种说法是不正确的。首先，在 Windows Server 2003 系统中，默认情况下，对用户账户口令是有复杂性要求的。如果服务器系统只是成员服务器，则通过 gpedit.msc 打开的本地组策略中的密码策略也是可以修改的，密码复杂性策略是否可以修改要看服务器当前的角色和所打开的组策略对象。总体来说，这个策略项还是可以修改的，无论是域控制器，还是成员服务器。

**17.【填空题】【Windows Server 服务配置】**如果想利用 Windows Server 2003 系统中 RRAS 的 NAT 功能，使局域网中的用户都能共享一条 ADSL 宽带上网，则需要新建（ ）接口。

请求拨号

**18.【简答题】【Windows Server 服务配置】**在 Windows Server 2003 系统中，默认情况下，用户的口令设置默认必须符合复杂性要求，且不可更改此账号策略，是否正确。并简要说明理由。

这种说法是不正确的。首先，在 Windows Server 2003 系统中，默认情况下，对用户账户口令是有复杂性要求的。如果服务器系统只是成员服务器，则通过 gpedit.msc 打开的本地组策略中的密码策略也是可以修改的，密码复杂性策略是否可以修改要看服务器当前的角色和所打开的组策略对象。总体来说，这个策略项还是可以修改的，无论是域控制器，还是成员服务器。

**19.【简答题】【网络基础】**你对互联网和万维网了解多少？

互联网是一个庞大的网络网络，网络基础设施包括多个广域网。万维网（或万维网）是一种通过互联网媒介访问信息的方式。

**20.【简答题】【网络基础】**什么是 VLAN？

VLAN 又称虚拟局域网，是指在网络层对局域网进行划分，一个 VLAN 组成一个逻辑子网，即一个独立的广播

域，各子网自己产生的广播网络流量被限制在各子网内部，降低数据帧的碰撞率，它可以覆盖多个网络设备，允许处于不同地理位置的网络用户加入到一个逻辑子网中

## 21.【简答题】【计算机基础】计算机中的端口共有多少个？

TCP0-65535，UDP0-65535 也就是共有  $65536 \times 2 = 131072$  个端口

## 22.【简答题】【网络基础】C类地址有多少ip地址？B类地址有多少ip地址？

C类：256 个 IP 地址

B类：65536 个 IP 地址

## 23.【简答题】【网络通信协议】ICMP是什么协议？处于哪一层？

ICMP 是（Internet Control Message Protocol）Internet 控制报文协议。它是 TCP/IP 协议簇的一个子协议，用于在 IP 主机、路由器之间传递控制消息。属于网络层协议。

## 24.【简答题】【网络通信协议】谈一谈 TCP 与 UDP 的区别。

TCP 与 UDP 都是传输层的协议，且都用端口号标识数据所达的进程。

TCP 提供的是面向连接服务，提供可靠交付。且具有流量控制和拥塞控制。可用于可靠要求高的场合如:SMTP，FTP，HTTP 等

UDP 提供的是无连接服务，提供不可靠交付，且无确认机制。主要用于即时强的场合如:视频聊天，语音电话等。

## 25.【简答题】【网络通信协议】网络协议的三个核心要素是什么？各有什么作用？

语法，定义了数据与控制信息的格式；

语义，定义了需要发出何种控制信息，完成何种响应动作以及作出何种响应；

同步，定义了事件实现顺序的详细说明。

## 26.【简答题】【网络通信协议】TCP/IP 网络协议的核心是什么？

TCP/IP 协议的核心是 TCP、UDP 和 IP 协议

## 27. 【简答题】【网络通信协议】你对 TCP / IP 模型了解多少？

TCP/IP 代表传输控制协议和 Internet 协议。它是当前 Internet 体系结构中使用的网络模型。协议是一组维护网络通信的规则。它由四层组成：分别是网络访问层，Internet 层，传输层和应用层

## 28. 【简答题】【网络通信协议】你对 OSI 模型了解多少？

OSI（开放系统互连）是应用程序如何通过网络进行通信的参考模型。它由七层组成：分别为物理，应用，表现，会话，传输，网络和数据链路层

## 29. 【简答题】【linux 常用命令】绝对路径用什么符号表示？当前目录、上层目录用什么表示？主目录用什么表示？切换目录用什么命令？

绝对路径：如/etc/init.d

当前目录和上层目录：./ ../

主目录：~/

切换目录：cd

## 30. 【简答题】【linux 常用命令】怎么查看当前进程？怎么执行退出？怎么查看当前路径？

查看当前进程：ps

执行退出：exit

查看当前路径：pwd

## 31. 【简答题】【linux 常用命令】怎么清屏？怎么退出当前命令？怎么执行睡眠？怎么查看当前用户 id？查看指定帮助用什么命令？

清屏：clear；

退出当前命令：ctrl+c 彻底退出；

执行睡眠：ctrl+z 挂起当前进程 fg 恢复后台；

查看当前用户 id：“id”：查看显示目前登陆账户的 uid 和 gid 及所属分组及用户名；

查看指定帮助：如 man adduser 这个很全 而且有例子；adduser --help 这个告诉你一些常用参数；info adduser。

**32. 【简答题】【Linux 常用命令】ls 命令执行什么功能？可以带哪些参数，有什么区别？**

ls 执行的功能：列出指定目录中的目录，以及文件

哪些参数以及区别：a 所有文件 l 详细信息，包括大小字节数，可读可写可执行的权限等

**33. 【简答题】【linux 常用命令】建立软链接(快捷方式)，以及硬链接的命令。**

软链接：ln -s link source

硬链接：ln link source

**34. 【简答题】【linux 常用命令】复制文件：cp 7. 文件权限修改用什么命令？格式是怎样的？**

权限修改：chmod

格式如下：

\$ chmod u+x file 给 file 的属主增加执行权限

\$ chmod 751 file 给 file 的属主分配读、写、执行(7)的权限，给 file 的所在组分配读、执行(5)的权限，给其他用户分配执行(1)的权限

\$ chmod u=rwx, g=rx, o=x file 上例的另一种形式

\$ chmod =r file 为所有用户分配读权限

\$ chmod 444 file 同上例

\$ chmod a-wx, a+r file 同上例

\$ chmod -R u+r directory 递归地给 directory 目录下所有文件和子目录的属主分配读的权限

**35. 【简答题】【linux 常用命令】目录创建用什么命令？创建文件用什么命令？复制文件用什么命令？**

创建目录：mkdir

创建文件：典型的如 touch, vi 也可以创建文件，其实只要向一个不存在的文件输出，都会创建文件

**36. 【简答题】【linux 常用命令】查看文件内容有哪些命令可以使用？**

vi 文件名 #编辑方式查看，可修改

cat 文件名 #显示全部文件内容

more 文件名 #分页显示文件内容



less 文件名 #与 more 相似，更好的是可以往前翻页

tail 文件名 #仅查看尾部，还可以指定行数

head 文件名 #仅查看头部，还可以指定行数

**37.【简答题】【Linux 常用命令】删除文件用哪个命令？如果需要连目录及目录下文件一块删除呢？删除空文件夹用什么命令？**

```
rm -r 目录名
```

**38.【简答题】【linux 常用命令】怎么使一个命令在后台运行？**

一般都是使用 `&` 在命令结尾来让程序自动运行。(命令后可以不追加空格)

**39.【简答题】【Linux 常用命令】哪个命令专门用来查看后台任务？**

```
jobs -l
```

**40.【简答题】【linux 软件包分类与管理】堆和栈的区别是？**

(1) 存储内容不同

栈：在函数调用时，栈中存放的是函数中（最底下是函数调用后的下一条指令）的各个参数（局部变量）。

堆：一般是在堆的头部用一个字节存放堆的大小。堆中的具体内容由程序员分配。

(2) 管理方式上不同

栈：由系统自动分配并释放空间。例如，声明在函数中一个局部变量 `int b;` 系统自动在栈中为 `b` 开辟空间，当对应的生存周期结束后栈空间被自动释放。

堆：需要程序员指定大小手动申请和手动释放，在 C 语言中使用 `malloc` 函数申请，使用 `free` 函数释放。

(3) 空间大小不同

栈：获取空间较小。在 Windows 下一般大小是 1M 或 2M，当剩余栈空间不足时，分配失败 `overflow`。

堆：获得空间根据系统的有效虚拟内存有关，比较灵活、大。

(4) 能否产生碎片不同

栈：不会产生碎片，空间连续。

堆：采用的是链表的存储方式，会产生碎片。

(5) 生长方向不同

栈：向低地址扩展的数据结构，是一块连续的内存区域。

堆：向高地址扩展的数据结构，是不连续的内存区域。这是由于系统是用链表空闲内存地址来存储的，自然不连续，而链表的遍历方向是由低地址向高地址。

(6) 分配方式不同

栈：有 2 种分配方式：静态分配和动态分配，静态由编译器完成，例如局部变量；动态由 malloc 函数实现，由编译器进行释放。

堆：都是动态分配的，没有静态分配的堆。

(7) 分配效率不同

栈：由系统自动分配，速度较快。但程序员无法控制。

堆：由 new 分配的内存，一般速度比较慢，而且容易产生内存碎片，不过用起来最方便。

#### 41. 【简答题】【linux 软件包分类与管理】Linux 中的文件包括哪些？

执行文件，普通文件，目录文件，链接文件和设备文件，管道文件。

#### 42. 【简答题】【linux 常用命令】查看/var/log 目录下文件数

```
ls /var/log/ -lR | grep "^-" | wc -l。
```

#### 43. 【简答题】【linux 软件包分类和管理】linux 系统下有几个包？分别是什么？

两个：源码包和二进制包。

#### 44. 【简答题】【linux 软件包分类与管理】请分别说明源码包和二进制包的优缺点？

源码包

优点：开源，可以修改源代码。

源码包先进行编译后再安装，更适合自己的系统，也更加稳定，效率更高。

可以自由选择所需功能。

卸载方便，不会遗留垃圾文件。

缺点：安装步骤多，如果安装较大的软件集合，容易出现拼写错误。

安装时间较长。

因为是编译安装，安装过程如果报错，新手很难解决。

RPM 包

#### 45. 【简答题】【linux 软件包分类与管理】源码包的卸载命令？

不需要卸载命令，直接删除安装目录即可，不会遗留任何垃圾文件。

```
rm -rf /usr/local/apache2/
```

#### 46. 【简答题】【linux 软件包分类与管理】linux 下的软件包的区别？

几乎所有的 Linux 发行版本都使用某种形式的软件包管理安装、更新和卸载软件。与直接从源代码安装相比，软件包管理易于安装和卸载；易于更新已安装的软件包；易于保护配置文件；易于跟踪已安装文件。

通常，用 tar 打包的，都是源程序；而用 rpm、dpkg 打包的则常是可执行程序。一般来说，自己动手编译源程序能够更具灵活性，但也容易遇到各种问题和困难。而相对来说，下载那些可执行程序包，反而是更容易完成软件的安装，当然那样灵活性就差多了。所以一般一个软件总会提供多种打包格式的安装程序的。你可以根据自己的情况来选择。

#### 47. 【简答题】【linux 软件包分类与管理】Linux 下软件包的格式主要有哪几种？

Linux 下的软件安装包主要有 rpm、deb、tar.gz 三种格式

## 2、Linux 系统管理与 shell 编程

### VIM 编辑器:

#### 48. 【填空题】vi 编辑器具有两种工作模式: ( 和 )。

命令模式 编辑模式

#### 49. 【简答题】在用 vim 编辑文本中，把 40 行以后的 123456 全部替换成 525421?51、Linux 系统中网络管理员对 WWW 服务器进行访问、控制存取和运行等控制，这些控制可在下面什么文件中体现？

\$s/123456/525421/g

#### 50. 【选择题】一台主机要实现通过局域网与另一个局域网通信，需要做的工作是 (C)。

- A 配置域名服务器
- B 定义一条本机指向所在网络的路由
- C 定义一条本机指向所在网络网关的路由
- D 定义一条本机指向目标网络网关的路由

51.【选择题】在局域网络内的某台主机用 `ping` 命令测试网络连接时发现网络内部的主机都可以连同，而不能与公网连通，问题可能是（C）。

- A 主机 IP 设置有误
- B 没有设置连接局域网的网关
- C 局域网的网关或主机的网关设置有误
- D 局域网 DNS 服务器设置有误

52.【选择题】下列提法中，不属于 `ifconfig` 命令作用范围的是（D）。

- A 配置本地回环地址
- B 配置网卡的 IP 地址
- C 激活网络适配器
- D 加载网卡到内核中

53.【选择题】Linux 有三个查看文件的命令，若希望在查看文件内容过程中可以用光标上下移动来查看文件内容，应使用（C）命令。

- A `cat`
- B `more`
- C `less`
- D `menu`

54.【选择题】改变文件所有者的命令为（C）。

- A `chmod`
- B `touch`
- C `chown`
- D `cat`

55.【简答题】用 `vim`，给文件 `test.sh` 的 132 到 186 行一次性添加#注释？

132, 186s/N/#/g

56.【简答题】`Sar` 提供了什么？`Sar` 日志的存储位置在哪里？

`Sar` 收集，报告，保存系统的活动信息。`Sar` 命令的默认版本（CPU 使用率报告）可能是用户执行系统活动调查时第一个用到的，因为它监控着大部分系统资源。当 CPU 利用率接近 100%时，采样的工作负载时 CPU 密集

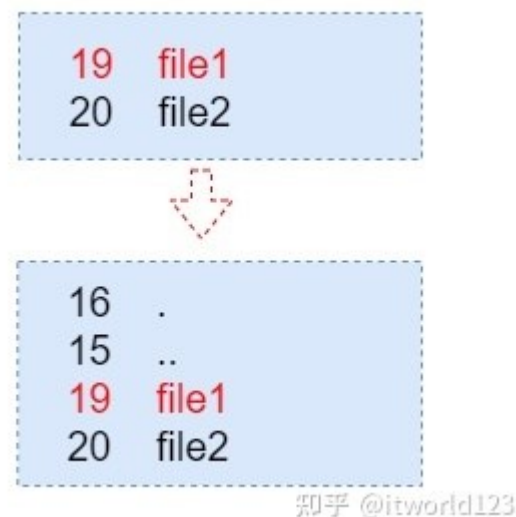
型的。

Sar 命令的默认日志文件位于 /var/log/sa/sadd 文件，其中 dd 代表着当前日期。

**57.【简答题】**我们可以为目录创建一个软链接，但是当我们为目录创建硬链接的时候将会出现如下错误提示：

```
ln: newdir/test_dir: hard link not allowed for directory
```

解析：为什么不能为目录创建硬链接呢？这是因为在目录中除了文件相关内容外，还有一个指向父目录的目录项，具体如下图所示的“.”目录。如果我们在其它目录创建该目录的硬链接，那么该内容就会被改写，从而导致混乱。



