03

第3章大数据实验环境搭建

授课教师：[李阳源]

ywlee1212@gmail.com

主页：https://sites.google.com/view/dalguri



SL-01:提纲

3.1 Linux系统的安装
3.2 Hadoop的安装
3.3 MySQL的安装
3.4 Kafka的安装
3.5 Anaconda的安装和使用方法





3.1 Linux系统的安装



3.1.1 下载安装文件
3.1.2 Linux系统的安装方式
3.1.3 安装Linux虚拟机

3.1.1 下载安装文件



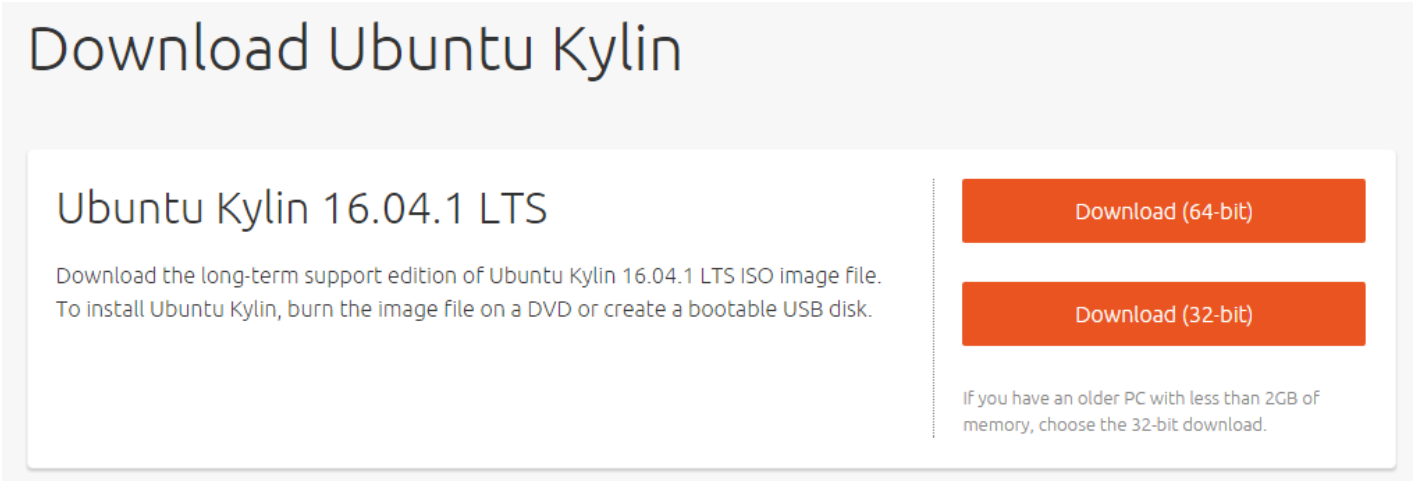
本教程采用的Linux发行版是Ubuntu，同时，为了更好支持汉化（比如，更容易输入中文），这里采用了Ubuntu Kylin发行版。Ubuntu Kylin是针对中国用户定制的Ubuntu发行版本，里面包含了一些中国特色的软件（比如中文拼音输入法），并根据中国人的使用习惯，对系统做了一些优化。

Ubuntu Kylin较新的版本是22.04 LTS，但是，在实际使用过程中发现，该版本对计算机的资源消耗较多，在使用虚拟机方式安装时，系统运行起来速度较慢。因此，本教程选择较低的版本Ubuntu Kylin 16.04 LTS，这个版本不仅降低了对计算机配置的要求，也可以保证大数据各种软件的顺利安装和运行，帮助读者很好完成本教程的各个实验。

3.1.1 下载安装文件



可以通过两种途径下载Ubuntu Kylin发行版的安装映像文件，第一种方式是进入Windows系统，访问Ubuntu官网下载（https://www.ubuntu.org.cn/download/ubuntu-kylin）。进入Ubuntu官网的下载页面后（如图2-1所示），会提供两种不同版本的安装映像文件的下载地址，即32位版本和64位版本。如果计算机硬件配置较低，内存小于2GB，建议选择32位版本；如果内存大于4GB，建议选择64位版本。



3.1.2 Linux系统的安装方式



Linux系统的安装主要有两种方式：虚拟机安装方式和双系统安装方式。

- （1）**虚拟机安装方式**：在Windows操作系统上安装虚拟机软件（比如VirtualBox或VMware），然后，在虚拟机软件之上安装Linux系统。采用这种安装方式时，Linux系统就相当于运行在Windows上的一个软件，如果要使用Linux系统，需要在电脑开机后，首先启动进入Windows操作系统，然后，在Windows操作系统中打开虚拟机软件（比如VirtualBox），最后，在虚拟机软件中启动Linux，之后才能使用Linux系统。

- （2）**双系统安装方式**：直接把Linux系统安装在电脑“裸机”上，而不是安装在Windows系统之上。采用这种安装方式时，Linux系统和Windows系统的地位是平等的，当电脑开机时，屏幕上会显示提示信息，让用户选择要启动的系统，如果用户选择Windows系统，电脑就继续启动进入Windows系统，如果用户选择Linux系统，电脑就继续启动进入Linux系统。

3.1.3 安装Linux虚拟机



1.安装虚拟机软件

常用的虚拟机软件包括VMware和VirtualBox等，VirtualBox属于开源软件，免费，VMware属于商业软件，需要付费。从易用性的角度，VMware要比VirtualBox更胜一筹，因此，本教程采用VMware，可以访问VMware官网（https://www.vmware.com/cn/products/workstation-pro.html）下载安装文件，或者也可以到本教程官网下载，位于“下载专区”的“软件”目录下，文件名是VMware-workstation-full-17.0.1.exe。下载后，请在Windows操作系统中安装VMware。

3.1.3 安装Linux虚拟机



2.安装Linux操作系统

请进入Windows系统，启动VMware软件，按照以下两大步骤完成Linux系统的安装：首先需要创建一个虚拟机，然后，在虚拟机上安装Linux系统。

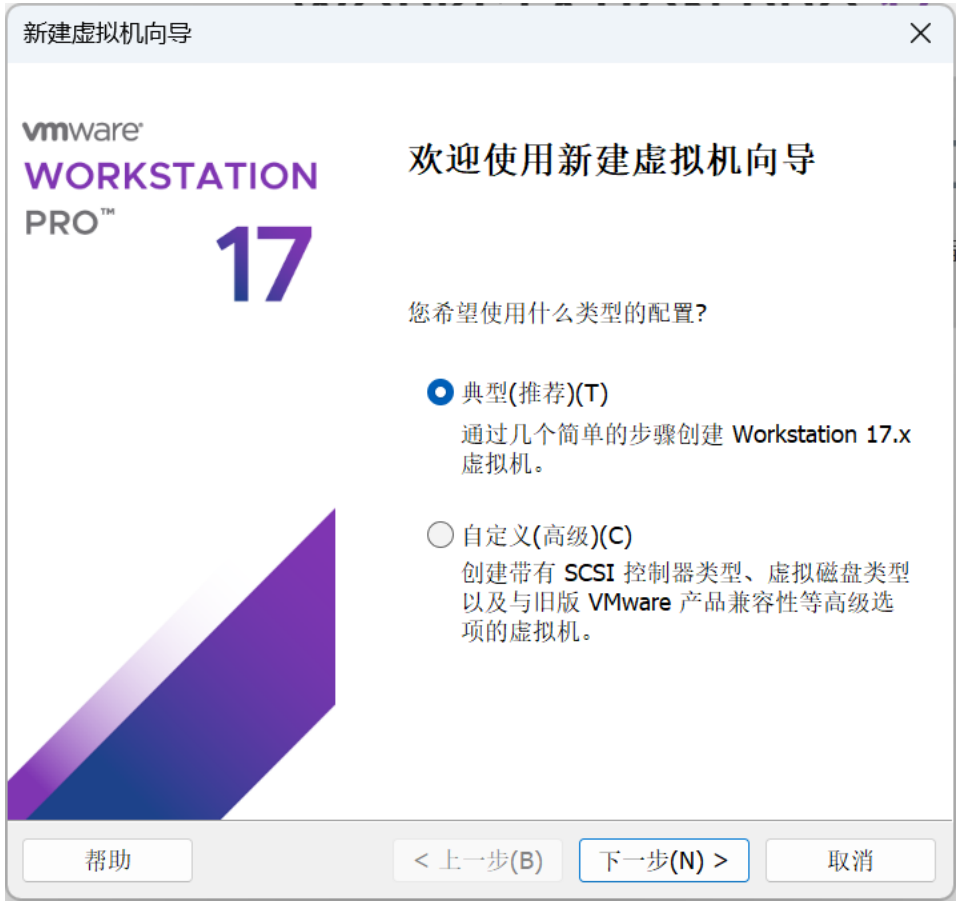
打开VMware，在“主页”选项卡中，可以看到如图2-2所示的界面，点击“创建新的虚拟机”。



3.1.3 安装Linux虚拟机



在出现的界面中（如图所示），选择“典型(推荐)(T)”。



3.1.3 安装Linux虚拟机



在出现的界面中（如图3-4所示），选中“安装程序光盘映像文件(iso)(M)”，点击“浏览”按钮，在弹出的界面中（如图3-5所示），找到之前已经准备好的Ubuntu安装映像文件ubuntukylin-16.04-desktop-amd64.iso，点击“打开”把它加入进来，然后点击“下一步”。

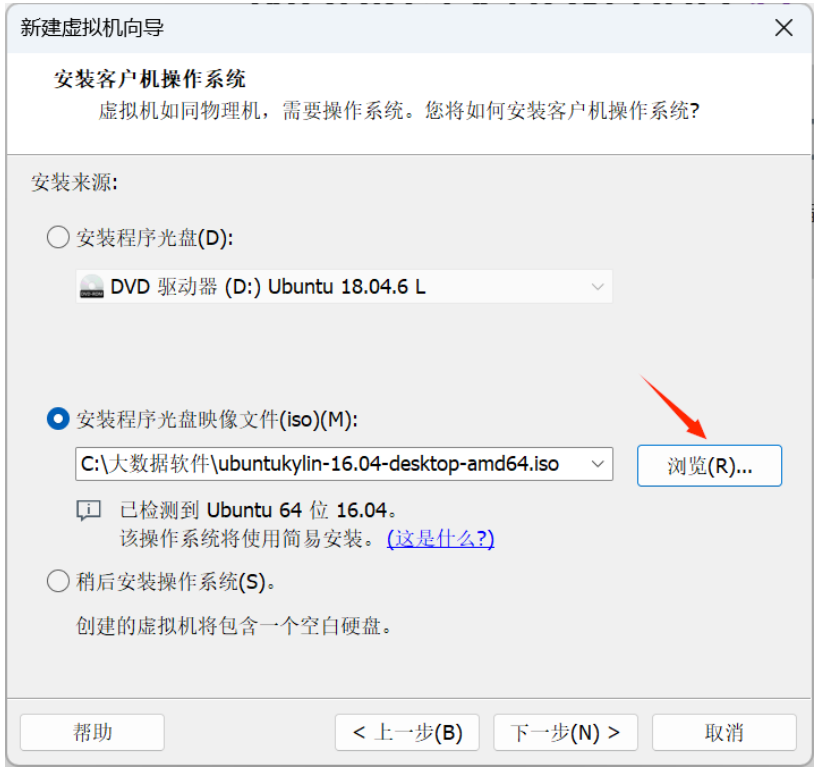


图3-4 安装程序光盘映像文件

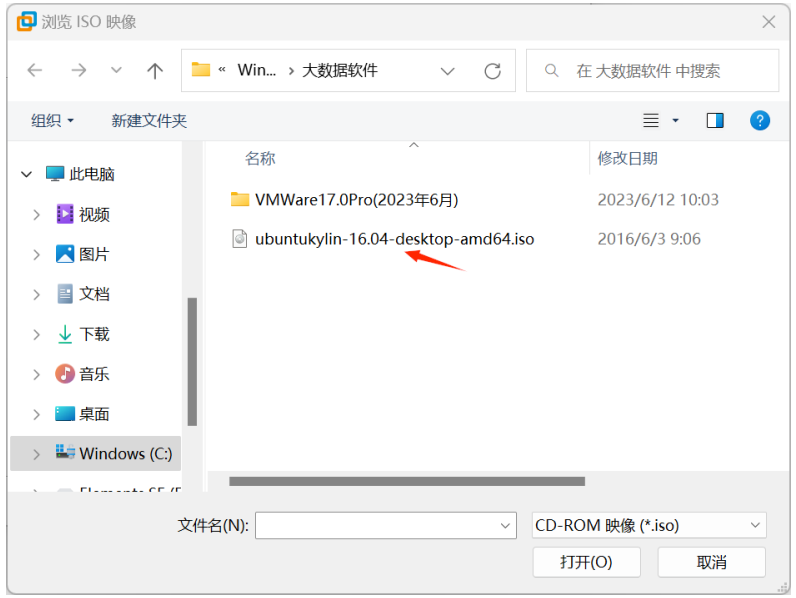


图3-5 打开映像文件

3.1.3 安装Linux虚拟机



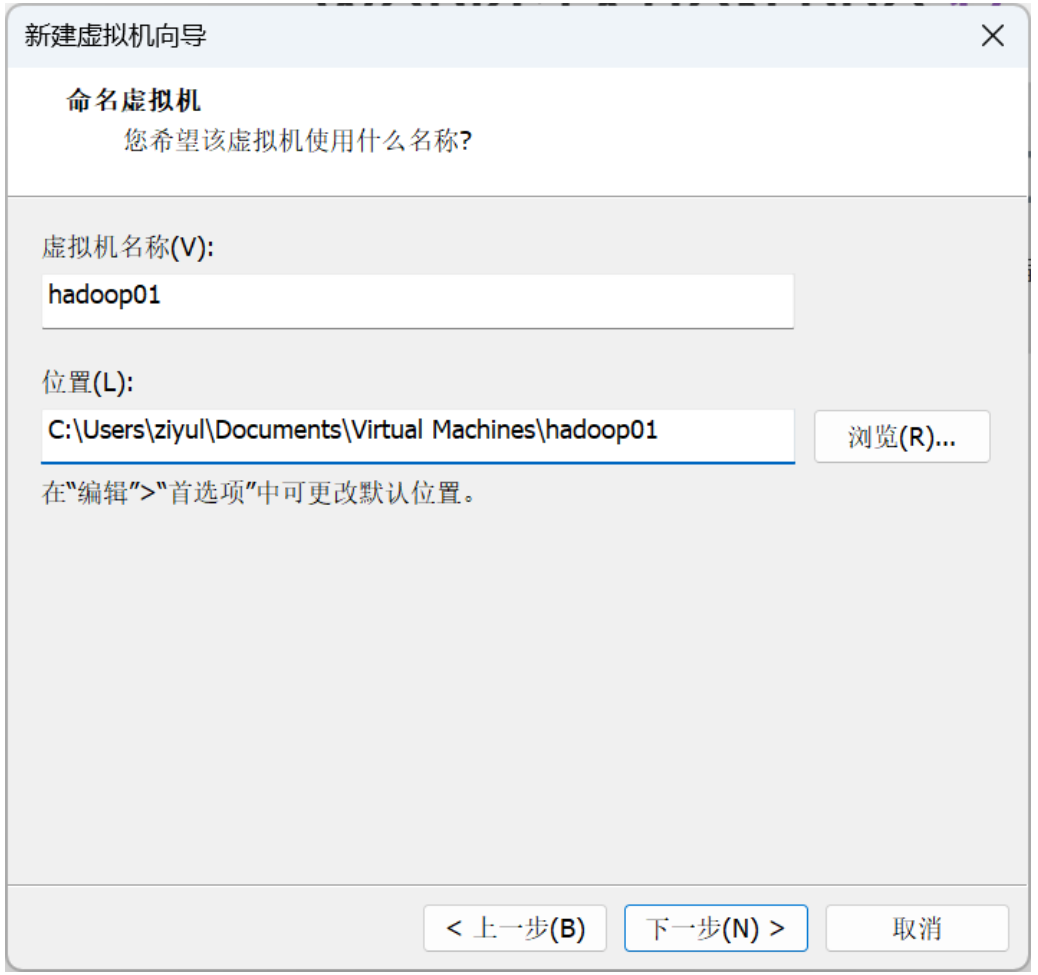
在弹出的界面中（如图3-6所示），进行个性化Linux设置，比如，可以把“全名”设置为“xmudblab”，“用户名”设置为“dblab”，“密码”设置为“123456”。然后，点击“下一步”。



3.1.3 安装Linux虚拟机



在弹出的界面中（如图3-7所示），设置“虚拟机名称”为“hadoop01”，并设置虚拟机文件保存位置。然后点击“下一步”。



3.1.3 安装Linux虚拟机



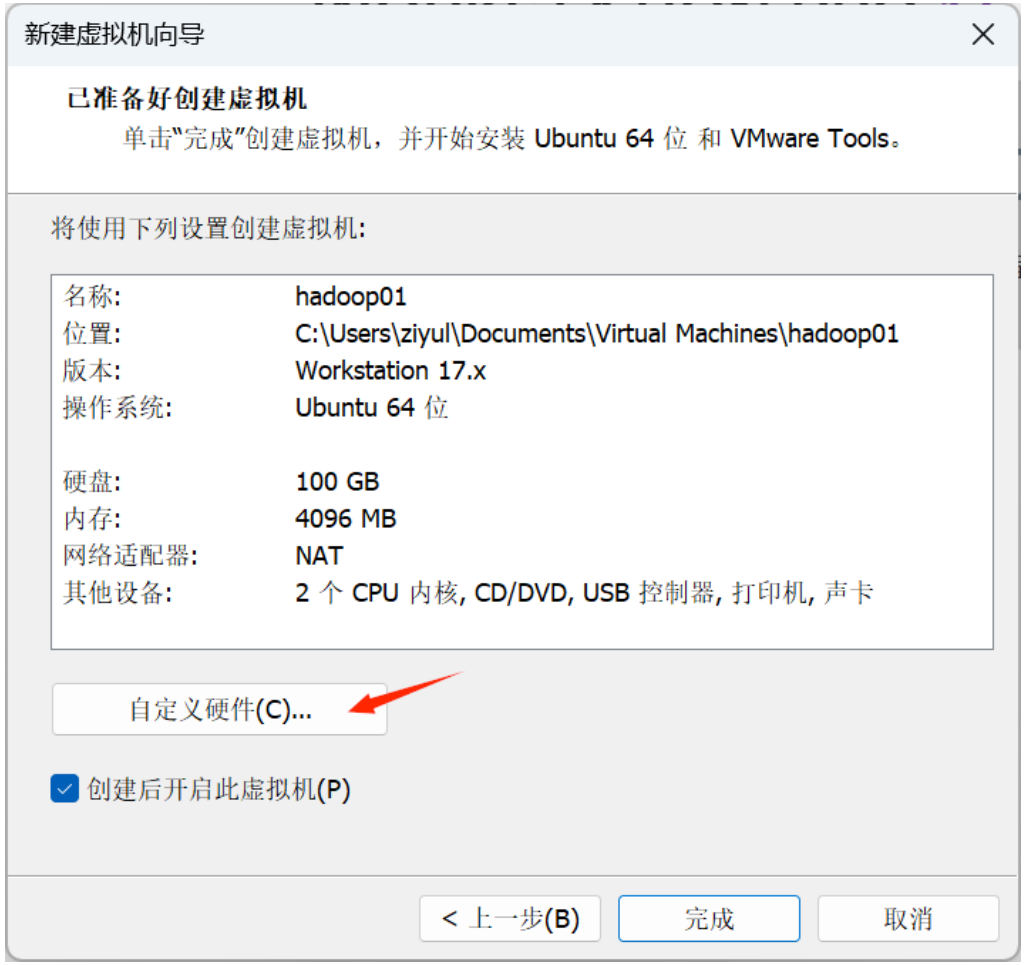
在弹出的界面中（如图3-8所示），对磁盘大小进行设置。一般而言，开展大数据实验，需要在虚拟机中安装各种大数据软件，至少需要消耗40GB磁盘空间，因此，建议把磁盘空间设置为50GB~100GB。同时，需要把“将虚拟磁盘存储为单个文件(O)”选中。然后，点击“下一步”。



