实验题目:

实验1: Hadoop 环境配置与安装;

实验环境:

阿里云服务器 Linux centos7 操作系统; Mac 端 SSH 连接阿里云服务器;

实验技术原理:

【案例 1-1】Linux 设置静态 IP 地址

Ip 的地址可以通过配置网卡文件进行设置,该文件是/etc/sysconfig/ network -scripts- /ifcfg-eth0,编辑该文件中的内容即可。

【案例 1-2】修改主机名和映射关系

通过配置文件/etc/sysconfig/network 把计算机的主机名和 ip 地址关联起来,访问计算机地址时,可以不用写 ip 地址,只需要计算机名即可实现,对 ip 地址有保护作用

【案例 1-3】Linux 常用命令: Linux 操作系统常用命令(必做)

在 linux 系统中,通过命令终端的各种指定和 linux 系统进行交互,以达到 用户希望系统做的事情。

【案例 1-5】ssh 免密码配置

SSH 是目前比较可靠的专为远程登录会话和其他网络服务提供安全的协议。不同主机之间在进行通信时,一般都是需要输入密码进行验证, ssh 免密码之后,只要通过指定主机地址和端口号就可以实现不同的计算机之间访问时,不需要密码实现直接访问。ssh 免密码登录主要采用算法有:对称加密算法和非对称加密算法。例如有两个 linux 系统 simple01 和 simple02:

1. 首先在 simple01 上 ssh-keygen -t rsa 生成一对钥匙 id_rsa (私 钥) 和 id_rsa. pub (公钥)

2. simple01 执行 ssh-copy-id simple02 在 simple02 端生成 autherired_keys 文件并保存了 simple01 机器上的 id_rsa. pub 信息内容

3. simple01 向 simple02 发送一个 ssh 请求

4. 请求在传输中带有 simple01 机器发送的指令信息,指令信息如 (root@simple01+指令等)

5. simple02 接收到 simple01 的信息后,会根据(root@simple01+ 指令等)到 authorized_keys 中进行查找,如果没找到,向 simple01 返回需 要输入密码,如果找到 simple02 会随机生成一个字符串,自已保存一份,然 后使用 simple01 的公钥进行加密。

6. 将加密后的字符串发送给 simple01

7. simple01 接收到加密后的字符串,用自己的私钥进行解密。

8. 将解密后的字符串再发送给 simple02

9. simple02 接收 simple01 发送的解密后的字符串与原来的字符串 进行比对,如果一致,可以免密码登录

【案例 2-1】 linux 系统 JDK 的安装和配置

Jdk 作为一个虚拟机安装在 linux 系统中,独立于操作系统,从而可以实现 java 编程的跨平台性, java 程序的执行过程都是 jvm 控制的,并且 jdk 提供 很多开发工具,让开发实现起来更加容易。

【案例 2-2】 linux 系统下 HelloWorld

通过 vim 编译后缀名为 Java 文件,再通过 Java 运行后缀名为 Java 的文件, 生成 class 结尾的文件,了解 Java 的运行机制。

【案例 2-3】hadoop 单机分布式配置与测试(必做)

在缺省情况下,hadoop 被配置为非分布式模式运行的,作为一个单一的 java 进程,这种配置一般用于调式作用。

【案例 2-4】hadoop 伪分布式配置与测试

Hadoop 伪分布式配置是在没有多台计算机节点的情况下,对 hadoop 的分布式存储和计算进行模拟安装和配置。通过在一台计算机节点上解压 hadoop 安装压缩包后,然后进行 hadoop 相关文件进行配置,让 hadoop 的核心进程都运行在同一台机器上并实现对数据存储和计算的测试支持。此时的配置环境并没有达到真实的分布式存储和分布式计算。



执行 systemctl restart network 命令,重启网络服务。
<pre>(base) [root@ferry ~]# (base) [root@ferry ~]#</pre>
二 手动设置 ip 地址
2.1 在 linux 系统命令终端,执行命令: /etc/sysconfig/network-scripts 切换到该目 录并查看该目录下的文件 ifcfg-eth0。如图所示
<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ ~]i cd /etc/sysconfig/network-scripts/ (base) [root@iZwz9j14gd1jp0mfm3206jZ network-scripts]# ls ifcfg-eth0 ifdown-ippp ifdown-routes ifup ifup-ipv6 ifup-ppp ifup-tunnel ifcfg-lo ifdown-ipv6 ifdown-sit ifup-aliases ifup-isdn ifup-routes ifup-wireless ifdown ifdown-isdn ifdown-Team ifup-bnep ifup-plip ifup-sit init.ipv6-global ifdown-bnep ifdown-post ifdown-TeamPort ifup-eth ifup-plusb ifup-Team network-functions ifdown-eth ifdown-ppp ifdown-tunnel ifup-ippp ifup-post ifup-TeamPort network-functions-ipv6 (base) [root@iZwz9j14gd1jp0mfm3206jZ network-scripts]#</pre>
2.2 在 linux 系统命令终端,执行命令: vim ifcfg-eth0 并修改文件的内容,按i进入 编辑内容,编译完成后按 Esc 退出编译状态,之后:wq 保存并退出。如图所示
(base) [root@iZwz9j14gd1jp0mfm3206jZ network-scripts]# vim ifcfg-eth0
ifcfg-eth0 1 DEVICE=eth0
2 BOOTPROTO=dhcp 3 ONBOOT=yes
~ 2.3 配置完毕 ip 地址之后,执行命令: ifconfig 查看配置效果。如图所示
<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ network-scripts]# ifconfig eth0: flags=4163<up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast></pre>
inet 172.18.242.188 netmask 255.255.240.0 broadcast 172.18.255.255 ether 00:16:3e:12:e3:b1 txgueuelen 1000 (Ethernet)
RX packets 3344403 bytes 1012404001 (965.5 MiB) RX errors 0 dronned 0 overrups 0 frame 0
TX packets 2703296 bytes 599786740 (572.0 MiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73 <up,loopback,running> mtu 65536</up,loopback,running>
inet 127.0.0.1 netmask 255.0.0.0 loop txaueuelen 1000 (Local Loopback)
RX packets 104620 bytes 50869813 (48.5 MiB)
RX errors 0 dropped 0 overruns 0 frame 0 TX packets 104620 bytes 50869813 (48.5 MiB)