除了导致目录数据混乱外，还有一种问题就是会形成路径环。至于如何形成路径环本文不再赘述，大家可以自己思考一下。

**58.【简答题】**如果一个文件有硬链接，删除源文件会怎样？

如果存在硬链接，删除源文件将不会删除文件的数据。通过硬链接仍然能够访问文件的数据。

解析：为什么会出现这种情况？这是因为当有目录项指向 inode 的时候，inode 会增加引用计数。通常情况下引用计数是 1，当有一个硬链接时为 2，再有一个硬链接的时候就变成了 3。

当我们删除普通文件（无硬链接）的时候，首先会删除目录项中的内容，然后将 inode 的引用计数减 1，如果此时 inode 中引用计数为 0，则删除该 inode，同时删除相关联的数据。如果非 0，则只删除目录项，而不删除相关内容。所以，如果存在硬链接则可以正常访问文件内容。

## Linux 高级文件系统管理

**59.【简答题】 Ls 命令执行什么功能？可以带哪些参数，有什么区别？**

ls 执行的功能：列出指定目录中的目录，以及文件

哪些参数以及区别：a 所有文件 l 详细信息，包括大小字节数，可读可写可执行的权限等

**60.【简答题】查看当前路径：pwd63、 ssh -o stricthostkeychecking = no - o ConnectTimeout = 60 分别代表什么意思？**

ssh 加接时不需要输入 key 检查，然后六十秒为超时时间

**61.【简答题】 shell 变量类型有哪些？列举出常用的 4 个预定义变量？**

1) 1、自定义变量 2、系统环境变量 3、预定义变量 4、位置变量

2) \$? \$# \$\* \$\$

**62.【简答题】系统配置文件是哪几个？有什么用？**

/etc/profile /etc/bashrc

可以在里面定义变量，用户每打开一个终端时加载的文件，针对所有用户生效

环境变量配置文件

**63.【简答题】写出正则表达式中，子表达式后面跟有量词，最后一次匹配到的内容，作为子表达式的值**

我们知道非全局匹配下的 match 方法的结果和 exec 方法的结果一样，会得到所有的匹配信息，包括子表达式的内容。

以下例子中，子表达式的值是 456，匹配多个 3 数字，最后一个 3 数字的内容。

```
var str = "12345678";
```

```
var reg = /(\d{3})*;/
```

```
console.log(str.match(reg))
```

```
["123456", "456", index: 0, input: "12345678", groups: undefined]
```

**64.【简答题】请用正则表达式匹配一个[1-255]的数字：**

括号内匹配符合条件的值都可以，但是要以匹配的括号内容开头且结尾。这样就不会匹配 345 中的 34 和 5。

代码：var str = "248";

```
var reg = /^[1]\d\d|[2][0-4]\d|[2]5[0-5]||[1-9]\d|[1-9])$/  
console.log(str.match(reg));
```

字符处理三剑客

65. 【简答题】显示/proc/meminfo 文件中以不区分大小的 s 开头的行；

```
grep -i ^s /proc/meminfo  
cat /proc/meminfo|grep -i ^s
```

66. 【简答题】 访问控制模型包括三个要素？

主体 (Subject) 指主动对其它实体施加动作的实体

客体(Object) 是被动接受其他实体访问的实体

控制策略(Policy)为主体对客体的操作行为和约束条件

shell 实例

67. 【选择题】 以下不合法的 shell 头是(不合法指运行会报错): (C)

- A、#!/bin/bash
- B、#-/bin/bash
- C、!#/bin/bas

68. 【选择题】在 shell 脚本中，用来读取文件内各个域的内容并将其赋值给 shell 变量的命令是 (D)

- A、fold
- B、join
- C、tr
- D、read

### 3、Linux 服务管理

69. 【选择题】【定时任务】 crontab 的使用中操作符/是什么意思？

- A、取值范围内的所有数字
- B、每过多少个数字
- C、从 X 到 Z
- D、散列数字

B

70.【判断题】【Linux 服务管理】Linux 服务管理有两种方式 service 和 systemctl.

(√)

71.【简答题】【Linux 系统修复】什么是 linux 修复模式

Linux 的修复模式就是 linux rescue；修复模式并不需要系统密码，当进入修复模式后，系统会提示我们要挂载哪个分区上的文件系统

72.【简答题】【Linux 服务管理】服务（service）的概念？

服务(service) 本质就是进程，但是是运行在后台的，通常都会监听某个端口，等待其它程序的请求，比如(mysql , sshd 防火墙等)，因此我们又称为守护进程

73.【简答题】【Linux 启动管理】 Linux 系统启动过程是哪些？

开机自检--MBR 引导—GRUB 菜单—加载内核—init 进程初始化

74.【选择题】【Linux 网络基础】查看所有网卡信息，无论是否激活？（A）

A、ifconfig -a    B、ifconfig -s    C、ifconfig -v    D、ifconfig

75.【判断题】【Linux 网络基础】虚拟网卡可以为一块真实网卡配置多个 IP 参数，可同时和多个网络通信。(√)

76.【简答题】【Linux 进程管理和工作管理】什么是进程管理？

进程是执行的程序。程序被执行后，执行者的权限和属性、程序代码与所需数据等都会被加载到内存中，操作系统赋予这个内存内的单元一个标识符 PID。

77.【简答题】【Linux 进程管理和工作管理】Linux 手工启动进程的方式有哪些？

前台启动和后台启动 2 种方式。



**78.【简答题】【定时任务】crond 是什么？**

crond 是 linux 系统中用来定期执行命令或指定程序的一种服务或软件。

**79.【简答题】【日志管理与 ELK 日志工具】Elasticsearch 是什么搜索引擎？**

Elasticsearch 是开源分布式搜索引擎。

**80.【选择题】【日志管理与 ELK 日志工具】ELK 由那几部分组成？（D）**

A、Elasticsearch    B、Logstash    C、Kibana    D、以上都是

**81.【简答题】【SSH 服务管理和私钥认证登录】什么是私钥？**

用来解密公钥加密的数据(公钥用来给数据加密，用公钥加密的数据只能使用私钥解密)

**82.【简答题】【SSH 服务管理和私钥认证登录】SSH 是如何工作的？**

1、连接建立 2、版本协商 3、算法协商 4、密钥交换 5、用户认证 6、会话请求 7、会话交互

**83.【填空题】【DHCP 服务管理】DHCP 服务器安装好后并不是立即就可以给 DHCP 客户端提供服务，它必须经过一个“\_\_\_\_\_”步骤。未经此步骤的 DHCP 服务器在接收到 DHCP 客户端索取 IP 地址的要求时，并不会给 DHCP 客户端分派 IP 地址。**

授权

**84.【选择题】【DHCP 服务管理】使用“DHCP 服务器”功能的好处是（D）**

A、降低 TCP/IP 网络的配置工作量  
B、增加系统安全与依赖性  
C、对那些经常变动位置的工作站 DHCP 能迅速更新位置信息  
D、以上都是

**85.【简答题】【DNS 服务管理】什么是 DNS？**

通过主机名，最终得到该主机名对应的 IP 地址的过程叫做域名解析。

**86. 【简答题】【DNS 服务管理】 DNS 有哪两种解析方式?如何解析?**

正向解析:由域名查找 IP 地址;

反向解析:由 IP 地址查找域名。

**87. 【简答题】【VSftp 文件服务器】如何限制用户和禁止浏览超出他们的主目录? 怎么样?**

可以将用户限制到其主目录并浏览超出主目录。这可以通过在 ftp 配置文件（即 vsftpd、conf）中启用 chroot 选项来实现。

chroot\_local\_user=YES

**88. 【简答题】 【VSftp 文件服务器】如何禁用从 FTP 服务器的下载?**

从 FTP 服务器下载禁用可以通过修改参数'download\_enable'来实现。如果设置为 NO，所有下载请求将被拒绝。默认值为 YES 表示，下载已启用。

**89. 【简答题】【NFS 网络文件系统】 TCP/IP 把网络通信分为四层，NFS 网络文件系统是哪一层?**

应用层

一共有四个层，分别为：链路层，网络层，传输层，应用层

**90. 【选择题】【Samba 配置】 Samba 服务器的配置文件是（D）**

- A、httpD、conf
- B、inetD、conf
- C、rC、samba
- D、smB、conf

**91. 【选择题】【Samba 配置】 以下关于 Samba 的描述中，不正确的是（D）。**

- A. Samba 采用 SMB 协议
- B. Samba 支持 WINS 名字解析
- C. Samba 向 Linux 客户端提供文件和打印机共享服务
- D. Samba 不支持 Windows 的域用户管理

## LAMP 网站开发平台部署题

### 92.【简答题】【apache 配置部署操作】如何配置 apache 的虚拟主机

```
vim httpD.conf
```

```
#Include conf/extra/httpd-vhosts.conf
```

把该行最前面的#去掉，然后修改：

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Order deny, allow
```

```
Deny from all
```

改为

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Order deny, allow
```

```
Allow from all
```

再 vim conf/extra/httpd-vhosts.conf 根据实际需求修改

### 93.【简答题】【apache 配置部署操作】如何在 Apache 中改变默认的端口，以及如何侦听其中的指令工作？

答案：在 httpD.conf 文件中有一个指令“Listen”可以让我们改变默认的 Apache 端口。在 Listen 指令的帮助下我们可以在不同的端口还有不同的接口进行 Apache 侦听。

### 94.【简答题】【文件查询】怎么确定你的 php.ini 在哪个目录下？

```
php -i |grep 'Configuration File'
```

## Apache 服务器问题

### 95.【简答题】【Apache】什么是 Apache web 服务器？

Apache web 服务器 HTTP 是一个非常流行、功能强大并且开源，用于管理 web 站点并向网络提供 web 文件服务。它基于 HTTP 超文本传输协议运行，这一协议提供了服务器和客户端 web 浏览器通信的标准。它支持 SSL，CGI 文件，虚拟主机还有许多其它的功能特性。

**96.【简答题】 【文件查询】怎么确定你的 php.ini 在哪个目录下?**

使用：php -i |grep 'Configuration File'命令

## **Apache 服务器问题**

**97.【简答题】 【Apache】什么是 Apache web 服务器?**

答案：Apache web 服务器 HTTP 是一个非常流行、功能强大并且开源，用于管理 web 站点并向网络提供 web 文件服务。它基于 HTTP 超文本传输协议运行，这一协议提供了服务器和客户端 web 浏览器通信的标准。它支持 SSL，CGI 文件，虚拟主机还有许多其它的功能特性。

**98.【简答题】【Apache 命令】如何在 Apache 中改变默认的端口，以及如何侦听其中的指令工作?**

答案：进入 apache 安装目录，找到 conf 配置文件夹，点击进入后找到 httpd.conf。你选择用记事本的方式打开后 ctrl+f 查找

listen 直到找到 listen 80，然后把 80 改成你想要更改的端口

## **LNMP 网站开发平台部署**

**99.【选择题】【邮件系统】以下均是邮件系统的工作过程，正确的顺序是（B）。**

- ①用户使用客户端创建新邮件；
- ②发送方邮件服务器使用 SMTP 将邮件发送到接收方的邮件服务器，接收方的邮件服务器将收到的邮件存储在用户的邮箱中，并等待用户处理；
- ③客户端软件使用 SMTP 将邮件发到发送方的邮件服务器；④接收方客户端软件使用 POP3/IMAP4 从邮件服务器中读取邮件。

- A: ①②③④
- B: ①③②④
- C: ①④②③
- D: ①②④③

**100.【选择题】【邮件系统】在邮件服务器之间实现邮件投递的协议是（ ）。**

- A、IMAP
- B、TFTP
- C、SMTP
- D、RARP

答案: C

解析: SMTP 称为简单 Mail 传输协议 (Simple Mail Transfer Protocol), 是向用户提供高效、可靠的邮件传输

## Extmail 综合邮件部署平台

### 101. 【简答题】【邮件系统】SMTP 服务器, 邮件传输代理(MTA) 由什么软件提供

Postfix

### 102. 【简答题】【环境搭建】如何在 Linux 系统上集成项目所需环境(mysql+jdk+tomcat)

安装 MySQL

# rpm -qa | grep mysql ## 查看该操作系统上是否已经安装了 mysql 数据库,  
有的话, 可以通过 rpm -e 命令 或者 rpm -e --nodeps 命令来卸载掉

```
# yum install mysql-server mysql mysql-devel
# service mysqld start
# chkconfig --list | grep mysqld
mysqld 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

用上面的命令查看到 MySQL 并没有设置开机启动, 所以需要设置开机启动

```
# chkconfig mysqld on
```

为了方便远程管理, 防火墙中打开 3306 端口

```
# vi /etc/sysconfig/iptables
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
```

重启防火墙, 使端口配置生效

```
# service iptables restart
```

设置 MySQL 数据库 root 用户的密码:

```
# mysqladmin -u root password 'mysql'
```

登录数据库:

```
# mysql -uroot -p
```

MySQL 授权远程访问 (先用 root 登录 mysql)

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysql' WITH GRANT
OPTION;

mysql> FLUSH PRIVILEGES;
```

安装 JDK

```
# rpm -qa | grep jdk ##查看该操作系统上是否已经安装了 jdk,
```

有的话, 可以通过 rpm -e 命令 或者 rpm -e --nodeps 命令来卸载

接下来在 windows 客户端使用 filezilla 工具将已经下载好的 jdk-7u80-linux-i586.rpm 上传到 root 目录下, 紧接着执行以下命令

```
#chmod +ux jdk-7u80-linux-i586.rpm

# rpm -ivh jdk-7u80-linux-i586.rpm

# rpm -qla jdk
```

配置环境变量:

```
# vi /etc/rC、local
```

加入

```
export PATH=/usr/java/jdk1.8.0_11/bin:$PATH

export CLASSPATH=.:usr/java/jdk1.8.0_11/jre/lib

export JAVA_HOME=/usr/java/jdk1.8.0_11
```

```
# source /etc/rC、local ##使配置文件生效
```

```
# echo $PATH
```

下载安装 Tomcat

```
# wgethttp://apache.fayeA、com/tomcat/tomcat-7/v7.0.85/bin/apache-tomcat-7.0.85.tar.gz

# tar -xzf apache-tomcat-7.0.85.tar.gz

# mv apache-tomcat-7.0.85 tomcat7
```

进入 tomcat 服务器的 bin 目录, 然后执行"./startup.sh"命令启动 Tomcat 服务器

查看 tomcat 服务器的 Log 信息, 看看 tomcat 服务器是否已经正常启动, 进入 tomcat 服务器下的 logs 目录, 打开 catalinA、out 文件进行查看