3.1.3 安装Linux虚拟机



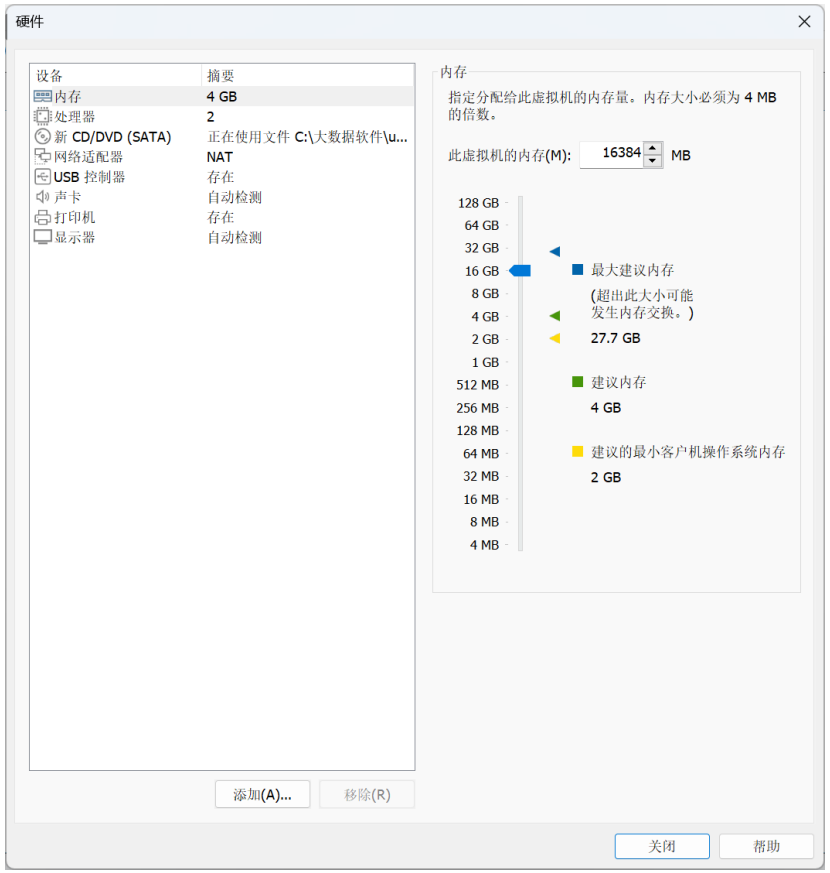
在弹出的界面中（如图3-9所示），点击“自定义硬件”按钮，对内存大小进行设置。



3.1.3 安装Linux虚拟机



在弹出的界面中（如图3-10所示），对虚拟机的内存进行设置，比如，如果计算机的内存有32GB，则可以把虚拟机内存设置为16GB，最小要设置为4GB。然后，点击“关闭”，返回到图2-9所示界面，点击“完成”。这时，VMWare就会开始自动安装Ubuntu系统。



3.1.3 安装Linux虚拟机



系统安装完成以后，会出现如图3-11所示的登录界面，点击“xmudblab”用户，输入密码，就可以登录Ubuntu系统了，登录后系统界面如图3-12所示。

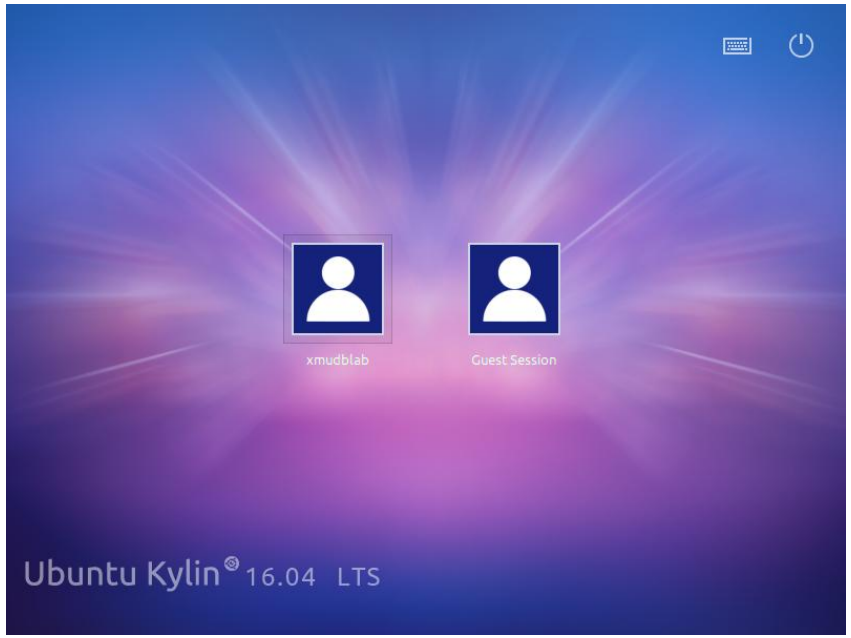


图3-11 Ubuntu登录界面



图3-12 Ubuntu系统界面

3.1.3 安装Linux虚拟机



在系统界面底部的一排按钮中，
有一个最经常被使用的按钮是“终端”（Terminal）按钮（如图3-13所示），
点击这个按钮，就可以新建一个Linux终端，
也就是Linux Shell环境。当然，也可以使用组合键“**CTRL+ALT+T**”新建一个终端。

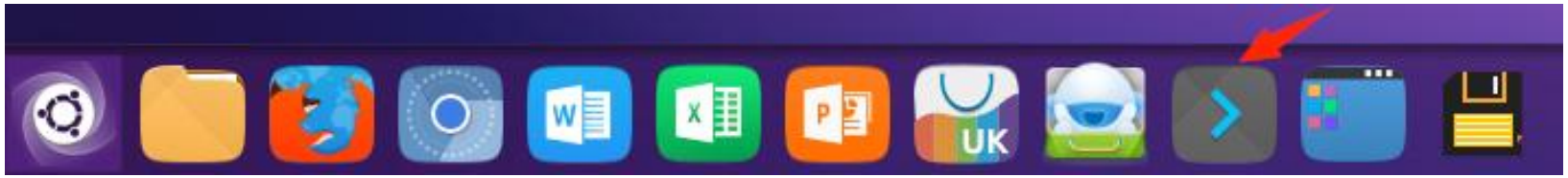


图3-13 终端按钮

3.1.3 安装Linux虚拟机



在系统界面底部的一排按钮中，
另一个经常被使用的按钮是“文件夹”按钮（如图3-14所示），
点击这个按钮，就可以打开文件夹管理界面（如图3-15所示），
就可以像Windows系统一样进行各种操作。
VMWare支持Windows系统和Linux虚拟机之间的双向复制，
可以把Windows系统中的文件或文字直接复制粘贴到Linux虚拟机中，
也可以把Linux虚拟机中的文件或文字复制粘贴到Windows系统中。



图3-14 文件夹按钮

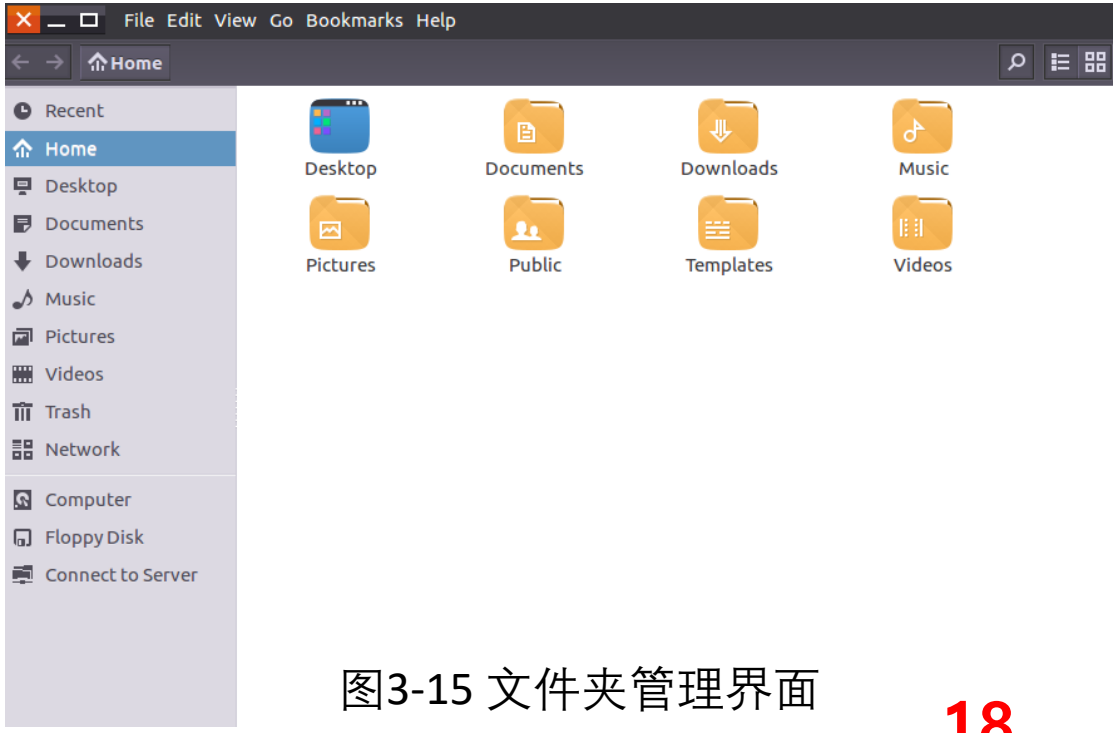


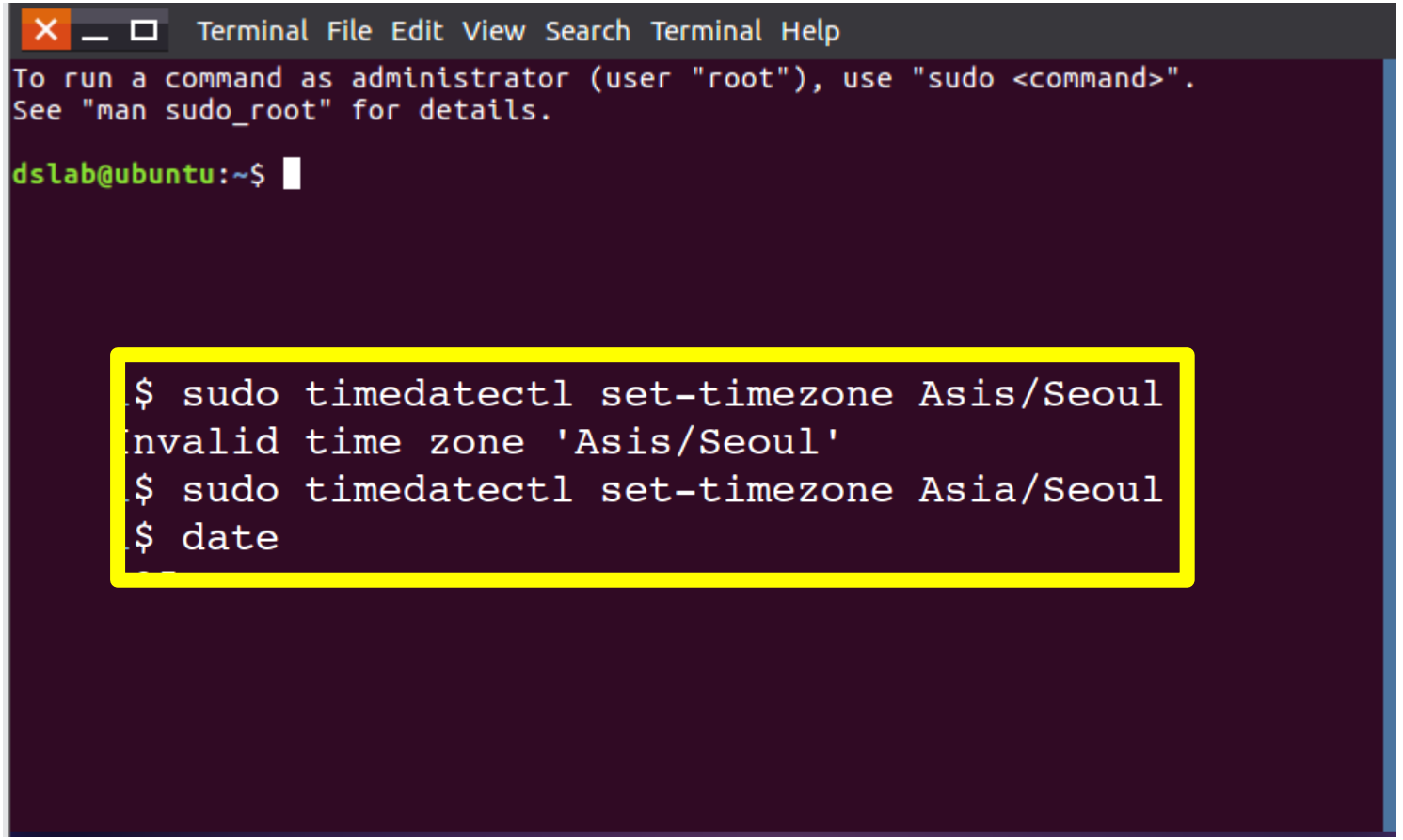
图3-15 文件夹管理界面

Time setting



timedatectl list-timezones | grep Asia
sudo timedatectl set-timezone Asia/Seoul

sudo timedatectl set-timezone Asia/Shanghai



3.1.3 安装Linux虚拟机



当需要退出登录或关机时，可以点击系统界面右上角的齿轮图标，会弹出如图3-16所示的界面，若点击“Log Out...”就可以退出当前登录，若点击“Shut Down...”就可以关机，也就是关闭Ubuntu系统，而不是关闭Windows系统。

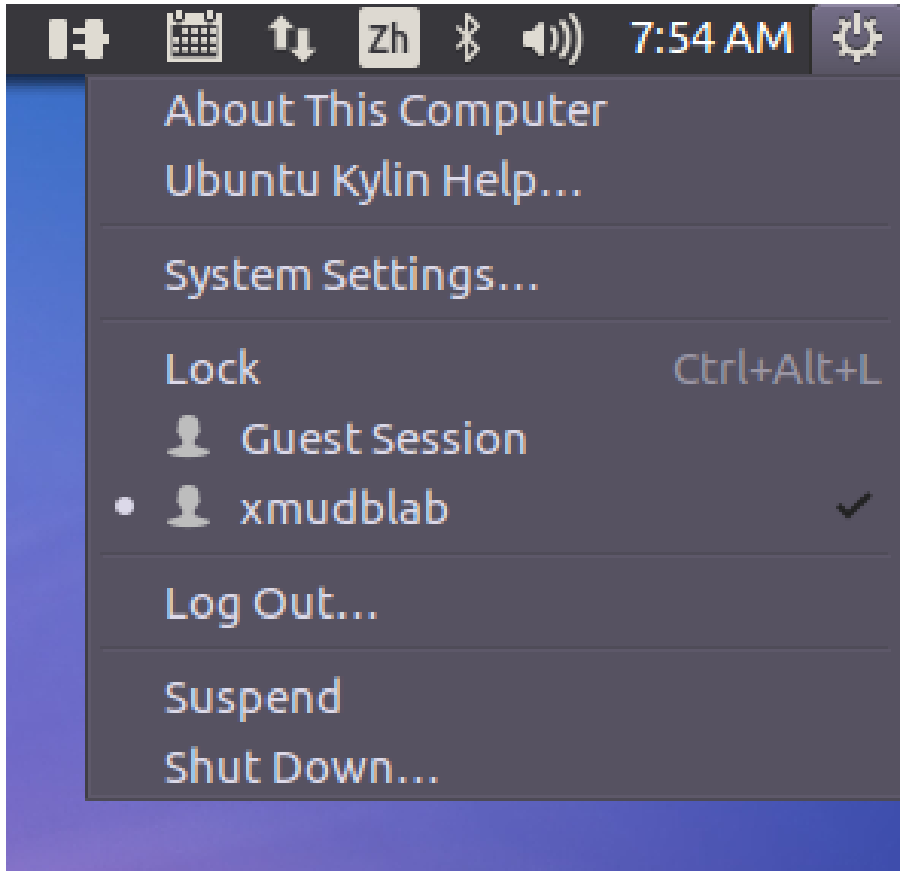


图3-16退出登录或关机界面

3.2 Hadoop的安装

3.2.1 Hadoop简介
3.2.2 安装Hadoop前的准备工作
3.2.3 安装Hadoop的三种模式
3.2.4 下载Hadoop安装文件
3.2.5 单机模式配置
3.2.6 伪分布式模式配置
3.2.7 分布式模式配置



3.2.1 Hadoop简介



- Hadoop是Apache软件基金会旗下的一个开源分布式计算平台，为用户提供了系统底层细节透明的分布式基础架构。Hadoop是基于Java语言开发的，具有很好的跨平台特性，并且可以部署在廉价的计算机集群中。Hadoop的核心是分布式文件系统（Hadoop Distributed File System，HDFS）和MapReduce。

- Apache Hadoop版本分为三代，分别是Hadoop 1.0、Hadoop 2.0和Hadoop3.0。

- 除了免费开源的Apache Hadoop以外，还有一些商业公司推出Hadoop的发行版。2008年，Cloudera成为第一个Hadoop商业化公司，并在2009年推出第一个Hadoop发行版。此后，很多大公司也加入了做Hadoop产品化的行列，比如MapR、Hortonworks、星环等。2018年10月，Cloudera和Hortonworks宣布合并。一般而言，商业化公司推出的Hadoop发行版也是以Apache Hadoop为基础，但是前者比后者具有更好的易用性、更多的功能以及更高的性能。

CLOUDERA Hortonworks

3.2.2 安装Hadoop前的准备工作



1. 创建hadoop用户

本课程需要创建一个名称为hadoop的普通用户，后续所有操作都会使用该用户名登录到Linux系统。上面在安装Linux系统时，设置了一个名称为xmudblab的用户，现在就可以使用xmudblab用户登录Linux 系统，
然后，打开一个终端（可以使用快捷键 Ctrl+Alt+T），使用如下命令创建一个用户hadoop：

```
$ sudo useradd -m hadoop -s /bin/bash
```

这条命令创建了可以登陆的 hadoop 用户，并使用 /bin/bash 作为 Shell。
接着使用如下命令为hadoop用户设置密码：

```
$ sudo passwd hadoop
```

然后，可为hadoop用户增加管理员权限，以方便部署，避免一些对于新手来说比较棘手的权限问题，命令如下：

```
$ sudo adduser hadoop sudo
```



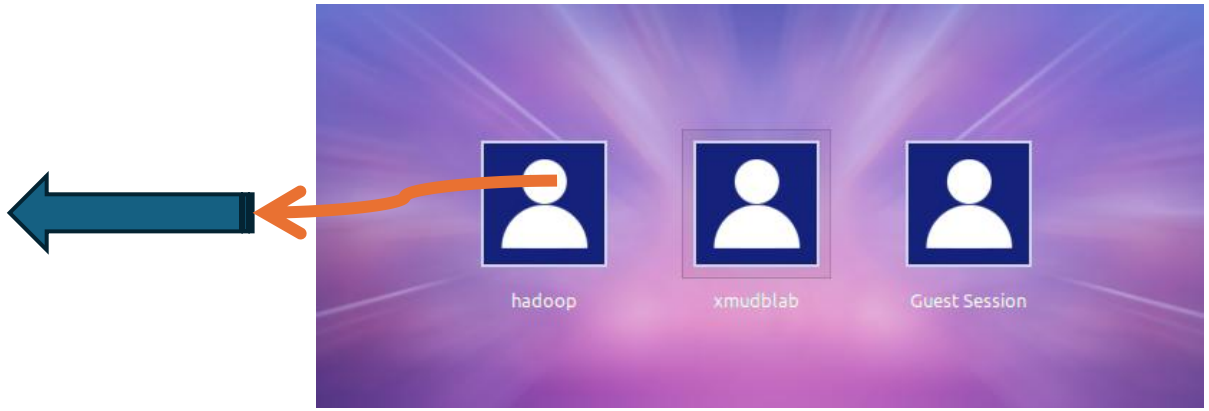
터미널 종료후에 Hadoop 사용자로 다시 로그인 할것
终端退出后，请重新以Hadoop用户身份登录

다시 로그인후 Screen saver Inactivation



防止屏幕关闭:

禁用屏幕保护程序:



禁用屏幕锁定:

```
Ubuntu Kylin Desktop
Terminal File Edit View Search Terminal Help
dblab@ubuntu:~$ gsettings set org.gnome.desktop.session idle-delay 0
dblab@ubuntu:~$ gsettings set org.gnome.desktop.screensaver idle-activation-enabled false
dblab@ubuntu:~$ gsettings set org.gnome.desktop.screensaver lock-enabled false
dblab@ubuntu:~$
```

3.2.2 安装Hadoop前的准备工作



2. 更新APT

为了确保Hadoop安装过程顺利进行，建议在Linux终端中执行下面命令更新APT软件：

```
$ sudo apt-get update
```



3.2.2 安装Hadoop前的准备工作



3. 安装SSH

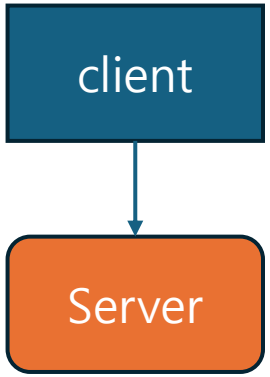
Ubuntu默认已安装了SSH客户端，因此，这里还需要安装SSH服务端，请在Linux的终端中执行以下命令：

```
$ sudo apt-get install openssh-server
```



安装后，可以使用如下命令登录本机：

```
$ ssh localhost
```



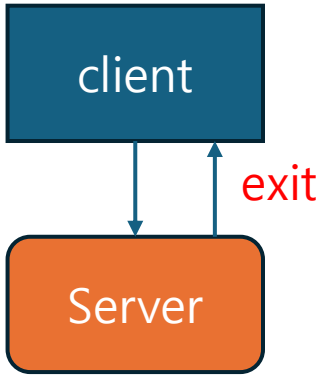
执行该命令后会出现如图3-17所示的提示信息(SSH首次登录提示)，输入“yes”，然后按提示输入密码hadoop，就登录到本机了。

```
hadoop@DBLab-XMU:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is a9:28:e0:4e:89:40:a4:cd:75:8f:0b:8b:57:79:67:86.
Are you sure you want to continue connecting (yes/no)? yes
```