深圳大学学生实验报告用纸

【案例 1-2】修改主机名和映射关系

一、修改配置文件

1.1 在 linux 系统中启动命令终端。

1.2 启动命令终端,在任何目录下执行命令: cd /etc/sysconfig 切换到该目录并查看 目录下的文件,可以发现存在文件 network。如图所示

(base) [root@iZwz9j14gd1jp0mfm3206jZ jdk]# cd /etc/sysconfig								
anaconda	cpupower	init	man-db	ntpdate	run-parts	system-config-users		
atd	crond	ip6tables-config		raid-check	samba	wpa_supplicant		
authconfig	ebtables-config	iptables-config	netconsole	rdisc	selinux			
cbq	fcoe	irqbalance	network	readonly-root	sshd			
chronyd	firewalld	kdump		rsyncd	sysstat			
console	grub	kernel	ntpd	rsyslog	sysstat.ioconf			
(base) [roo	(base) [root@iZwz9j14gd1jp0mfm3206jZ sysconfig]#							

1.3 在/etc/sysconfig 目录下找到文件 network 并执行命令:vim network 编辑其中 的内 容。按 i 进入编辑内容,编译完成后按 Esc 退出编译状态,之后:wq 保存并退出。 如图所示

(base) [root@iZwz9j14gd1jp0mfm3206jZ sysconfig] # vim network

network

🛿 🏄 Created by anaconda

1.4 在/etc/sysconfig 目录下执行命令: cat network 进行查看 network 文件中的内 容。 如图所示

1.5 如果要修改主机名和 ip 地址具有映射关系。执行命令: vim /etc/hosts 并编辑其中的内容。按 i 进入编辑内容,编译完成后按 Esc 退出编译状态,把主机名修改为ferry.szu 之后:wq 保存并退出。如图所示

(base) [root@iZwz9j14gd1jp0mfm3206jZ sysconfig]# vim /etc/hosts







<pre>(base) [root@i2wz9j14gd1jp0mfm3206 (base) [root@i2wz9j14gd1jp0mfm3206</pre>	jZ ~]# mkdir AAA jZ ~]# ls		
AAA anaconda3	General hadoop-2.4.1.tar.az	jichu.ipynb Movina	Text, vim-8.1.tor.bz2
Anaconda3-2020.02-Linux-x86_64.sh	hello	python基础-张.ipynb	【Y1】python训练营-基础知识.ipynb
Colors Editing	hello.c.html Helper	Spell test	未命名1.ipynb 未命名.ipynb
ersion	hl.c	test.c	
(base) [root@iZwz9j14gd1jp0mfm3206	jZ ~] ¥ cd AAA		
(base) [roote:2w29]14ga1]pomtm3206 mkdir: 无法创建目录"BBB/CCC": 没有	JZ AAAJ# mkdir BBB/(那个文件或目录		
(base) [root@iZwz9j14gd1jp0mfm3206	jZ AAA]# mkdir -p BB	B/CCC	
<pre>(base) [root@iZwz9j14gd1jp0mfm3206 .:</pre>	jZ AAA]# ls -R		
BBB			
./BBB: CCC			
./BBB/CCC: (base) [root@iZwz9j14gd1jp0mfm3206	jz AAA]#		
1.6 touch 命令和 echo 命	☆ 。		
如果希望创建一个文件	‡,可以执行命	令:touchtian.t	xt。如果希望把命令终端输入
的信息写入到创建的文件	中,可以执行命	令:echo'this	isa tian file'>>tian.txt
如图所示			
(base) [root@iZwz9j14gd	ljp0mfm3206jZ AAA	<pre># touch tian.txt # acho !this is a</pre>	tion filel as tion but
(base) [root@l2wz9]14ga	1jp0mtm3206j2 AAA 1jp0mfm3206j7 AAA	# ecno 'this is a]# cat tian tyt	tian file' >> tian.txt
this is a tian file			
(base) [root@iZwz9j14gd	1jp0mfm3206jZ AAA]#	
17 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			
1./ rm 命令。			
如果想删除指定的文件	‡或文件夹,可I	以执行命令:rmti	an.txt。如果希望强制删除或
递归删除,可以通过增加参	参数−f 或−r。如	1图所示	
(hasa) [root@i7w	noil/adlin0mf		16
(Duse) [rooterzw	29J14gu1Jpolin	m3200]2 AAA]#	15
BBB tlan.txt			.
(base) [root@izwa	29]14gd1]p0mf	m3206jZ AAA]#	rm -f tlan.txt
(base) [root@iZw	z9j14gd1jp0mf	m3206jZ AAA]#	ls
BBB			
(base) [root@iZwz9j14gd1jp0mfm3	206jZ AAA]# ls		
BBB (base) Froot@iZwz9i14ad1ip0mfm3	206iZ AAA]# cd		
(base) [root@iZwz9j14gd1jp0mfm3	206jZ ~]# rm -r -f AAA		
<pre>(base) [root@iZwz9j14gd1jp0mfm3 anaconda3</pre>	206jZ ~]# ls hadoop-2.4.1.tar.gz	: Moving vi	im-8.1.tar.bz2
Anaconda3-2020.02-Linux-x86_64.	sh hello	python基础-张.ipynb 【	[γ1] python训练营-基础知识.ipynb
Editing	Helper	test 未	命名.ipynb
ersion General	hl.c jichu.ipynb	test.c Text,	
(base) [root@iZwz9j14gd1jp0mfm3	206jZ ~]#		
1.8 service iptables 命名	令 。		
如果想管理防火墙,同	可以通过执行命	令:service ipta	ables start 启动防火墙;执行
	细	10	