Rsync 同步服务器

### 103.【选择题】NFS 是（ C ）系统

- A、文件
- B、磁盘
- C、网络文件

## D、操作

### 104.【简答题】Nginx FCGI 运行原理是什么？

Nginx 不支持对外部动态程序的直接调用或者解析。所有的外部程序(包括 PHP)必须通过 FastCGI 接口来调用。

FastCGI 接口在 Linux 下是一个 socket，为了调用 CGI 程序，还需要一个 FastCGI 的 wrapper(可以理解为用户启动另一个程序的程序)，这个 wrapper 绑定在某个固定的 socket 上。

当 Nginx 将 CGI 请求发送给这个 socket 的时候，通过 FastCGI 接口，wrapper 接受到请求，然后派生出一个新的线程，这个线程调用解释器或外部的程序处理脚本来读取返回的数据。

然后 wrapper 再将返回的数据通过 FastCGI 接口，沿着固定的 socket 传递给 Nginx，最后 Nginx 将返回的数据发送给客户端。

FastCGI 的主要优点就是把动态的语言和 HTTP 服务器分离开来，使 Nginx 专门处理静态的请求，动态的请求直接使用 PHP/PHP-FPM 服务器专门处理。

## Nginx 服务器

### 105.【简答题】Nginx 应用场景？

- (1) http 服务器。Nginx 是一个 http 服务可以独立提供 http 服务。可以做网页静态服务器。
- (2) 虚拟主机。可以实现在一台服务器虚拟出多个网站，例如个人网站使用的虚拟机。
- (3) 反向代理，负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用 nginx 做反向代理。并且多台服务器可以平均分担负载，不会应某台服务器负载高宕机而某台服务器闲置的情况。
- (4) nginx 中也可以配置安全管理、比如可以使用 Nginx 搭建 API 接口网关，对每个接口服务进行拦截。

## 邮件服务器

### 106.【简答题】什么是 FCGI？

FastCGI 是一个可伸缩的、高速的在 HTTP 服务器的动态脚本语言间通信的接口(在 Linux 下，FastCGI 就是 socket，这个 socket 可以是文件 socket 或 IPsocket)。

FastCGI 采用 C/S 架构，它可以将 HTTP 服务器和脚本服务器分开，同时还可以在脚本解析服务器上启动一个或多个服务器来解析守护进程。

当 HTTP 服务器遇到动态程序的时候，可以将其直接交付给 FastCGI 进程来执行，然后将得到的结果返回给浏览器。这种方式可以让 HTTP 服务器专一的处理静态的请求。

这会很高的提高整个应用系统的性能。

### 107.【简答题】Nginx 的优缺点？

优点：

占内存小，可实现高并发连接，处理响应快

可实现 http 服务器、虚拟主机、方向代理、负载均衡

Nginx 配置简单

可以不暴露正式的服务器 IP 地址

缺点：

动态处理差：nginx 处理静态文件好，耗费内存少，但是处理动态页面则很鸡肋，现在一般前端用 nginx 作为反向代理抗住压力，

### 108.【简答题】安装依赖 rpm 包 (这里使用了本地 cdrom 源, 如果你不是本地 yum 源, 请直接 yum install 后边的 rpm 包)

```
yum --disablerepo=* --enablerepo=c5-media install httpd php php-mysql mysql mysql-server mysql-devel openssl-devel perl-DBD-MySQL tcl tcl-devel libart_lgpl libart_lgpl-devel libtool-ltdl libtool-ltdl-devel expect
```

### 109.【简答题】ssh2 的默认监听端口是：

22

### 110.【简答题】ssh 的配置文件是

/etc/ssh/ssh——config

### 111.【选择题】要配置 NFS 服务器，在服务器端主要配置（ C ）文件

A、/etc/rC、d/rC、inetl

B、/etc/rC、d/rC、m

C、/etc/exports

D、/etc/rC、d/rC、s

## 4、数据库管理

mysql 管理：



**112.【简答题】【MySQL】**请简洁地描述下 MySQL 中 InnoDB 支持的四种事务隔离级别名称，以及逐级之间的区别？

读未提交(read uncommitted)

读已提交(read committed)

可重复读(repeatable read)

串行(serializable)

**113.【简答题】【MySQL】**explain 的作用？

- ①表的读取顺序。(对应 id)
- ②数据读取操作的操作类型。(对应 select\_type)
- ③哪些索引可以使用。(对应 possible\_keys)
- ④哪些索引被实际使用。(对应 key)
- ⑤表直接的引用。(对应 ref)
- ⑥每张表有多少行被优化器查询。(对应 rows)

**114.【简答题】【MySQL】**mysql 主从集群：

(1) 配置 mysql 主从模式的原因

Mysql 内建的复制功能是构建大型、高性能应用程序的基础。在实际企业应用环境当中，单台 mysql 数据库是不足以满足日后业务需求的。

譬如当服务器发生故障，而没有备份服务器来提供服务时，业务就必须得停止，这样会对企业带来巨大的损失。

为了提高数据库服务器的稳定性，加快数据处理的效率，保护数据免受意外的损失，我们采用 mysql 的主从复制方式，分离对数据库的查询和更新操作，使用从服务器上备份的数据保证来数据的安全性和稳定性。

(2) Mysql 主从复制的原理是什么

MySQL 的 Replication 是一个异步的复制过程，从一个 MySQL instace(我们称之为 Master)复制到另一个 MySQL instance(我们称之 Slave)。

在 Master 与 Slave 之间的实现整个复制过程主要由三个线程来完成，其中两个线程(Sql 线程和 IO 线程)在 Slave 端，另外一个线程(IO 线程)在 Master 端。

在执行这个主从复制之前，首先必须打开 Master 端的 Binary Log(MySQL-bin.xxxxxx)功能，否则无法实现。

在启动 MySQL Server 的过程中使用“log-bin”参数选项

在 my.cnf 配置文件中的 MySQLd 参数组中增加“log-bin”参数项。

### 115.【简答题】【MySQL】mysql 主主集群:

(1) 多主自增长 ID 重复怎么解决

在主主同步配置时, 需要将两台服务器的 `auto_increment_increment` 设置为 `n`, `auto_increment_offset` 分别配置为 `1 2 ...`, 第一台从 1 开始, 第二台就是 2, 以此类推

(2) 在主主复制结构中, 两台服务器的任何一台上面的数据库发生了改变都会同步到另一台服务器上, 这样两台服务器互为主从, 并且都能向外提供服务。配置的时候需要注意什么

主键重复, `server-id` 不能重复。

### 116.【简答题】【Amoeba】部署实施 Amoeba 读写分离集群:

(1) 什么是 Amoeba?

Amoeba 是 mysql 的代理应用, 可以用 Amoeba 来灵活的代理 mysql 或 mysql 集群, 客户端直接访问 Amoeba 服务就能访问到它所代理的 mysql 数据库了, 访问方式是用 mysql 指令远程访问。

(2) Amoeba 能实现什么功能?

1.mysql 集群的读写分离

2.mysql 集群的负载均衡

3.mysql 集群的数据切片

### 117.【简答题】【MHA】MHA 高可用性架构:

(1) 说说 MHA 集群结构

MHA 一般至少是需要 3 台数据库服务器, 而且额外还需要一台服务器来做为 Manager 端

(2) 谈谈服务角色

主要是接收管理节点所发出指令的代理, 代理需要运行在每一个 mysql 节点上。简单讲 `node` 就是用来收集从节点服务器上所生成的 `bin-log`。对比打算提升为新的主节点之上的从节点的是否拥有并完成操作, 如果没有发给新主节点在本地应用后提升为主节点。每个复制组内部和 Manager 之间都需要 ssh 实现无密码互连, 只有这样, 在 Master 出故障时, Manager 才能顺利的连接进去, 实现主从切换功能。

### 118.【简答题】【Redis】Redis 基础:

(1) redis 有哪五大类型

String (字符串)

Hash (哈希)

List (列表)

Set (集合)

zset(sorted set: 有序集合)

## (2) 如何理解 String 类型

string 是 redis 最基本的类型，你可以理解成与 Memcached 一模一样的类型，一个 key 对应一个 value。value 其实不仅是 String，也可以是数字。string 类型是二进制安全的。意思是 redis 的 string 可以包含任何数据。比如 jpg 图片或者序列化的对象。string 类型是 Redis 最基本的数据类型，string 类型的值最大能存储 512MB。

## 119. 【简答题】【redis-cluster 集群】redis-cluster 集群部署：

### (1) 如何 rediscluster 集群部署添加新节点

准备新的节点 在集群目录 redis\_cluster 目录下增加 redis7006 和 redis7007 目录 mkdir redis7006 mkdir redis7007  
增加完成后的目录 复制端口 7000 的 redis.co...

添加主节点 (1)向集群中添加节点 7006，注意一定要保证节点里面没有添加过任何数据，不然添加会报错。 cd /usr/local

### (2) rediscluster 集群的概念是什么

由多个 Redis 服务器组成的分布式网络服务集群；  
集群之中有多个 Master 主节点，每一个主节点都可读可写；  
节点之间会互相通信，两两相连；  
Redis 集群无中心节点。

## 120. 【简答题】【redis】redis 优化 mysql:

### (1) redis 主要消耗什么物理资源

内存

### (2) redis 与 memcached 相比有哪些优势

“memcached 所有的值均是简单的字符串，redis 作为其替代者，支持更为丰富的数据类型 redis 的速度比 memcached 快很多 redis 的速度比 memcached 快很多 redis 可以持久化其数据 redis 可以持久化其数据”

## 121. 【简答题】【KickStart+PXE+DHCP】KickStart+PXE+DHCP 无人值守安装：

### (1) KickStart+PXE+DHCP 无人值守安装的原理是什么

利用 PXE 协议使用服务器通过网络方式启动，后向 DHCP 服务器请求 IP，连接后将服务端的文件下载到本地，执行一系列的操作

### (2) 举例 KickStart+PXE+DHCP 无人值守安装的过程

第一步：PXE-client 发送请求

将支持 PXE 启动方式的服务器设置成 PXE 启动，PEX 客户端通过 PXE boot ROM 以 udp 的形式发送一个广播包，请求 DHCP 服务器分配 IP 地址

第二步：DHCP 应答请求并回应

DHCP 服务器收到请求后，验证是否来自合法的 PXE 客户端请求，验证通过后，回应 PXE 客户端，回应信息

中包括分配的 IP 地址，pxelinux 启动程序（TFTP）的位置，以及配置文件的位置

第三步：PXE-client 请求下载启动文件

PXE 客户端收到回应后，向 TFTP 服务器请求下载所需的启动系统安装文件（文件包括：pxelinux.0、pxelinux.cfg/default、vmlinuz、initrd、img 等文件）

第四步：TFTP 服务器响应请求并传送文件

当 TFTP 服务器收到请求后，服务器会响应请求并应答请求，之后传送所需的文件给客户端

第五步：PXE-client 请求下载自动应答文件

PXE 客户端通过 pxelinux.cfg/default 文件成引导 linux 安装后，安装程序必须先确定通过什么方式安装系统，如果是通过网络，则会在此时进行初始化网络，并定位安装系统所需要的二进制包以及配置文件的位置，接着读取文件中指定的自动应答文件 ks.cfg，然后根据文件位置请求下载文件

第六步：客户端安装系统

将 ks.cfg 下载到本地，通过文件找到安装系统的 ISO 文件位置，并请求下载所需的软件包，正常连接后，开始传输软件包，最终开始安装操作系统，安装完成后重新启动

## 122.【简答题】Nosql 数据库概述：

（1）什么是 NoSQL？

NoSQL 涵盖了各种不同的数据库技术，这些技术是根据现代应用程序的要求而开发的。

开发人员正在使用一些创建大量的、快速变化的数据类型的应用程序，数据类型有结构化的，半结构化的，非结构化的和多态的样式

12 到 18 个月的瀑布发展周期已经过去了。现在，小团队在没日没夜地工作着，软件快速更新换代，每周或两周推送代码，甚至会有每天多次的推送代码，以更新软件

曾经服务于有限受众的应用程序，现在必须一直处于开放状态提供服务，可从不同的设备访问并且来自世界各地的访问者多达百万级别

现在，各个组织都在使用开源软件、商品服务器和云计算，而不是大型单片机服务器和存储基础设施的横向扩展架构

关系型数据库的设计不是为了应对现有应用程序的规模和速度的挑战，也不是为了利用当代可用的商品存储和处理能力而建立的。

（2）NoSQL 数据库的类型有哪些

document databases：每一组 key/value，其中 value 可以是复杂的数据结构，可以称之为文档。文档可以包括不同的键值对，或者键阵列，甚至是嵌套关系的文档 例如 MongoDB

Graph stores 是被用来存储数据网络的信息，例如社交连接。Graph stores 包括 Neo4J 和 Girapha

key-value stores 是最简单的 NoSQL 数据库。每一个单独的项以 key 和 value 的形式存储在数据库里。典型的这种数据库有：Riak 和 BerkeleyDB。有些 key-value stores，例如 Redis，允许 value 有特定的数据类型

wide-column stores，例如 Cassandra 和 Hbase，优化了大型数据库的查询，将数据以列的形式存储而不是传统的以行形式存储

### 123.【简答题】Redis 部署：

(1) redis 分布式部署的三种方式是哪三种

主从模式

哨兵模式

集群 (Cluster) 模式

(2) 哨兵模式的特点是什么

哨兵模式是主从模式的升级版，他会在后台有一个监控，监控当前的主机，巡逻主机下面的从机，如果某一时刻主机挂掉了，那么他会通过一种投票的机制从从机之中选举一台作为新的主机，并且，其余的从机将会连接到这个新的主机上面。

## 5、集群管理

### 124.【选择题】【集群概念详解】下面哪个不属于 LVS 集群 3 层结构 (A)

- A Real-server
- B load balancer
- C server pool
- D shard storage

### 125.【选择题】【集群概念详解】下面关于 GFS 集群的叙述中，正确的是 ( C)

- A GFS 所有的元数据都存放在 chunk server 上
- B GFS 所有的元数据都是放在硬盘上的
- C 一个 GFS 集群由一个 master 和大量 chunk server 构成
- D chunk 数据块的默认大小为 100M