3.2.2 安装Hadoop前的准备工作

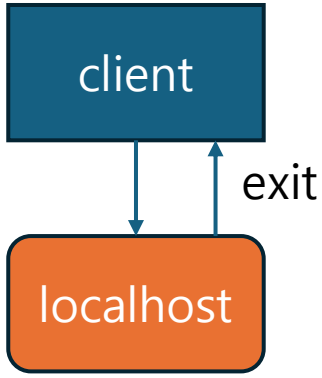


首先，请输入命令“exit”退出刚才的SSH，就回到了原先的终端窗口；
然后，可以利用ssh-keygen生成密钥，并将密钥加入到授权中，命令如下：




```
$ cd ~/.ssh/          # 若没有该目录，请先执行一次ssh localhost
$ ssh-keygen -t rsa     # 会有提示，都按回车即可
$ cat ./id_rsa.pub >> ./authorized_keys  # 加入授权
```

此时，再执行ssh localhost命令，无需输入密码就可以直接登录了，如图所示。

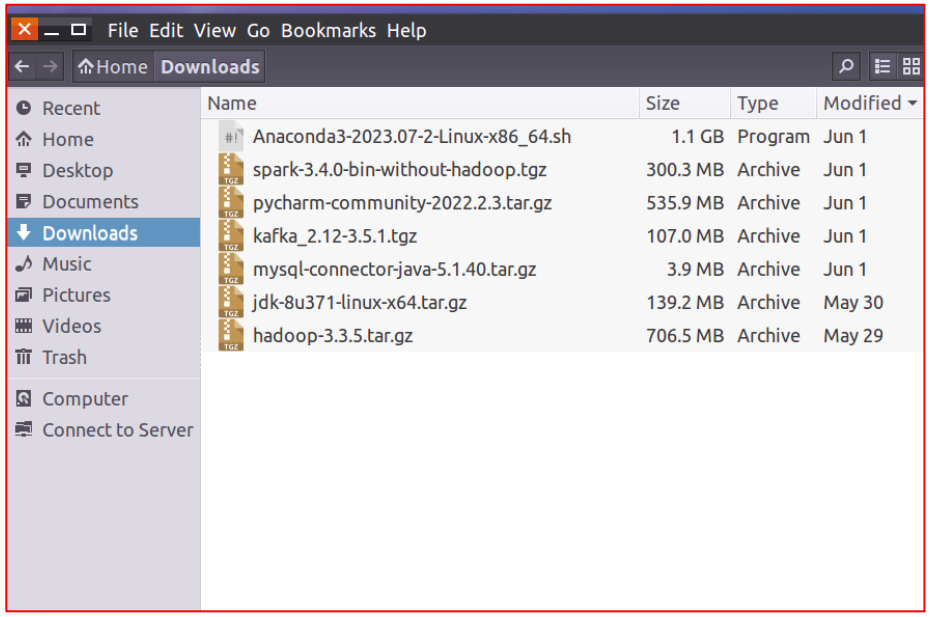


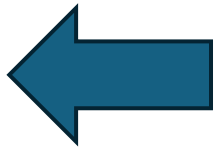

```
hadoop@ubuntu:~$ ssh localhost
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

31 packages can be updated.
2 updates are security updates.

Last login: Mon Jun 12 00:24:04 2023 from 127.0.0.1
```

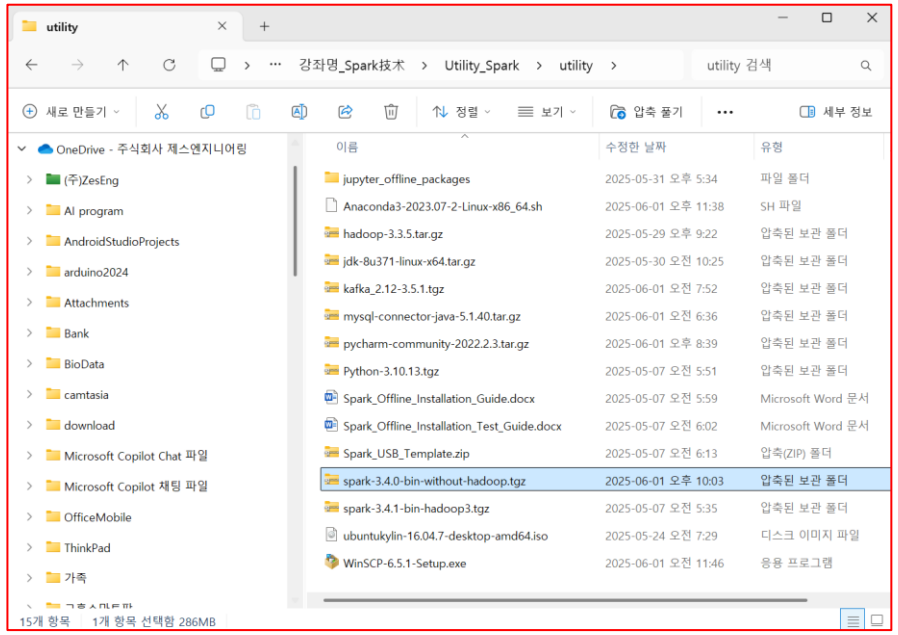
3.2.2 安装Hadoop前的准备工作



在Windows上下载Linux所需的软件









Ubuntu에서 Hadoop을 설치하기 전에 SSH 설정은 필수입니다. Hadoop은 분산 모드에서 작동할 때 노드 간 통신을 위해 패스워드 없이 SSH 접속이 가능해야 합니다. 아래는 단계별 설정 방법입니다.

1. OpenSSH 서버 설치

bash 复制 下载

```
sudo apt-get update
sudo apt-get install openssh-server openssh-client -y
```



```
Terminal File Edit View Search Terminal Help
d64 DEP-11 Metadata [2,464 B]
Get:25 https://mirrors.tuna.tsinghua.edu.cn/ubuntu xenial-security/multiverse DE
P-11 64x64 Icons [2,638 B]
Err:26 http://archive.ubuntukylin.com:10006/ubuntukylin trusty InRelease
  Could not connect to archive.ubuntukylin.com:10006 (124.126.103.228). - connec
t (111: Connection refused)
Hit:27 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:28 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:29 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 4,111 kB in 24s (164 kB/s)
Reading package lists... Done
W: Failed to fetch http://archive.ubuntukylin.com:10006/ubuntukylin/dists/trusty
/InRelease  Could not connect to archive.ubuntukylin.com:10006 (124.126.103.228)
. - connect (111: Connection refused)
W: Some index files failed to download. They have been ignored, or old ones used
 instead.
dslab@ubuntu:~$ sudo apt-get install openssh-server openssh-client -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.2p2-4ubuntu2.10).
openssh-server is already the newest version (1:7.2p2-4ubuntu2.10).
0 upgraded, 0 newly installed, 0 to remove and 203 not upgraded.
dslab@ubuntu:~$
```

2. SSH 서비스 시작 및 상태 확인



bash 复制 下载

```bash
sudo systemctl start ssh    # SSH 서비스 시작
sudo systemctl enable ssh   # 부팅 시 자동 시작
sudo systemctl status ssh   # 상태 확인 (활성화되면 초록색 "active (running)")
```

```
Terminal File Edit View Search Terminal Help
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.2p2-4ubuntu2.10).
openssh-server is already the newest version (1:7.2p2-4ubuntu2.10).
0 upgraded, 0 newly installed, 0 to remove and 203 not upgraded.
dslab@ubuntu:~$ sudo systemctl start ssh
dslab@ubuntu:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV init with /lib/systemd/systemd-sysv
-install...
Executing /lib/systemd/systemd-sysv-install enable ssh
dslab@ubuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Tue 2025-06-10 23:22:20 PDT; 4min 30s ago
 Main PID: 9659 (sshd)
   CGroup: /system.slice/ssh.service
           └─9659 /usr/sbin/sshd -D

Jun 10 23:22:20 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Jun 10 23:22:20 ubuntu sshd[9659]: Server listening on 0.0.0.0 port 22.
Jun 10 23:22:20 ubuntu sshd[9659]: Server listening on :: port 22.
Jun 10 23:22:20 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Jun 10 23:26:23 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```



Terminal File Edit View Search Terminal Help

```
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.2p2-4ubuntu2.10).
```

Terminal File Edit View Search Terminal Help

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

204 packages can be updated.
170 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

dslab@ubuntu:~$ su - hadoop
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hadoop@ubuntu:~$
```



```
Terminal File Edit View Search Terminal Help
Last login: Tue Jun 10 23:29:37 2025 from 127.0.0.1
dslab@ubuntu:~$ cd ~/.ssh/
dslab@ubuntu:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dslab/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dslab/.ssh/id_rsa.
Your public key has been saved in /home/dslab/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:rB34+QwWnMY52PQ1qXozJ3Cz/xOpk4zrtPEaRN7ZqbU dslab@ubuntu
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|              .  |
|         . . +   |
|        O * + + .|
|       o S B o +.|
|        = X o oo.|
|       . B XoooE.|
|        . *.@= . |
|          .Booo..|
+----[SHA256]-----+
dslab@ubuntu:~/.ssh$
```



Terminal File Edit View Search Terminal Help

```
|        = X o oo. |
|       . B XoooE. |
|        . *.@= .  |
|          .Booo.. |
+----[SHA256]-----+
dslab@ubuntu:~/.ssh$ cat ./id_rsa.pub >> ./authorized_keys
dslab@ubuntu:~/.ssh$ ls -la
total 24
drwx------  2 dslab dslab 4096 Jun 10 23:34 .
drwxr-xr-x 17 dslab dslab 4096 Jun 10 23:29 ..
-rw-rw-r--  1 dslab dslab  394 Jun 10 23:34 authorized_keys
-rw-------  1 dslab dslab 1679 Jun 10 23:33 id_rsa
-rw-r--r--  1 dslab dslab  394 Jun 10 23:33 id_rsa.pub
-rw-r--r--  1 dslab dslab  222 Jun 10 23:29 known_hosts
dslab@ubuntu:~/.ssh$ chmod 600 ~/.ssh/authorized_keys
dslab@ubuntu:~/.ssh$ ls -la
total 24
drwx------  2 dslab dslab 4096 Jun 10 23:34 .
drwxr-xr-x 17 dslab dslab 4096 Jun 10 23:29 ..
-rw-------  1 dslab dslab  394 Jun 10 23:34 authorized_keys
-rw-------  1 dslab dslab 1679 Jun 10 23:33 id_rsa
-rw-r--r--  1 dslab dslab  394 Jun 10 23:33 id_rsa.pub
-rw-r--r--  1 dslab dslab  222 Jun 10 23:29 known_hosts
dslab@ubuntu:~/.ssh$
```

VMware共享文件夹设置及在Linux中获取文件的方法



共享文件夹设置步骤

1. 基本设置步骤

1.1 打开VMware设置

•选择正在运行的虚拟机

•点击顶部菜单 VM > Settings

1.2 选择Options选项卡

在左侧菜单中选择 Options 选项卡
从列表中选择 Shared Folders

1.3 启用共享文件夹

在右侧选择 Always enabled
检查 Map as a network drive in Windows guests 选项（Windows客户机适用）

1.4 添加文件夹

•点击 Add... 按钮

•选择要共享的主机（Windows）文件夹

•设置文件夹名称（默认或自定义名称）

•确认勾选 Enable this share

•点击 Finish

Virtual Machine Settings

Hardware Options

Settings	Summary
General	hadoop01
Power	
Shared Folders	Enabled
Snapshots	
AutoProtect	Disabled
Guest Isolation	
Access Control	Not encrypted
VMware Tools	Time sync off
VNC Connections	Disabled
Unity	
Appliance View	
Autologin	Not supported
Advanced	Default/Default

Folder sharing

Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

- Disabled
- Always enabled
- Enabled until next power off or suspend

Folders

Name	Host Path
Camtasia	C:₩Users₩ebi₩Documents₩Camtasia

Add... Remove Properties

OK Cancel Help



在Linux中获取文件的多种方法



1. 使用终端命令

基本命令示例：

```
# 查看共享文件夹内容
ls /mnt/hgfs/

# 查看特定共享文件夹内容
ls /mnt/hgfs/共享文件夹名称

# 复制文件（共享文件夹 → Linux主目录）
cp /mnt/hgfs/共享文件夹名称/文件名 ~/

# 复制文件到下载文件夹
cp /mnt/hgfs/共享文件夹名称/文件名 ~/Downloads/

# 复制整个目录
cp -r /mnt/hgtfs/共享文件夹名称/目录名 ~/
```

```
Terminal File Edit View Search Terminal Help
(base) hadoop@hadoop01:~$ ls /mnt/hgfs
Camtasia
(base) hadoop@hadoop01:~$ ls -la /mnt/hgfs/Camtasia/
total 124463
drwxrwxrwx 1 root root     4096 Aug 19 09:01 .
dr-xr-xr-x 1 root root     4192 Sep 17 07:38 ..
drwxrwxrwx 1 root root        0 Aug 19 08:52 Audiate
-rwxrwxrwx 1 root root 78629673 Aug 19 09:00 Rec 2025-08-19----10-00-21.trec
-rwxrwxrwx 1 root root 48514834 Aug 19 08:51 Rec 2025-08-19----9-51-39.trec
-rwxrwxrwx 1 root root   296010 Aug 19 09:43 ubuntu01.tscproj
(base) hadoop@hadoop01:~$ cp /mnt/hgfs/Camtasia/ubuntu01.tscproj ~/Downloads/
(base) hadoop@hadoop01:~$ ls -la ~/Downloads/ub*
-rwxrwxr-x 1 hadoop hadoop 296010 Sep 17 07:39 /home/hadoop/Downloads/ubuntu01.t
scproj
(base) hadoop@hadoop01:~$
```

3.2.2 安装Hadoop前的准备工作



4. 安装Java环境

执行如下命令创建“/usr/lib/jvm”目录用来存放JDK文件：

```
$cd /usr/lib
$sudo mkdir jvm #创建/usr/lib/jvm目录用来存放JDK文件
```

执行如下命令对安装文件进行解压缩：

```
$cd ~ #进入hadoop用户的主目录
$cd Downloads
$sudo tar -zxvf ./jdk-8u371-linux-x64.tar.gz -C /usr/lib/jvm
```

3.2.2 安装Hadoop前的准备工作



下面继续执行如下命令，设置环境变量：

```
$vim ~/.bashrc
```

上面命令使用vim编辑器打开了hadoop这个用户的环境变量配置文件，请在这个文件的开头位置，添加如下几行内容：

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_371
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

3.2.2 安装Hadoop前的准备工作



保存.bashrc文件并退出vim编辑器。然后，继续执行如下命令让.bashrc文件的配置立即生效：

```
$source ~/.bashrc
```

这时，可以使用如下命令查看是否安装成功：

```
$java -version
```

如果能够在屏幕上返回如下信息，则说明安装成功：

```
java version "1.8.0_371"
Java(TM) SE Runtime Environment (build 1.8.0_371-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)
```

3.2.3 安装Hadoop的三种模式



Hadoop包括三种安装模式：

- **单机模式：只在一台机器上运行，存储是采用本地文件系统，没有采用分布式文件系统HDFS；**
- 伪分布式模式：存储采用分布式文件系统HDFS，但是，HDFS的名称节点和数据节点都在同一台机器上；
- 分布式模式：存储采用分布式文件系统HDFS，而且，HDFS的名称节点和数据节点位于不同机器上。

3.2.4 下载Hadoop安装文件



本课程采用的Hadoop版本是3.3.5，可以到Hadoop官网下载安装文件（https://archive.apache.org/dist/hadoop/core/hadoop-3.3.5）

请使用hadoop用户登录Linux系统，打开一个终端，执行如下命令：

```
$ sudo tar -zxvf ~/Downloads/hadoop-3.3.5.tar.gz -C /usr/local    # 解压到/usr/local中
$ cd /usr/local/
$ sudo mv ./hadoop-3.3.5/ ./hadoop       # 将文件夹名改为hadoop
$ sudo chown -R hadoop ./hadoop          # 修改文件权限
```

Hadoop解压后即可使用，可以输入如下命令来检查 Hadoop是否可用，成功则会显示 Hadoop版本信息：

```
$ cd /usr/local/hadoop
$ ./bin/hadoop version
```

运行结果

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hadoop version
Hadoop 3.3.5
Source code repository https://github.com/apache/hadoop.git -r 706d88266abcee09ed78fbaa0ad5f74d818ab0e9
Compiled by stevel on 2023-03-15T15:56Z
Compiled with protoc 3.7.1
From source with checksum 6bbd9afcf4838a0eb12a5f189e9bd7
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.5.jar
hadoop@ubuntu:/usr/local/hadoop$
```

3.2.5 单机模式配置



Hadoop默认模式为非分布式模式（本地模式， local mode），
无需进行其他配置即可运行。
Hadoop附带了丰富的例子，运行如下命令可以查看所有例子：

```
$ cd /usr/local/hadoop
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar
```

```
Ubuntu Kylin Desktop
Terminal File Edit View Search Terminal Help
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input
 files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in t
he input files.
hadoop@ubuntu:/usr/local/hadoop$
```

3.2.5 单机模式配置



这里选择运行grep例子

이것의 의미

```
$ cd /usr/local/hadoop
$ mkdir input
$ cp ./etc/hadoop/*.xml ./input   # 将配置文件复制到input目录下
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-ex
amples-*.jar grep ./input ./output 'dfs[a-z.]+'
$ cat ./output/*          # 查看运行结果
```

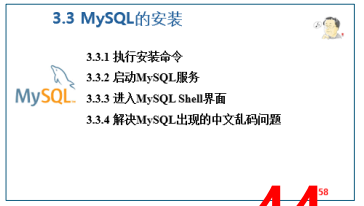
```
hadoop@ubuntu:/usr/local/hadoop$ cat ./output/*
1       dfsadmin
```

需要注意的是，Hadoop默认不会覆盖结果文件，因此，再次运行上面实例会提示出错。如果要再次运行，需要先使用如下命令把output文件夹删除：

```
$ rm -r ./output
```



Go MySQL



install

3.2.6 伪分布式模式配置



1. 修改配置文件

修改以后，/usr/local/Hadoop/etc/hadoop/core-site.xml文件的内容如下：

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

install

3.2.6 伪分布式模式配置



同样，需要修改配置文件hdfs-site.xml，修改后的内容如下：

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/data</value>
    </property>
</configuration>
```

install

3.2.6 伪分布式模式配置



2. 执行名称节点格式化

修改配置文件以后，要执行名称节点的格式化，命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs namenode -format
```

如果格式化成功，会看到“successfully formatted”的提示信息

```
2023-06-16 20:15:26,505 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/df
s/name has been successfully formatted.
2023-06-16 20:15:26,562 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/loc
al/hadoop/tmp/dfs/name/current/fsimage.ckpt_0000000000000000000 using no compression
2023-06-16 20:15:26,764 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hado
op/tmp/dfs/name/current/fsimage.ckpt_0000000000000000000 of size 401 bytes saved in 0 s
econds .
2023-06-16 20:15:26,788 INFO namenode.NNStorageRetentionManager: Going to retain 1 imag
es with txid >= 0
2023-06-16 20:15:26,840 INFO namenode.FSNamesystem: Stopping services started for activ
e state
2023-06-16 20:15:26,841 INFO namenode.FSNamesystem: Stopping services started for stand
by state
2023-06-16 20:15:26,847 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 wh
en meet shutdown.
2023-06-16 20:15:26,847 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
************************************************************/
```

install

3.2.6 伪分布式模式配置



3. 启动Hadoop

执行下面命令启动Hadoop：

```
$ cd /usr/local/hadoop
$ ./sbin/start-dfs.sh   #start-dfs.sh是个完整的可执行文件，中间没有空格
```

如果出现如图3-5所示的SSH提示，输入yes即可：

```
hadoop@DBLab-XMU:/usr/local/hadoop$ sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-na
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-da
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is a9:28:e0:4e:89:40:a4:cd:75:8f:0b:8b:57:79:67:86.
Are you sure you want to continue connecting (yes/no)? yes
```

install

3.2.6 伪分布式模式配置



Hadoop启动完成后，可以通过命令 jps 来判断是否成功启动，命令如下：

```
$ jps
```

若成功启动，则会列出如下进程

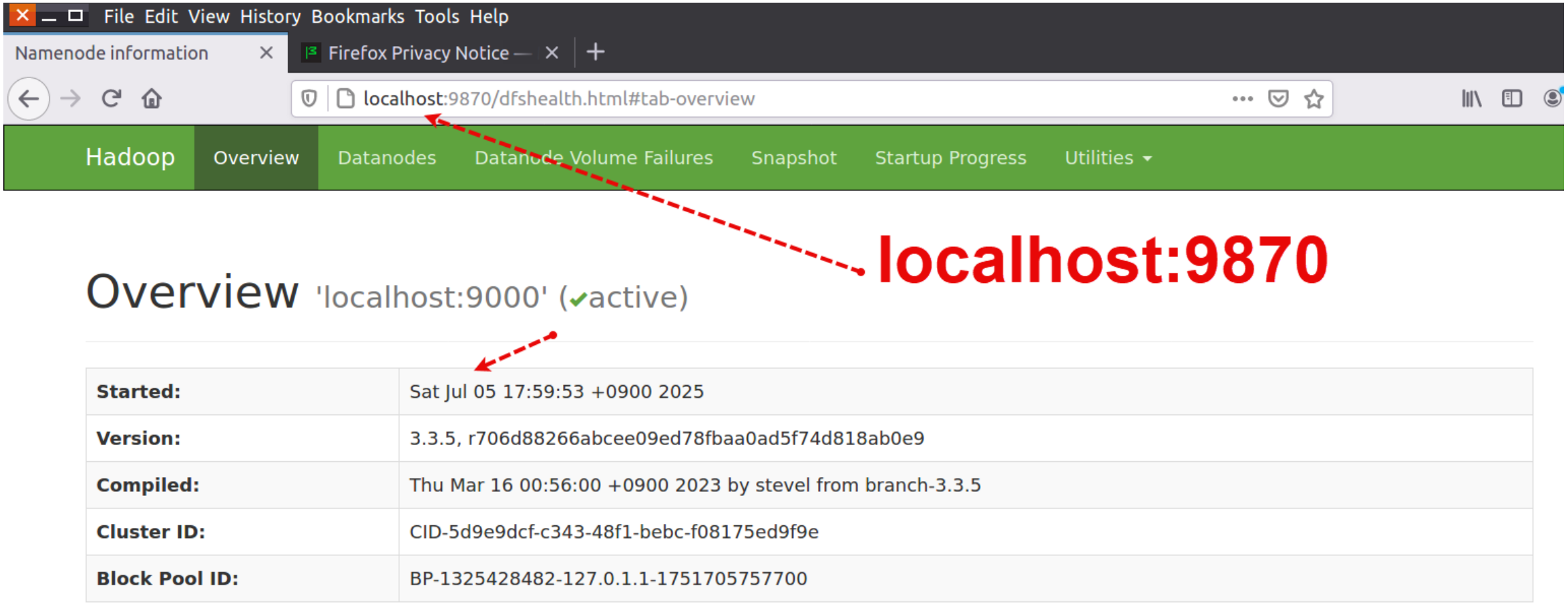
```
hadoop@ubuntu:/usr/local/hadoop$ jps
3638 DataNode
4011 Jps
3516 NameNode
3852 SecondaryNameNode
```

install

3.2.6 伪分布式模式配置



4. 使用Web界面查看HDFS信息



install

3.2.6 伪分布式模式配置



5. 运行Hadoop伪分布式实例

要使用HDFS，首先需要在HDFS中创建用户目录（本教程全部统一采用hadoop用户名登录Linux系统），命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -mkdir -p /user/hadoop
```