2.2 查看公钥和私钥内容, 执行 cat id_rsa 和 cat id_rsa.pub 。如图所 示





	ORACLE	订阅 我的帐户	帮助					
	尊敬的 谢晓锋: 要完成 Oracle 账户注册,请先确认你的电子部件地址。							
	要完成 Oracle 統广注册,请先确认愿的电子部件地址。							
	确认电子邮件地址							
	此链接有效期为 24 /	小时。如果按钮已过期,请尝试请求向您发送新的确认电子邮件	•					
	我们收到了一个使用您的电子 或发送电子邮件至 profilehel;	邮件地址创建 Oracle 帐户的请求。如果您没有提出此请求,请提交帮 9_ww@oracle.com。	即助请求					
	谢谢! Oracle 帐户团队							
	Linux x86 RPM Package	171.28 MB	jdk-8u241-linux-i586.rpm					
	Linux x86 Compressed Archive	186.1 MB	أيل -8u241-linux-i586.tar.gz					
	Linux x64 RPM Package	170.65 MB	°↓ jdk-8u241-linux-x64.rpm					
	Linux x64 Compressed Archive	185.53 MB	°↓ jdk-8u241-linux-x64.tar.gz					
	macOS x64	254.06 MB	°₩ jdk-8u241-macosx-x64.dmg					
	Solaris SPARC 64-bit (SVR4 package)	133.01 MB	jdk-8u241-solaris-sparcv9.tar.Z					
jdk 文 (base) (base) (base)	<pre>i.2 在 simple 日來下回建一下 jak 又忤夾, 并把下氧好的安装包工传到 jdk 文件夹, 执行解压命令 (base) [root@iZwz9j14gd1jp0mfm3206jZ ~]# cd simple (base) [root@iZwz9j14gd1jp0mfm3206jZ simple]# mkdir jdk (base) [root@iZwz9j14gd1jp0mfm3206jZ simple]# tar -zxvf jdk-8u241-linux-x64.tar.gz</pre>							
1. 3. 3	日命令 vim /etc/profi base) [root@iZwz9j14gd /e/profile 70 71 fi 72 fi 73 done 74 75 unset i 76 unset -f pathmunge 77 78 PATH=/root/anacondo3 79 export PATH 80 81 # JAVA HOME =/simple 82 export JAVA_HOME =/root 83 export CLASSPATH=S:0 84 export PATH=SPATH:SJ 85 export HADOOP_HOME=/root	ile 配置环境变量 1jp0mfm3206jZ jdk]# vim /dev/null /bin:\$PATH /jdk1.7.0.79 ot/simple/jdk/jdk1.8.0_241 LASSPATH:\$JAVA_HOME/lib/ AVA_HOME/bin root/simple/hadoop-2.4.1	/etc/profile					
	87 export PATH=\$JAVA_HO 88 /etc/profile CWD: /roo "/etc/profile" 88L, 21370	ME/bin:\$HADOOP_HOME/bin:\$HADOOP_HOM t Line: 87 Column: 1	Æ/sbin:\$PATH					