### 126.【简答题】【负载均衡性集群管理】lvs 有哪几种模式？

正确答案：DR 模式 NAT 模式 TUN 隧道模式 FULL NAT 模式

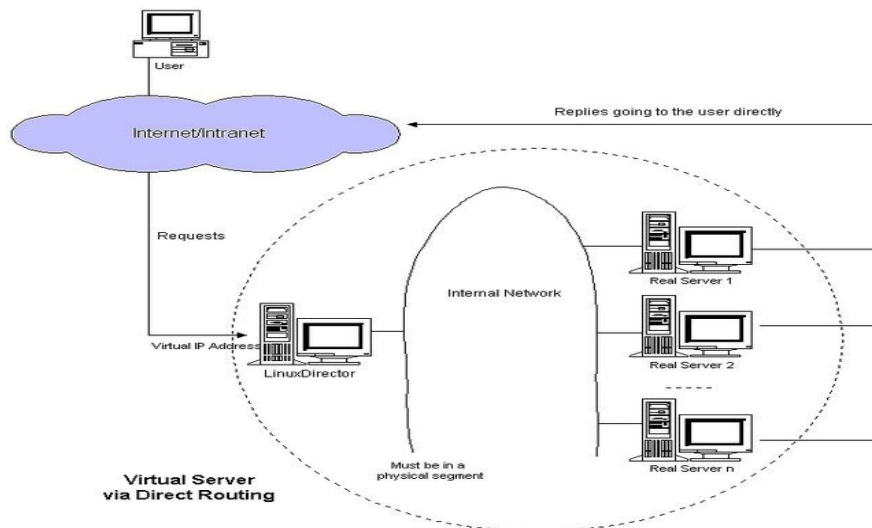
### 127.【简答题】【负载均衡性集群管理】请写出 lvs 原理？

LVS 通过工作于内核的 ipvs 模块来实现功能，其主要工作于 netfilter 的 INPUT 链上。而用户需要对 ipvs 进行操作配置则需要使用 ipvsadm 这个工具。ipvsadm 主要用于设置 lvs 模型、调度方式以及指定后端主机。

128. 【简答题】【LVS 工作模式与算法】 LVS 有哪几种工作模式？

DR : 直接路由模式 tun: 隧道模式 NAT: 路由转发模式 fullnat 模式

129. 【简答题】【LVS 工作模式与算法】 请说出 LVS 中 DR 模式的工作基本原理？



client 发送一个 pv 请求给 VIP; VIP 收到这请求后会跟 LVS 设置的 LB 算法选择一个 LB 比较合理的 realserver, 然后把此请求的 package 的 MAC 地址修改为 realserver 的 MAC 地址。

130. 【简答题】【LVS 实例】海量文件如何存储？

使用分布式存储, 如 mfs hadoop

131. 【选择题】【LVS 实例】下面哪个不是 LVS 的集群组成部分 (B)

- A lvs 内核模块
- B lvsServer
- C lvsadmin
- D 控制端软件

132. 【选择题】【keepalived 原理与实例】下面关于 keepalived 下面的说法错误的是 (B)

- A keepalived 的作用是检测 web 服务器的状态
- B keepalived 使用 layer4 得方式工作时, keepalived 会定期像服务器群中的服务器发送一个 icmp 的数据包
- C 如果有一台 web 服务器死机, 或工作出现故障, keepalived 将检测到, 并将有故障的 web 服务器从系统中剔除。

133. 【选择题】【keepalived 原理与实例】异步串口可以设为 (A)

- A 专线方式
- B 语言接入方式
- C PRL 接入方式
- D APR 接入方式

134. 【选择题】【heartbeat 原理与实例】关于 Heartbeat 下面说法错误的是 (D)

- A heartbeat 通过插件技术实现了集群的串口，多播，广播和组播通信
- B heartbeat 通过冗余通信通道和消息重传机制来保证通信的可靠性
- C heartbeat 核心的包括两个部分。心跳监测部分和资源接管部分
- D 心跳监测部分可以通过网络链路和串口进行，但不支持冗余链路

135. 【选择题】【heartbeat】openssl 是一个 (A)

- A 加密软件
- B 邮件系统
- C 数据库管理系统
- D 嵌入式脚本编程语言

136. 【简答题】【squid 代理】简述 Squid 的工作流程？

通过缓存的方式为用户提供 Web 访问加速对用户的 Web 访问进行过滤控制

137. 【选择题】【squid 代理】下面哪个不是 HAProxy 配置中包含的组件 (B)

- A frontend
- B daemon
- C listen
- D global

138. 【选择题】【|scs|详解】下列哪个不可作为负载均衡器 (D)

- A F5-BLG-LP
- B LVS
- C Nginx
- D Nagios

139. 【选择题】【|scs|详解】下面哪个不是 Session 共享及实现的方法（D）

- A 基于 Cookie 的 Session 共享
- B 基于数据库的 session 共享
- C 基于 Memcache 的 Session 共享
- D 基于 web 的 session 共享

140. 【简答题】【RHCS 集群部署与实例】请说出 RHCS 集群系统中三种集群构架

高可用性集群 负载均衡集群 存储集群。

141. 【简答题】【RHCS 集群部署与实例】请描述出 Tcp/ip 协议中主机与主机之间通信的三要素

ip 地址 子网掩码 ip 路由。

142. 【简答题】【MFS 分布式文件系统】请描述出 MFS 文件系统的组成

元数据服务器 元数据日志服务器 数据存储服务器 客户端。

143. 【简答题】【MFS 分布式文件系统】请描述出 MFS 特征

mfs 是一个分层的目录树结构

存储支持 POSIX 标准的文件属性（权限，最后访问和修改时间）

支持特殊的文件，如：块设备，字符设备，管道和套接字，链接文件（符号链接和硬链接）

支持基于 IP 地址和密码的方式访问文件系统

## 6、安全防护

144. 【简答题】【LoadRunner 负载测试工具、JMeter 压力测试工具】比较负载测试，容量测试和强度测试的区别？

负载测试：在一定的工作负荷下，系统的负荷及响应时间。

强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。

容量测试：容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测



试 还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。

**145.【选择题】【加密和安全】？ 下列关于计算机病毒的叙述中，正确的是（）？**

- A、计算机病毒只感染.exe 或.com 文件
- B、计算机病毒可通过读/写移动存储设备或通过 Internet 网络进行传播
- C、计算机病毒是通过电网进行传播的
- D、计算机病毒是由于程序中的逻辑错误造成的

答案：B。

解析：计算机病毒可以通过软盘、光盘或 Internet 网络进行传播。

病毒依附存储介质软盘、 硬盘等构成传染源。病毒传染的媒介由工作的环境来定。病毒激活是将病毒放在内存， 并设置触发条件，触发的条件是多样化的， 可以是时钟，系统的日期，用户标识符，也可以是系统一次通信等。条件成熟病毒就开始自我复制到传染对象中，进行各种破坏活动等。

病毒的传染是病毒性能的一个重要标志。在传染环节中，病毒复制一个自身副本到传染对象中去。

**146.【多选题】【运维】Apache 设置虚拟主机服务通常采用的方式有()?**

- A、 基于名字的虚拟主机
- B、 基于链接的主机
- C、 基于 IP 地址的虚拟主机
- D、 基于服务的虚拟主机

A， C， D。

Apache 虚拟主机的实现方式有 3 种。 基于 IP 的虚拟主机 基于端口的虚拟主机 基于域名的虚拟主机

**147.【简答题】DRBD 镜像数据的特点有那些？**

- 1.实时：当应用程序修改设备上的数据时，复制将连续进行。
- 2.透明：应用程序不需要知道数据存储多个主机上。
- 3.同步或异步：使用同步镜像，在所有主机上执行写操作后，将通知应用程序写完成。使用异步镜像，当本地完成写入时（通常在它们传播到其他主机之前），会通知应用程序写入完成。

**148.【简答题】DRBD 的工作原理。**

DRBD 是 linux 内核的存储层中的一个分布式存储系统，使用 DRBD 在两台 Linux 服务器之间共享块设备，共享文件系统和数据。

**149.【填空题】DRBD 为了管理的便捷性，目前通常会将会些配置文件分成多个部分，且都保存至/etc/drbd、d 目录中，主配置文件中仅使用\_\_\_\_\_指令将这些配置文件片断整合起来。**

include

**150.【简答题】zabbix 监控了多少客户端 客户端是怎么进行批量安装的根据实际公司台数回答。**

- 1、使用命令生成密钥。
- 2、将公钥发送到所有安装 zabbix 客户端的主机。
- 3、安装 ansible 软件，（修改配置文件，将 zabbix 客户机添加进组）。
- 4、创建一个安装 zabbix 客户端的剧本。
- 5、执行该剧本。
- 6、验证。

**151.【简答题】zabbix 是怎么微信报警的。**

- 1、首先，需要有一个微信企业号。（一个实名认证的[微信号]一个可以使用的[手机号]一个可以登录的[邮箱号]
- 2、下载并配置微信公众平台私有接口。
- 3、配置 Zabbix 告警，（增加示警媒介类型，添加用户报警媒介，添加报警动作）。

**152.【简答题】zabbix 自定义发现是怎么做的。**

- 1、首先需要在模板当中创建一个自动发现的规则，这个地方只需要一个名称和一个键值。
- 2、过滤器中间要添加你需要的用到的值宏。
- 3、然后要创建一个监控项原型，也是一个名称和一个键值。
- 4、然后需要去写一个这样的键值的收集。

自动发现实际上就是需要首先去获得需要监控的值，然后将这个值作为一个新的参数传递到另外一个收集数据的 item 里面去。

**153.【简答题】nagios/zabbix 优缺点对比。**

nagios 核心功能是监控报警，是一个轻量化的监控系统。

如果需要图标显示，需要增加图标显示插件（如 pnp4nagios）；

如果需要存入数据库，需要相应的插件（NDOUtils）；

如果需要分布式，需要 NSCA。。。

优点：监控很多协议，邮件和短信通知，服务抖动检测

缺点：只能在终端配置，基于文件的配置方式，不方便扩展，易读性差，管理耗时 zabbix 比较大而全，已经把图标、数据库、分布式等等添加进系统，可以直接配置使用。

优点：图表显示、数据库存储、分布式等功能已经整合，页面配置主动发现主机，内置插件比较全面，监控模块化，邮件短信报警，功能全面，图表显示比较细腻

缺点：安装稍微复杂，WEB 操作方便，没有服务抖动检测，配置较复杂两个系统各有自己的优缺点，还需要深入使用去发现。

#### 154.【简答题】Nagios 的主要功能特点。

1. 监视网络服务 (SMTP, POP3, HTTP, NNTP, PING 等)
2. 监视主机资源 (进程, 磁盘等)
3. 简单的插件设计可以轻松扩展 Nagios 的监视功能
4. 服务等监视的并发处理
5. 错误通知功能 (通过 email, pager, 或其他用户自定义方法)
6. 可指定自定义的事件处理控制器
7. 可选的基于浏览器的 WEB 界面以方便系统管理人员查看网络状态, 各种系统问题, 以及日志等等
8. 可以通过手机查看系统监控信息

#### 155.【填空题】Nagios 是一个监视\_\_\_\_\_和\_\_\_\_\_的监视系统。Nagios 能监视所指定的本地或远程主机以及服务，同时提供异常通知功能等

系统运行状态 网络信息

#### 156.【选择题】哪个不属于 iptables 的表（D）？

- A、 filter
- B、 nat
- C、 mangle
- D、 INPUT

#### 157.【选择题】rule permit ip source 210.78.1.1 0.0.255.255 destination 202.38.5.2 0.0.0.0 的含义是（D）。

- A、 允许主机 210.78.1.1 访问主机 202.38.5.2

- B、 允许 210.78.0.0 的网络访问 202.38.0.0 的网络
- C、 允许主机 202.38.5.2 访问网络 210.78.0.0
- D、 允许 210.78.0.0 的网络访问主机 202.38.5.2

158.【选择题】以下对防火墙的描述正确的是：(C)

- A、 完全阻隔了网络
- B、 能在物理层隔绝网络
- C、 仅允许合法的通讯
- D、 无法阻隔黑客的侵入

159.【多选题】在防火墙上允许 tcp 和 和 udp 端口 21 、 23 、 25 访问内网，下列那张协议包可以进来 (ACD)。

- A、 SMTP #25 简单邮件传输协议
- B、 STP #
- C、 FTP #21 20
- D、 Telnet #23
- E、 HTTP #80
- F、 POP3 #110

160.【填空题】\_\_\_\_\_ 是设置在被保护网络和外部网络之间的一道 屏障，以防止破坏性侵入

防火墙

161.【填空题】在 CentOS7 下，我想关闭掉防火墙，应该用命令\_\_\_\_\_来关闭掉。如果以后开机都不想它启动起来，执行\_\_\_\_\_命令

systemctl stop firewalld systemctl disable firewalld、

162.【填空题】在 CentOS7 配置 ip 转发需要在\_\_\_\_\_ 里加入\_\_\_\_\_ 执行\_\_\_\_\_命令后生效

/etc/sysctl.conf net.ipv4.ip\_forward=1 sysctl -p

**163.【简答题】**防火期墙策略，开放服务器 80 端口，禁止来自 10.0.0.188 的地址访问服务器 80 端口的请求。

```
iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT  
iptables -t filter -I INPUT -p tcp --dport 80 -s 10.0.0.188 -j DROP  
或 iptables -A INPUT -p tcp --dport 80 ! -s 10.0.0.188 -j ACCEPT
```

**164.【简答题】**防火墙策略，实现把访问 10.0.0.3:80 的请求转到 172.16.1.17:8080 上。

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 172.16.1.17:8080
```

**165.【简答题】**防火墙策略配置说明。阐述出 10.10.10.1 访问 192.168.1.1 所有端口策略需要的配置过程

```
iptables -A INPUT -p all -s 10.10.10.1 -d 192.168.1.1 --dport 1:65535 -j ACCEPT  
iptables -t filter -I INPUT -p tcp --dport 1-65535 -s 10.10.10.1 -d 192.168.1.1 -j ACCEPT
```

**166.【简答题】**请写出查看 iptables 当前所有规则的命令。

```
iptables -nL （默认查看 filter 表）  
iptables -nL -t nat (插卡 NAT 表的规则)
```

**167.【简答题】**禁止来自 10.0.0.188 ip 地址访问 80 端口的请求

```
iptables -A INPUT -p tcp -s 10.0.0.188 --dport 80 -j DROP
```

**168.【简答题】**如何在命令行执行的 iptables 规则永久生效？

```
方法 1：  
/etc/init.d/iptables save  
方法 2：  
iptables-save >/etc/sysconfig/iptables
```

**169.【简答题】**实现把访问 10.0.0.3:80 的请求转到 172.16.1.17:80 在 10.0.0.3 主机上做 IP 映射，具体命令：