接着需要把本地文件系统的“/usr/local/hadoop/etc/hadoop”目录中的所有xml文件作为输入文件，复制到分布式文件系统HDFS中的“/user/hadoop/input”目录中，命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -mkdir input  #在HDFS中创建hadoop用户对应的input目录
$ ./bin/hdfs dfs -put ./etc/hadoop/*.xml input  #把本地文件复制到HDFS中
```

install

3.2.6 伪分布式模式配置



现在就可以运行Hadoop自带的grep程序，命令如下：

```
$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar grep input output 'dfs[a-z.]+'
```

运行结束后，可以通过如下命令查看HDFS中的output文件夹中的内容：

```
$ ./bin/hdfs dfs -cat output/*
```

执行结果如图所示

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat output/*
1       dfsadmin
1       dfs.replication
1       dfs.namenode.name.dir
1       dfs.datanode.data.dir
```




```
$ ./bin/hdfs dfs –rm –r output
```

install

3.2.6 伪分布式模式配置



6. 关闭Hadoop

如果要关闭Hadoop，可以执行下面命令：

```
$ cd /usr/local/hadoop
$ ./sbin/stop-dfs.sh
```

install

3.2.6 伪分布式模式配置



7. 配置PATH变量

首先使用vim编辑器打开“~/.bashrc”这个文件，然后，在这个文件的最前面位置加入如下单独一行：

```
export PATH=$PATH:/usr/local/hadoop/sbin
```

在后面的学习过程中，如果要继续把其他命令的路径也加入到PATH变量中，也需要继续修改“~/.bashrc”这个文件。当后面要继续加入新的路径时，只要用英文冒号“:”隔开，把新的路径加到后面即可，比如，如果要继续把“/usr/local/hadoop/bin”路径增加到PATH中，只要继续追加到后面，如下所示：

```
export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin
```

添加后，执行命令“source ~/.bashrc”使设置生效。设置生效后，在任何目录下启动Hadoop，都只要直接输入start-dfs.sh命令即可，同理，停止Hadoop，也只需要在任何目录下输入stop-dfs.sh命令即可。

install



```
#Java
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_371
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib

#Hadoop
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

#Spark
export SPARK_HOME=/usr/local/spark
export PYSPARK_PYTHON=/home/hadoop/anaconda3/envs/nanyang/bin/python3.8
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9.5-src.zip:$PYTHONPATH

#path
export PATH=$PATH:${JAVA_HOME}/bin:$HADOOP_HOME/bin:${HADOOP_HOME}/sbin:$SPARK_HOME/bin
```

HDFS常用操作

- Hadoop 集群启动与关闭
- HDFS 基本命令使用方法
- WordCount 示例运行
- 文件系统操作实验



1. 文件/目录操作

命令	说明	示例
-ls [路径]	列出目录内容	hdfs dfs -ls /data
-mkdir [路径]	创建目录	hdfs dfs -mkdir /user/hadoop
-put [本地文件] [HDFS路径]	上传本地文件到HDFS	hdfs dfs -put file.txt /data/
-get [HDFS文件] [本地路径]	下载HDFS文件到本地	hdfs dfs -get /data/file.txt ./
-cp [源路径] [目标路径]	HDFS内复制文件	hdfs dfs -cp /data/a.txt /backup/
-mv [源路径] [目标路径]	HDFS内移动文件	hdfs dfs -mv /data/a.txt /archive/
-rm [-r] [路径]	删除文件/目录 (-r递归删除)	hdfs dfs -rm /data/old.log
-cat [文件路径]	查看文件内容	hdfs dfs -cat /data/log.txt
-tail [-f] [文件路径]	实时查看文件尾部 (-f持续跟踪)	hdfs dfs -tail -f /data/log.txt







3.3 MySQL的安装



3.3.1 执行安装命令

3.3.2 启动MySQL服务

3.3.3 进入MySQL Shell界面

3.3.4 解决MySQL出现的中文乱码问题

3.3.1执行安装命令



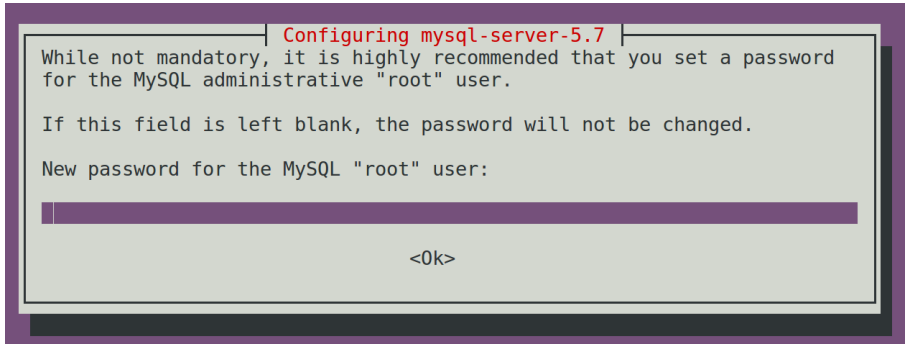
在安装MySQL之前，需要更新一下软件源以获得最新版本，命令如下：

```
$ sudo apt-get update
```

然后，执行如下命令安装MySQL：

```
$ sudo apt-get install mysql-server
```

然后，密码输入

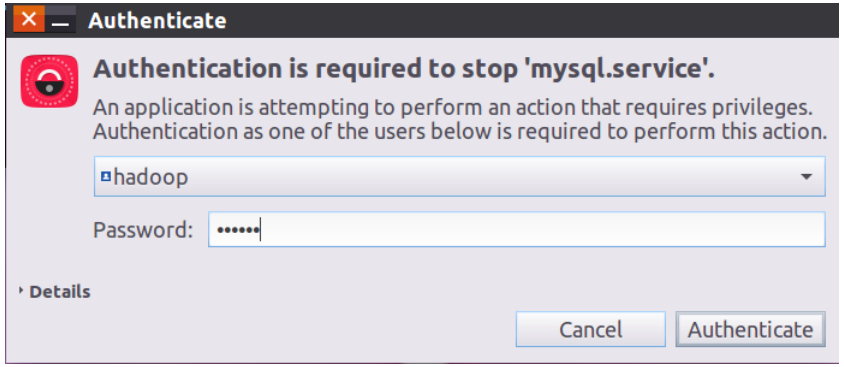


3.3.2启动MySQL服务



默认情况下，安装完成就会自动启动MySQL。
可以手动关闭MySQL服务，然后再次启动MySQL服务，
命令如下：

```
$ service mysql stop
$ service mysql start
```



可以执行如下命令来确认是否启动成功：

如图3-43所示，如果MySQL节点处于LISTEN状态，则表示启动成功：

```
hadoop@ubuntu:~$ cd /usr/local/hadoop
hadoop@ubuntu:/usr/local/hadoop$ sudo netstat -tap |grep  mysql
tcp        0      0 localhost:mysql         *:*                     LISTEN
5024/mysqld
hadoop@ubuntu:/usr/local/hadoop$
```

图3-43 netstat命令执行结果

3.3.3进入MySQL Shell界面



执行如下命令进入MySQL Shell界面：

```
$ mysql -u root -p
```

该命令执行以后，系统会提示输入MySQL数据库的root用户的密码，本课程把密码统一设置为hadoop。
然后，就进入了“mysql>”命令提示符状态，如图3-44所示。

```
hadoop@hadoop01:/usr/local/idea/bin$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.33-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

图3-44 启动进入 MySQl Shell 界面

3.3.4 解决MySQL出现的中文乱码问题



当向MySQL数据库导入数据时，可能会出现中文乱码问题，
原因是character_set_server默认设置是latin1，
不是中文编码。
要查询MySQL数据库当前的字符编码格式，可以使用如下命令：

```
mysql> show variables like 'char%';
```

执行该命令以后，会出现类似图3-45所示的信息。

```
mysql> show variables like "char%";
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | latin1                     |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)
```

图3-45查看数据库字符编码格式

3.3.4 解决MySQL出现的中文乱码问题



可以单个设置修改编码方式，比如，使用如下命令：

```
mysql> set character_set_server=utf8;
```

但是，通过这种方式设置字符编码格式，
重启MySQL服务以后就会失效。
因此，建议按照如下方式修改编码格式：

（1）修改配置文件

在Linux系统中新打开一个终端，
使用vim编辑器编辑“/etc/mysql/mysql.conf.d/mysqld.cnf”文件，命令如下：

```
$ sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

3.3.4 解决MySQL出现的中文乱码问题



注意，上面的命令是在Linux Shell命令提示符下执行的，不是在“mysql>”命令提示符下执行，一定要注意区分。打开mysqld.cnf文件以后，请在[mysqld]下面添加一行“character_set_server=utf8”，如图3-46所示。

```
[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
character_set_server=utf8
skip-external-locking
```

图3-46 修改配置文件

3.3.4 解决MySQL出现的中文乱码问题



（3）重启MySQL服务

在Linux终端的Shell命令提示符（不是“mysql>”命令提示符）下执行如下命令重启MySQL服务：

```
$ service mysql restart
```

（4）登录MySQL查看当前编码格式

重启MySQL服务以后，再次使用如下命令查询MySQL数据库当前的字符编码格式：

```
$ mysql -u root -p
```

```
mysql> show variables like 'char%';
```

3.3.4 解决MySQL出现的中文乱码问题



执行该命令以后，会出现类似图3-47所示的信息。

```
mysql> show variables like "char%";
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | utf8                       |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)
```

图3-47 修改数据库字符编码格式以后的提示信息

3.4 Kafka的安装



3.4.1 Kafka简介

3.4.2 Kafka的安装和使用

3.4.1 Kafka简介



Kafka是一种高吞吐量的分布式发布订阅消息系统，为了更好地理解和使用 Kafka，这里介绍一下Kafka的相关概念：

- Broker：Kafka集群包含一个或多个服务器，这些服务器被称为Broker。
- Topic：每条发布到Kafka集群的消息都有一个类别，这个类别被称为Topic。

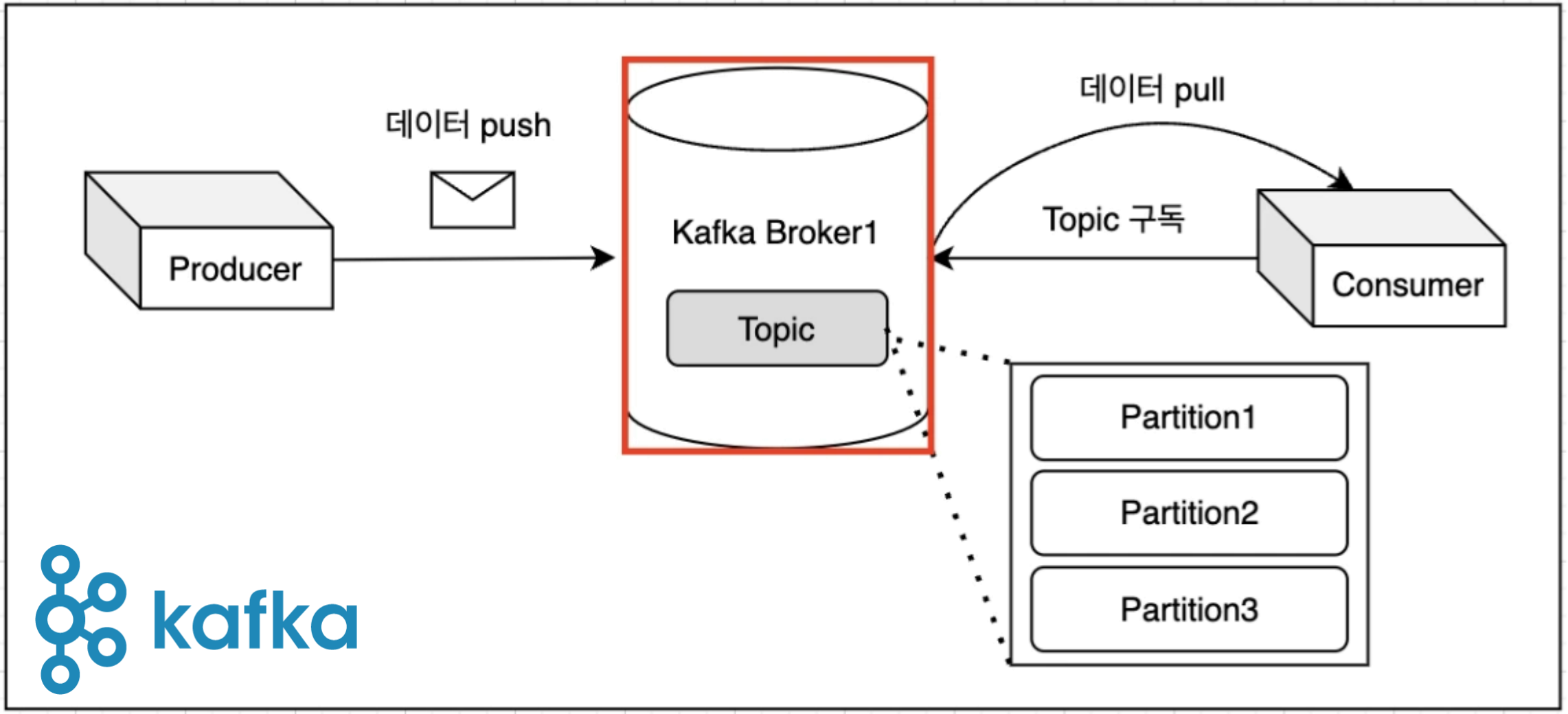
  物理上不同Topic的消息分开存储，逻辑上一个Topic的消息虽然保存于一个或多个Broker上，但用户只需指定消息的Topic，即可生产或消费数据，而不必关心数据存于何处。
- Partition：是物理上的概念，每个Topic包含一个或多个Partition。
- Producer：负责发布消息到Kafka Broker。
- Consumer：消息消费者，向Kafka Broker读取消息的客户端。

Consumer Group：每个Consumer属于一个特定的Consumer Group，

可为每个Consumer指定Group Name，若不指定Group Name，则属于默认的Group。

3.4.1 Kafka简介




데이터 push
데이터 pull
Producer
Kafka Broker1
Topic
Topic 구독
Consumer
Partition1
Partition2
Partition3
kafka

3.4.2 Kafka的安装和使用



访问Kafka官网下载页面（https://kafka.apache.org/downloads），

下载Kafka稳定版本kafka_2.12-3.5.1.tgz，

为了让Spark应用程序能够顺利使用Kafka数据源，

在下载Kafka安装文件的时候要注意，

Kafka版本号一定要和自己电脑上已经安装的Scala版本号一致才可以。

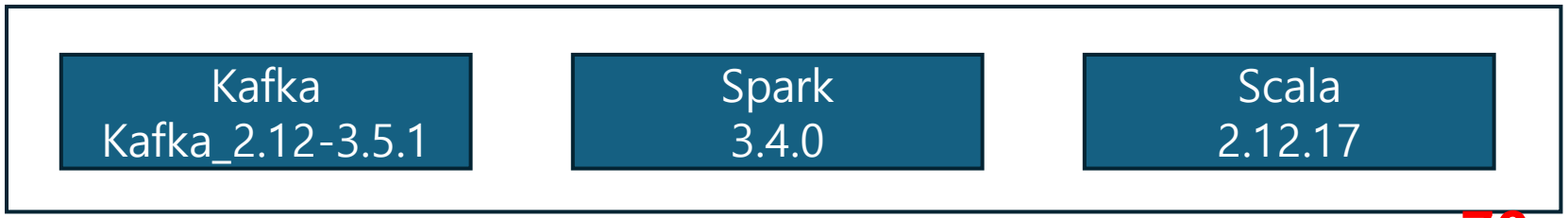
本课程安装Spark版本号是3.4.0，Scala版本号是2.12.17，

所以，一定要选择Kafka版本号是2.12开头的。

比如，到Kafka官网中，可以下载安装文件kafka_2.12-3.5.1.tgz，

前面的2.12就是支持的Scala版本号，后面的3.5.1是Kafka自身的版本号。





3.4.2 Kafka的安装和使用



执行如下命令安装Kafka：

```
$ cd ~/Downloads #假设安装文件放在这个目录下
$ sudo tar -zxvf kafka_2.12-3.5.1.tgz -C /usr/local
$ cd /usr/local
$ sudo mv kafka_2.12-3.5.1 kafka
$ sudo chown -R hadoop ./kafka
```



首先需要启动Kafka。请登录Linux系统（本课程统一使用hadoop用户登录），打开一个终端，输入下面命令启动Zookeeper服务：

```
$ cd  /usr/local/kafka
$ ./bin/zookeeper-server-start.sh  config/zookeeper.properties
```

注意，执行上面命令以后，终端窗口会返回一堆信息，然后就停住不动了，没有回到Shell命令提示符状态，这时，不要误以为死机了，而是Zookeeper服务器已经启动，正在处于服务状态。

所以，不要关闭这个终端窗口，一旦关闭，Zookeeper服务就停止了。

```
[2025-07-03 13:18:52,054] INFO zookeeper.request_throttler.shutdownTimeout = 100
00 (org.apache.zookeeper.server.RequestThrottler)
[2025-07-03 13:18:52,059] INFO PrepRequestProcessor (sid:0) started, reconfigEna
bled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2025-07-03 13:18:52,196] INFO Using checkIntervalMs=60000 maxPerMinute=10000 ma
xNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2025-07-03 13:18:52,200] INFO ZooKeeper audit is disabled. (org.apache.zookeepe
r.audit.ZKAuditProvider)
```

3.4.2 Kafka的安装和使用



请另外打开第二个终端，然后输入下面命令启动Kafka服务：

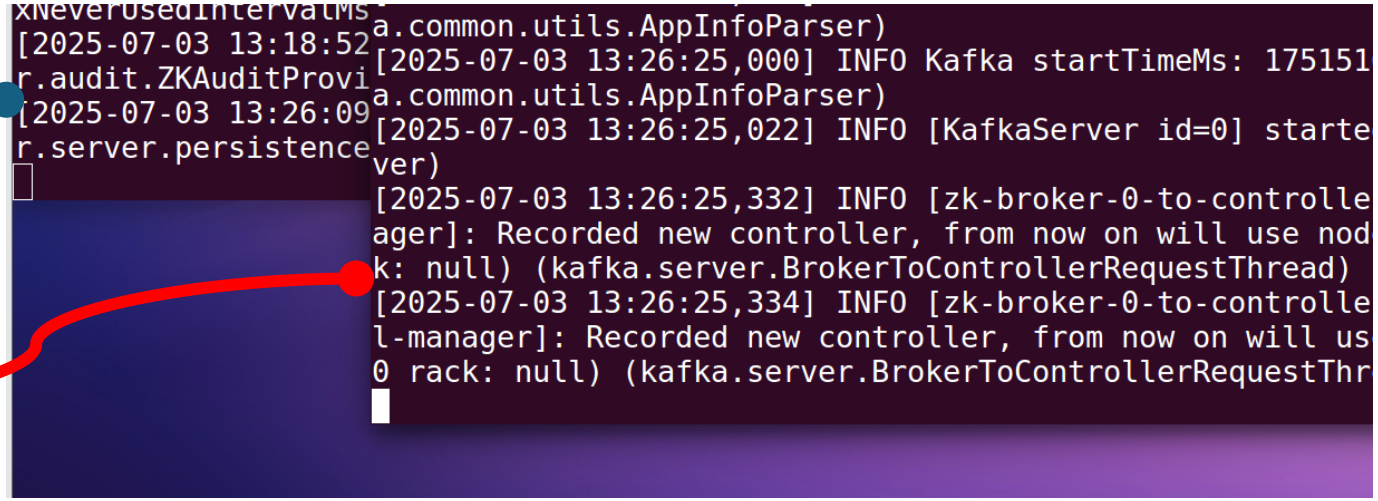
```
$ cd  /usr/local/kafka
$ ./bin/kafka-server-start.sh  config/server.properties
```