<pre>(base) [root@iZwz9j14gd (base) [root@iZwz9j14gd capacity-scheduler.xml configuration.xsl container-executor.cfg core-site.xml hadoop-env.cmd hadoop-env.sh (base) [root@iZwz9j14gd</pre>	<pre>i1jp0mfm3206jZ output # cd ~/ i1jp0mfm3206jZ hadoop]# ls hadoop-metrics2.properties hadoop-metrics.properties hadoop-policy.xml hadoop-vnv.sh hdfs-site.xml httpfs-env.sh 11jp0mfm3206jZ hadoop]#</pre>	<pre>/simple/hadoop-2.4.1/etc/h httpfs-log4j.properties httpfs-signature.secret httpfs-site.xml log4j.properties mapred-env.cmd mapred-env.sh</pre>	madoop/ mapred-queues.xml.template mapred-site.xml.template slaves ssl-client.xml.example ssl-server.xml.example yarn-env.cmd	yarn-env.sh yarn-site.xml
2.2 在\$HADOOP_ i键之后进入编	HOME/etc/hadoop 辑状态,在文件中	p 目录下执行命 中添加如下内容	で令: vim hadoop :export	-env.sh,按
JAVA_HOME=/roo	ot/simple/jdk1.	8. 0_241。如图	所示	
(base) [root@iZwz9j14gd1 (base) [root@iZwz9j14gd1	.jp0mfm3206jZ ~ <mark> </mark> # cd ~/simpl .jp0mfm3206jZ hadoop]# ls	e/hadoop-2.4.1/etc/hadoop		
capacity-scheduler.xml configuration.xsl container-executor.cfg core-site.xml hadoop-env.cmd hadoop-env.sh (base) [root@iZwz9j14gd1	hadoop-metrics2.properties hadoop-metrics.properties hadoop-policy.xml hadoop-vnv.sh hdfs-site.xml httpfs-env.sh .jp0mfm3206jZ hadoop] / vim h	httpfs-log4j.properties httpfs-signature.secret httpfs-site.xml log4j.properties mapred-env.cmd mapred-env.sh adoop-env.sh	<pre>mapred-queues.xml.template mapred-site.xml.template slaves ssl-client.xml.example ssl-server.xml.example yarn-env.cmd</pre>	e yarn-env.sh yarn-site.xml
+ hadoop-env.sh				
71 72 # The direct 73 # NOTE: this 74 # the 75 # pot 76 export HADO 77 export HADO 78 79 # A string 80 export HADO 81 82 export JAVA ~/simple/hadoop	tory where pid files a s should be set to a d user that will run th ential for a symlink a OP_PID_DIR=\${HADOOP_PI OP_SECURE_DN_PID_DIR=\$ representing this inst OP_IDENT_STRING=\$USER _HOME=/root/simple/jdk -2.4.1/etc/hadoop/hado	re stored. /tmp by d irectory that can on e hadoop daemons. 0 ttack. D_DIR} {HADOOP_PID_DIR} ance of hadoop. \$USE /idk1.8.0_241 op-env.sh[+] CWD:	efault. ly be written to by therwise there is the R by default. /root/simple/hadoop-2.	4.1/etc/hado
2.3 在\$HADOOP_ 后把/etc/hadoo 执行命令:命令	HOME/bin 目录下 p 目录下的所有以 :	首先执行 mkdi 以 xml 结尾的文	r input 创建 inp 件拷贝到 input	ut 文件,然 目录中,之后
Hadoop jar ~/s	imple/hadoop-2	.4.1/share/ha	doop/mapreduce/	/
hadoop-mapredu	ce-examples-2.4	4.1.jar grep	input output 'c	lfs[a-z.]+'
<pre>(base) [root@iZwa (base) [root@iZwa (base) [root@iZwa container-executor hadoop (base) [root@iZwa (base) [root@iZwa . container-exe . hadoop</pre>	29j14gd1jp0mfm3206jZ hada 29j14gd1jp0mfm3206jZ bin 29j14gd1jp0mfm3206jZ bin or hadoop.cmd hdfs.cmd hdfs input 29j14gd1jp0mfm3206jZ bin 29j14gd1jp0mfm3206jZ bin ecutor hadoop.cmd hdfs. hdfs input	<pre>pop] / cd ~/simple/hado # mkdir input # LS mapred rcc mapred.cmd test-con]# cp ~/simple/hadoop-]# ls -a .cmd mapred rcc t mapred.cmd test</pre>	op-2.4.1/bin yarn tainer-executor yarn.cmw 2.4.1/etc/hadoop/*.xml in yarn -container-executor yarn	d nput n n.cmd