```
iptables -t nat -A PREROUTING -d 10.0.0.3 -p tcp --dport 80 -j DNAT --to-destination 172.16.1.17:80
```

## 170.【简答题】详述 iptables 工作流程以及规则过滤顺序？

iptables 是采用数据包过滤机制工作的，所以他会请求的数据包的包头数据进行分析，并根据我们预先设定的规则来匹配进行相关操作。

当防火墙收到数据包时：

- 1.防火墙是一层一层的过滤的，规则顺序从上到下，从前到后进行归路
- 2.如果匹配上规则（ACCEPT，DROP）就不会再向下匹配了。
- 3.如果匹配规则没有明确表明是阻止或者是通过这个数据包，也就是没有匹配上规则，就会继续向下执行下一跳规则。
- 4.如果以上所有规则都不能匹配上，最后会执行默认规则。

## 171.【简答题】iptables 有几个表以及每个表有几个链？

表和链（四表五链）

- 1.filter（过滤）进行包过滤处理的一张表
- 2.nat（映射）对数据地址信息进行转换/数据包端口信息进行转换实现内网用户访问外网实现外网用户访问内网
- 3.mangle（不常用）对数据包信息进行标记
- 4.raw（不常用）

## 172.【简答题】selinux 是怎么工作的。

它通过 operation (操作)的主和宾打一个“标签”，在一个“类型”里只能由特定的目录或用户执行。这样可以限定一个进程的执行范围只能在一个“沙箱”（sandbox）内了（最小权限法则）

###operation: 读，写，执行，等。其实是一个主谓宾的结构 Subject Operation Object

###最小权限法则：例如定义 Subject httpd 的目录/var/www 为 public\_content\_t 类型，这样，httpd 只能访问有这个特定类型的文件或目录，这些文件就叫沙箱“sendbox”

使用 selinx，必须精心设计一套访问法则

给不同的进程打上不同的“域”，给不同的目录打上不同的“类型”“通过定义“类型”和“域”的对应规则，来实现。selinx 需要域和标签的对应关系，一对一对应。但是系统上有上千个进程，实现起来很麻烦，所以一般情况下，并不是用 selinx 作为安全防范

173.【填空题】selinux 描述的是\_\_\_\_\_和\_\_\_\_\_间的访问控制的安全上下文。一个进程或者一个文件，目录相当于有了独有的一个\_\_\_\_\_，其他的文件，进程要访问具有安全上下文的文件或者进程，需要有安全上下文标识的标签，才能互通有无。

主体（进程）    客体（文件，socket， 进程间通信的管道）    安全域

174.【填空题】Security-Enhanced Linux (SELinux)是强制访问控制\_\_\_\_\_的一种实现——在自主访问控制(DAC)检查之后，再次检查允许的操作。SELinux 通过定义的策略，强限制 Linux 系统中对文件的操作、进程的权限。

MAC 在 Linux 内核中

175.【简答题】使用 selinux 相关命令查看和修改状态

`selinux=enforcing`

#此项定义 selinux 状态

#enforcing-是强制模式系统，它受 selinux 保护。就是违反了策略你就无法继续操作下去。

#permissive-是提示模式系统不会受到 selinux 保护，只是收到警告信息。permissive 就是 selinux 有效，但是即使你违反了策略的话它让你继续操作，但是把你违反的内容记录下来（警告信息）

#disabled-禁用 selinux

`selinuxtype=targeted`

#此项定义 selinux 使用哪个策略模块保护系统。targeted 只对 Apache, sendmail, bind, postgresql, nfs, cifs 等网络服务保护。

以上策略配置都放置在/etc/selinux 目录中，目录和策略名称相同。

使用 selinux 相关命令查看和修改状态：（属于立即生效但临时性的）

176.【简答题】查看 selinux 工作状态命令

`getenforce` 查看 selinux 状态 `setenforce` 设定 selinux 运行状态，1 开启（Enforce），0 关闭（Permissive）

177.【选择题】哪个 vSphere 组件可用于为每台虚拟机创建实时卷影实例，以便在虚拟机出现故障时可由卷影实例取代虚拟机（B）？

A、High Availability

B、Fault Tolerance

- C、Data Protection
- D、Distributed Resources Scheduler

178.【选择题】以下哪项是对虚拟机的最佳描述？（B）

- A、执行虚拟化软件测试程序的物理机
- B、通过软件实施的计算机，可以像物理机一样执行程序
- C、一种旨在提供网络故障切换和故障恢复功能的计算机工具
- D、一种软件计算机，其中封装了物理硬件

179.【选择题】阅读以下说法并选择与其对应的虚拟机属性：“如果一个虚拟机因为错误而崩溃，同一主机上的其他虚拟机不会受到影响”（A）

- A、隔离性
- B、兼容性
- C、独立于硬件
- D、统一性

## 7、虚拟化与云计算

180.【选择题】云计算的一大特征是（B），没有高效的网络云计算就什么都不是，就不能提供很好的使用体验

- A、按需自助效劳
- B、无处不在的网络接入
- C、资源池化
- D、快速弹性伸缩

181.【选择题】要使端口组到达其他 VLAN 上的端口组，必须将 VLAN ID 设置为（B）

- A、80
- B、4095
- C、8080
- D、3306

182.【选择题】对于公有边缘节点，通常以（A）的形式部署于（A）

- A、小型数据中心，地市及以下的自有机房



- B、大型数据中心，公有云机房
- C、大型数据中心，私有云机房
- D、大型数据中心，地市及以下的自有机房

## 7. 虚拟化和云计算

### 183. 【简答题】云计算的不同层包括哪些？

- 1) SaaS：软件即服务，它让用户可以直接访问云应用程序，不必在系统上安装任何东西。
- 2) IaaS：基础设施即服务，它从硬件（比如内存和处理器速度等）等层面提供了基础设施。
- 3) PaaS：平台即服务，它为开发人员提供了云应用程序平台。

### 184. 【简答题】计算节点上的集成桥（br-int）的作用是什么？

集成桥（br-int）对来自和运行在计算节点上的实例的流量执行 VLAN 标记和取消标记。

数据包从实例的 n/w 接口发出使用虚拟接口 qvo 通过 Linux 桥（qbr）。qvb 接口是用来连接 Linux 桥的，qvo 接口是用来连接集成桥的。集成桥上的 qvo 端口有一个内部 VLAN 标签，这个标签是用于当数据包到达集成桥的时候贴到数据包头部的

### 185. 【简答题】如何查看控制节点和计算节点的 OVS 桥配置？

控制节点和计算节点的 OVS 桥配置可使用以下命令来查看：

```
~]# ovs-vsctl show
```

### 186. 【简答题】如何使用命令行在 OpenStack 中向存在的安全组 SG（安全组）中添加新规则？

可以使用 neutron 命令向 OpenStack 已存在的安全组中添加新规则：

```
~# neutron security-group-rule-create --protocol <tcp or udp> --port-range-min <port-number> --port-range-max <port-number> --direction <ingress or egress> --remote-ip-prefix <IP-address-or-range> Security-Group-Name
```

### 187. 【简单题】假如计算节点上为数据流量配置了一些 DPDK 端口，你如何检查 DPDK 端口的状态呢？

因为我们使用 openvSwitch (OVS) 来配置 DPDK 端口，因此可以使用如下命令来检查端口的状态：

```
root@compute-0-15:~# ovs-appctl bond/show | grep dpdk
```

```
active slave mac: 90:38:09:ac:7a:99(dpdk0)
slave dpdk0: enabled
slave dpdk1: enabled
root@compute-0-15:~#
root@compute-0-15:~# dpdk-devbind.py --status
```

### 188.【简答题】如何使用命令行查看 OpenStack 实例的控制台日志？

使用如下命令可查看实例的控制台日志。

首先获取实例的 ID，然后使用如下命令：

```
openstack console log show {Instance-id}
```

### 189.【简答题】计算节点上虚拟机的默认地址是什么？

虚拟机存储在计算节点的 `/var/lib/nova/instances`。

### 190.【简答题】如何在特定可用区域中或在计算主机上配置虚拟机？

假设我们想在 `compute-02` 中的可用区 `NonProduction` 上配置虚拟机，可以使用如下命令：

```
openstack server create --flavor m1.tiny --image cirros --nic net-id=e0be93b8-728b-4d4d-a272-7d672b2560a6 --
security-group NonProd_SG --key-name linuxtec --availability-zone NonProduction:compute-02 nonprod_testvm
```

### 191.【简答题】说一下如何使用命令行启动一个虚拟机？

我们可以使用如下 OpenStack 命令来启动一个新的虚拟机：

```
openstack server create --flavor {flavor-name} --image {Image-Name-Or-Image-ID} --nic net-id={Network-ID} --
security-group {Security_Group_ID} --key-name {Keypair-Name} <VM_Name>
```

### 192.【简答题】什么服务通常在计算节点上运行？

以下服务通常在计算节点运行：

Nova compute

网络服务，比如 OVS

### 193.【简答题】什么服务通常在控制节点上运行？

以下服务通常在控制节点上运行：

认证服务 (KeyStone)

镜像服务 (Glance)

Nova 服务比如 Nova API、Nova Scheduler 和 Nova DB

块存储和对象存储服务

Ceilometer 服务

MariaDB / MySQL 和 RabbitMQ 服务

网络(Neutron)和网络代理的管理服务

编排服务 (Heat)

## 194.【简答题】说一下 OpenStack 及其主要组件？

OpenStack 是一系列开源软件，这些软件组成了一个云供给软件，也就是 OpenStack，意即开源软件或项目栈。

下面是 OpenStack 的主要关键组件：

Nova - 用于在计算级别管理虚拟机，并在计算或管理程序级别执行其他计算任务。

Neutron - 为虚拟机、计算和控制节点提供网络功能。

Keystone - 为所有云用户和 OpenStack 云服务提供身份认证服务。换句话说，我们可以说 Keystone 是一个提供给云用户和云服务访问权限的方法。

Horizon - 用于提供图形用户界面。使用图形化管理界面可以很轻松地完成各种日常操作任务。

Cinder - 用于提供块存储功能。通常来说 OpenStack 的 Cinder 中集成了 Chef 和 ScaleIO 来共同为计算和控制节点提供块存储服务。

Swift - 用于提供对象存储功能。通常来说，Glance 管理的镜像存储在对象存储空间。像 ScaleIO 这样的外部存储也可以提供对象存储，可以很容易的集成 Glance 服务。

Glance - 用于提供镜像服务。使用 Glance 的管理平台来上传和下载云镜像。

Heat - 用于提供编排服务或功能。使用 Heat 管理平台可以轻松地将虚拟机作为堆栈，并且根据需要可以将虚拟机扩展或收缩。

Ceilometer - 用于提供计量与监控功能。

## 195.【简答题】如何使用命令行来获取可使用的浮动 IP 列表？

```
~]# openstack ip floating list | grep None | head -10
```

## 196.【简答题】如何在特定可用区域中或在计算主机上配置虚拟机？

假设我们想在 compute-02 中的可用区 NonProduction 上配置虚拟机，可以使用如下命令：

```
~]# openstack server create --flavor m1.tiny --image cirros --nic net-id=e0be93b8-728b-4d4d-a272-7d672b2560a6  
--security-group NonProd_SG --key-name linuxtec --availability-zone NonProduction:compute-02 nonprod_testvm
```

### 197. 【简答题】如何在特定计算节点上获取配置的虚拟机列表？

假设我们想要获取在 `compute-0-19` 中配置的虚拟机列表，可以使用如下命令：

命令格式: `openstack server list - all-projects - long -c Name -c Host | grep -I {Compute-Node-Name}`：

```
~# openstack server list - all-projects - long -c Name -c Host | grep -I compute-0-19
```

### 198. 【简答题】如何使用命令行查看 OpenStack 实例的控制台日志？

使用如下命令可查看实例的控制台日志。

首先获取实例的 ID，然后使用如下命令：

```
~# openstack console log show {Instance-id}
```

### 199. 【简答题】如何获取 OpenStack 实例的控制台的 URL 地址？

可以使用以下 OpenStack 命令从命令行检索实例的控制台 URL 地址：

```
~# openstack console url show {Instance-id}
```

### 200. 【简答题】如何使用命令行创建可启动的 cinder / block 存储卷？

假设创建一个 8GB 可启动存储卷，可参考如下步骤：

使用如下命令获取镜像列表

```
~# openstack image list | grep -I cirros  
| 89254d46-a54b-4bc8-8e4d-658287c7ee92 | cirros | active |
```

使用 `cirros` 镜像创建 8GB 的可启动存储卷

```
~# cinder create - image-id 89254d46-a54b-4bc8-8e4d-658287c7ee92 - display-name cirros-bootable-vol 8
```

### 201. 【简答题】如何列出所有在你的 OpenStack 中创建的项目或用户？

```
~# openstack project list - long
```

### 202. 【简答题】如何显示 OpenStack 服务端点列表？

OpenStack 服务端点被分为 3 类：

公共端点

内部端点

管理端点

使用如下 OpenStack 命令来查看各种 OpenStack 服务端点：

```
~# openstack catalog list
```

可通过以下命令来显示特定服务端点（比如说 keystone）列表：

```
~# openstack catalog show keystone
```

### 203. 【简答题】在控制节点上你应该按照什么步骤来重启 nova 服务？

应该按照如下步骤来重启 OpenStack 控制节点的 nova 服务：

```
Service nova-api restart
Service nova-cert restart
Service nova-conductor restart
Service nova-consoleauth restart
Service nova-scheduler restart
```

### 204. 【简答题】假如计算节点上为数据流量配置了一些 DPDK 端口，你如何检查 DPDK 端口的状态呢？

因为我们使用 openvSwitch (OVS) 来配置 DPDK 端口，因此可以使用如下命令来检查端口的状态：

```
root@compute-0-15:~# ovs-appctl bond/show | grep dpdk
Active slave mac: 90:38:09:ac:7a:99(dpdk0)
Slave dpdk0: enabled
Slave dpdk1: enabled
root@compute-0-15:~#
root@compute-0-15:~# dpdk-devbind.py - status
```

### 205. 【简答题】如何使用命令行在 OpenStack 中向存在的安全组 SG（安全组）中添加新规则？

可以使用 neutron 命令向 OpenStack 已存在的安全组中添加新规则：

```
~# neutron security-group-rule-create - protocol <tcp or udp> --port-range-min <port-number> --port-range-max <port-number> --direction <ingress or egress> --remote-ip-prefix <IP-address-or-range> Security-Group-Name
```

## 206.【简答题】如何查看控制节点和计算节点的 OVS 桥配置？