同样，执行上面命令以后，终端窗口会返回一堆信息，然后就会停住不动，没有回到Shell命令提示符状态，这时，同样不要误以为死机了，而是Kafka服务器已经启动，正在处于服务状态。所以，不要关闭这个终端窗口，一旦关闭，Kafka服务就停止了。



① 这是 Zookeeper 正在运行

② 这是 Kafka 正在运行



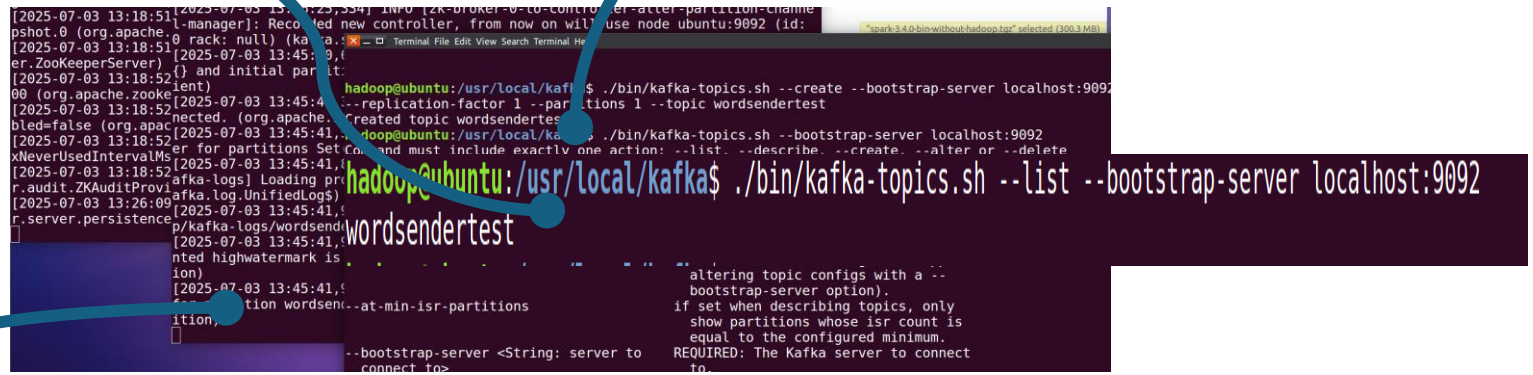
3.4.2 Kafka的安装和使用



下面先测试一下Kafka是否可以正常使用。再打开第三个终端，然后输入下面命令创建一个自定义名称为“wordsendertest”的Topic：

```
$ cd  /usr/local/kafka
$ ./bin/kafka-topics.sh  --create  --bootstrap-server  localhost:9092  ₩
> --replication-factor  1  --partitions  1  --topic  wordsendertest
#这个Topic叫wordsendertest，2181是Zookeeper默认的端口号，--partitions是Topic里面的分区数，--replication-factor是备份的数量，在Kafka集群中使用，由于这里是单机版，所以不用备份
#可以用list列出所有创建的Topic，来查看上面创建的Topic是否存在
$ ./bin/kafka-topics.sh  --list  --bootstrap-server  localhost:9092
```





kafka

3.4.2 Kafka的安装和使用



这个名称为“wordsendertest”的Topic，就是专门负责采集发送一些单词的。下面用生产者（Producer）来产生一些数据，请在当前终端内继续输入下面命令：

```
$ ./bin/kafka-console-producer.sh  --broker-list  localhost:9092 ₩
>  --topic  wordsendertest
```

上面命令执行后，就可以在当前终端（假设名称为“生产者终端”）内用键盘输入一些英文单词，比如可以输入：

```
hello hadoop
hello flink
```

要结束 Kafka 控制台生产者的消息输入并安全退出，请使用以下方法：

标准退出指令

Ctrl + C

3.4.2 Kafka的安装和使用



这些单词就是数据源，会被Kafka捕捉到以后发送给消费者。现在可以启动一个消费者，来查看刚才生产者产生的数据。请另外打开第四个终端，输入下面命令：

```
$ cd /usr/local/kafka
$./bin/kafka-console-consumer.sh  --bootstrap-server  localhost:9092  ₩
> --topic  wordsendertest  --from-beginning
```

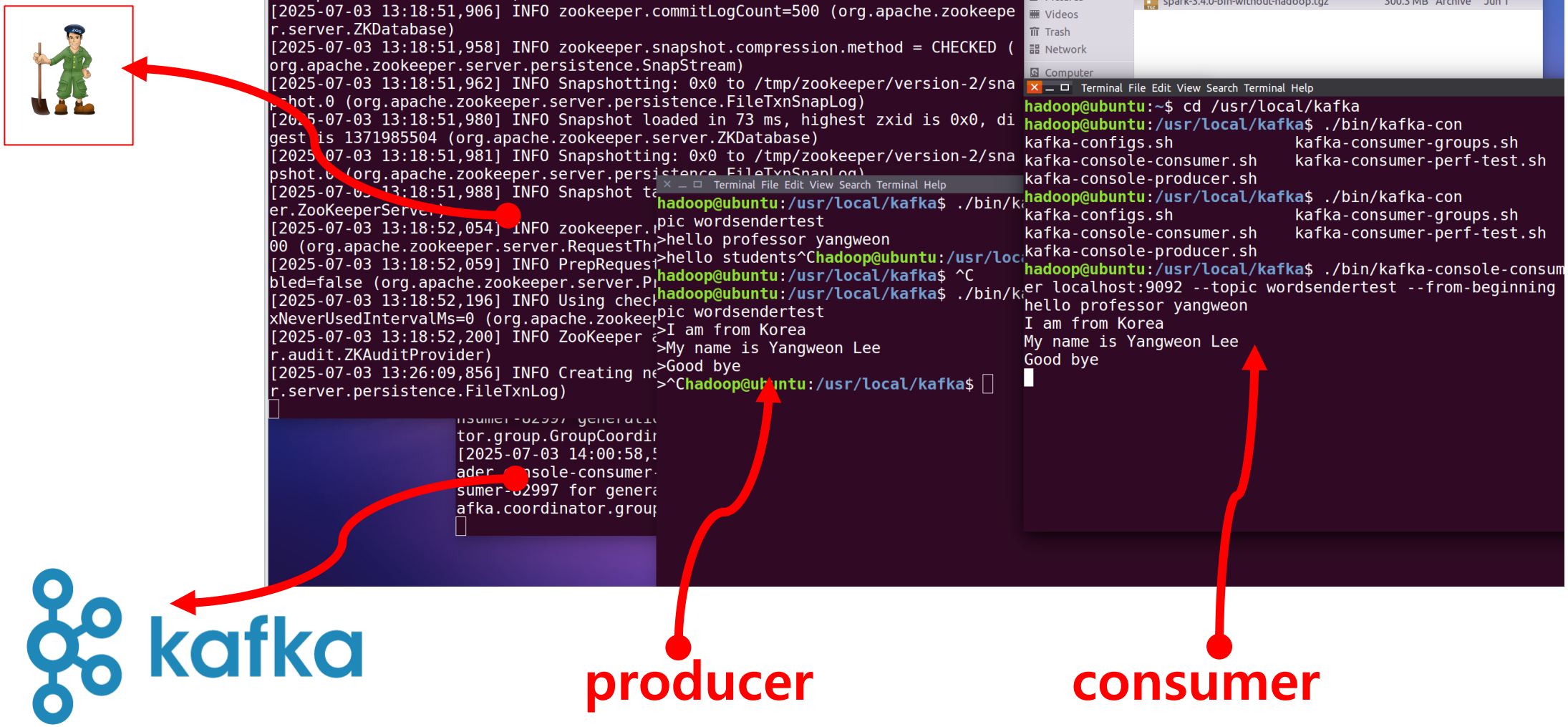
可以看到，屏幕上会显示出如下结果，也就是刚才在另外一个终端里面输入的内容：

```
hello hadoop
hello flink
```

kafka

3.4.2 Kafka的安装和使用





3.5 Anaconda的安装和使用方法



Anaconda官方网站下载地址是https://www.anaconda.com/，但是，由于访问国外网站速度较慢，所以，我们一般选择国内的镜像网站进行下载，地址是https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/。可以选择下载Anaconda3-2023.07-2-Linux-x86_64.sh。

在Linux终端中执行如下命令安装Anaconda：

```
$ cd /home/hadoop/Downloads #假设安装文件在这个目录下
$ sh ./Anaconda3-2023.07-2-Linux-x86_64.sh
```

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/

3.5 Anaconda的安装和使用方法



执行上述命令后，会出现如图3-48所示的效果，要求阅读许可文件，
可以按回车键开始阅读许可文件，许可文件比较长，
可以按空格键向下翻页，
当出现“Do you accept the license terms? [yes|no]”时，输入“yes”即可。

```
hadoop@hadoop01:/usr/local$ cd /home/hadoop/Downloads cd /home/hadoop/Downloads
hadoop@hadoop01:~/Downloads$ sh ./Anaconda3-2023.07-2-Linux-x86_64.sh

Welcome to Anaconda3 2023.07-2

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

图3-48 安装命令执行后出现的界面

3.5 Anaconda的安装和使用方法

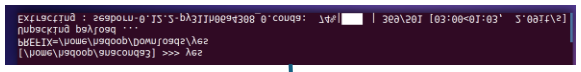


当出现设置安装路径提示时（如图3-49所示），可以自己设置一个路径，也可以选择默认的路径“/home/hadoop/anaconda3”，如果选择默认路径，可以直接按回车键。

```
Anaconda3 will now be installed into this location:
/home/hadoop/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/hadoop/anaconda3] >>> 
```

Enter key

엄청 시간이 오래 걸림



当询问是否需要初始化时（如图3-50所示），可以输入“**yes**”并回车，这样，Anaconda就安装成功了。

```
done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> yes
```

3.5 Anaconda的安装和使用方法

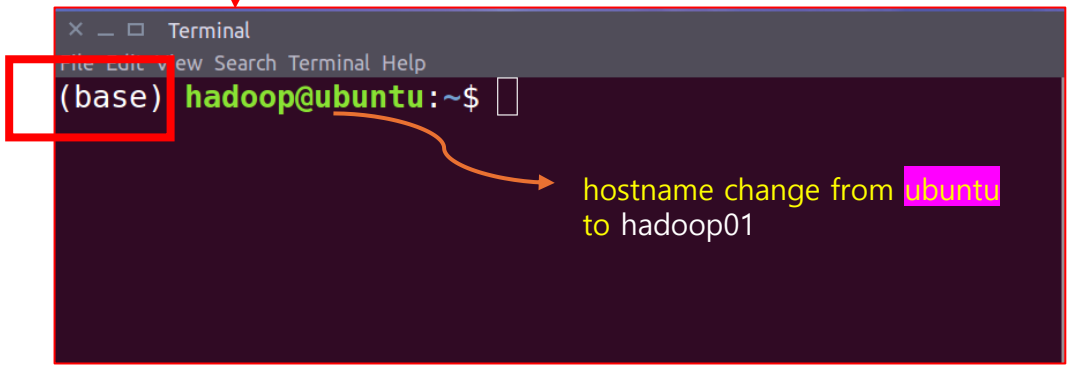


关闭当前的Linux终端，然后，再新建一个终端，可以看到，命令提示符变成了类似如下的格式（多了一个前缀base）：

```
(base) hadoop@hadoop01:~$
```

由于Anaconda是国外软件，Anaconda后期安装其他模块时，都需要访问国外网站下载文件，速度很慢，所以，可以更改Anaconda的数据源，从而加快访问速度，需要执行如下命令新建一个.condarc文件：

```
$ vim ~/.condarc
```



ubuntu→hadoop01



```
Terminal
File Edit View Search Terminal Help
(base) hadoop@ubuntu:~$ hostnamectl
   Static hostname: ubuntu
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 59ab03b21463489984b01e90a83220cc
           Boot ID: e35fafe1bfc04507b047f4e23b79653f
    Virtualization: vmware
  Operating System: Ubuntu Kylin 16.04.7 LTS
            Kernel: Linux 4.15.0-112-generic
(base) hadoop@ubuntu:~$
```

sudo vim /etc/hostname
sudo systemctl restart systemd-hostnamed
exec bash

```
Terminal
File Edit View Search Terminal Help
(base) hadoop@ubuntu:~$ hostnamectl
   Static hostname: ubuntu
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 59ab03b21463489984b01e90a83220cc
           Boot ID: e35fafe1bfc04507b047f4e23b79653f
    Virtualization: vmware
  Operating System: Ubuntu Kylin 16.04.7 LTS
            Kernel: Linux 4.15.0-112-generic
(base) hadoop@ubuntu:~$ sudo vim /etc/hostname
[sudo] password for hadoop:
Sorry, try again.
(base) hadoop@ubuntu:~$ sudo systemctl restart systemd-hostnamed
(base) hadoop@ubuntu:~$ exec bash
(base) hadoop@hadoop01:~$ hostnamectl
   Static hostname: hadoop01
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 59ab03b21463489984b01e90a83220cc
           Boot ID: e35fafe1bfc04507b047f4e23b79653f
    Virtualization: vmware
  Operating System: Ubuntu Kylin 16.04.7 LTS
            Kernel: Linux 4.15.0-112-generic
      Architecture: x86-64
(base) hadoop@hadoop01:~$
```

3.5 Anaconda的安装和使用方法



然后，在该文件中输入如下内容并保存：

```
channels:- defaultsshow_channel_urls: truechannel_alias:
https://mirrors.tuna.tsinghua.edu.cn/anacondadefault_channels:-
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main-
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free-
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r-
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro-
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2custo
m_channels:conda-forge:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloudmsys2:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloudbioconda:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloudmenpo:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloudpytorch:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloudsimpleitk:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

当 **.condarc** 文件创建或修改后，
conda 会 **自动检测并立即应用** 其中的配置，
无需重启终端或手动刷新。

3.5 Anaconda的安装和使用方法



在Linux终端内执行“python”命令（如图3-51所示），
可以看到当前的Python版本信息。

```
(base) hadoop@hadoop01:~$ python
Python 3.11.4 (main, Jul  5 2023, 14:15:25) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python 3.11.4这个Python版本太高，会与本课程安装的Spark3.4.0存在兼容性问题，
所以，安装结束后，需要额外安装可以和Spark3.4.0兼容的Python3.8，
可以在Anaconda3目录下执行如下命令安装Python3.8：

```
$ conda create -n pyspark python=3.8
```

1. 命令结构解析

命令部分	含义
`conda create`	指示Conda创建新虚拟环境
`-n pyspark`	将环境命名为 `pyspark`（`-n` 是 `--name` 的缩写）
`python=3.8`	在该环境中安装 **Python 3.8版本**

上面命令中“pyspark”是我们给这个版本的Python环境起的名字。
当安装过程出现“Proceed ([y]/n)? ”的询问时，输入“y”即可。
安装Python3.8结束以后，可以把Python环境切换到pyspark，命令如下：

```
$ conda activate pyspark
```

```
(base) hadoop@hadoop01:~$
(base) hadoop@hadoop01:~$ conda activate pyspark
(pyspark) hadoop@hadoop01:~$
```

3.5 Anaconda的安装和使用方法



然后执行如下命令测试是否切换成功:

```
$ python
```

如果切换成功，会显示如图3-52所示的效果。

```
(pyspark) hadoop@hadoop01:~$ python
Python 3.8.18 (default, Sep 11 2023, 13:40:15)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

最后，可以执行如下命令退出Python环境：

```
>>> exit()
```

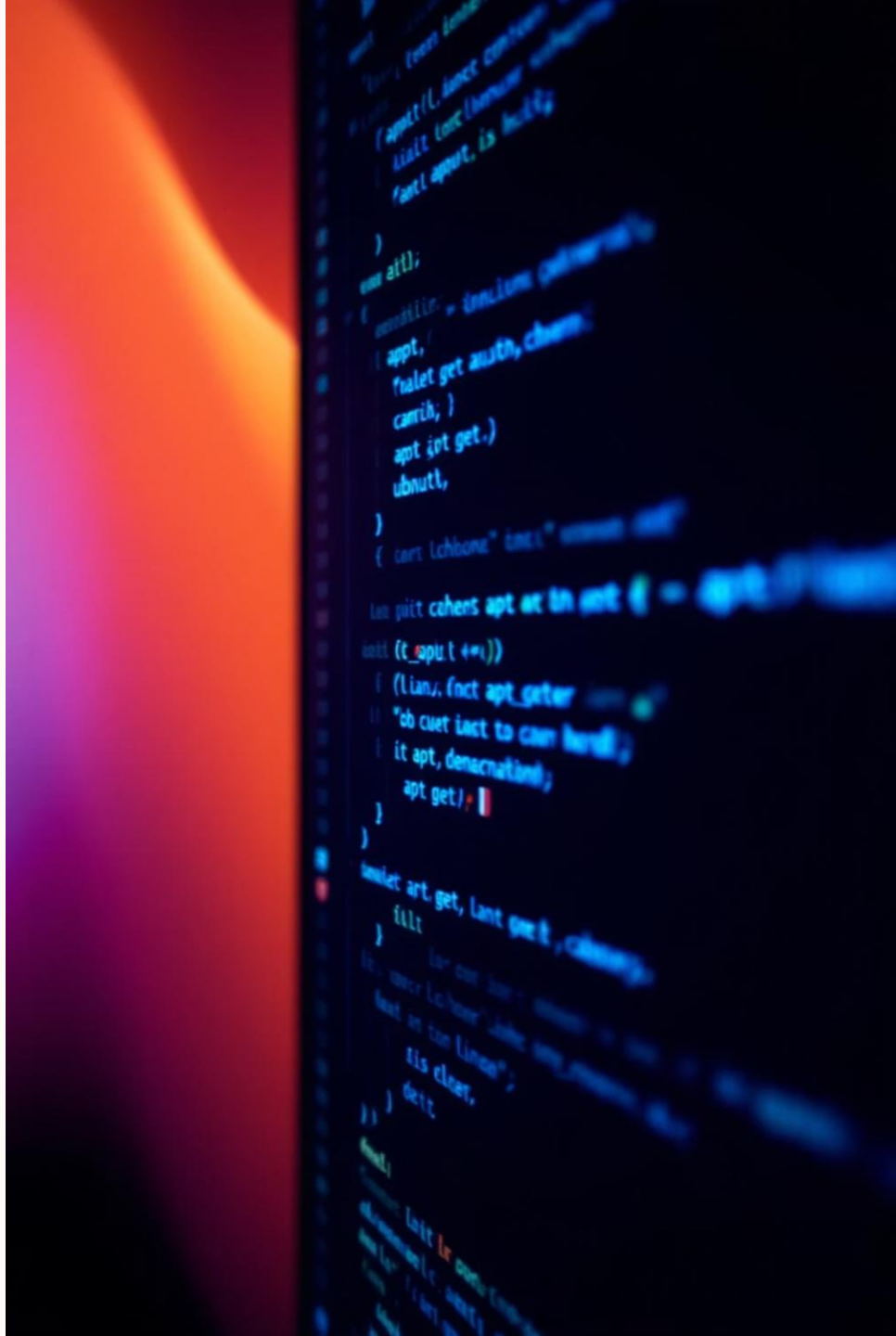
3.6 本章小结



1. 本课程所有软件，都是安装和运行在Linux操作系统之上，
   因此，顺利安装好Linux系统并且掌握Linux系统的基本使用方法，
   是开展后续授课内容学习的前提和基础。
2. Linux系统可以采用双系统方式安装，可是我们采用虚拟机方式安装。
   本课详细介绍了如何安装Linux虚拟机。
3. Hadoop是当前流行的分布式计算框架，
   在企业中得到了广泛的部署和应用。
   Hadoop和Flink可以配合使用，
   由Hadoop负责数据存储，由Flink负责数据计算，
   本课介绍了如何安装Hadoop。
4. 本课最后介绍了MySQL、Kafka和Anaconda的安装和使用方法。

Ubuntu Linux 中的 apt-get 软件件包管理工具

apt-get 是 Ubuntu Linux 系统中最经典的软件包管理工具，它能帮助用户轻松地安装、升级、卸载和管理各种软件包。通过命令行界面的操作，apt-get 显著提高了系统维护的效率，让软件管理变得简单而强大。

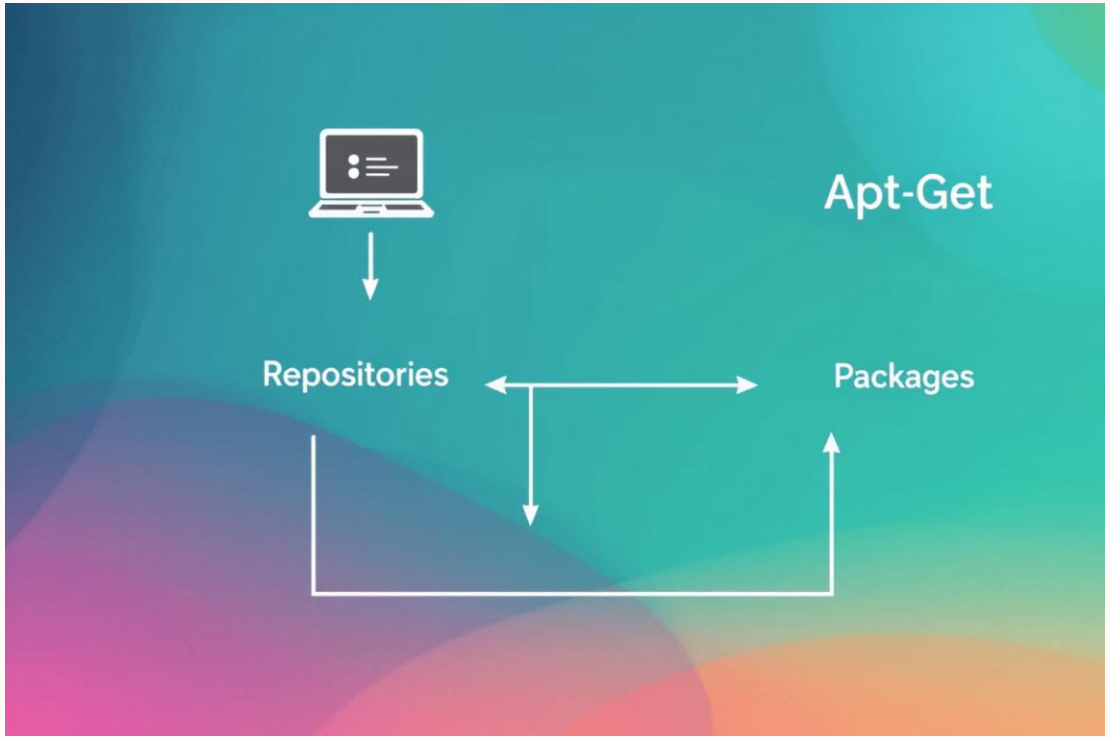


apt-get 的基本概念

基础特性

apt-get 是基于 Debian 的 Linux 发行版中默认的软件包管理工具，主要支支持 .deb 格式的软件包。它能够自动解决软件包之间复杂的依赖关系，确保系统运行稳定。

作为系统管理的核心工具，apt-get 通过访问预配置的软件源来获取最新的软件包信息，实现软件的便捷管理。



apt-get 系统架构图示



为什么软件包管理很重要?

