



达内 Linux 云计算 Architecture 阶段必备知识

达内 Linux 云计算学院

2017/6/18

目 录

DOCKER.....	3
1 为什么要使用 Docker? 使用 Docker 能给企业带来哪些好处?	3
2 dockerfile 的作用是什么 FROM RUN COPY CMD ENTRYPOINT EXPOSE 指令作用是什么? CMD ENTRYPOINT 区别是什么? EXPOSE 22 后是不是直接就能访问容器的 22 端口?	3
3 什么是联合文件系统? 构建镜像的时候, 每一个镜像 ID 有什么作用?	4
4 什么是 宿主机数据卷, 有什么特性?	4
5 如何为不同的容器分配不同的 cpu 内存等物理资源?	4
SALTSTACK.....	4
1.简述 Saltstack	4
2.Saltstack 是架构.....	5
3.Saltstack 两大核心功能	5
4.Saltstack 的两个数据系统以及区别(重点).....	5
5.saltstack 编写 state 文件时如何使用 jinja 模板传递变量。	5
6.Saltstack 常用的状态模块	5
7.如何解决模块之间的依赖关系?	6
8.如何给 100 台服务器安装 Salt-minion 客户端软件, 并完成与 salt-master 的认证, 接受 salt-master 的 管理?	6
ELK	6
1.ELK 三大组件以及其各自的作用.....	6
2.Elasticsearch 常用的插件及其特点	6
3.什么是倒排索引	7
4.如何批量的创建和获取 es 中的数据.....	7
5.Logstash 的三大配置区域以及常用的插件	7
6.叙述 ELK 是如何工作的.....	7
7.为什么使用 filebeat 而不使用 logstash 进行客户端数据(日志)采集?	7
8.如何区分不同类型的数据, 例如 tomcat 报错日志和 nginx 访问日志? (重点)	8
Hadoop	8
1、Hadoop 的部署模式有哪几种?	8
2、hadoop 的三大核心组件是什么?	8
3、HDFS 里面的角色有哪几种,作用是什么?	8
4、怎么样格式化 hdfs namenode ?	8
5、hdfs 上传、下载、显示文件的命令是什么?	9
6、zookeeper 是什么?	9
7、zookeeper 下列角色的作用是什么?	9
8、kafka 是什么?	9
9、简要列出 hadoop 的安装步骤.....	9
10、spark 是什么?	9
11、简述 zookeeper 集群配置过程.....	10

DOCKER

1 为什么要使用 Docker? 使用 Docker 能给企业带来哪些好处?

1、更简单、更易用、更快速:

与虚拟机相比,部署多个应用无需安装创建多个虚拟机,安装多个操作系统,所有的应用模块都寄存于宿主机的操作系统,共享宿主机硬件资源,多个应用模块类似于宿主机上多个“应用进程”;

与虚拟机相比,容器的启动、停止等能都实现秒级的速度,从而实现业务系统服务的快速弹性扩展;

所有的运维基础服务(如 mysql redis 等)以及公司自有模块都可以随意封装为镜像,只要有宿主机系统环境,就可以无需部署,直接启动服务,从而实现“一次封装,到处运行”;

2、更灵活的硬件资源调整、管控:

所有容器能够针对 CPU,内存、磁盘 IO、网络等进行更细化的调整控制,让所有容器更能充分利用宿主机的硬件资源;

3、透明部署,促进企业实现真正的 DevOps:

docker 的出现,让运维、测试、研发等企业技术人员,彻底跳出“环境部署的苦海”,实现真正的企业内部业务系统的“透明部署”,简化了技术部门产品发布的流程;

2 dockerfile 的作用是什么 FROM RUN COPY CMD ENTRYPOINT EXPOSE 指令作用是什么? CMD ENTRYPOINT 区别是什么? EXPOSE 22 后是不是直接就能访问容器的 22 端口 ?

dockerfile 类似于创建镜像的配置文件,里面定义了多个指令,每一个指令实现一种操作,在创建镜像的时候,所有的命令依次执行,最终创建成一个完整的镜像;

FROM: 定义哪个镜像作为本镜像的基础镜像,基础镜像的所有数据会被直接加载到当前镜像

RUN: 定义在当前基础镜像基础上执行的命令,并提交给新的镜像;

COPY: 把镜像工作目录中的指定文件或目录所有数据加载到新镜像指定的路径下面;

CMD: 指定一个容器启动时自动执行的命令;

ENTRYPOINT: 指定一个容器启动时自动执行的命令;

使用 `docker run` 命令启动容器,并手工指定执行一个命令,那么这个命令是不能覆盖 `dockerfile` 中定义的 `ENTRYPOINT` 命令,但是能够覆盖 `CMD` 命令不能直接访问! `docker` 处于安全考虑, `EXPOSE` 并不会直接打开端口,必须要在运行容器的时候明确开启端口映射才会真正开放此端口。