```
~|# ovs-vsctl show
```

## 207.【简答题】计算节点上的集成桥（br-int）的作用是什么？

集成桥（br-int）对来自和运行在计算节点上的实例的流量执行 VLAN 标记和取消标记。

数据包从实例的 n/w 接口发出使用虚拟接口 qvo 通过 Linux 桥（qbr）。Qvb 接口是用来连接 Linux 桥的，qvo 接口是用来连接集成桥的。集成桥上的 qvo 端口有一个内部 VLAN 标签，这个标签是用于当数据包到达集成桥的时候贴到数据包头部的。

## 208.【简答题】隧道桥（br-tun）在计算节点上的作用是什么？

隧道桥（br-tun）根据 OpenFlow 规则将 VLAN 标记的流量从集成网桥转换为隧道 ID。

隧道桥允许不同网络的实例彼此进行通信。隧道有利于封装在非安全网络上传输的流量，它支持两层网络，即 GRE 和 VXLAN。

## 209.【简答题】外部 OVS 桥（br-ex）的作用是什么？

顾名思义，此网桥转发来往网络的流量，以允许外部访问实例。Br-ex 连接物理接口比如 eth2，这样用户网络的浮动 IP 数据从物理网络接收并路由到用户网络端口。

## 210.【简答题】什么是 Neutron 代理？如何显示所有 Neutron 代理？

OpenStack Neutron 服务器充当中心控制器，实际网络配置是在计算节点或者网络节点上执行的。Neutron 代理是计算节点或者网络节点上进行配置更新的软件实体。Neutron 代理通过 Neutron 服务和消息队列来和中心 Neutron 服务通信。

可通过如下命令查看 Neutron 代理列表：

```
~# openstack network agent list -c 'Agent type' -c Host -c Alive -c State
```

## 211.【简答题】讲讲 Opnstack 启动虚拟机实例后，虚拟机的状态是怎么变化的？

- （1）管理员发出创建虚拟机的命令，决定从镜像文件或是快照文件进行启动。
- （2）当创建后，虚拟机实例进入 Build 状态，任务状态是 Spawning 孵化。
- （3）期间，将会从控制节点上的 Glance 组件把相应的镜像文件从中下载到 Nova 计算节点，并进行一些虚拟机的初始参数配置，如内存、CPU、磁盘空间。

(4) 当一切正常后，虚拟机将会进入 Active 状态，此后，用户便可以使用云主机了。创建实例的时间一般由镜像文件的大小、网络传输带宽、以及创建的 Hypervisor 磁盘性能大小。

(5) 虚拟机创建完毕，可以经过 Horizon 的 web 界面进行管理，也可以基于 Python Nova client 的命令行管理。

## 212. 【简答题】概述创建虚拟机的流程。(这个问题，非常值得思考，很容易被问到!)

(1) Horizon 通过 Keystone 获取 nova-Compute 组件的访问地址 (即 URL)，并获取令牌 token。

(2) Horizon 携带授权令牌，发送创建虚拟机指令。

(3) Nova-compute 组件通过 glance-api 下载虚拟机镜像，glance 镜像中有缓存机制，通常把缓存文件放入名为 \_base 的目录，如果 \_base 缓存没有镜像文件

那么就会从 glance 下载镜像到 base 缓存，然后再从 base 缓存通过 TCP/IP 网络复制到计算节点云主机实例下的镜像目录里。

(4) glance 检索后端镜像，glance 后端存储不一定要使用 swift，只要是存放镜像文件的系统都可以。

(5) 获取网络信息，决定虚拟机的网络模式以及建立网络连接。

nova-compute 发送启动虚拟机指令，至此经过虚拟机服务任务状态的变化，正常过后创建即可完成。

## 8、Docker

### 213. 【多选题】【运维】Apache 设置虚拟主机服务通常采用的方式有 ( ) ?

- A、 基于名字的虚拟主机
- B、 基于链接的主机
- C、 基于 IP 地址的虚拟主机
- D、 基于服务的虚拟主机

A, C, D。

Apache 虚拟主机的实现方式有 3 种。 基于 IP 的虚拟主机 基于端口的虚拟主机 基于域名的虚拟主机

### 214. 【简答题】【基础环境搭建】什么是 Docker 容器?

Docker 容器包括应用程序及所有的依赖项，作为操作系统的独立进程运行

### 215. 【简答题】【基本命令操作】dockerFile 中最常见的指令是什么?

FROM (指定基础镜像)

LABEL (功能为镜像指定标签)

RUN（运行指定命令）

CMD（容器启动时要运行的命令）

## 216.【简答题】【基本命令操作】解释一下 dockerfile 的 ONBUILD 指令？

当镜像用作另一个镜像构建的基础时，ONBUILD 指令像镜像添加将在稍后执行的触发指令。如果要构建将用作构建其他镜像的基础的镜像（例如，可以使用特定于用户的配置自定义的应用程序构建环境或守护程序）

## 217.【简答题】【基本命令操作】Docker 的常用命令？

docker pull ：拉去或更新指定的镜像

docker push ：将镜像推送到远程仓库

docker rm ：删除容器

docker rmi ：删除镜像

docker images ：列出所有镜像

docker ps ：列出所有容器

## 218.【简答题】【基本命令操作】请依次列举 docker 镜像操作

1、搜索镜像：docker search 镜像名

2、下载镜像：docker pull 镜像名

3、查看已下载的镜像：docker images

4、获取镜像的详细信息：docker inspect 镜像 ID 号

5、添加镜像标签：docker tag nginx:latest nginx:web （类似于别名）

6、删除镜像：docker rmi 仓库名:TAG 标签

7、导出镜像：docker save -o 文件名 镜像名

8、载入镜像：docker load < 镜像名

9、上传镜像到公有仓库（必须要有 docker 官方账号，且创建公有仓库）：

docker login

Username: docker

password:

Email: xxx@xxx.com （输入账号、密码）

上传：

docker push 镜像名:标签名(一般公司使用私有仓库)



## 219.【简答题】【基本命令操作】请依次列举容器操作

- 1、创建容器（创建前必须有该镜像）：`docker create -it 镜像名 运行程序`
- 2、查看容器运行状态：`docker ps -a`
- 3、开启容器，关闭容器：`docker start 容器 ID 号`；`docker stop 容器 ID 号`
- 4、`docker run` 命令：`docker run 镜像名 /bin/bash -c ls /`
- 5、进入容器与退出容器：`docker exec -it 容器 ID /bin/bash`
- 6、导出、导入容器：`docker export 容器 ID > 定义容器名`(注意：导入容器只会生成镜像)
- 7、删除容器：`docker rm 容器 ID 号`
- 8、批量删除容器:第一种：`docker rm $(docker ps -aq)`

## 220.【简答题】【知识面】docker 网络模式有几种，哪几种？

有 4 种，分别是：

**HOST 模式：**host 模式容器不会虚拟自己的网卡，配置自己的 ip，而是使用宿主机的 ip 和端口，如果启动容器的时候使用了 host 模式，这个容器将不会获得一个独立的网络名称空间，而是和宿主机使用同一个网络名称空间，容器将不会虚拟出自己的网卡以及配置等，但是容器的其他方面，比如文件，进程等还是和宿主机隔离的，host 模式最大的优势就是网络性能好，但是 docker host 上已经使用的端口就不能再使用了，所以网络的隔离性并不是很好

**Container：**这个模式指定新创建的容器和已存在的一个容器共享同一个网络名称空间，而不是和宿主机共享，新创建的容器不会创建自己的网卡和 IP，而是和一个指定的容器公用，同样两个容器除了网络方面，其他的文件或者是进程还是隔离的，且两个容器之间的进程可以通过 lo 网卡设备通信

**None：**这种模式下容器只有 lo 回环网口，没有其他的网卡 none 模式可以再容器刚刚创建的时候通过 `network=none` 参数指定，但是这种类型的网络无法访问外网，这也叫就体现了他的有点，封闭也就体现了可以给容器提供一个很好的安全。

**Bridge：**这个模式会为每一个容器分配 IP、端口等，并将容器连接到一个 docker 虚拟网桥，通过 docker0 网桥以及 iptables 的 net 表配置和宿主机通信，当 docker 进程启动时，会在主机上创建一个名为 docker0 的虚拟网桥，此主机上启动的 docker 容器会连接到这个虚拟网桥上，虚拟网桥的工作方式和物理交换机类似，这样主机的所有容器就会通过交换机连接在一个二层网络中，当 docker0 子网中分配一个 IP 和容器中，并设置 docker0 的 IP 地址为容器的默认网关，在主机中创建一对虚拟网卡设备，docker 将设备的一端放在新创建的容器中，并命名为容器的网卡，另一端放在主机中，以 vethxx 这种类似的名字命名，并将这个网络设备加入到 docker 网桥中，可以通过 `brctl show` 命令查看，bridge 模式是 docker 的默认网络模式，不屑 `-net` 参数就是网桥模式，使用 `docker run -p` 时，docker 实际是在防火墙做了 DNAT 规则，实现端口转发功能可以使用 `iptables -t nat -vnL` 查看。

## 221.【简答题】【知识面】docker 网络基本命令有哪些？

1. 显示 docker 网络：`docker network ls`

2.查看其中一个网络: `docker network inspect` (网络名)

3.创建自定的网络: `docker network create -d bridge my-bridge` ; `docker network create my-bridge`

创建一个网络名为 `my_net` 且 `driver` 为 `bridge` 的网络, 默认就是创建 `driver` 为 `birdge` 模式, 创建自定义的网络后, 将容器连接上两个容器就可以互相通信了。

4. 创建连接自定义网络上的容器, 默认是连接上 `docker` 启动时默认启动的 `bridge` 上

```
docker run -d --name test1 --network my-bridge busybox /bin/sh -c "while true; do sleep 3600; done"
```

```
docker run -d --name test2 --network my-bridge busybox /bin/sh -c "while true; do sleep 3600; done"
```

注: `--network my-bridge`

验证 `test1` 是否能够连接上 `test2`