1 保证安全与效率

软件包管理器确保所有软件安装和升级过程安全可控，避免系统混乱。从可信源获取软件包，减少恶意软件的风险。

2 节省时间

手动编译和安装软件非常耗时。apt-get 通过自动化过程大幅缩短了软件安装和升级的时间，提高系统管理效率。

3 自动处理依赖关系

Linux 软件通常有复杂的依赖关系。apt-get 能自动分析并安装所需的依赖包，大大降低了系统冲突和错误的可能性。

apt-get 主要命令示例

安装软件包

```
sudo apt-get install 包名
```

从软件源下载并安装指定的软件包及其所有依赖

更新软件源

```
sudo apt-get update
```

更新本地软件包列表数据库，获取最新的软件包信息

升级系统

```
sudo apt-get upgrade
```

将系统中所有已安装的软件包升级到最新版本

卸载软件包

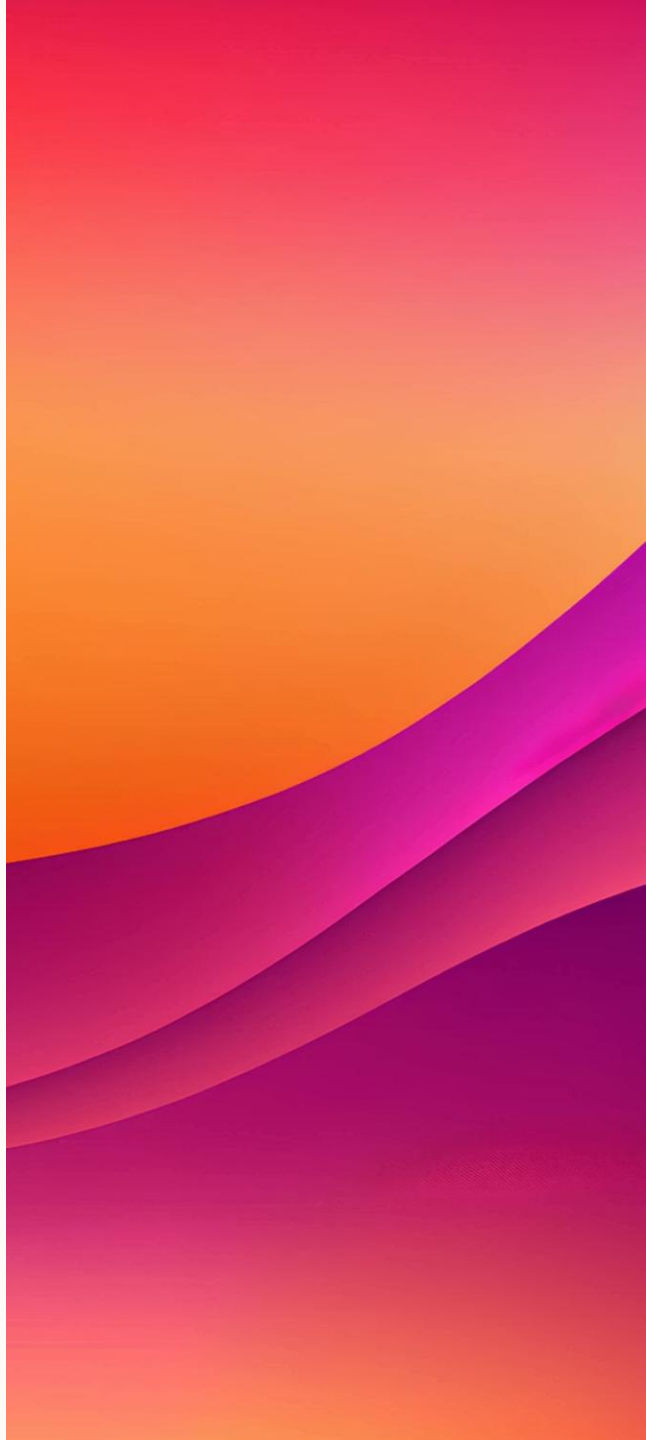
```
sudo apt-get remove 包名
```

从系统中删除指定的软件包，但保留配置文件

清理无用包

```
sudo apt-get autoremove
```

自动移除系统中不再需要的依赖包，释放磁盘空间



apt-get 的工作原理

检查仓库列表

apt-get 首先检查 `/etc/apt/sources.list` 文件中配置的软件源列表，确确定从哪些仓库获取软件包信息。

下载软件包

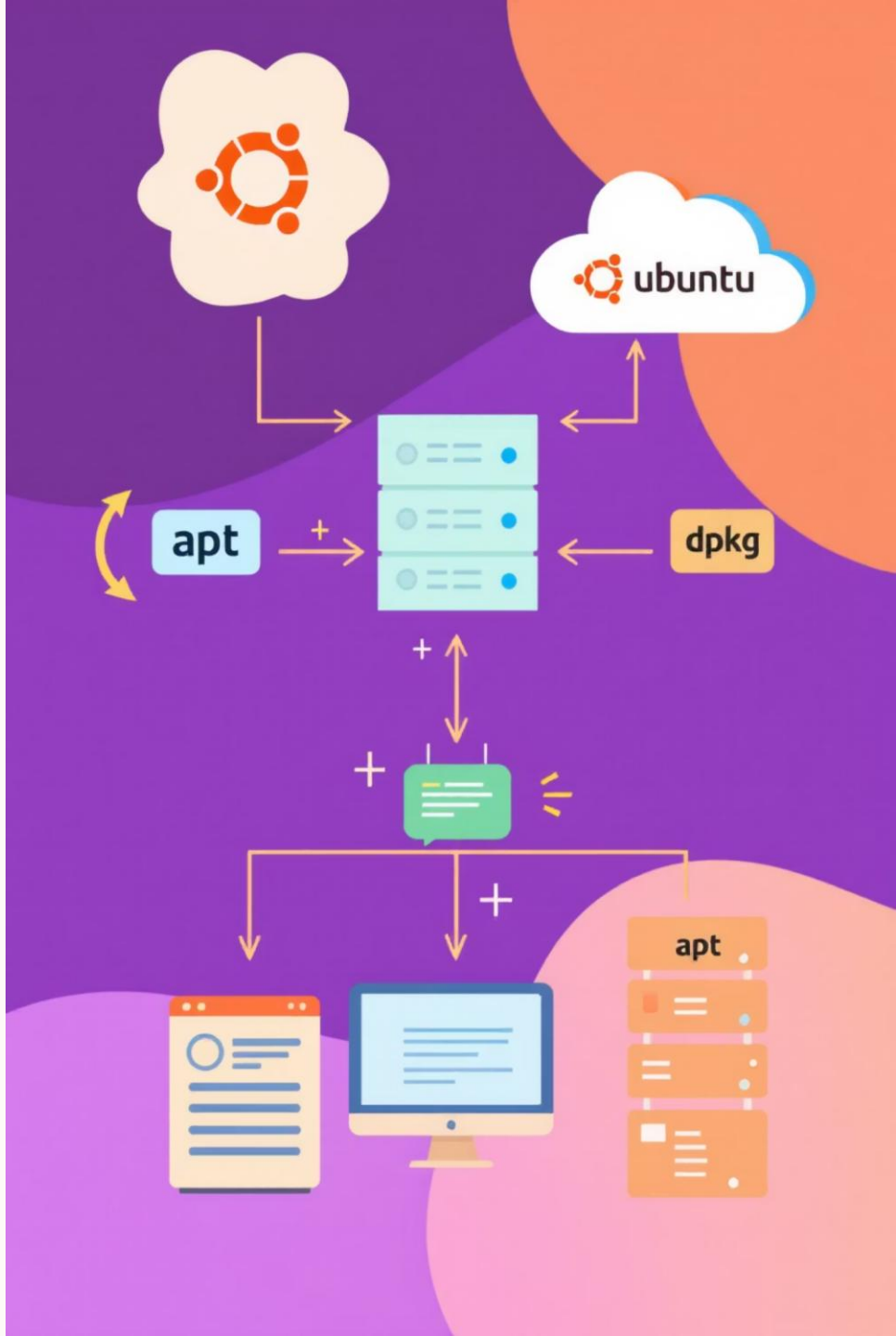
根据用户请求，从网络仓库下载所需的软件包及其元数据，确保获取最新版本。

解析依赖关系

自动分析软件包的依赖关系，确定需要额外安装哪些软件包才能满足系统要求。

安装与配置

按照正确的顺序安装所有软件包，完成配置，并更新系统数据库。



apt-get 与 apt 的区别

apt-get 特点

- 更适合在脚本中使用，命令参数丰富
- 提供更细致的控制选项
- 历史悠久，兼容性好
- 适合高级用户和系统管理员

apt 特点

- 提供更友好的用户界面体验
- 命令更简洁，易于记忆
- 默认显示进度条和颜色提示
- 集成了多个工具的常用功能
- 推荐新用户使用

使用 apt-get 时的注意事项

1 更新软件源

任何软件包安装或升级操作前，都应该先运行 `sudo apt-get update` 命令更新软件源信息，确保获取最新的软件包列表。否则可能会安装到过时的软件版本。

2 使用管理员权限

软件包的安装、升级和卸载操作需要管理员权限，必须在命令前加上 `sudo`。不正确的权限使用可能导致操作失败或系统文件损坏。

3 关注返回信息

在执行 apt-get 命令时，务必仔细阅读终端返回的信息，特别是错误和警告信息。这些信息对于判断操作是否成功以及排除故障非常重要。



总结

核心价值

apt-get 作为 Ubuntu 系统的核心管理工具，极大地简化了软件的安装与维维护流程。它通过自动化处理依赖关系，提供了一种安全、高效的软件管理方式。

深入理解 apt-get 的工作原理和常用命令，可以显著提升 Linux 系统的运维运维效率，减少系统管理的复杂性。

60%

时间节省

相比手动管理软件，使用apt-get可可节省的系统维护时间

50K+

软件包

Ubuntu官方仓库中可用的软件包数数量

1998年

诞生

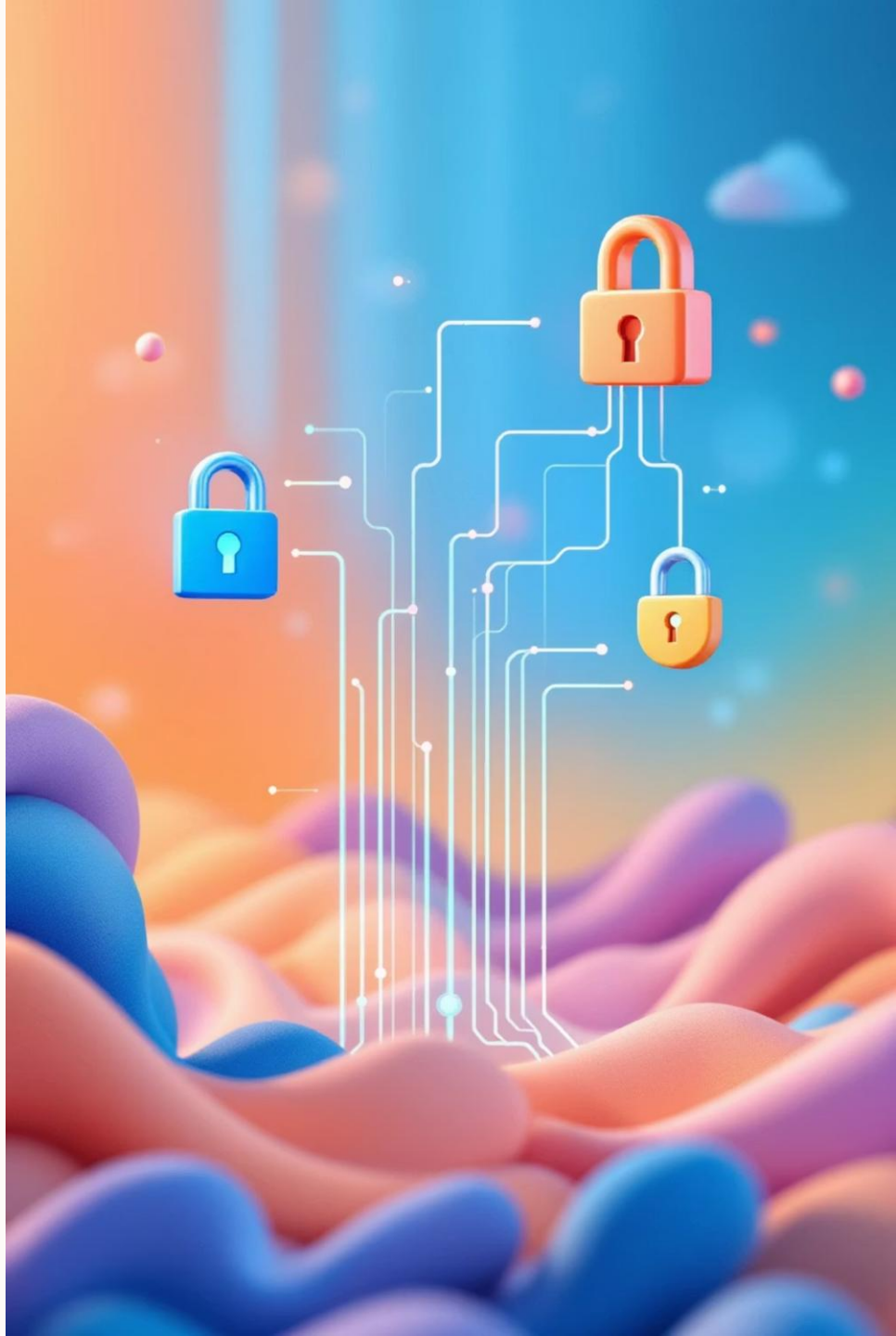
APT系统最初开发年份

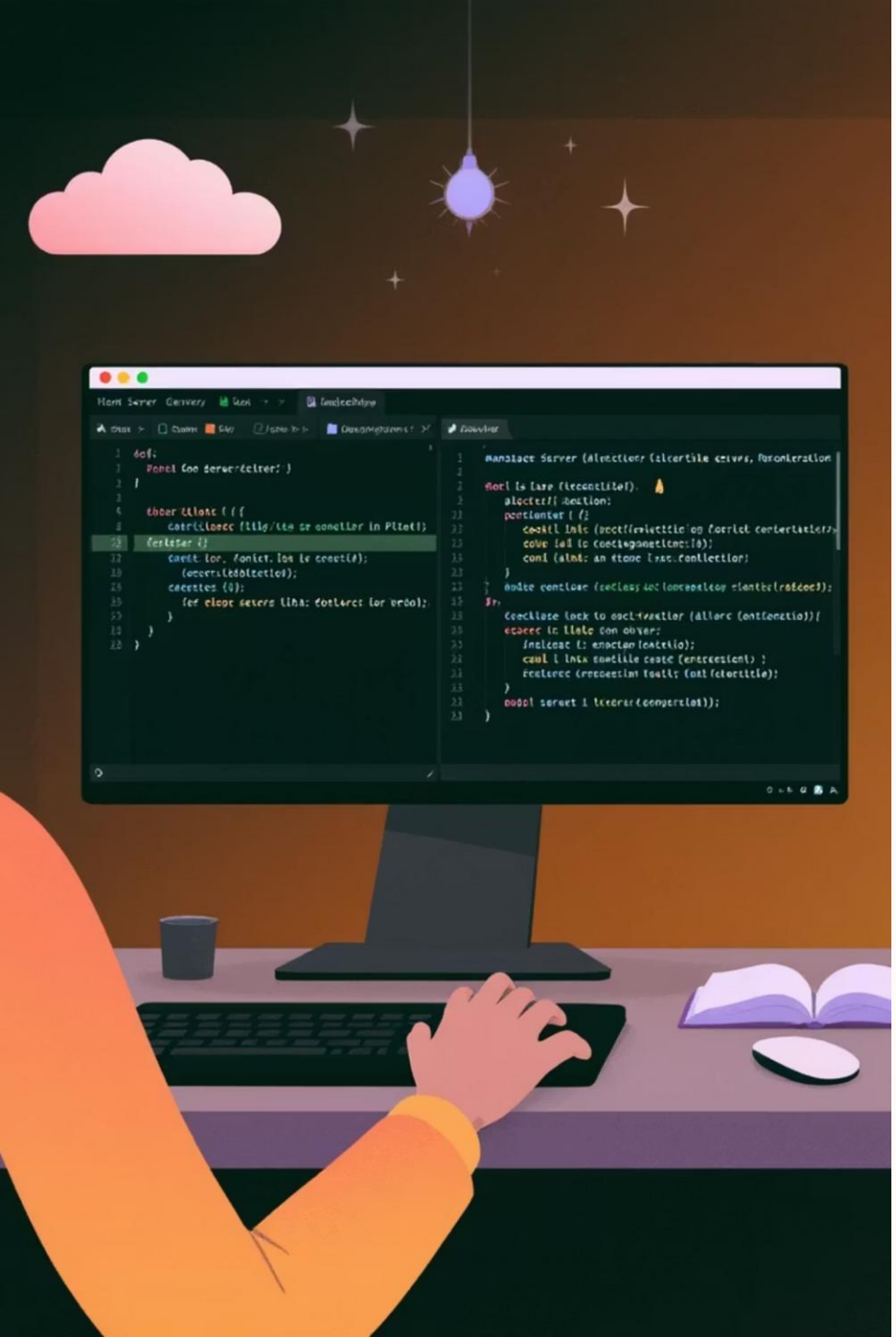


Linux Ubuntu SSH 连接概述

SSH(Secure Shell)是在Linux Ubuntu环境中用于安全远程连接的协议。SSH提供客户端（尝试连接的计算机）和服务器（被连接的目标计算机）之间的安全通信。

SSH支持两种主要的认证方式：传统的基于密码的认证和更安全的公钥/私钥认证。通过这种结构，即使在远程位置也能保持安全性并连接到服务器。

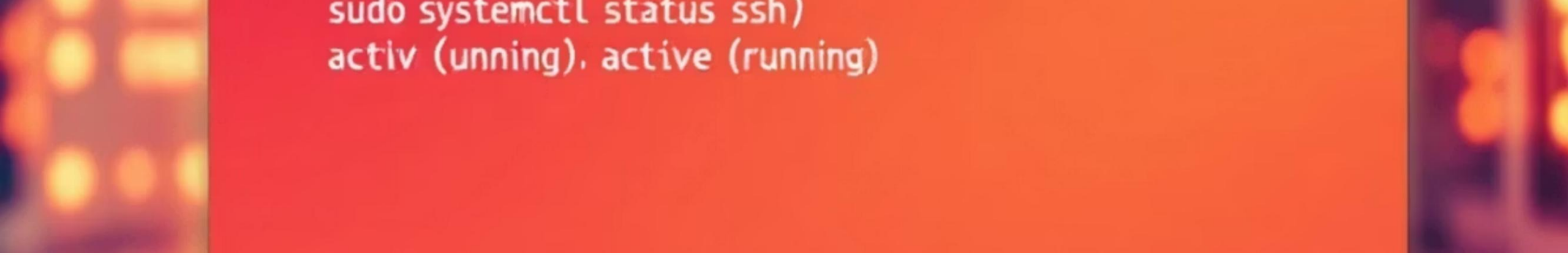




为什么需要SSH

- 通过终端或命令行界面可以高效地管理远程服务器
- 无需物理显示器或键盘也能完美运行无头(Headless)服务器
- 所有数据传输都经过加密，保护信息免受中间人攻击
- 网络管理员可以从任何地方安全地访问服务器，便于维护