<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# hadoop jar ~/simple/hadoop-2.4.1/share/hadoop/mapreduce/hadoop-mapreduce-examp les-2.4.1.jar grep input output 'dfs[a-z.]+' Java HotSpot(TM) 64-Bit Server VM warning: You have loaded library /root/simple/hadoop-2.4.1/lib/native/libhadoop.so.1.0. 0 which might have disabled stack guard. The VM will try to fix the stack guard now. It's highly recommended that you fix the library with 'execstack -c <libfiles', '-z="" it="" link="" noexecstack'.<br="" or="" with="">20/04/03 13:47:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform using builtin-jav a classes where applicable 20/04/03 13:47:17 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id 20/04/03 13:47:17 INFO jym.JymMetrics: Initializing JYM Metrics with processName=JobTracker, sessionId=</libfiles',></pre>
2.4 执行 cd output 密令进行查看,如图所示
<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# ls container-executor hadoop.cmd hdfs.cmd mapred output test-container-executor yarn.cmd hadoop hdfs input mapred.cmd rcc yarn (base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# cd output (base) [root@iZwz9j14gd1jp0mfm3206jZ output]# ls part-r-00000 _SUCCESS (base) [root@iZwz9j14gd1jp0mfm3206jZ output]#</pre>
 【案例 2-4】hadoop 伪分布式配置与测试 一 解压/usr/hadoop/hadoop-2.4.1.tar.gz 压缩包 这部分内容前面已完成 二 配置 hadoop 文件 2.1 切换到\$HADOOP_HOME/etc/hadoop 目录下并查看。如图所示
<pre>(base) [root@ferry ~]# cd ~/simple/hadoop-2.4.1/etc/hadoop/ (base) [root@ferry hadoop]# ls capacity-scheduler.xml hadoop-metrics2.properties httpfs-log4j.properties mapred-queues.xml.template yarn-env.cmd configuration.xsl hadoop-metrics.properties httpfs-signature.secret mapred-site.xml yarn-env.sh container-executor.cfg hadoop-policy.xml httpfs-site.xml mapred-site.xml.template yarn-site.xml core-site.xml hadoop-vnv.sh log4j.properties slaves hadoop-env.cmd hdfs-site.xml mapred-env.cmd ssl-client.xml.example hadoop-env.sh httpfs-env.sh mapred-env.sh ssl-server.xml.example (base) [root@ferry hadoop]#</pre>
 2.2 在\$HADOOP_HOME/etc/hadoop 目录下执行命令: vi hadoop-env.sh,按 i 键之后进 入编辑状态,在文件中添加如下内容: export JAVA_HOME=/simple/jdk1.7.0_79。如图所示
<pre># The directory where pid files are stored. /tmp by default. # NOTE: this should be set to a directory that can only be written to by # the user that will run the hadoop daemons. Otherwise there is the # potential for a symlink attack. export HADOOP_PID_DIR=\${HADOOP_PID_DIR} export HADOOP_SECURE_DN_PID_DIR=\${HADOOP_PID_DIR}J</pre>
<pre># A string representing this instance of hadoop. \$USER by default. export HADOOP_IDENT_STRING=\$USER</pre>
export JAVA_HOME=/root/simple/jdk/jdk1.8.0_241

Г

2.3 在 \$HADOOP_HOME/etc/hadoop 目录下执行 vi core-site.xml 并修改配 置文件 core-site.xml 的 内容如下(主机名查看虚拟机的主机名)。



2.4 在 \$HADOOP_HOME/etc/hadoop 目录下执行 vi hdfs-site.xml 并修改配 置文件 hdfs-site.xml

```
<configuration>
<!-- 指定 HDFS 副本的数量 -->
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/root/simple/Hadoop-2.4.1/hdfs/name</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/root/simple/hadoop-2.4.1/hdfs/data</value>
</property>
<configuration>
```

2.5 在\$HADOOP_HOME/etc/hadoop 目录下查看是否有配置文件 mapred-site.xml。目录下 默 认 情 况 下 没 有 该 文 件 , 可 通 过 执 行 如 下 命 令 : mv mapred-site.xml.template mapred-site.xml 修改 一个文件的命名,然后执行编辑文件命令: vi mapred-site.xml 并修改该文 件内容

<!-- 指定 YARN 的老大(ResourceManager)的地址 --> <property>

<name>yarn.resourcemanager.hostname</name> <value>ferry.szu</value>

<!-- Site specific YARN configuration properties -->

</property>

<!-- reducer 获取数据的方式 -->

<property>

<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>

</property>

</configuration>

2.7 执行 vi /etc/profile。把 hadoop 的安装目录配置到环境变量中。如 图 6 所示 然后让配置文件生效:source /etc/profile。如图所示

export PATH=\$JAVA_HOME/bin:\$PATH
export CLASSPATH=.:\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar

export HAD00P_HOME=/root/simple/hadoop-2.4.1
export PATH=\$JAVA_HOME/bin:\$HAD00P_HOME/bin:\$HAD00P_HOME/sbin:\$PATH
export HAD00P_COMMON_LIB_NATIVE_DIR=\${HAD00P_HOME}/lib/native
export HAD00P_0PTS="-Djava.library.path=\$HAD00P_HOME/lib"