```
docker exec -it test1 ping test2
```

5.查询网桥信息: `brctl show`

6.端口绑定: 作用: 当在创建一个 `nginx` 容器时, 如果不做端口映射在宿主主机上就不能通过 `curl 127.0.0.1` 访问

## 222. 【简答题】【知识面】 Docker Hub 有哪些功能?

Docker Hub 具有以下主要功能。

(1) 镜像仓库, 用于上传和下载容器镜像;

(2) 团队和组织, 管理对容器镜像的私有仓库的访问;

(3) 提供官方镜像;

(4) 提供发布者镜像;

(5) 镜像构建, 支持源代码托管平台自动构建容器镜像并将它们上传到 Docker HuB;

(6) 提供 Webhooks. 这是一个自动化构建特性, 在一个镜像上传成功后, Webhooks 会触发操作, 将 Docker Hub 与其他服务进行整合。

## 223. 【简答题】【知识面】 Docker 构建镜像有哪几种方法?

Docker 提供了两种构建镜像的方法, 一是将现有容器转化为镜像, 二是通过 `Dockerfile` 构建镜像。

## 224. 【简答题】【知识面】 COPY 指令应遵守哪些复制规则?

(1) 源路径必须位于构建上下文中。

(2) 如果源是目录, 则复制目录的整个内容, 包括文件系统元数据。

(3) 如果源是任何其他类型的文件, 则它会与其元数据被分别复制。

(4) 如果直接指定多个源, 或者源中使用了通配符, 则目的路径必须是目录, 并且必须以斜杠(/)结尾。

(5) 如果目的路径不以斜杠结尾, 则它将被视为常规文件, 源内容将被写入目录路径。(6) 如果目的路径不存在, 则会与其路径中所有缺少的目录起被创建。

## 225.【简答题】【知识面】RUN、CMD 和 ENTRYPOINT 指令有何不同？

RUN 指令执行命令并创建新的镜像层经常用于安装应用程序和软件包，RUN 先于 CMD 或 ENTRYPOINT 指令在构建镜像时执行，并被固化有所生成的镜像中。

CMD 和 ENTRYPOINT 指令在每次启动容器时才执行，两者的区别在于 CMD 指令会被 docker run 命令所覆盖。两个指令起使用时，ENTRYPOINT 指令作为可执行文件，而 CMD 指令则为 ENTRYPOINT 指令提供默认参数。

## 226.【简答题】【知识面】用户自定义桥接网络和默认桥接网络有哪些不同？

主要不同点体现在以下几个方面：

(1) 用户自定义桥接网络能提供容器化应用程序之间更好的隔离和互操作性，连接到同一个用户自定义桥接网络的容器会自动互相暴露所有端口，但不会将端口暴露到外部。而默认桥接网络上运行应用栈，Docker 主机需要通过其他方式来限制对端口的访问。

(2) 用户自定义桥接网络提供容器之间自动 DNS 解析功能，可以通过名称或别名互相访问。而默认桥接网络上的容器只能通过 IP 地址互相访问。

(3) 容器可以在运行时与用户自定义网络连接和断开。要断开与默认桥接网络的连接，需要停止容器并使用不同的网络选项重新创建该容器。

(4) 每个用户可通过自定义网络创建一个可配置的网桥，而默认桥接网络不行。

(5) 用户自定义网络中所连接的容器无法直接共享环境变量，而默认桥接网络中所连接的容器可以共享环境变量。用户自定义网络有更好的方式来实现共享环境变量。

## 227.【简答题】【知识面】描述 Docker 容器的生命周期？

Docker 容器经历以下阶段：

创建容器、运行容器、暂停容器（可选）、取消暂停容器（可选）、启动容器、停止容器、重启容器、杀死容器、销毁容器

## 228.【简答题】【知识面】为什么生产环境不建议使用默认桥接网络？

默认桥接网络相当于 VMware 虚拟机网络连接的 NAT 模式，容器拥有独立的网络名称空间和隔离的网络栈，不支持基于名称的服务发现和用户指定的 IP 地址，所连接的容器只能通过 IP 地址互相访问，除非创建容器时使用 -link 选项建立容器连接。由于功能受限，默认桥接网络只适合一些演示或实验场合，不建议用于生产

## 229.【简答题】【知识面】容器之间有哪些通信解决方案？

容器之间的通信方案列举如下：

(1) bridge 模式让同一个 Docker 网络上的所有容器在所有端口上都可以相互连接。默认桥接网络所连接的容器只能通过 IP 地址互相访问，除非创建容器时使用--link 选项建立容器连接。

(2) host 模式让所有容器都位于同一个主机网络空间中，并共用主机的 IP 地址栈，在该主机上的所有容器都可通过主机的网络接口相互通信。

(3)在用户自定义桥接网络中，容器之间可以通过名称或别名互相访问。

(4)容器通过端口映射对外部提供连接。

(5)container 模式让容器共用个 IP 网络，两个容器之间可通过回环网络接口相互通

(6)容器之间使用--link 选项建立传统的容器互联。

(7)容器之间通过挂载主机目录来实现相互之间的数据通信。

### 230.【简答题】【知识面】 Docker 的容器支持哪几种网络模式？

Docker 的容器支持 5 种网络模式，分别是 bridge 模式、host 模式、container 模式、none 模式和自定义网络。

### 231.【简答题】【知识面】 选择 Docker 网络驱动应遵守哪些原则？

选择 Docker 网络驱动应遵守以下原则：

(1)用户自定义桥接网络最适合用于同一个 Docker 主机上运行的多个容器之间需要通信的场景。(2)host 网络最适用于当网络栈不能与 Docker 主机隔离，而容器的其他方面需要被隔离的场景。(3)overlay 网络适用于不同 Docker 主机上运行的容器需要通信的场景，或者多个应用程序通过 Swarm 集群服务起工作的场景。(4)macvlan 网络适用于从虚拟机迁移过来的场景，或者容器需要像网络上的物理机一样，拥有独立 MAC 地址的场景。(5)第三方网络插件适用于将 Docker 与专用网络栈进行集成的场景。

### 232.【简答题】【知识面】 如何控制容器占用系统资源(cpu、内存)的份额？

在使用 docker create 命令创建容器或使用 docker run 创建并运行的时候、可以使用 -c|-cpu-shares[=0]参数来调整同期使用 cpu 的权重、使用-m|-memory 参数来调整容器使用内存的大小

### 233.【简答题】【知识面】 仓库(Repository)、则注册索引(Index)有何关系？

首先、仓库是存放一组关联镜像的集合、比如同一个应用的不同版本的镜像、注册服务器是存放实际镜像的地方、注册索引则负责维护用户的账号、权限、搜索、标签等管理。注册服务器利用注册索引来实现认证等管理。

## 9、kuberneres

### 234.【简答题】什么是 kubernetes 资源清单管理

Kubernetes 中所有的内容都抽象为资源，资源实例化（被调用、被执行了）之后，叫做对象。

大致地可以总结为：对象描述了什么容器化应用在运行（以及在哪个 Node 上）；可以被应用使用的资源；关于应用如何表现的策略，比如重启策略、升级策略，以及容错策略。

Kubernetes 对象是“目标性记录”：一旦创建对象，Kubernetes 系统将持续工作以确保对象存在。通过创建对象，可以有效地告知 Kubernetes 系统，所需要的集群工作负载看起来是什么样子的，这就是 Kubernetes 集群的期望状态。

### 235.【简答题】与 Kubernetes 对象工作是否创建、修改，或者删除？

需要使用 Kubernetes API。当使用 kubectl 命令行接口时，比如，CLI 会使用必要的 Kubernetes API 调用，也可以在程序中直接使用 Kubernetes API。

### 236.【简答题】什么是 kubernetes 资源控制器

Kubernetes 中内建了很多 controller（控制器），这些相当于一个状态机，用来控制 Pod 的具体状态和行为

### 237.【简答题】什么是 kubernetes service

Kubemetes 服务是一种为一组功能相同的 pod 提供单一不变的接入点资源。默认情况下，您会获得稳定的集群 IP 地址，集群内部的客户端可以使用该 IP 地址与 Service 中的 Pod 通信。客户端向稳定 IP 地址发送请求，然后请求会被路由到 Service 的其中一个 Pod。

### 238.【简答题】Service 怎么样来识别 pod

Service 通过选择器 selector 来识别器成员 Pod。为使得 Pod 成为 Service 的成员，该 Pod 必须具有选择器中指定的所有标签 label。

### 239.【简答题】kubernetes 存储有几类？

本地存储有 3 种：hostPath、local、emptyDir

## 240.【简答题】kubernetes 为什么需要本地存储

特殊使用场景需求，如需要个临时存储空间，运行 cAdvisor 需要能访问到 node 节点/sys/fs/cgroup 的数据，做本机单节点的 k8s 环境功能测试等等。

容器集群只是做小规模部署，满足开发测试、集成测试需求。

作为分布式存储服务的一种补充手段，比如我在一台 node 主机上插了块 SSD，准备给某个容器吃小灶。

目前主流的两个容器集群存储解决方案是 ceph 和 glusterfs，二者都是典型的网络分布式存储，所有的数据读、写都是对磁盘 IO 和网络 IO 的考验，所以部署存储集群时至少要使用万兆的光纤网卡和光纤交换机。如果你都没有这些硬货的话，强上分布式存储方案的结果就是收获一个以”慢动作”见长的容器集群啦。

分布式存储集群服务的规划、部署和长期的监控、扩容与运行维护是专业性很强的工作，需要有专职的技术人员做长期的技术建设投入。

## 241.【简答题】kubernetes 集群存储有几类？

集群存储有 3 种：1.secret 2.configMap 3.downwardAPI

## 242.【简答题】kubernetes 远程存储有几类？

persistentClaim、nfs、gitRepo、flexVolume、rbd、Cephfs

## 243.【简答题】什么是 kubernetes 集群调度管理？

集群（Cluster）：这个系统管理的 Linux 主机组成一个资源池，用来运行任务，这个资源池就是集群。

## 244.【简答题】什么是 kubernetes 集群调度管理里的作业？

作业（Job）：就是定义集群如何去执行任务，在例子里面 Crontab 就是一个简单的作业，里面明确的告诉了集群需要在什么时间（时间间隔），做什么事情（执行的脚本）。一些作业的定义会复杂很多，比如还会定义一个作业分几个任务做完，以及任务之间的依赖关系，还包括每一个任务对资源的需求。

## 245.【简答题】什么是 kubernetes 集群调度管理里的任务？

任务（Task）：作业需要被调度成具体的执行任务，如果我们定义了一个作业是每天晚上凌晨 1 点执行一个脚本，那么在每天凌晨 1 点被执行的这个脚本进程就是任务。

## 246.【简答题】什么是 kubernetes 集群调度管理的核心任务？

在设计集群调度系统的时候，这个调度系统的核心任务也就是 2 个：

任务调度。作业提交给集群调度系统之后，需要对提交的作业拆分成具体的执行任务，并且跟踪和监控任务的执行结果。在分布式 Cron 的例子中，调度系统需要按照作业的要求定时启动进程，如果进程执行失败，需要重试等，一些复杂的场景，比如 Hadoop 的 Map Reduce，调度系统需要把 Map Reduce 任务拆分成相应的多个 Map 和 Reduce 任务，并且最终拿到任务执行结果的数据。

资源调度：本质上是对任务和资源做匹配，根据集群中主机的资源使用情况，分配合适的资源来运行任务。和操作系统的进程调度算法比较类似，资源调度的主要目标是，在固定的资源供给的情况下，尽可能提高资源使用率，减少任务等待的时间（任务等待资源去执行的时间），减少任务运行的延迟或者响应时间（如果是批量任务的话，就是任务从开始执行到结束的时间，如果在线响应式任务的话，比如 Web 应用，就是每一次响应请求的时间），尽可能公平（资源公平的被分配到所有任务）的同时，还需要考虑任务的优先级。这些目标里面有一些是有冲突的，需要平衡，比如资源利用率和响应时间，公平和优先级。

## 247.【简答题】什么是 kubernetes 安全策略管理

Pod 安全策略由设置和策略组成，它们能够控制 Pod 访问的安全特征。

## 248.【简答题】kubernetes 安全策略管理分为哪几类？

基于布尔值控制：这种类型的字段默认为最严格限制的值。

基于被允许的值集合控制：这种类型的字段会与这组值进行对比，以确认值被允许。

基于策略控制：设置项通过一种策略提供的机制来生成该值，该机制能够确保指定值落在被允许的这组值中。

## 249.【简答题】kubernetes Helm 及其他功能性组件？

在没使用 helm 之前，向 kubernetes 部署应用，我们要依次部署 deployment，svc 等，步骤较繁琐。况且随着很多项目微服务化，复杂的应用在容器中部署以及管理显得较为复杂，helm 通过打包的方式，支持发布的版本管理和控制，很大程度上简化了 Kubernetes 应用的部署和管理。

## 250.【简答题】什么是 Helm 的本质？

Helm 本质就是让 K8s 的应用管理 (Deployment, Service 等)可配置，能动态生成。通过动态生成 K8s 资源清单文件(deployment.yaml, service.yaml)。然后调用 Kubectl 自动执行 K8s 资源部署。

Helm 是官方提供的类似于 YUM 的包管理器，是部署环境的流程封装。Helm 有两个重要的概念:chart 和 release。

chart 是创建一个应用的信息集合，包括各种 Kubernetes 对象的配置模板、参数定义、依赖关系、文档说明等。

chart 是应用部署的自包含逻辑单元。可以将 chart 想象成 apt. yum 中的软件安装包

release 是 chart 的运行实例，代表了一个正在运行的应用。当 chart 被安装到 Kubernetes 集群，就生成一个 release， chart 能够多次安装到同一个集群，每次安装都是一个 release

Helm 包含两个组件:Helm 客户端和 Tiller 服务器

1.MetricServer: 是 kubernetes 集群资源使用情况的聚合器，收集数据给 kubernetes 集群内使用，如 kubectl, hpa, scheduler 等。

2.PrometheusOperator:是一个系统监测和警报工具箱，用来存储监控数据

3.NodeExporter:用于各 node 的关键度量指标状态数据。

4.KubeStateMetrics:收集 kubernetes 集群内资源对象数据，制定告警规则。

5.Prometheus: 采用 pull 方式收集 apiserver, scheduler, controller-manager, kubelet 组件数据，通过 http 协议传输。

6.Grafana:是可视化数据统计和监控平台。

## 251.【简答题】什么是 kubernetes CI/CD 流程？

CI/CD 其实早在很多年前就已经诞生了，主要是为了将频繁的变更测试、构建、交付和部署等进行自动化，减少手工操作。

CI 持续集成主要是在代码更改时自动分支合并、构建并执行一系列的测试（包括单元测试、集成测试、端到端测试等），确保这些变更不会破坏原来的应用。

CD 持续交付和部署则是 CI 测试通过之后把构建结果存档、发布到预布环境和生产环境、最后再进行验收测试的过程。

CI/CD 是 DevOps 的基础，CI/CD 侧重于软件开发过程中的自动化，而 Devops 则是侧重于文化构建，旨在减少开发、运维、QA 之间的沟通鸿沟，促进快速可靠发布的同时还保证产品质量。

CI/CD 一系列流程通常会组成一个流水线，docker 和 Kubernetes 则可以简化这些流水线中的很多流程，比如 Docker 容器可以很容易把有冲突的环境隔离开来，而 Kubernetes 则更进一步简化整个流水线的构建、执行和维护工作。

## 252.【简答题】kubernetes 证书可用时间修改

```
[root@k8s-master ~]# cd /etc/kubernetes/pki/
[root@k8s-master pki]# ls
apiserver.crt apiserver-etcd-client.key apiserver-kubelet-client.crt ca.crt etcd front-proxy-ca.key front-proxy-client.key sa.pub
apiserver-etcd-client.crt apiserver.key apiserver-kubelet-client.key ca.key front-proxy-ca.crt front-proxy-client.crt sa.key
[root@k8s-master pki]# openssl x509 -in apiserver.crt -text -noout |grep Not
```



**Not Before: Apr 5 05:33:55 2021 GMT**

**Not After : Apr 5 05:33:55 2022 GMT**

## 2.部署 Go 语言环境

Go 中文社区: <https://study.golang.com/dl>

Go 官网: <https://golang.org/dl/>

```
cd /opt/src
wget https://studygolang.com/dl/golang/go1.16.3.linux-amd64.tar.gz
tar zxvf go1.16.3.linux-amd64.tar.gz -C /usr/local/
echo "export PATH=$PATH:/usr/local/go/bin" >> /etc/profile
source /etc/profile
go version
```

## 下载源码

```
cd /data && git clone https://github.com/kubernetes/kubernetes.git
cd kubernetes
git checkout -b remotes/origin/release-1.19.0 v1.19.0
```

## 4.修改 Kubeadm 源码包更新证书策略

## 5.更新 kubeadm

## 6.更新各节点至 Master 节点

```
cp -r /etc/kubernetes/pki /etc/kubernetes/pki.old
cd /etc/kubernetes/pki
kubeadm alpha certs renew all --config=/root/kubeadm-config.yaml
openssl x509 -in apiserver.crt -text -noout | grep Not
```

## 7.HA 集群其余 master 节点证书更新

```
#!/bin/bash
masterNode="192.168.33.157 192.168.33.167"
#for host in ${masterNode}; do
# scp /etc/kubernetes/pki/{cA, crt, cA, key, sA, key, sA, pub, front-proxy-cA, crt, front-proxy-cA,
key}
# "${USER}"@$host:/etc/kubernetes/pki/
# scp /etc/kubernetes/pki/etcd/{cA, crt, cA, key} "root"@$host:/etc/kubernetes/pki/etcd
# scp /etc/kubernetes/admin.conf "root"@$host:/etc/kubernetes/
#done
```

```
for host in ${CONTROL_PLANE_IPS}; do
    scp /etc/kubernetes/pki/{cA, crt, cA, key, sA, key, sA, pub, front-proxy-cA, crt, front-proxy-cA,
key}
    "${USER}"@$host:/root/pki/
    scp /etc/kubernetes/pki/etcd/{cA, crt, cA, key} "root"@$host:/root/etcd
    scp /etc/kubernetes/admin.conf "root"@$host:/root/kubernetes/
done
```

## 253. 【简答题】什么是 kubernetes 高可用集群构建？

### ①. 操作系统的配置项

swapoff -a

避免开机启动交换空间

vi /etc/fstab

注释 swap 开头的行

关闭防火墙

ufw disable

配置 DNS

vi /etc/systemd/resolved.conf

取消 DNS 行注释，并增加 DNS 配置如：114.114.114.114，修改后重启下计算机

reboot

### ②. 安装 Docker

构建 kubernetes 基础系统镜像-安装 Docker

### ③. 配置 Docker 加速器

构建 kubernetes 基础系统镜像-安装 Docker(国内镜像加速器可能会很卡，请替换成你自己阿里云镜像加速器)

### ④. 安装 kubeadm, kubelet, kubectl

构建 kubernetes 基础系统镜像-基于 kubeadm 安装 kubernetes 环境

### ⑤. 同步时间

设置时区

dpkg-reconfigure tzdata

选择 Asia (亚洲)

选择 Shanghai (上海)

时间同步

安装 ntpdate

apt-get install ntpdate

设置系统时间与网络时间同步 (cn.pool.ntp.org 位于中国的公共 NTP 服务器)