SSH已成为现代服务器管理环境中必不可少的工具。该协议提供了上述关键优势。



服务器端：SSH服务器安装与配置

SSH服务器安装

在终端中执行以下命令：

```
sudo apt install openssh-server
```

此命令在Ubuntu服务器上安装SSH服务器软件。

服务器状态检查

安装后确认服务是否正常运行：

```
sudo systemctl status ssh
```

如果显示active (running)消息，则表示正常运行。

服务器启动与IP确认

如果服务未运行，启动命令：

```
sudo systemctl start ssh
```

确认IP地址的命令：

```
ip a
```

在输出信息中查找'inet'项以确认IP地址。

服务器端：防火墙与安全设置

防火墙设置

使用Ubuntu的基本防火墙工具UFW(Uncomplicated Firewall)允许SSH连接：

```
sudo ufw allow ssh
```

此命令开放默认SSH端口22。更改设置后重新加载防火墙：

```
sudo ufw reload
```

安全增强设置

可以修改配置文件增强SSH服务器的安全性：

```
sudo nano /etc/ssh/sshd_config
```

主要安全设置选项：

- PermitRootLogin no (阻止root账户直接登录)
- PasswordAuthentication no (仅允许密钥认证)
- Port 2222 (更改默认22端口)

设置更改后：

```
sudo systemctl restart ssh
```

客户端：SSH客户端准备



Ubuntu客户端

大多数Ubuntu发行版已预装SSH客户端。如果未安装：

```
sudo apt-get install openssh-client
```

安装后可以直接在终端中使用SSH命令。



Mac OS客户端

Mac OS已默认包含SSH客户端。打开终端应用即可直接使用SSH命令，无需额外安装。



Windows客户端

最新的Windows 10/11包含OpenSSH客户端。或者可以安装PuTTY、MobaXterm等专用SSH客户端程序。







客户端：服务器连接方法

基本SSH连接命令

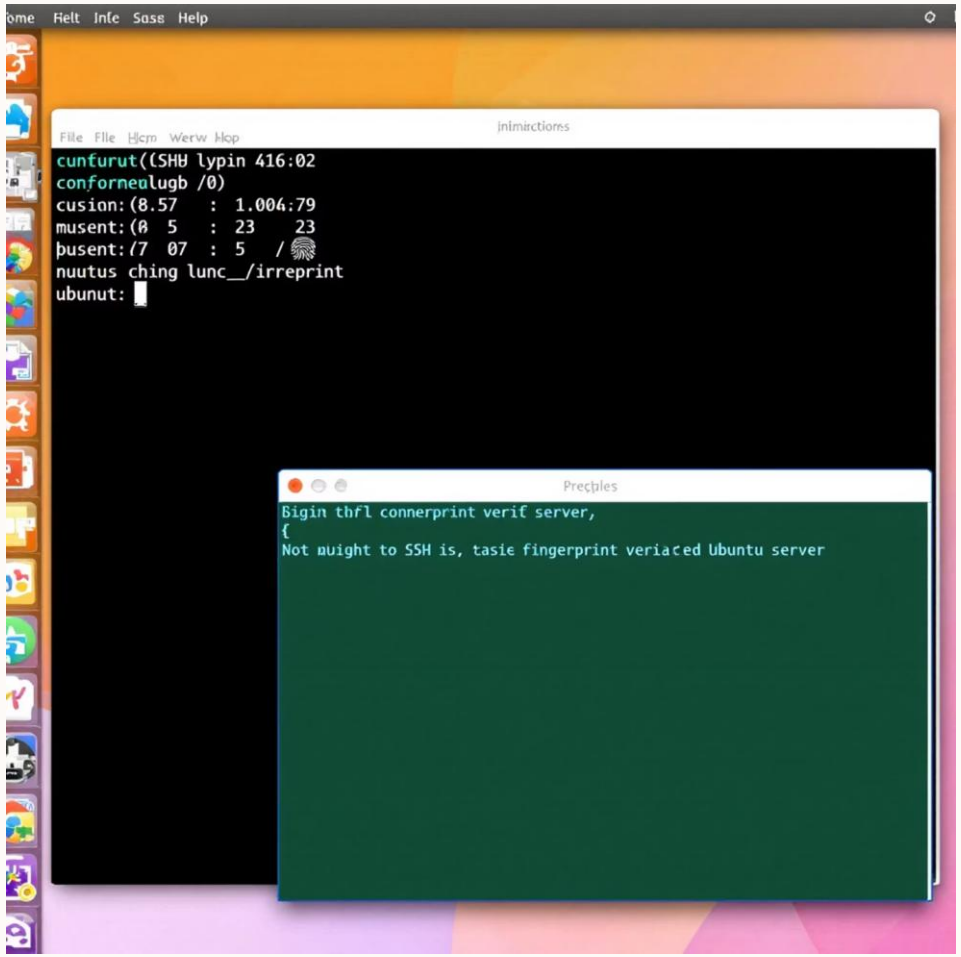
```
ssh 用户名@服务器IP
```

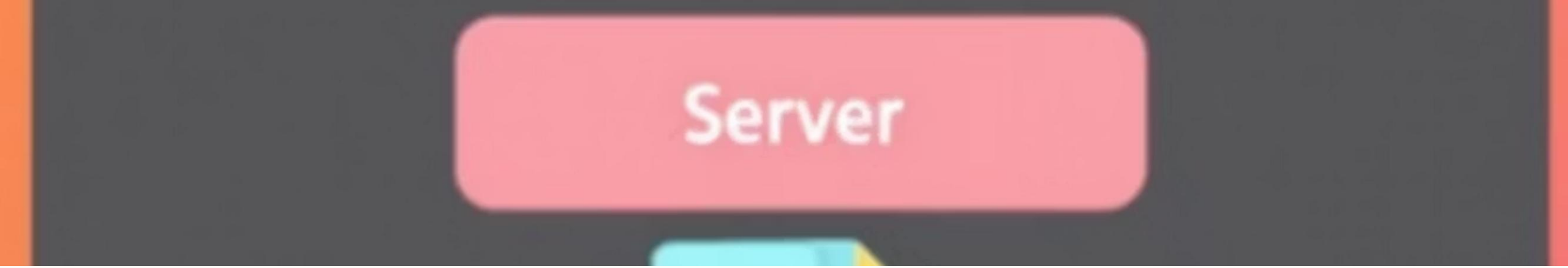
示例：ssh admin@192.168.1.100

连接过程

1. 输入命令后显示服务器指纹(fingerprint)确认消息
2. 输入'yes'确认信任服务器指纹
3. 输入用户账户密码
4. 认证成功后，显示服务器欢迎消息并完成连接

连接后可以像本地计算机一样执行命令。





SSH连接技巧与实用功能

基于密钥的认证增强安全性

无需输入密码即可安全连接的SSH密钥认证设置：

1. 生成密钥：ssh-keygen -t rsa -b 4096
2. 复制密钥到服务器：ssh-copy-id 用户@服务器IP

此后连接时将跳过密码输入过程。

端口转发扩展应用

本地端口转发：ssh -L 8080:localhost:80 user@server

远程端口转发：ssh -R 8080:localhost:80 user@server

通过这种方式可以绕过防火墙或安全访问内部服务。

文件传输功能

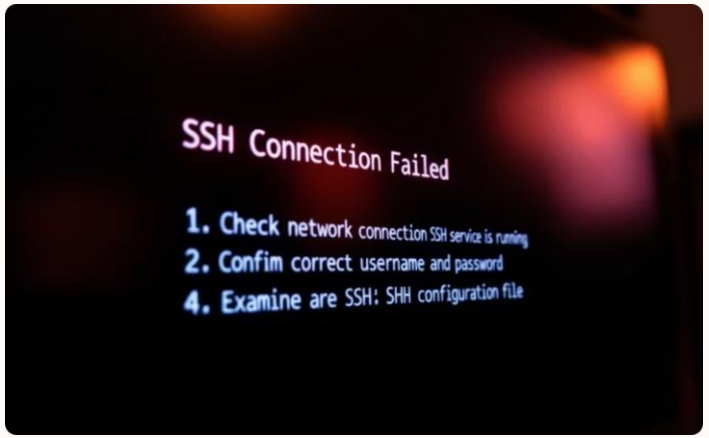
使用SCP传输文件：scp 文件名 用户@服务器IP:路径

使用SFTP进行交互式文件传输：sftp 用户@服务器IP

通过SSH连接安全地收发文件。



连接故障及解决方法



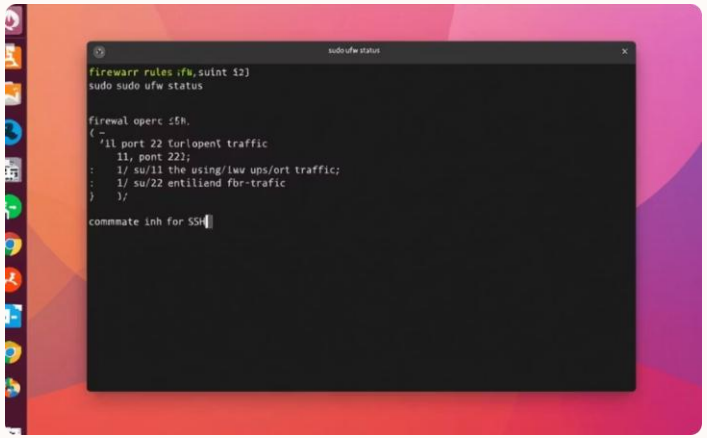
服务问题

SSH连接被拒绝时，检查服务器上SSH服务是否运行：

```
sudo systemctl status ssh
```

如果服务停止，重新启动：

```
sudo systemctl restart ssh
```



防火墙设置检查

防火墙可能阻止SSH连接。检查UFW状态：

```
sudo ufw status
```

如果未允许SSH端口，添加：

```
sudo ufw allow ssh
```



网络连接检查

确认服务器IP地址或主机名是否正确。测试网络连接：

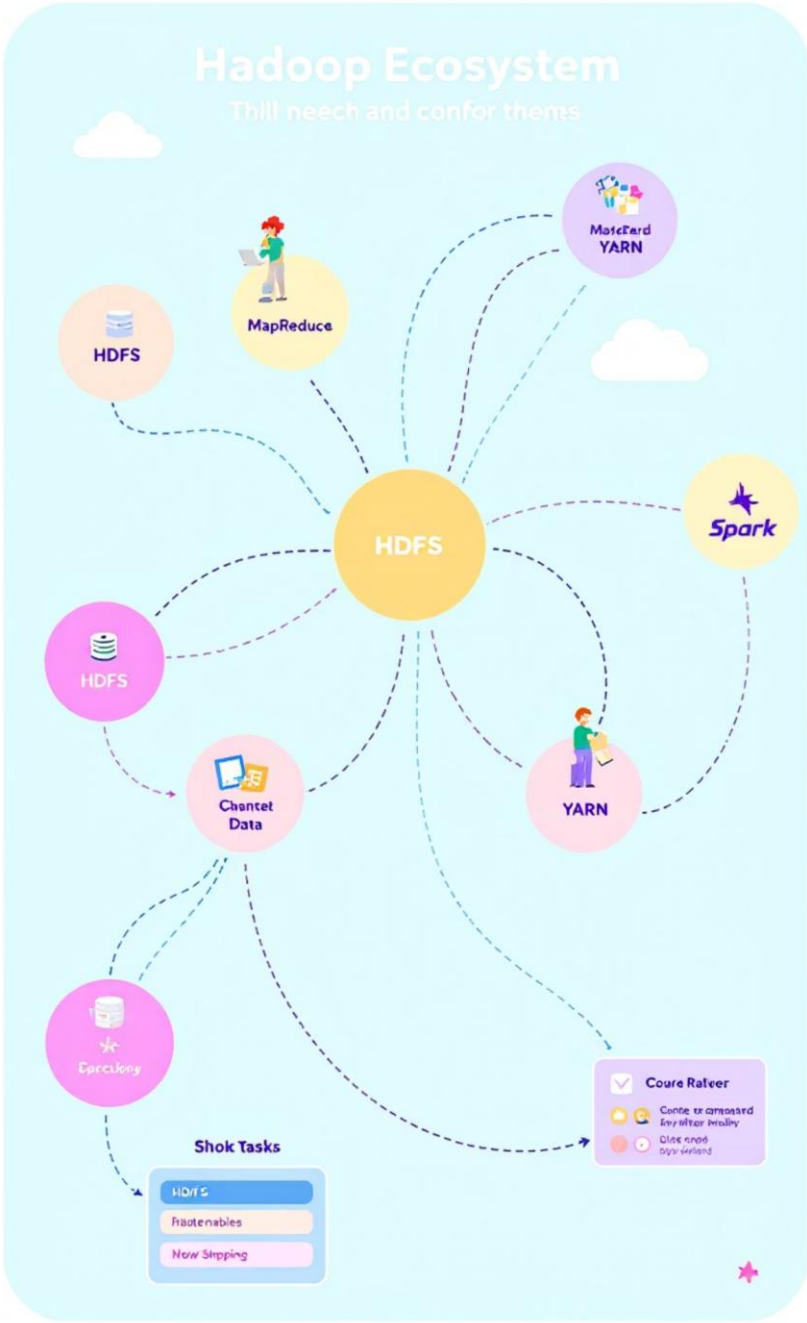
```
ping 服务器IP
```

确认用户账户名和密码是否正确。检查日志文件：

```
sudo tail -f /var/log/auth.log
```

Hadoop 命令解析

本演示将详细解析 Hadoop 命令结构，帮助您理解如何在分布式环境中执行 MapReduce 任务。我们将分析一个实际的 grep 示例，并逐步解释每个命令部分的功能和意义。

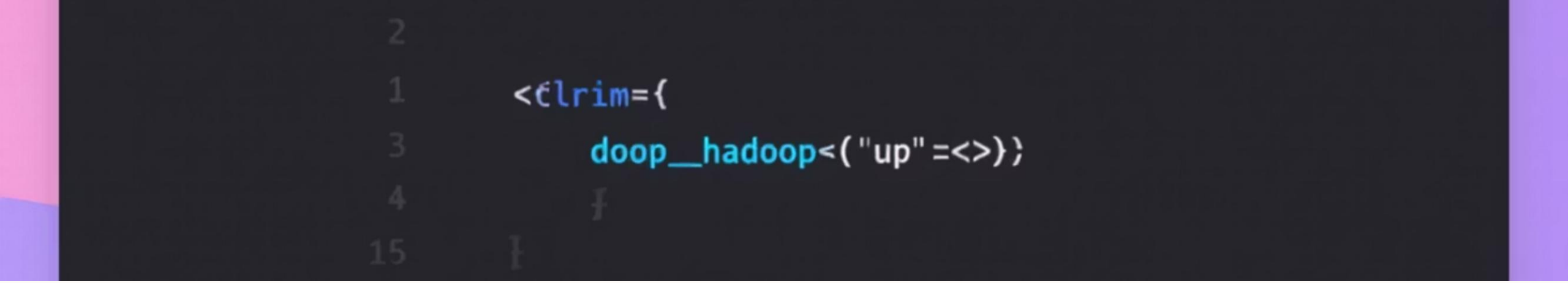


示例命令概览

让我们首先看一下我们将要分析的完整 Hadoop 命令：

```
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar grep ./input ./output 'dfs[a-z.]+'
```

这个命令使用 Hadoop 的 MapReduce 框架在指定的输入目录中搜索匹配特定正则表达式的文本。接下来，我们将逐个分析命令的各个组成部分。



命令组成部分

./bin/hadoop

这是 Hadoop 的可执行脚本，位于 Hadoop 安装目录的 bin 子目录下。它是启动所有 Hadoop 相关操作的入口点。

jar

这个参数表示要运行一个 JAR 文件。在 Hadoop 环境中，MapReduce 作业通常打包为 JAR 文件。

./share/hadoop/mapreduce/hadoadoop-mapreduce-examples-*.jar *.jar

这是 Hadoop 内置的 MapReduce 示例程序包的路径，例如 hadoop-mapreduce-examples-3.3.6.jar。

命令参数解析

grep

这个参数指定要运行的是示例程序中的分布式 grep 功能。它类似于 Unix/Linux 中的 grep 命令，但在分布式环境中执行。

./input

这是输入目录的路径，可以是 HDFS 中的路径或本地文件系统中的路径。Hadoop 将从这个目录读取要处理的数据。

./output

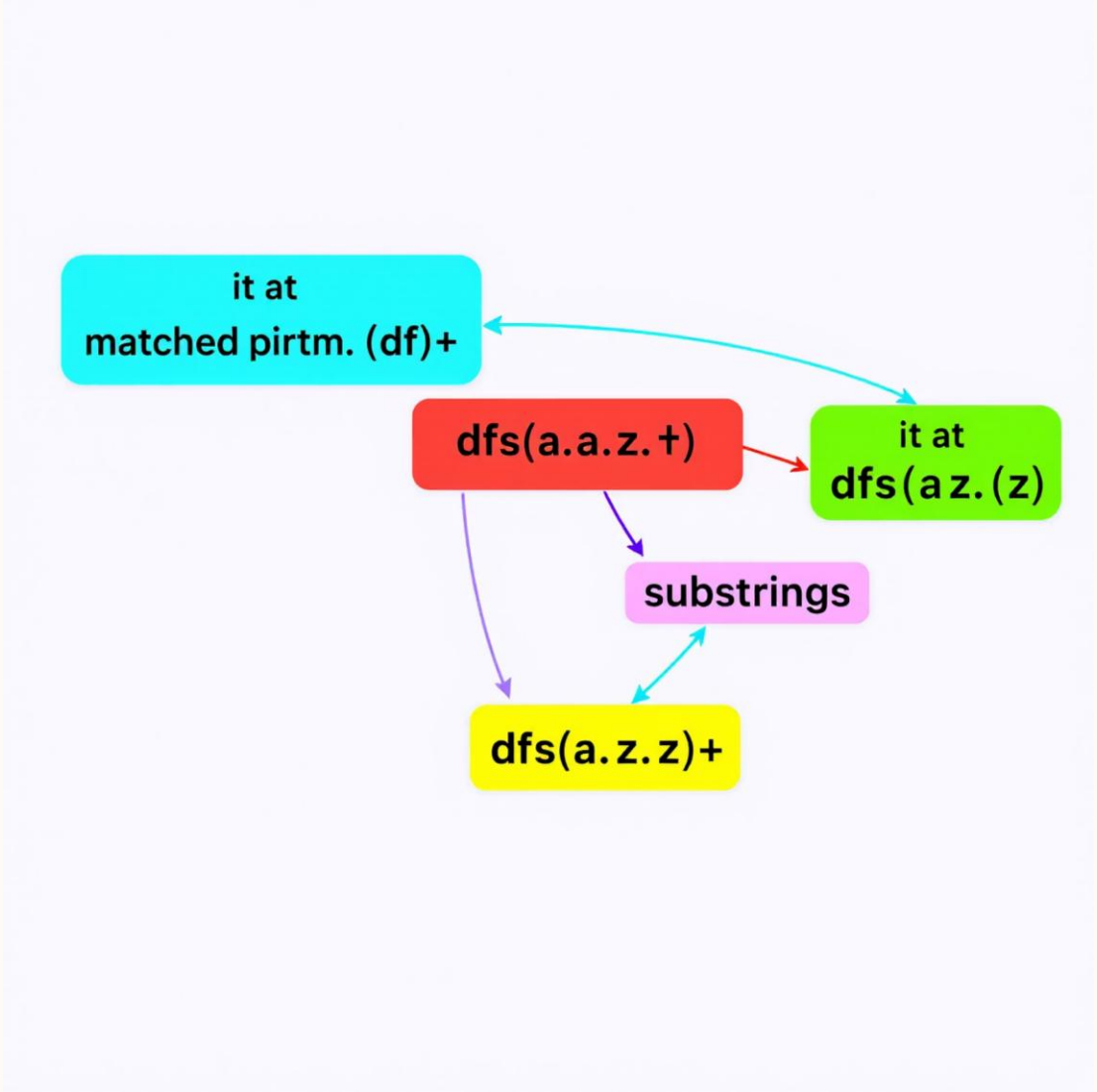
这是输出目录的路径，用于存储处理结果。重要的是，这个目录必须不存在，否则作业会失败。

正则表达式参数

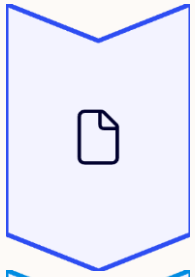
'dfs[a-z.]+'