3 什么是联合文件系统？构建镜像的时候，每一个镜像 ID 有什么作用？

镜像通过联合文件系统（Union Filesystem）将各层文件系统叠加在一起，在用户看来就是一个完整的文件系统，镜像中的每一层可以看做是一个“数据快照”，所有数据层叠加就形成一个完整的镜像。多个镜像中，数据完全一样的独立成一个层（如基础镜像，或者不同的镜像中具有相同操作得部分），相同部分只需要存储一份。因此镜像所有占用的磁盘空间（docker images 显示的大小）远远小于所有镜像之和。

每一个镜像 ID,就是在创建镜像时候的一个数据分层的对应的 ID，利用这个 ID，可以恢复到这个 ID 对应的数据分层快照，这个 ID 类似于版本控制，从而能够恢复到任何一个“数据快照”节点。

4 什么是 宿主机数据卷， 有什么特性？

数据卷是宿主机上面的某个特殊目录或者文件，它可供一个或多个容器使用（类似于 Linux 下对目录或文件进行 mount），它绕过了 UFS（联合文件系统）。

特性：

- 1、数据卷中所有的数据是保存在宿主机磁盘本地的，它可以被一个或多个容器使用；在权限允许的前提下，宿主机+各个容器对数据卷都可以进行读、写、删除等各项操作，并且在宿主机或者各个容器任何一个节点上面变更数据（增、删、改），将会实时同步到宿主机本地以及其它所有挂载了此数据卷的其它容器内部，实现数据的“全局同步”；
- 2、对数据卷的更新，不会影响镜像；
- 3、卷会一直存在，直到没有容器使用；

5 如何为不同的容器分配不同的 cpu 内存等物理资源？

--cpu-shares 设置容器使用的 cpu 权重，在多个容器对 cpu 发生资源争用的时候，按照权重分配资源；
--cpuset-cpus 把容器使用 cpu 资源固定在一个或多个指定的 cpu 核上；
-m 设置容器使用宿主机的物理内存容量大小
--memory-swap 设置容器使用宿主机的物理内存+物理交换分区的总容量大小；

SALTSTACK

1.简述 Saltstack

Saltstack 是基于 Python 开发的一套 C/S 架构配置管理工具，使用 SSL 证书进行认证管理，底层通过



ZeroMQ 进行(master 与 minion)通信。

2.Saltstack 是架构

Saltstack 基于 C/S 架构，服务端称作 Master，客户端称作 Minion。Saltstack 服务模式分为两类：

一种是客户端主动请求服务端模式。(客户端主动模式，主动拉取服务端配置)

一种是消息队列的 pub/sub 服务模式。(客户端被动模式，服务端主动下发配置)

*建议：尽量人为触发服务端配置下发执行，提高可控性。

3.Saltstack 两大核心功能

远程执行 通过命令行使用常用的远程执行模块操作客户端。

配置管理 通过编写 state 模块文件，执行该模块文件操作客户端。

4.Saltstack 的两个数据系统以及区别(重点)

grains 存储 minion 端的信息，包括一些网络、硬件等信息，保存在 minion 端。一般为静态信息，非经常变化的数据。

pillar 存储在 master 端，存放需要提供给 minion 的信息。常用于敏感信息，每个 minion 只能访问 master 分配给自己的 pillar 信息，更多的是经常动态的变化信息。

区别：

- ①.grains 是静态、不常变化的；pillar 则是动态的。
- ②.grains 是存储在 minion 本地，而 pillar 存储在 master 本地。
- ③.minion 有权限操作自己的 grains 值，如增加、删除，但 minion 只能查看自己的 pillar，无权修改。

5.saltstack 编写 state 文件时如何使用 jinja 模板传递变量。

- ①.在 state 文件中使用"- template: jinja"声明。
- ②.在模版文件中使用变量"{{ name }}"声明,name 为变量，自己定义。
- ③.在 state 文件中使用"- defaults: name: value"声明。

6.Saltstack 常用的状态模块

file/pkg/cmd/service 等模块



7.如何解决模块之间的依赖关系？

可以使用如下方法解决模块之间的依赖关系：

watch: 关注某一事件变化

watch_in: 被关注自身事件变化

require: 依赖某一事件发生

require_in: 被依赖自身事件发生

unless: 判断是否为 false，如果为 false 则执行。常用在 cmd 中 name 的命令。

onlyif: 判断是否为 true，如果为 true 则执行。

8.如何给 100 台服务器安装 Salt-minion 客户端软件，并完成与 salt-master 的认证，接受 salt-master 的管理？

①.在 master 节点安装 Saltstack 的 salt-master 以及 Salt-ssh 组件

②.在/etc/roster 文件中定义需要安装 minion 客户端的 ip 地址，用户名和密码

③.将 master 端的 key 信息写入到所有客户端，然后删除 roster 文件中的密码，避免密码泄露

④.编写 state 状态文件并执行，等待执行完毕，所有客户端会安装好 minion 软件，并启动服务。

⑤.在 master 端上执行 salt-key -L 查看，会显示所有需要认证的客户端列表，完成认证即完成了 minion 客户端软件的批量部署。

ELK

1.ELK 三大组件以及其各自的作用

ELK 是由 Elasticsearch、Logstash 以及 Kibana 三个组件组成的日志收集、处理、检索、展示平台。

es 是一个采用 RESTful API 标准，实现分布式、可扩展以及高可用的实时数据存储分析的全文搜索工具。底层使用 lucene 全文搜索框架。

logstash 是一个数据采集、加工处理以及传输(输出)的工具。拥有丰富的数据收集处理输出插件。

kibana 是一个数据可视化平台工具。能够完成数据的检索、图表绘制以及展示等功能

2.Elasticsearch 常用的插件及其特点

常用插件: head/bigdesk/kopf/marvel

head 插件: head 插件是对 es 进行各种操作的客户端。es 集群的健康状态: green 表示健康, yellow 代表少量分片未进行分配, 红色代表大量分片未进行分配。

head 插件能对索引进行创建删除关闭优化等操作，还可以查询浏览相应索引的文档内容。

bigdesk: bigdesk 插件能够实现对节点实时的状态监控，包括 jvm、Linux 以及 elasticsearch 的情况。

kopf: 集成了 head 和 bigdesk 的功能于一体，友好的显示和操作界面。但是 kopf 的节点监控功能弱于



bigdesk。

3.什么是倒排索引

概念：倒排索引(英语：Inverted index)，也常被称为反向索引、置入档案或反向档案。是一种索引方法，被用来存储在全文搜索下某个单词在一个文档或者一组文档中的存储位置的映射。它是文档检索系统中最常用的数据结构。

正向索引和倒排索引对比：

内容 --正向--> 关键字/词

内容 <--倒排-- 关键字/词

4.如何批量的创建和获取 es 中的数据

通过使用 bulk 批量创建导入 ES 数据

通过使用 mget 批量获取 ES 数据

5.Logstash 的三大配置区域以及常用的插件

input{}: 该区域主要配置数据从哪里来。常用的插件有 file、tcp/udp、syslog、redis、beats 等

filter{}: 该区域主要配置数据如何处理。常用的插件有 json、grok、mutate、date 等。

output{}: 该区域主要配置数据到哪里去。常用插件有 file、tcp/udp、redis、elasticsearch 等。

codec: 编解码插件，常用的编解码插件：plain、json、multiline、rubydebug 等。

6.叙述 ELK 是如何工作的

①.在被收集数据(日志)的节点，安装 filebeat，配置监听日志文件路径，配置转发数据到 logstash/redis/elasticsearch 等。

②.在 logstash 部署节点编写配置文件，接收来自 filebeat 发送的数据(日志)，并按照规定处理，将处理后的数据输出到 elasticsearch。

③.kibana 通过调用 es 的 api 接口展示数据，可以在 kibana 界面完成对数据的检索，统计，绘图，展示等。

7.为什么使用 filebeat 而不使用 logstash 进行客户端数据(日志)采集？

logstash 需要安装 jvm 环境，比较重，占用内存高，一般占用 5-6 百 M。

filebeat 不需要安装 jvm 环境，轻量，占用内存更低，一般也就是几十 M。官方也推荐使用 filebeat。



8.如何区分不同类型的数据,例如 tomcat 报错日志和 nginx 访问日志? (重点)

在 filebeat 收集日志的同时为日志添加 type 或者 tags 等字段, filebeat 配置文件中对此定义。
在使用 type 作为不同日志类型判断依据时, 源数据中不能存在 type 字段。

Hadoop

1、Hadoop 的部署模式有哪几种?

- a)单机
- b)伪分布式
- c)完全分布式

2、hadoop 的三大核心组件是什么?

- a) YARN
- b) HDFS
- c) MapReduce

3、HDFS 里面的角色有哪几种,作用是什么?

- a) NameNode 元数据服务器, 其中包含每个文件、文件位置以及这些文件及其所在的 DataNode 内的所有数据块的内存映射。
- b) Datanode 真实的数据存储节点。
- c) secondaryNameNode 它的作用是定期合并 NameNode 的 fsimage 和 日志, 在更新 NameNode 映像。

4、怎么样格式化 hdfs namenode ?

hdfs namenode -format



5、hdfs 上传、下载、显示文件的命令是什么？

```
hadoop fs -put /urfile
hadoop fs -get /urfile
hadoop fs -cat /urfile
```

6、zookeeper 是什么？

ZooKeeper 是一个分布式的，分布式应用程序协调服务。

7、zookeeper 下列角色的作用是什么？

leader 领导者，负责进行投票的发起和决议，更新系统状态

follower 用于接收客户请求并向客户端返回结果，在选主过程中参与投票

observer 可以接收客户端连接，将写请求转发给 leader 节点，但 Observer 不参加投票过程。

8、kafka 是什么？

Kafka 是一个分布式消息系统。

9、简要列出 hadoop 的安装步骤

```
安装 java 环境 (openjdk)
配置 ssh 免密码登录
设置 JAVA_HOME 变量 hadoop-env.sh
配置 hadoop core-site hdfs-site mapred-site yarn-site slaves
同步到所有机器
创建 数据目录和格式化 namenode
启动集群
```

10、spark 是什么？

Spark 是一个基于内存计算的开源的集群计算系统



11、简述 zookeeper 集群配置过程

配置 zoo.cfg

在每台机器生成 myid 里面的数字不能相同

同步配置

在所有机器上启动