```
ntpdate cn.pool.ntp.org
```

将系统时间写入硬件时间

```
hwclock --systohc
```

确认时间

```
date
```

自行对照与系统时间是否一致

## ⑥. 配置 IPVS

安装系统工具

```
apt-get install -y ipset ipvsadm
```

配置并加载 IPVS 模块

```
mkdir -p /etc/sysconfig/modules/
```

```
vim /etc/sysconfig/modules/ipvs.modules
```

输入如下内容

```
#!/bin/bash
```

```
modprobe -- ip_vs
```

```
modprobe -- ip_vs_rr
```

```
modprobe -- ip_vs_wrr
```

```
modprobe -- ip_vs_sh
```

```
modprobe -- nf_conntrack_ipv4
```

2. 执行脚本，注意：如果系统重启则需要重新运行该脚本

```
chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules && lsmod | grep -e
```

```
ip_vs -e nf_conntrack_ipv4
```

设置脚本权限，执行脚本，查看信息

执行脚本输出如下：

image.png

## ⑦. 配置内核参数

配置参数

```
vim /etc/sysctl.d/k8s.conf
```

输入如下内容

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
net.bridge.bridge-nf-call-iptables = 1
```

```
net.ipv4.ip_nonlocal_bind = 1
```

```
net.ipv4.ip_forward = 1
```

```
vm.swappiness=0
```

应用参数

```
sysctl -system
```

2.应用参数输出如下（找到 Applying /etc/sysctl.d/k8s.conf 开头的日志）

image.png

⑧. 修改 clouD、cfg

vim /etc/cloud/clouD、cfg

该配置默认为 false，修改为 true 即可

preserve\_hostname: true

以上配置完成后作为 Kubernetes 基础系统镜像，以此为基础复制其他集群

## 10、python 自动化

254.【选择题】以下代码运行结果是什么（假设在 python 2.X 下）（ ）

```
print type(1/2)
```

A、 int      B、 float      C、 0      D、 1 E.0.5

答案：A

解析：在 Py3.x 的版本中，结果是 float 型

255.【选择题】关于 Python 内存管理，下列说法错误的是（ ）

A、变量不必事先声明      B、变量无须先创建和赋值而直接使用  
C、变量无须指定类型      D、可以使用 del 释放资源

答案：B

解析：（不先赋值会报错，is not defined

256.【选择题】下面哪个不是 Python 合法的标识符（ ）

A、int32      B、40XL      C、self      D、name

答案：B

解析：合法的标识符不能以数字开头

257.【选择题】下列哪种说法是错误的（ ）

A、除字典类型外，所有标准对象均可以用于布尔测试  
B、空字符串的布尔值是 False

- C、空列表对象的布尔值是 False
- D、值为 0 的任何数字对象的布尔值是 False

答案: D

解析:  $a=0$ ,  $0$ ,  $\text{bool}(a)$  为 True

258.【选择题】下列表达式的值为 True 的是 ( )

- A、 $5+4j > 2-3j$
- B、 $3 > 2 > 2$
- C、 $(3, 2) < ('a', 'b')$
- D、 $'abc' > 'xyz'$

答案: C

解析: 在 Py2.x 版本中正确, 在 Py3.x 运行错误

259.【选择题】Python 不支持的数据类型有 ( )

- A、char
- B、int
- C、float
- D、list

答案: A

解析: python 里无 char 型数据, 有 string 字符串类型; 但 C 语言中有 char 数据类型

260.【选择题】关于 Python 中的复数, 下列说法错误的是 ( )

- A、表示复数的语法是  $\text{real} + \text{image}j$
- B、实部和虚部都是浮点数
- C、虚部必须后缀  $j$ , 且必须是小写
- D、方法 `conjugate` 返回复数的共轭复数

答案: C

解析: 复数虚部的后缀也可以是大写的 J

261.【选择题】关于字符串下列说法错误的是 ( )

- A、字符应该视为长度为 1 的字符串
- B、字符串以 `\0` 标志字符串的结束
- C、既可以用单引号, 也可以用双引号创建字符串
- D、在三引号字符串中可以包含换行回车等特殊字符

答案: B

解析: python 因为字符串有长度限制, 到了长度就标志字符串的结束

262. 【选择题】 以下不能创建一个字典的语句是（ ）

- A、dict1 = {}                                      B、dict2 = { 3 : 5 }  
C、dict3 = {[1, 2, 3]: "uestc" }      D、dict4 = {(1, 2, 3): "uestc" }

答案：C

解析：字典的键必须是不变的，而列表是可变的

263. 【选择题】 以下代码运行结果是什么？（ ）

```
print r"nwoow"
```

- A、new line then the string: woow  
B、the text exactly like this: r" nwoow"  
C、the text like exactly like this: nwoow  
D、the letter r and then newline then the text: woow  
E、the letter r then the text like this: nwoow

答案：C

解析：字符串前面加 r，表示禁止字符串转义

264. 【选择题】 以下代码运行结果是什么？（ ）

```
numbers = [1, 2, 3, 4]
numbers.append([5, 6, 7, 8])
print len(numbers)
```

- A、4      B、5      C、8      D、12      E.An exception is thrown

答案：B

解析：numbers 为[1, 2, 3, 4, [5, 6, 7, 8]]

265. 【选择题】 假设有个文件/usr/lib/python/person.py 内容如下，执行以下命令输出结果是什么？（ ）

```
python /usr/lib/python/person.py
class Person:
def __init__(self):
```

**A、 Person** **B、 getAge**  
**C、 usr.lib、 python.person** **D、 main E.Anexception is thrown**

解析：当内部单独执行模块时，`name__`的返回值就是`__main__`，当被外部模块调用时，返回的是模块的名字，在此即为 `person`

**A、 min = x if x< y= y**  
**B、 max = x > y ?x:y**  
**C、 if (x >y) print x**  
**D、 while True :pass**

```
kvps={ '1' :1, '2' :2 }
theCopy=kvps.copy()
kvps[ '1' ]=5
sum = kvps[ '1' ]+ theCopy[ '1' ]
print sum
```

解析：字典的浅拷贝，只拷贝父对象，即 theCopy 拷贝了 kvps，当 kvps 改变表面时，theCopy 不改变相应值

A、  $x=y=z=1$       B、  $x=(y=z+1)$   
C、  $x, y=y, x$       D、  $x+=y$

269.【选择题】以下选项中，不是 Python 语言保留字的是（ ）

- A、 do
- B、 pass
- C、 except
- D、 while

答案：A

解析：查看保留字的方法如下：import keyword; keyword.kwlist

270.【选择题】以下代码运行结果是什么？（C）

```
counter = 1
def doLotsOfStuff():
    global counter
    for i in (1, 2, 3):
        counter += 1
    doLotsOfStuff()
print counter
```

- A、 1
- B、 3
- C、 4
- D、 7
- E、 none of the above

271.【选择题】以下代码运行结果是什么？（C）

```
for i in range(2):
    print i
for i in range(4, 6):
    print i
```

- A、 2, 4, 6
- B、 0, 1, 2, 4, 5, 6
- C、 0, 1, 4, 5
- D、 0, 1, 4, 5, 6, 7, 8, 9
- E、 1, 2, 4, 5, 6

272.【选择题】以下代码运行结果是什么？（B）

```
import math
```



```
print math.floor(5.5)
```

A、 5      B、 5.0      C、 5.5      D、 6      E.6.0

273.【选择题】以下代码运行结果是什么？（B）

```
names1 = [ 'Amir', 'Barry', 'Chales', 'Dao' ]  
if 'amir' in names1:  
    print 1  
else:  
    print 2
```

A、 1      B、 2      C、 An exception is thrown      D、 Nothing

274.【简答题】Python 是如何进行内存管理的？

从三个方面来说，一对象的引用计数机制，二垃圾回收机制，三内存池机制

一、对象的引用计数机制

Python 内部使用引用计数，来保持追踪内存中的对象，所有对象都有引用计数。

引用计数增加的情况：

- 1，一个对象分配一个新名称
- 2，将其放入一个容器中（如列表、元组或字典）

引用计数减少的情况：

- 1，使用 del 语句对对象别名显示的销毁
- 2，引用超出作用域或被重新赋值

sys.getrefcount() 函数可以获得对象的当前引用计数

多数情况下，引用计数比你猜测得要大得多。对于不可变数据（如数字和字符串），解释器会在程序的不同部分共享内存，以便节约内存。

二、垃圾回收

- 1，当一个对象的引用计数归零时，它将被垃圾收集机制处理掉。

2，当两个对象 a 和 b 相互引用时，del 语句可以减少 a 和 b 的引用计数，并销毁用于引用底层对象的名称。然而由于每个对象都包含一个对其他对象的应用，因此引用计数不会归零，对象也不会销毁。（从而导致内存泄露）。为解决这一问题，解释器会定期执行一个循环检测器，搜索不可访问对象的循环并删除它们。

三、内存池机制

Python 提供了对内存的垃圾收集机制，但是它将不用的内存放到内存池而不是返回给操作系统。

1，Pymalloc 机制。为了加速 Python 的执行效率，Python 引入了一个内存池机制，用于管理对小块内存的申请和释放。

2, Python 中所有小于 256 个字节的对象都使用 pymalloc 实现的分配器, 而大的对象则使用系统的 malloc。

3, 对于 Python 对象, 如整数, 浮点数和 List, 都有其独立的私有内存池, 对象间不共享他们的内存池。也就是说如果你分配又释放了大量的整数, 用于缓存这些整数的内存就不能再分配给浮点数。

### 275. 【简答题】什么是 lambda 函数? 它有什么好处?

lambda 表达式, 通常是在需要一个函数, 但是又不想费神去命名一个函数的场合下使用, 也就是指匿名函数

lambda 函数: 首要用途是指点短小的回调函数

lambda [arguments]:expression

```
a=lambdax, y:x+y
```

```
a(3, 11)
```

### 276. 【简答题】Python 里面如何实现 tuple 和 list 的转换?

直接使用 tuple 和 list 函数就行了, type() 可以判断对象的类型

### 277. 【编程题】请写出一段 Python 代码实现删除一个 list 里面的重复元素

1, 使用 set 函数, set(list)

2, 使用字典函数,

```
a=[1, 2, 4, 2, 4, 5, 6, 5, 7, 8, 9, 0]
b={}
b=B.fromkeys(a)
c=list(B.keys())
c
```

### 278. 【编程题】编程用 sort 进行排序, 然后从最后一个元素开始判断

```
a=[1, 2, 4, 2, 4, 5, 7, 10, 5, 5, 7, 8, 9, 0, 3]
a.sort()
last=a[-1]
for i in range(len(a)-2, -1, -1):
    if last==a[i]:
        del a[i]
    else: last=a[i]
```

```
print(a)
```

## 279.【编程题】Python 里面如何拷贝一个对象？（赋值，浅拷贝，深拷贝的区别）

赋值（=），就是创建了一个新的引用，修改其中任意一个变量都会影响到另一个。

浅拷贝：创建一个新的对象，但它包含的是对原始对象中包含项的引用（如果用引用的方式修改其中一个对象，另外一个也会修改改变）{1，完全切片方法；2，工厂函数，如 `list()`；3，`copy` 模块的 `copy()` 函数}

深拷贝：创建一个新的对象，并且递归的复制它所包含的对象（修改其中一个，另外一个不会改变）{`copy` 模块的 `deep.deepcopy()` 函数}

## 280.【简答题】介绍一下 `except` 的用法和作用？

```
try...except...except...[else...][finally...]
```

执行 `try` 下的语句，如果引发异常，则执行过程会跳到 `except` 语句。对每个 `except` 分支顺序尝试执行，如果引发的异常与 `except` 中的异常组匹配，执行相应的语句。如果所有的 `except` 都不匹配，则异常会传递到下一个调用本代码的最高层 `try` 代码中。

`try` 下的语句正常执行，则执行 `else` 块代码。如果发生异常，就不会执行

如果存在 `finally` 语句，最后总是会执行。

## 281.【简答题】Python 中 `pass` 语句的作用是什么？

`pass` 语句不会执行任何操作，一般作为占位符或者创建占位程序，`while False: pass`

## 282.【简答题】介绍一下 Python 下 `range()` 函数的用法？

列出一组数据，经常用在 `for in range()` 循环中

## 283.【简答题】如何用 Python 来进行查询和替换一个文本字符串？

可以使用 `re` 模块中的 `sub()` 函数或者 `subn()` 函数来进行查询和替换，

格式：`sub(replacement, string[, count=0])`（`replacement` 是被替换成的文本，`string` 是需要被替换的文本，`count` 是一个可选参数，指最大被替换的数量）

```
import re
p=re.compile( 'blue|white|red' )
print(p.sub( 'colour', 'blue socks and red shoes' ))
colour socks and colourshoes
print(p.sub( 'colour', 'blue socks and red shoes', count=1))
```

subn()方法执行的效果跟 sub()一样，不过它会返回一个二维数组，包括替换后的新的字符串和总共替换的数量

## 284. 【简答题】Python 里面 match()和 search()的区别？

re 模块中 match(pattern, string[, flags])，检查 string 的开头是否与 pattern 匹配。

re 模块中 re.search(pattern, string[, flags])，在 string 搜索 pattern 的第一个匹配值。

```
print(re.match( 'super',    'superstition' ).span())
(0, 5)
print(re.match( 'super',    'insuperable' ))
None
print(re.search( 'super',    'superstition' ).span())
(0, 5)
print(re.search( 'super',    'insuperable' ).span())
(2, 7)
```

## 285. 【简答题】用 Python 匹配 HTML tag 的时候，<.>和<.>?>有什么区别？

术语叫贪婪匹配(<.>)和非贪婪匹配(<.>?)

## 286. 【简答题】Python 里面如何生成随机数？

random 模块

随机整数：random.randint(a, b): 返回随机整数 x，a<=x<=b

random.randrange(start, stop, [, step]): 返回一个范围在(start, stop, step)之间的随机整数，不包括结束值。

随机实数：random.random():返回 0 到 1 之间的浮点数

random.uniform(a, b):返回指定范围内的浮点数。

## 287. 【简答题】有没有一个工具可以帮助查找 python 的 bug 和进行静态的代码分析？

PyChecker 是一个 python 代码的静态分析工具，它可以帮助查找 python 代码的 bug，会对代码的复杂度和格式提出警告

Pylint 是另外一个工具可以进行 codingstandard 检查

## 288.【简答题】如何在一个 function 里面设置一个全局的变量？

解决方法是在 function 的开始插入一个 global 声明：

```
def f()  
    global x
```

## 289.【简答题】单引号，双引号，三引号的区别

单引号和双引号是等效的，如果要换行，需要符号()，三引号则可以直接换行，并且可以包含注释

如果要表示 Let' s go 这个字符串

单引号：s4 = 'Let\' s go'

双引号：s5 = "Let' s go"

s6 = 'I realy like "python" !'