这是要搜索的正则表达式，它定义了我们要在输入数据中查找的模式：

- 以 "dfs" 开头
- 后面跟着一个或多个小写字母或点号
- 例如会匹配：dfsadmin, dfs.replication 等

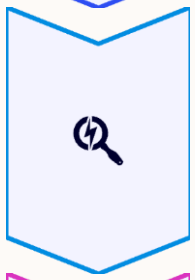


MapReduce 执行流程



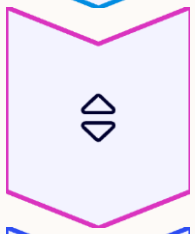
输入阶段

Hadoop 从 ./input 目录读取数据文件，并将其分割成多个输入分片，分配给不同的 map 任务。



Map 阶段

每个 map 任务处理一个输入分片，搜索匹配 'dfs[a-z.]+' 正则表达式的行，并输出匹配结果。



Shuffle & Sort

中间结果被排序和合并，准备进入 reduce 阶段。



Reduce & 输出

Reduce 任务处理排序后的数据，并将最终结果写入 ./output 目录。



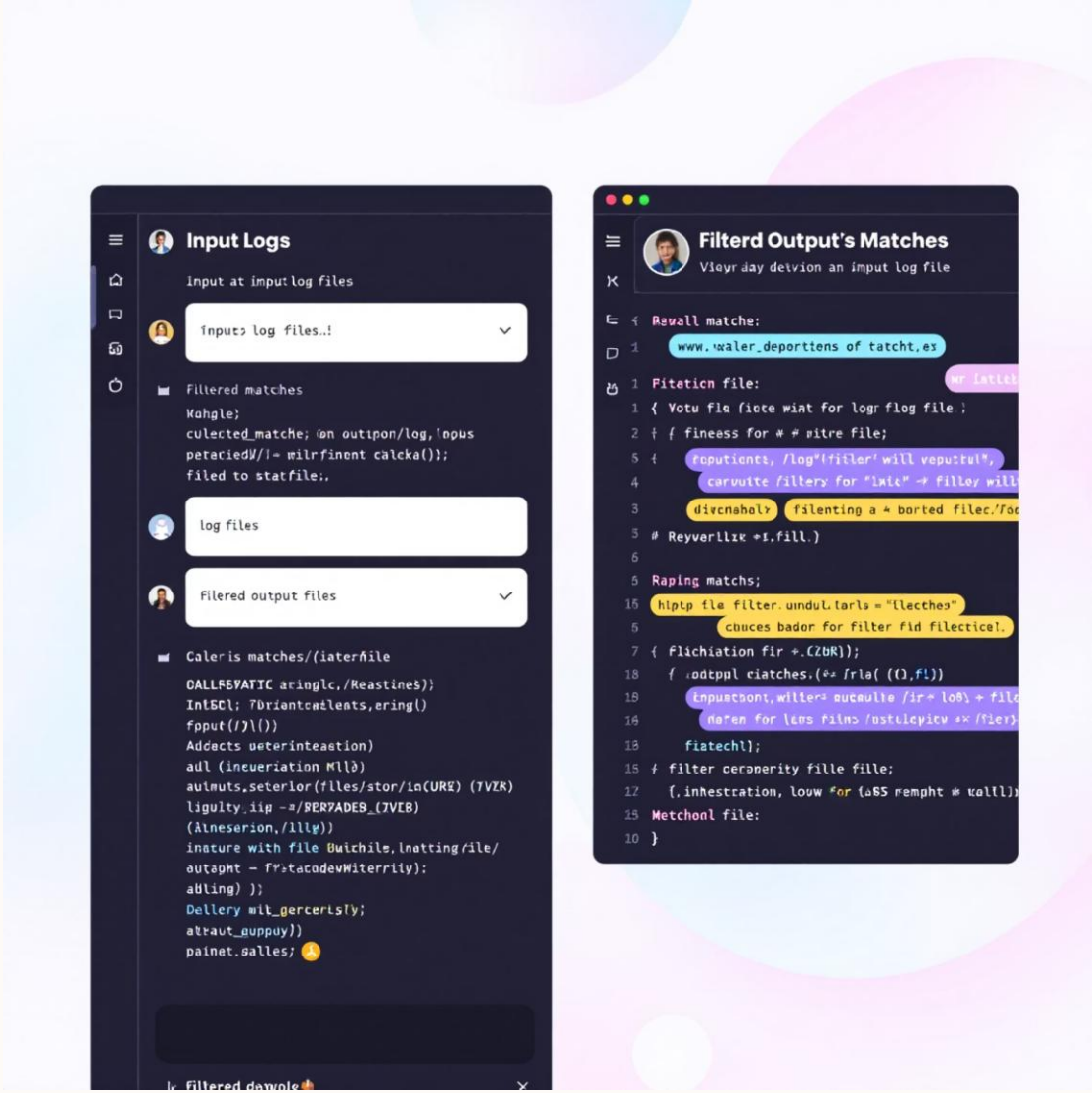
命令执行示例

假设我们有以下日志文件内容：

```
2023-10-15 INFO dfsadmin -report2023-10-15 INFO datanode started2023-10-15 INFO dfs.replication set to 32023-10-15 INFO hdfs directory created
```

执行我们的命令后，输出目录中将包含匹配 'dfs[a-z.]+' 的行：

```
2023-10-15 INFO dfsadmin -report2023-10-15 INFO dfs.replication set to 3
```



总结与最佳实践

命令结构

Hadoop 命令遵循一致的结构：hadoop [命令选项] [主命令] [命令参数]，理解这种结构有助于构建和调试复杂命令。

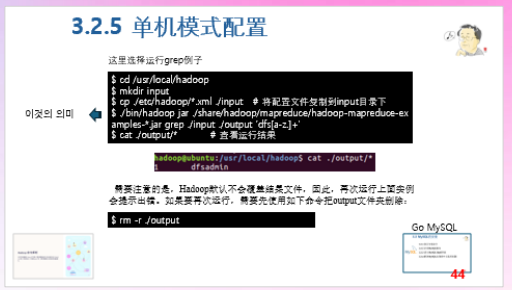
路径考虑

输入路径可以是已存在的，但输出路径必须不存在。始终确保指定正确的 HDFS 或本地文件系统路径。

正则表达式

正则表达式需要用引号括起来，以防止 shell 解释特殊字符。测试复杂的正则表达式前，先在小数据集上验证。

通过本演示，您应该能够理解并构建自己的 Hadoop MapReduce 命令，以执行各种分布式数据处理任务。





Hadoop MapReduce 框架：grep 命令解析

本演示将深入解析 Hadoop MapReduce 框架中 grep 示例的执行过程，探讨分布式数据处理的核心机制。我们将分析命令结构、执行流程、内部工作原理及性能优化技巧，帮助开发者全面理解 Hadoop 的数据处理能力。

Made with GAMMA

命令结构分析

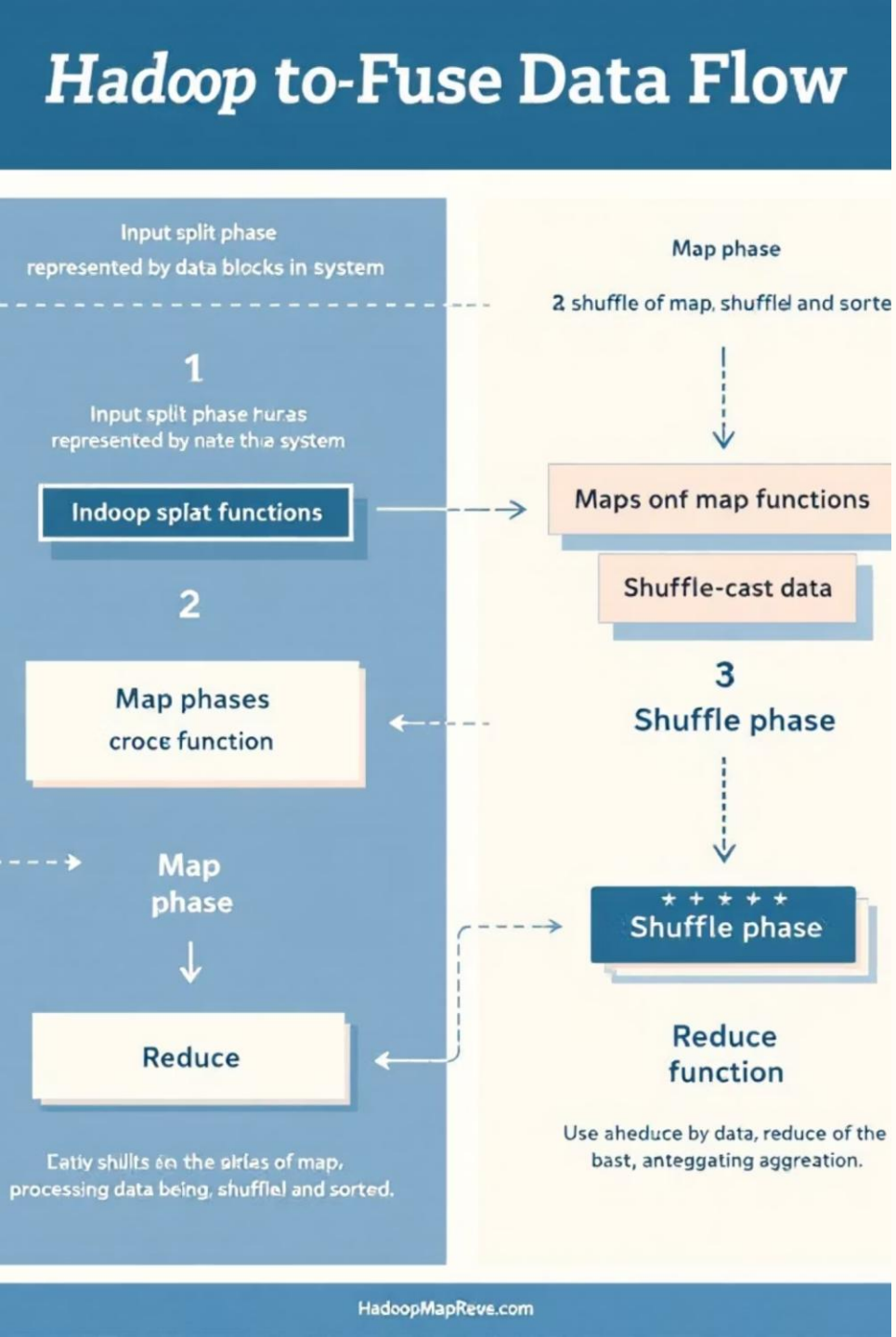
```
./bin/hadoop jar \./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar \grep \input \output \'dfs\[a-z.\]+'
```

这条命令通过 Hadoop MapReduce 框架执行 grep 示例程序，在 HDFS 的 input 目录中搜索匹配正则表达式 'dfs[a-z.]+'的内容，并将结果保存到 output 目录。

1 命令参数解析

- hadoop jar：执行 JAR 文件的命令
- hadoop-mapreduce-examples-3.3.5.jar：包含示例程序的 JAR 包
- grep：要执行的示例程序
- input：HDFS 上的输入目录路径
- output：HDFS 上的输出目录路径
- 'dfs\[a-z.\]+'：搜索的正则表达式模式

Made with GAMMA



MapReduce 执行流程

JAR 文件加载

系统加载 hadoop-mapreduce-examples-3.3.5.jar 文件中的 Grep 类，准备从输入文件中提取匹配正则表达式的行。

输入/输出路径验证

验证 input 目录存在于 HDFS 上，确保 output 目录不存在（必须是新目录，否则任务将失败）。

MapReduce 作业启动

提交作业到 Hadoop 集群，根据配置决定在本地模式或分布式模式下运行。

执行结果输出

将匹配结果写入 output/part-r-00000 文件，格式为"匹配字符串 出现次数"。

正则表达式 'dfs\[a-z.\]+' 匹配所有以 "dfs" 开头，后跟一个或多个小写字母或点号的字符串，如 dfsadmin、dfs.name、dfs.data 等。

Made with GAMMA

MapReduce 处理阶段详解

输入分割

将输入文件分割成大小为 128MB（默认值）的数据块，分配给不同的 Mapper 处理。每个数据块会由集群中的一个节点处理。

Map 阶段

对每行数据应用正则表达式 'dfs\[a-z.\]+'，对匹配的内容输出键值对：(匹配字符串, 1)。

Shuffle 与排序

将相同键的结果（如所有 "dfsadmin" 或所有 "dfs.name" ）分组并发送到同一个 Reducer。

Reduce 阶段

对每个键计算出现总次数，将最终结果 (匹配字符串, 总次数) 写入输出文件。

示例输入 (input/file1.txt):

```
dfsadmin commands datanode startupdfs.name.dir configuratio
n
```

示例输出 (output/part-r-00000):

```
dfsadmin 1dfs.name 1
```

Made with GAMMA

MapReduce 内部代码实现

Mapper 实现 (GrepMapper)

```
public void map(LongWritable key, Text value,
Context context) {  String text = value.toString();  M
atcher matcher = pattern.matcher(text);    while (matc
her.find()) {    context.write(      new Text(matcher.
group()),       new IntWritable(1)    );  }}
```

Mapper 将输入文本与正则表达式进行匹配，每找到一个匹配项就输出键值对 (匹配文本, 1)。

Reducer 实现 (IntSumReducer)

```
public void reduce(Text key,                Iterable
values,                Context context) {  int sum = 0
;  for (IntWritable val : values) {    sum += val.get(
);  }  context.write(key, new IntWritable(sum));}
```

Reducer 接收同一个键的所有值，计算它们的总和，然后输出 (键, 总和) 对。这个过程实现了对匹配模式出现次数的统计。

Made with GAMMA

性能优化与配置选项

输入/输出格式

默认使用 TextInputFormat 和 TextOutputFormat，适用于文本处理。可以根据数据类型选择其他格式如 SequenceFileInputFormat 或自定义格式。

日志监控

使用命令查看详细执行日志：yarn logs -applicationId <application_id>

通过 Web UI（默认 8088 端口）可以实时监控作业状态和资源使用情况。

性能调优

调整 Reducer 数量：-Dmapreduce.job.reduces=2

增加内存分配：-Dmapreduce.map.memory.mb=2048

适当的配置可以显著提高处理效率。

优化 MapReduce 作业需要考虑数据大小、集群资源和作业特性。对于正则表达式匹配这类 I/O 密集型任务，增加并行度（Mapper 数量）通常比增加单个任务的资源更有效。

Made with GAMMA

注意事项与最佳实践



1 输出目录必须不存在

执行前需要确保输出目录不存在，否则作业会失败。可以使用以下命令删除已有目录：

```
hdfs dfs -rm -r /user/hadoop/output
```

2 HDFS 路径使用

使用完整 HDFS 路径时的命令格式：

```
hadoop jar ... grep /input /output 'dfs\[a-z.\]+'
```

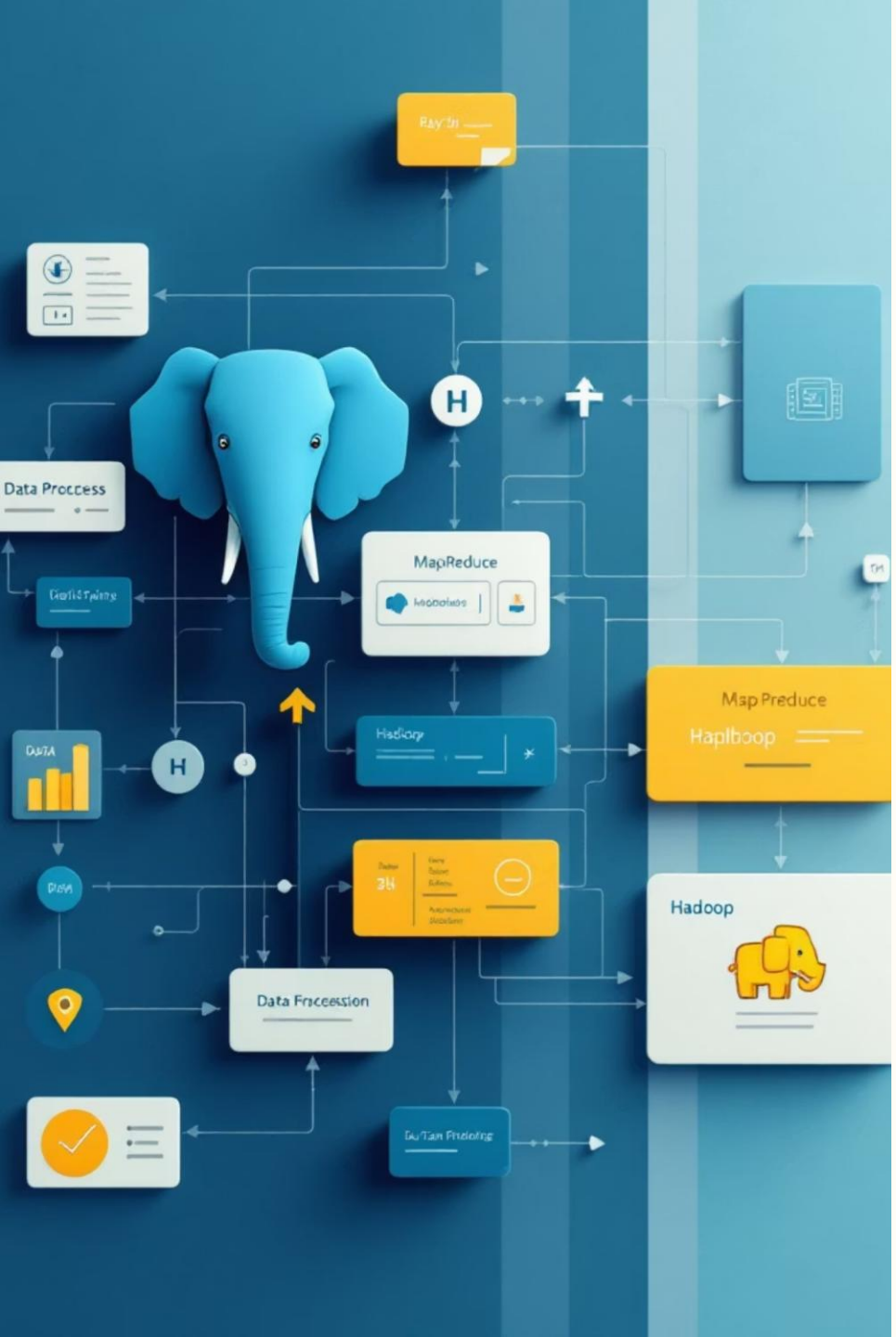
3 本地模式与分布式模式

系统会根据 mapred-site.xml 中的配置自动决定在本地运行还是分布式运行作业。对于小数据集，本地模式速度更快；大数据集则需要分布式处理。



grep 示例展示了 Hadoop MapReduce 的基本工作原理，实际应用中，开发者通常会实现自定义的 Mapper 和 Reducer 类来处理更复杂的数据分析任务。理解这些基础概念对于掌握大数据处理技术至关重要。

Made with GAMMA



Hadoop MapReduce 명령어 분석

이 명령어는 **Hadoop MapReduce 프레임워크를 이용해 grep 예제를 실행**하는 작업입니다. dfs\[a-z.\]+ 정규표현식으로 input 디렉토리의 파일에서 패턴을 검색하고, 결과를 output 디렉토리에 저장합니다. 단계별 동작 원리를 상세히 설명드리겠습니다.

Made with GAMMA

명령어 구조 분석

```
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar \\ grep \\ # 실행할 예제 프로그램 input \\ # 입력 디렉토리 (HDFS 경로) output \\ # 출력 디렉토리 (HDFS 경로) 'dfs\[a-z.\]+' # 검색할 정규표현식
```

명령어 구성요소

- 실행 파일: ./bin/hadoop
- JAR 파일: hadoop-mapreduce-examples-3.3.5.jar
- 예제 프로그램: grep

파라미터

- 입력 경로: input
- 출력 경로: output
- 정규표현식: 'dfs\[a-z.\]+'

Made with GAMMA

단계별 실행 과정

JAR 파일 로드

- hadoop-mapreduce-examples-3.3.5.jar 내장된 Grep 클래스 실행
- **기능**: 입력 파일에서 정규표현식 매칭 라인 추출

입력/출력 경로 확인

- input: HDFS 상의 입력 디렉토리 (예: /user/hadoop/input)
- output: 결과 저장 디렉토리 (자동 생성되며 **기존에 없어야 함**)

MapReduce 작업 실행

단계 설명 **Input Split** 입력 파일을 128MB 블록(default) 단위로 분할 **Map** 각 줄을 읽어 정규식 dfs\[a-z.\]+와 매칭되는지 검사 → (키: 매칭된 문자열, 값: "1") 출력 **Shuffle & Sort** 같은 키(dfsadmin, dfs.name 등)를 같은 리듀서로 그룹핑 **Reduce** 각 키의 등장 횟수 합계 계산 **Output** 결과를 output/part-r-00000 파일에 저장

정규표현식 매칭

- dfs\[a-z.\]+: "dfs"로 시작하고 소문자/점(.)이 1회 이상 반복되는 문자열 (예: dfsadmin, dfs.name, dfs.data 등)

Made with GAMMA

예시 결과

입력 파일 (input/file1.txt)

```
dfsadmin commands datanode startup dfs.name.dir config
uration
```

출력 (output/part-r-00000)

```
dfsadmin 1dfs.name 1
```

Made with GAMMA

내부 동작 상세

Mapper (GrepMapper)

```
public void map(LongWritable key, Text value, Context context) {    String text = value.toString();    Matcher matcher = pattern.matcher(text);    while (matcher.find()) {        context.write(new Text(matcher.group()), new IntWritable(1));    }}
```

Reducer (IntSumReducer)

```
public void reduce(Text key, Iterable values, Context context) { int sum = 0; for (IntWritable val : values) { sum += val.get(); } context.write(key, new IntWritable(sum));}
```

Made with GAMMA

주요 설정 및 옵션

입력/출력 형식

- 입력: TextInputFormat (기본값)
- 출력: TextOutputFormat

실행 로그 확인

```
# YARN 애플리케이션 로그yarn l
ogs -applicationId
```

성능 향상 팁

```
# 리듀서 개수 지정 (예: 2개)-D
mapreduce.job.reduces=2
```

Made with GAMMA

주의사항

출력 디렉토리 미리 존재하면 안 됨

```
hdfs dfs -rm -r /user/hadoop/output # 기존 출력 삭제
```

HDFS 경로 사용 시

```
hadoop jar ... grep /input /output 'dfs\[a-z.\]+'
```

로컬 모드 vs 분산 모드

- mapred-site.xml 설정에 따라 로컬/분산 처리 자동 결정

이 예제는 **Hadoop의 기본 작동 원리**를 이해하는 데 가장 적합합니다. 실제로는 사용자 정의 Mapper/Reducer를 구현해 더 복잡한 분석이 가능합니다.