2.8 俗式化 namenode。1	生任意目录下执行如下命令进行格式化: hdfs
namenode -Iormat 以有	nadoop namenode -format 。 如图用小
(base) [root@ferry h	adoop]≰ hadoop namenode -format
2.9 启动 hadoop, 首先	启动 hdfs: start-dfs.sh
(base) [root@ferry hadoop <mark>]# start-dfs.sh</mark> 20/04/25 13:12:39 WARN util.NativeCodeLog	ader: Unable to load native-hadoop library for your platform using builtin-java
classes where applicable	
ferry.szu: namenode running as process 1	3619. Stop it first.
Starting secondary namenodes [0.0.0.0]	/root/stmpte/nadoop-z.4.1/togs/nadoop-root-autanode-rerry.szu.out
0.0.0.0: secondarynamenode running as pro 20/04/25 13:12:49 WARN util.NativeCodeLoo	ocess 15563. Stop it first. ader: Unable to load native-hadoop library for your platform using builtin-java
classes where applicable 2 10 自动 warns start	vorn ch
2.10 /□ 4/J yain: Start (base) [root@ferry hadoop]# start-yar	yarıı. Silo
starting yarn daemons	
localhost: starting nodemanager, logging to	ing to /root/simple/hadoop-2.4.1/logs/yarn-root-resourcemanager-terry.szu.out
启动之后,在任意目录下	执行 jps 命令验证进程是否正常启动。如图所示
(base) [root@ferry	/ hadoop]# jps
11280 NodeManager	
12610 Nemeblede	
15019 Namenoae	
17847 Jps	
15563 SecondaryNam	leNode
10972 ResourceMana	iger i i i i i i i i i i i i i i i i i i i
10972 ResourceMana 20351 DataNode	iger
10972 ResourceMana 20351 DataNode	iger
10972 ResourceMana 20351 DataNode 2.10 测试 hdfs 和 yarn	ger (推荐火狐浏览器),首先在浏览器
10972 ResourceMana 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输入:http	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面)(本
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面)(本 IP, 端口不变)。 如图所示
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datan	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面) (本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress Utilities -
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面) (本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress Utilities -
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面)(本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress Utilities -
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datan Overview 'ferry.	nger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面)(本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress UtilIties - szu:9000' (active)
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar Overview 'ferry.	Iger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47. 113. 102. 106:50070 (HDFS 管理界面)(本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datan Overview 'ferry.	Iger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47.113.102.106:50070 (HDFS 管理界面)(本 IP, 端口不变)。 如图所示 odes Snapshot Startup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020 2.4.1, r1604318
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar Overview 'ferry.	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47. 113. 102. 106: 50070 (HDFS 管理界面) (本 IP, 端口不变)。如图所示 odes Snapshot Startup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020 2.4.1, r1604318 2014-06-21T05:43Z by jenkins from branch-2.4.1
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar Overview 'ferry.	ger (推荐火狐浏览器),首先在浏览器 ://47.113.102.106:50070 (HDFS 管理界面) (本 IP,端口不变)。如图所示 odes Snapshot Starup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020 24.1, r1604318 2014-06-21T05:432 by jenkins from branch-2.4.1 CID-89483fbf-6eb6-4fa7-83d5-7643f5670372
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datan Overview 'ferry.	ger (推 荐 火 狐浏 览 器), 首 先 在 浏览 器 :// 47. 113. 102. 106: 50070 (HDFS 管理界面) (本 IP, 端口不变)。如图所示 odes Snapshot Startup Progress Utilities - Sat Apr 25 12:30:11 CST 2020 2.4.1, r1604318 2014-06-21T05:43Z by jenkins from branch-2.4.1 CID-89483fbf-6eb6-4fa7-83d5-7643f5670372 BP-127175167-172.18.242.188-1586624673125
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar Noverview 'ferry.	lger (推荐火狐浏览器),首先在浏览器 :// 47.113.102.106:50070 (HDFS 管理界面) (本 IP,端口不变)。如图所示 odes Snepshot Startup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020 2.4.1, r1604318 2014-06-21T05:432 by jenkins from branch-2.4.1 CID-89483fbf-6eb6-4fa7-83d5-7643f5670372 BP-127175167-172.18.242.188-1586624673125
10972 ResourceMand 20351 DataNode 2.10 测试 hdfs 和 yarn 地 址栏中 输 入 : http IP 为自己虚拟机上面的 Hadoop Overview Datar Overview 'ferry. Started: Version: Compiled: Cluster ID: Block Pool ID:	Iger (推荐火狐浏览器),首先在浏览器 :// 47.113.102.106:50070 (HDFS 管理界面) (本 IP,端口不变)。如图所示 odes Snapshot Startup Progress Utilities - szu:9000' (active) Sat Apr 25 12:30:11 CST 2020 2.41, r1604318 2014-06-21T05:432 by jenkins from branch-2.4.1 CID-89483fbf-6eb6-4fa7-83d5-7643f5670372 BP-127175167-172.18.242.188-1586624673125

在浏览器的:	地址栏中输入: http:// 47.113.102.106:8088 (MR 管理界面)
(本 IP 为	自 己虚拟机上面的 IP,端口不变)。如图所示
\$hed	All Applications
- Cluster	Cluster Metrics
About Nodes Applications NEW	Apps Apps Apps Apps Containers Memory Memory Memory Active Decommissioned Lost U Submitted Pending Running Completed Running Used Total Reserved Nodes Nodes Nodes Nodes Nodes 0 <td< td=""></td<>
NEW_SAVING SUBMITTED ACCEPTED	Show 20 ; entries Sear
RUNNING FINISHED FAILED	The second seco
KILLED Scheduler	Showing 0 to 0 of 0 entries First F
这里值得说	明的是,由于使用的是阿里云服务器,所以要在后台管理处添加
端口安全组	,如图所示:
安全组内实例列表 安全组内弹性网卡	访问规则
	入方向 出方向
	手动波加 快速渐加 全部编辑
	授权策略 优先级 ○ 协议类型 第口范围 ○ 授权对象 ○ 描述 操作
	允许 1 自定义 TCP 目約:8088/8088 源:0.0.0.0/0 編輯 复制 删除
	允许 1 自定义 TCP 目約:50070/50070 源:0.0.0.0/0 编辑 重制 副除

实验结论与分析:

1. 粘贴最终结果图,并对结果做分析说明

【案例 1-1】Linux 设置静态 IP 地址

采用修改配置文件的方式来设置静态 IP。首先注意,路由网关处必须设置可以配置静态 IP或者使用指定 IP绑定 MAC 才可以配置,否则配置不成功。网络配置的配置文件在/etc/sysconfig/network-scripts/下,可以使用 if config 查看。这时候如果想修改成静态的,首先把 BOOTPROTO="dhcp"改 BOOTPROTO="static"表示静态获取,然后再修改下面的配置: BROADCAST 设置的是局域网广播地址, IPADDR 就是静态 IP, NETMASK 是子网掩码, GATEWAY 就是网关或者路由地址;需要说明,原来还有个 NETWORK 配置的是局域网网 络号,这个是 if calc 自动计算的,所以这里配置这些就足够了,最终配置如 下图:

+ /0	e/s/n/ifcfg-eth0
1	DEVICE=eth0
2	BOOTPROTO=static
3	ONBOOT=yes
4	IPADDR=172.18.242.188
5	GATEWAY=172.18.255.253
6	NETMASK=255.255.240.0

【案例 1-2】修改主机名和映射关系

Linux 系统安装好后,都会有默认的主机名,这里以 CentOS 系统为例,为了

便于使用,我们常常需要修改主机名,主要步骤为修改 network 文件,重启服务器,最后建立 IP 和 hostname 映射,建立 IP 和 hostname 映射关系很简单,只需要修改一个文件(/etc/hosts)即可。我将主机名设置为 ferry.szu,这里由于 shell 显示长短的原因只显示了 ferry。



深圳大学学生实验报告用纸

据命令的不同,参数分为可选的或必须的;所有的命令从标准输入接受输入,

命令	参数	示例	说明
cd		cd /home	切换目录
pwd		pwd	显示当前工作目录目录
touch	h touch 1.txt		创建空文件
mkdir	kdir mkdir testdir		创建一个新目录
	-р	mkidr -p dir1/dir2/dir3/	创建多级目录,父目录不存在情况下先生成父目录
cp		cp 1.txt	复制文件或目录
	-f	cp -r dir1/	递归处理,将指定目录下的文件与子目录一并拷贝
mv		mv dir1 dir2	移动文件或目录、文件或目录改名
rm		rm 1.txt	删除文件
	-r -f	rm -rf dir1	r同时删除该目录下的所有文件, 例a制删除文件或目录
rmdir	mdir rmdir dir1		删除空目录
命令	翻	改 示例	说即
top		top	显示当前系统中耗费资源最多的进程
date		date	显示系统当前时间
ps			较少单独使用,配参数根据需求,ps -ef 或者ps-aux
	-е /-А	ps -e	显示所有进程,环境变量
	-f	ps -ef	全格式显示
	-a	ps -a	显示所有用户的所有进程(包括其它用户)
	-u	ps -au	按用户名和启动时间的顺序来显示进程
	-x	ps -aux	显示无控制终端的进程
kill	-9	kill -9 pid	强制杀死一个进程
df		df	显示文件系统磁盘空间的使用情况
	-h	df -h	以人类可读的方式显示,Kb,Mb,GB等

输出 结果显示在标准输出, 而错误信息则显示在标准错误输出设备。可使用

【案例 1-5】ssh 免密码配置

假如我们现在有两台机器: ServerA 和 ServerB,现在想要让 ServerA 不用输入密码就能够进行访问。我们使用 ssh-keygen 在 ServerA 上生成 private 和 public 密钥,将生成的 public 密钥拷贝到远程机器 ServerB 上后,就可以使用 ssh 命令无需密码登录到另外一台机器 ServerB 上。在 linux 系统中, ssh 是远程登录的默认工具,因为该工具的协议使用了 RSA/DSA 的加密算法【默认 是 DSR 算法】,该工具做 linux 系统的远程管理是非常安全的。最终完成 ssh 免密码配置:

(base) [root@ferry .ssh]# ssh ferry.szu Last login: Sat Apr 25 10:38:31 2020 from 223.74.231.76

Welcome to Alibaba Cloud Elastic Compute Service !

(base) [root@ferry ~]#

【案例 2-1】linux 系统 JDK 的安装和配置

Linux 下 jdk 主要有两种安装方式; 1、yum 安装; 2、从官网下载包安装 使用 yum 安装是非常简单的,但是安装路径是系统默认的,不利于后面对 Hadoop 的配置。所以这里选择从 oracle 官网下载 java 安装包手动安装,这 里容易出错的地方是对 jdk 的配置,要根据安装路径进行修改,要记住 source /etc/profile 命令刷新配置,最后用 java -version 命令可以检查 是否安装成功,这里可以看到安装成功了,版本是 1.8.0

(base) [root@iZwz9j14gd1jp0mfm3206jZ ~] # source /etc/profile (base) [root@iZwz9j14gd1jp0mfm3206jZ ~] # java -version java version "1.8.0_241" Java(TM) SE Runtime Environment (build 1.8.0_241-b07) Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode) (base) [root@iZwz9j14gd1jp0mfm3206jZ ~]#

【案例 2-2】 linux 系统下 HelloWorld

到这一步,安装了完整的 JDK,可以通过手动建立和编译源文件,跑一次 HelloWorld,看看 JDK 安装正确与否。在路径下建立文件 hello.java,用命 令行先 touch hello.java,然后 vi hello.java,进入到 hello.java 的编辑, 编写 hello.java 程序,按照上面安装 jdk 一步步做完,那么 javac 编译 hello.java 是可以通过的,在 hello.java 相同的路径下会得到 hello.class 文件,注意这里对 jdk 的配置需要把路径写全,不然会出错,而且虽然 java 字节码文件是.class 后缀的,但是执行时不需要,这点要注意,执行后,输 出如下:

(base) [root@ferry jdk]# java HelloWorld HelloWorld

【案例 2-3】hadoop 单机分布式配置与测试(必做)

Hadoop 是大数据处理主流框架,如果要学习大数据处理与存储,hadoop 是必须掌握的一种框架技术。动手安装是最基础的实践。首先要注意的是:在安装 hadoop 之前需要先完成以下一些设置 1.静态 ip 设置 2.设置 ip 连接用户名 3. ssh 免密登陆,这些我们前面已经全部完成。其次容易出错的地方是在修改系统环境变量配置文件,将 HADOOP 的路径加入 profile 时,这里要主要路径要与安装路径一致。安装完成后,可以通过 hadoop version 命令测试 hadoop 是否安装成功。

这里安装及配置表明本机可以当做 hadoop 服务器处理程序,也即是单机运行,不是分布式的,map 和 reduce 使用同一进程不同阶段来执行。Hadoop 可以在单节点上以伪分布式的方式运行,Hadoop 进程以分离的 Java 进程来运行,节点既作为 NameNode 也作为 DataNode,同时,读取的是 HDFS 中的文件。下图命令中 input 为输入的文件夹,将处理 input 中的所有 xml 文件, output 为输出结果文件夹。

<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# hadoop jar ~/simple/hadoop-2.4.1/share/hadoop/mapreduce/hadoop-mapreduce-examp les-2.4.1.jar grep input output 'dfs[a-z.]+' Java HotSpot(TM) 64-Bit Server VM warning: You have loaded library /root/simple/hadoop-2.4.1/lib/native/libhadoop.so.1.0. 0 which might have disabled stack guard. The VM will try to fix the stack guard now. It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'. 20/04/03 13:47:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform using builtin-jav a classes where applicable 20/04/03 13:47:17 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id 20/04/03 13:47:17 INFO intializing JVM Metrics with processName=JobTracker, sessionId=</libfile></pre>
通过 cd output 密令进行查看,下图说明成功了
<pre>(base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# ls container-executor hadoop.cmd hdfs.cmd mapred output test-container-executor yarn.cmd hadoop hdfs input mapred.cmd rcc yarn (base) [root@iZwz9j14gd1jp0mfm3206jZ bin]# cd output (base) [root@iZwz9j14gd1jp0mfm3206jZ output]# ls part-r-00000 _SUCCESS (base) [root@iZwz9j14gd1jp0mfm3206jZ output]#</pre>
【案例 2-4】hadoop 伪分布式配置与测试 伪分布式配置,主要需要修改配置文件然后启动相应的集群服务。包括对 5 个配置文件修改,配置文件修改并保存后,就可以开始启动 Hadoop 了。首先 进行 hadoop 的初始化,然后启动所有进程,可通过 jps 命令查看进程状态
(base) [root@ferry hadoop]# jps
11280 NodeManager
13619 NameNode 17847 Jps
15563 SecondaryNameNode
10972 ResourceManager
20351 DataNode

至此,单机上伪分布式 Hadoop 环境配置好了,一台机器上配置 Hadoop 集群: NameNode、DataNode、SecondaryNameNode, YARN 集群: ResourceManager、 NodeManager。Hadoop 集群负责分布式存储和预算, YARN 集群负责任务资源 管理和调度。

可以在外部浏览器登录 hadoop,由于是阿里云服务器,没有图形界面,因此 无法从服务器本机测试,需要外部浏览器测试。首先要确定防火墙关闭,然 后 hadoop 开始采用端口为:50070。因此在外部浏览器地址栏输入: http://47.113.102.106:50070,这里 47.113.102.106 是服务器的 ip 地址, 50070 为 hadoop 所用端口。通过这个 web 界面浏览整个集群的状态。

Hadoop Ov	erview Datanodes	Snapshot	Startup Progress	Utilities -				
Overvie	Overview 'ferry.szu:9000' (active)							
Started:	Started: Sat Apr 25 12:30:11 CST 2020							
Version:	Version: 2.4.1, r1604318							
Compiled:	Compiled: 2014–06–21T05:43Z by jenkins from branch–2.4.1							
Cluster ID:	Cluster ID: CID-89483fbf-6eb6-4fa7-83d5-7643f5670372							
Block Pool ID:	BP-1	BP-127175167-172.18.242.188-1586624673125						

Summary

2. 谈谈自己对这个实验的体会感受

通过这次实验,我对 linux 和 Hadoop 有了基本的了解。本来以为对照着实 验手册操作会很容易,但实际上在实验过程中遇到了很多很多的问题,这次 的实验关键是运行环境的设置,只要一个地方出错了那么就不能运行,所以 做每一步都要很仔细认真,但最后通过查找资料和想助教老师请教,基本解 决了问题。总的来说,由于我使用的是阿里云服务器,和实验手册有些出入, 而且为了把实验报告做好,这次实验真的花了很多时间和精力,但也学习到 了很多!对大数据方向也越来越有兴趣了!

指导教师批阅意见:

成绩评定: