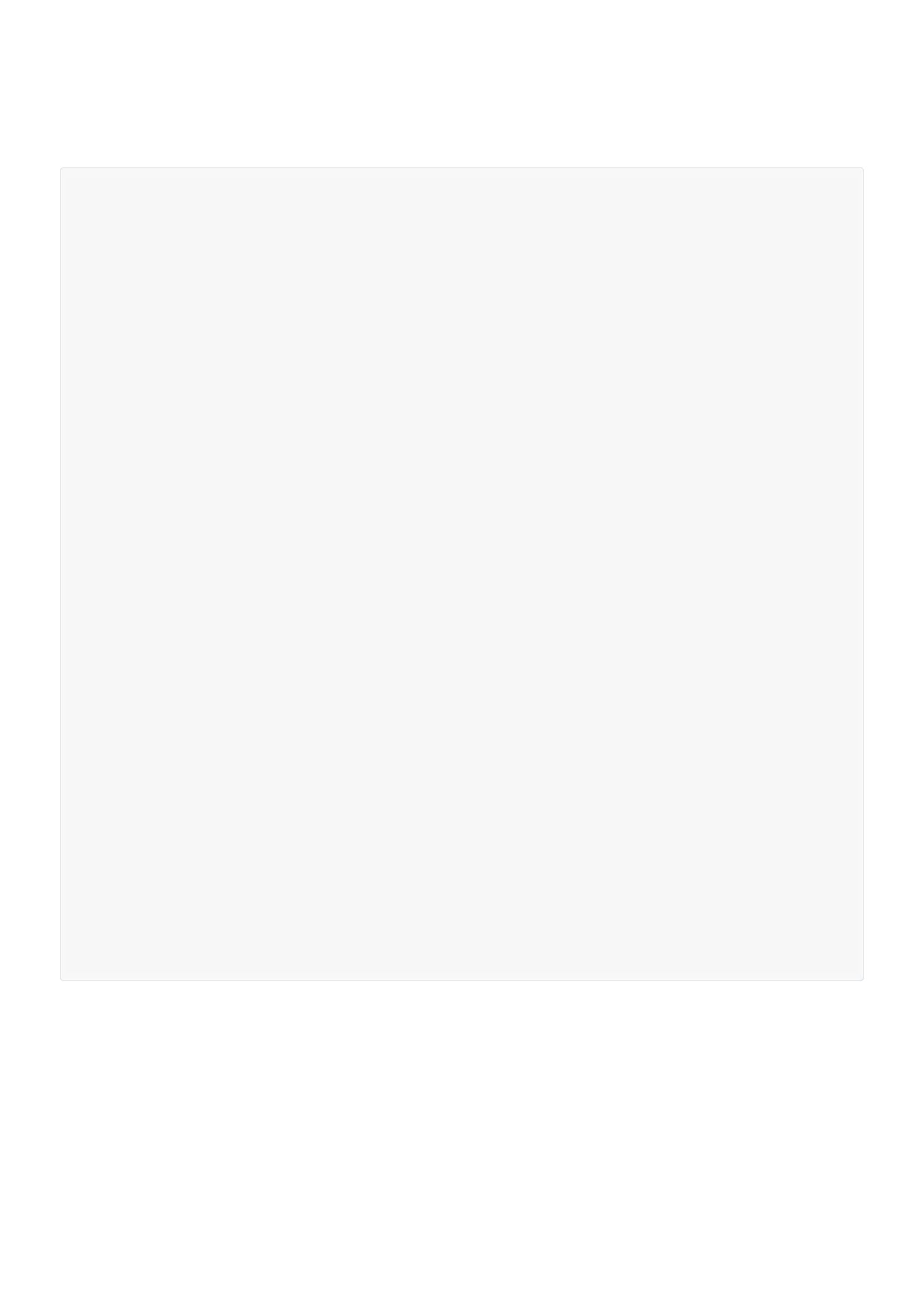
本⽂档遵循知识共享许可协议(Creative Commons license 4.0),以下简称CC 4.0，使⽤者需遵循以下授权内



容：

https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh

您必须遵守下列条件：

1.署名：使⽤本系列⽂档及⽂档内容，您必须注明源作者信息(⻢哥教育张⼠杰⽼师，

Email:2973707860@qq.com)，并提供本许可证的链接(https://creativecommons.org/licenses/by-nc-

sa/4.0/deed.zh),

同时需要声明您的内容是否经过修改，你可以采⽤适当的⽅式将相关修改信息表述出来，但是您不能在信息中以任

何形式暗示授权⼈已正式认可了您以及您的使⽤⾏为。

2.⾮商业性使⽤：您不能将本系列⽂档⽤于其他任何商业⽤途。

3.相同⽅式授权：如您对本系列⽂档进⾏修改或使⽤本⽂档中内容，那么您必须基于本协议分发您的作品(署名源

作则⻢哥教育张⼠杰⽼师并使⽤本许可协议)。

您的权利：

当遵循本协议后，您将有以下权限：

共享 — 允许以⾮商业性质复制本作品。

改编 — 在原基础上修改、复制或以本作品为基础进⾏重新编辑并⽤于个⼈⾮商业使⽤。

只要您遵守本许可协议条款，许可⼈(⻢哥教育张⼠杰⽼师)将授权您使⽤本系列⽂档，如果您违背本许可协议条款，许

可⼈(⻢哥教育张⼠杰⽼师)将有权收回您的授权并保留进⼀步追究其法律责任的权利。

https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en

Under the following terms:

Attribution — You must give appropriate credit(www.magedu.com,Zhang

ShiJie,Email:2973707860@qq.com), provide a link to the license, and indicate if changes

were made. You may do so in any reasonable manner, but not in any way that suggests the

licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute

your contributions under the same license as the original.

⼀：Debian与Ubuntu简介：



1.1：Debian简介：

是从 1993 年由 Ian Murdock(伊恩·默多克) 发起的，受到当时 Linux 与 GNU 的⿎舞，⽬标是成为⼀个公开的发⾏版，经过⼆⼗⼏年的迭代更新Debian 从⼀个⼩型紧密的⾃由软件骇客（hacker）⼩组，逐渐成⻓成今⽇庞⼤且运作良好的开发者与⽤户社群，Debian 的名字是由 Debian 的创始⼈ Ian Murdock (伊恩·默多克) 和他的爱妻Debra(黛布拉) 两⼈的名字组合⽽成的。

Debian 是由⼤约⼀千个分布在世界各地的开发者⽆偿地利⽤他们的业余时间开发的，⽽这些开发者实际上⼤部分都没⻅过⾯，彼此之间的通讯⼤多是通过电⼦邮件（lists.debian.org 上的邮件列表）和 IRC（irc.debian.org 上的#debian 频道）来完成的，⽬前Debian 提供59000多个软件包的维护与更新。

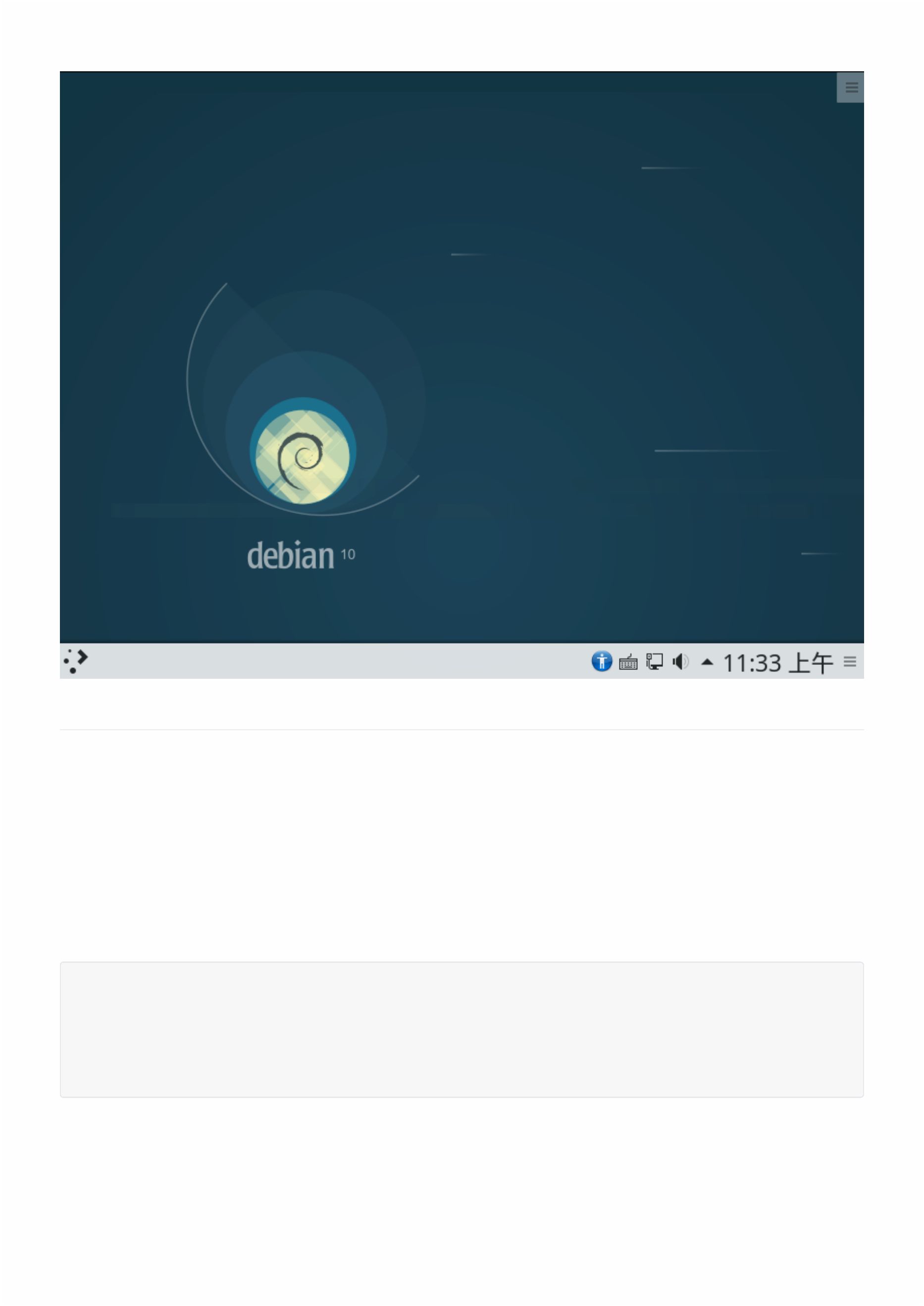
Debian官⽹： https://www.debian.org/

官⽅镜像地址： https://www.debian.org/mirror/list

清华⼤学下载地址： https://mirrors.tuna.tsinghua.edu.cn/debian-cd/10.1.0-live/amd64/iso-

hybrid/

1.2：Ubuntu简介：



Ubuntu（友帮拓、优般图、乌班图）早期是⼀个开源的GNU/Linux操作系统，Ubuntu 是基于DebianGNU/Linux，⽀持x86、amd64（即x64）和ppc架构，由全球化的专业开发团队（Canonical Ltd）打造的，其名称来⾃⾮洲南部祖鲁语或豪萨语的“ubuntu”⼀词，类似儒家“仁爱”的思想，意思是“⼈性”、“我的存在是因为⼤家的存在”，是⾮洲传统的⼀种价值观， Ubuntu基于Debian发⾏版和GNOME桌⾯环境，⽽从11.04版起，Ubuntu发⾏版放弃了Gnome桌⾯环境，改为Unity，与Debian的不同在于它每6个⽉会发布⼀个新版本，Ubuntu的⽬标在于为⼀般⽤户提供⼀个最新的、同时⼜相当稳定的主要由⾃由软件构建⽽成的操作系统，Ubuntu具有庞⼤的社区⼒量，⽤户可以⽅便地从社区获得帮助，Ubuntu对GNU/Linux的普及特别是桌⾯普及作出了巨⼤贡献，由此使更多 ⼈共享开源的成果与精彩。

Canonical公司是⼀个私营公司，由南⾮的企业家⻢克·沙特尔沃思(Mark Shuttleworth)创建，主要为了促进开源

软件项⽬，Canonical在⻢恩岛登记注册，其雇员分布在世界各地，其主要办事处在伦敦，在蒙特利尔也有办事处，

Canonical公司⽀持并创建了⼏个项⽬，主要都是⾃由/开源软件（FOSS）或是⼀些旨在改善⾃由软件开发者和贡献者

的⼯具。

http://www.techweb.com.cn/business/2013-08-20/1318187.shtml #ubuntu公司⽬前并未实现盈利

Ubuntu是各种公有云上最受欢迎的操作系统，这是因为Ubuntu与各⼤云运营商密切合作，针对云服务商的基础架

构优化内核和系统，加快安全更新速度，并最⼤限度降低⽤户默认需要的⽹络或存储成本。

https://cn.ubuntu.com/ #中⽂官⽹



https://ubuntu.com/download #英⽂官⽹

1.3：Ubuntu历史发⾏版本：

Ubuntu 没有像其它软件⼀样有 1.0 版本，是因为其第⼀个版本是发布于 2004 年，所以Ubuntu的⽣⽇是2004年的10⽉20⽇。

ubuntu的版本号有个固定的特征，其使⽤⼀个形容词和⼀个动物名称组成，并且形容词和名词的⾸字⺟都是⼀致的，Ubuntu版本的命名规则是根据正式版发布的年⽉命名，Ubuntu 16.04 也就意味着 2016年4⽉发布的Ubuntu，研发⼈员与⽤户可从版本号码就知道正式发布的时间。

每两年的 4 ⽉份，都会推出⼀个⻓期⽀持版本（LTS），其⽀持期⻓达五年，⽽⾮ LTS 版本的⽀持期通常只有半

年。

https://baike.baidu.com/item/ubuntu #Ubuntu历史版本⼀览表

版本 别名 发布⽇期

4.10 Warty Warthog(⻓疣的疣猪) 2004年10⽉20⽇

5.04 Hoary Hedgehog(灰⽩的刺猬) 2005年4⽉8⽇

5.10 Breezy Badger(活泼的獾) 2005年10⽉13⽇

6.06(LTS) Dapper Drake(整洁的公鸭) 2006年6⽉1⽇

6.10 Edgy Eft(急躁的⽔蜥) 2006年10⽉6⽇

7.04 Feisty Fawn(坏脾⽓的⼩⿅) 2007年4⽉19⽇

7.10 Gutsy Gibbon(勇敢的⻓臂猿) 2007年10⽉18⽇

8.04(LTS) Hardy Heron(耐寒的苍鹭) 2008年4⽉24⽇

8.10 Intrepid Ibex (勇敢的野⼭⽺) 2008年10⽉30⽇

9.04 Jaunty Jackalope(得意洋洋的怀俄明野兔) 2009年4⽉23⽇

9.10 Karmic Koala(幸运的考拉) 2009年10⽉29⽇

10.04(LTS) Lucid Lynx(清醒的猞猁) 2010年4⽉29⽇

11.10 Oneiric Ocelot(梦幻的豹猫) 2010年10⽉13⽇

11.04 Natty Narwhal(敏捷的独⻆鲸) 2011年4⽉28⽇

12.04(LTS) Precise Pangolin(精准的穿⼭甲) 2012年的4⽉26⽇

12.10 Quantal Quetzal(量⼦的绿咬鹃) 2012年的10⽉20⽇

13.04 Raring Ringtail(铆⾜了劲的猫熊) 2013年4⽉25⽇

13.10 Saucy Salamander(活泼的蝾螈) 2013年10⽉17⽇



14.04(LTS) Trusty Tahr (可靠的塔尔⽺) 2014年4⽉18⽇

Utopic Unicorn(乌托邦独⻆兽) 2014年10⽉23⽇

15.04 Vivid Vervet (活泼的⼩猴) 2015年4⽉

15.10 Wily Werewolf (狡猾的狼⼈) 2015年10⽉

16.04(LTS) Xenial Xerus (好客的地松⿏) 2016-04-21

16.10 Yakkety Yak(喋喋不休的牦⽜) 2016-10-13

17.04 Zesty Zapus(热情的美洲林跳⿏) 2017-04-13

17.10 Artful Aardvark(巧妙的⼟豚) 2017-10-19

18.04(LTS) Bionic Beaver(仿⽣的海狸) 2018-04-26

18.10 Cosmic Cuttlefish(宇宙的墨⻥) 2018-10-18

19.04 Disco Dingo(迪斯可的澳洲野⽝) 2019-04-18

19.10 Eoan Ermine(短尾的⻩⿏狼) 2019-10-17

1.4：Ubuntu使⽤场景：

Ubuntu系统在物联⽹、桌⾯系统、云环境、服务器系统与容器有⼴泛的使⽤场景，如下：

1.4.1：Ubuntu 桌⾯系统：

Ubuntu作为个⼈电脑既快速⼜安全、开源、免费，有数以千计的应⽤程序可供选择。从⼯作学习、到娱乐游戏，

⽆所不能。

1.4.1.1：Ubuntu桌⾯系统简介：

1.4.1.2：基于Ubuntu的优麒麟：



Canonical，⼯业和信息化部软件与集成电路促进中⼼（CSIP）以及国防科学技术⼤学（NUDT）携⼿打造，⽬前，开

发团队正在与开源社区合作打造Ubuntu的核⼼功能，以更好地适应中国⽤户，除了Ubuntu软件中⼼推出的本⼟化版本

以外，优麒麟还带最常⽤的中⽂程序。

1.4.1.3：基于Ubuntu的Elementary OS：

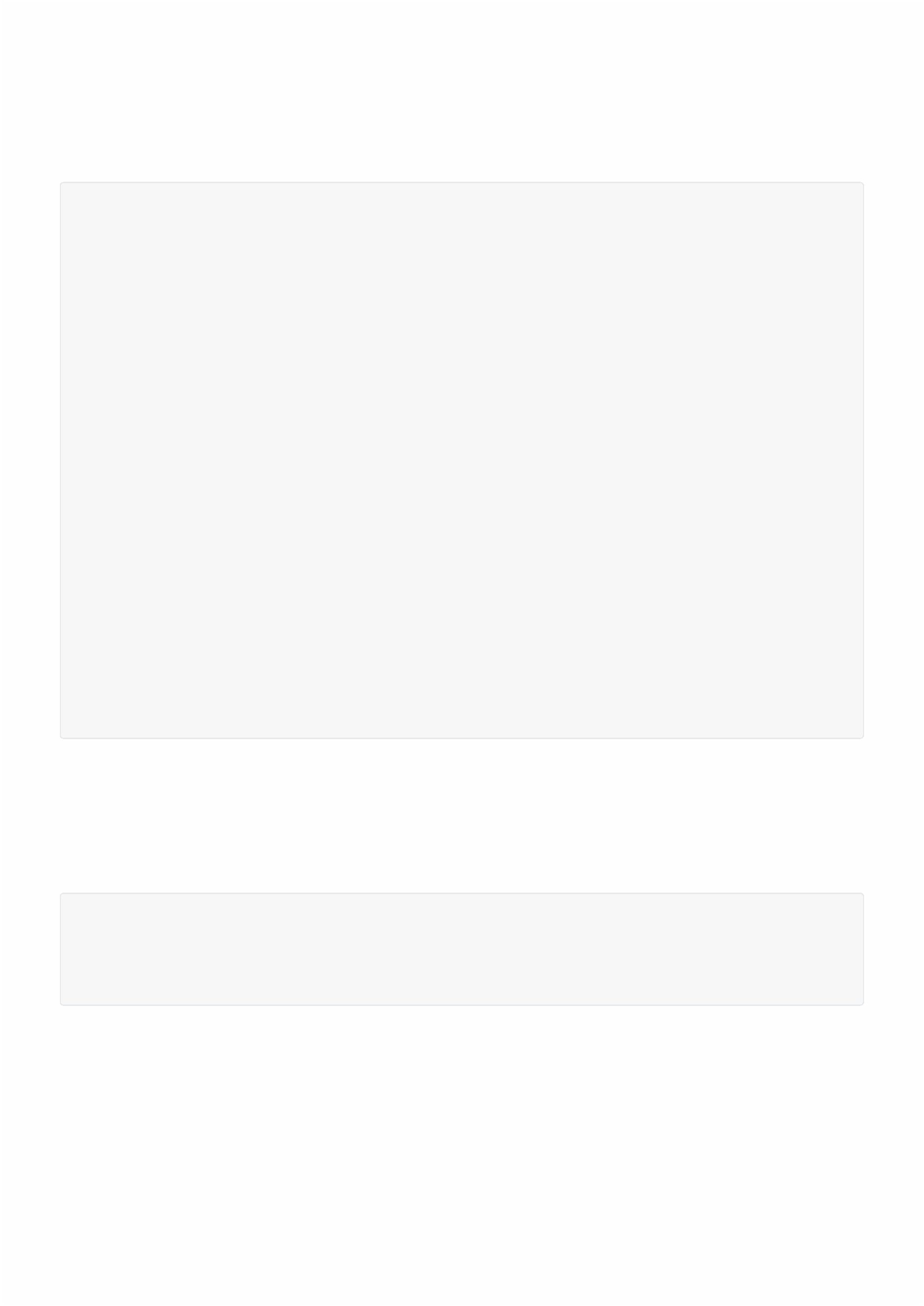
Elementary OS系统是⼀款基于Ubuntu精⼼打磨美化的桌⾯ Linux 发⾏版，号称“最漂亮的Linux系统”，它最早

是Ubuntu的⼀个美化主题项⽬，后来太受欢迎演变成了独⽴的发⾏版，Elementary OS 由设计师主导开发，⻛格与

macOS 相似，它借鉴了ChromeOS的菜单设计，系统核⼼依然是 Ubuntu并完全兼容所有Ubuntu软件包，

elementary OS从界⾯外观到系统设计都很精美，视觉效果很好！

1.4.1.4：基于Ubuntu的其他分⽀：



Ubuntu版本之外，Ubunt除u官了⽅标还准有⼏⼤主要分⽀，分别是Edubuntu、Kubuntu、Lubuntu、 Mythbuntu、Ubuntu MATE，Ubuntu GNOME、Ubuntu Kylin、Ubuntu Studio和Xubuntu

Edubuntu是Ubuntu的教育发⾏版，专注于学校(教育)的需求，是由Ubuntu社区和K12-LTSP社区合作开发的，适合

⼉童、学⽣、教师使⽤的基础发⾏版，其内置了⼤量适合教学的应⽤软件和游戏。

Kubuntu是使⽤KDE桌⾯管理器取代GNOME桌⾯管理器作为其默认的桌⾯管理器的版本。Kubuntu的推出，为喜爱KDE

桌⾯环境的使⽤者的安装和使⽤带来了很⼤的便利。

Lubuntu是⼀个后起之秀，以轻量级桌⾯环境LXDE替代Ubuntu默认的Unity。由于LXDE是⼀个轻量级桌⾯环境，所以

Lubuntu所需的计算机资源很少，⼗分适合追求简洁或速度，以及还在使⽤⽼旧硬件的朋友选⽤。

Mythbuntu是⼀个⽤来实现媒体中⼼的Ubuntu发⾏版本，其核⼼组件是MythTV，所以Mythbuntu可以视为Ubuntu和

MythTV的结合体。

Ubuntu GNOME是采⽤GNOME3作为Ubuntu默认桌⾯管理器的发⾏版本。由于Ubuntu的默认桌⾯环境是Unity，为了

满⾜Linux⽤户的不同需求和使⽤习惯，Ubuntu GNOME项⽬应运⽽⽣。

Ubuntu Kylin(优麒麟)是⼀个专⻔为中⽂⽤户定制的Ubuntu版本，预置了⼤量中国⽤户熟悉的应⽤，是开箱即⽤的

Ubuntu官⽅中国定制版本，适合中国⽤户使⽤。

Ubuntu Studio则是--个为专业多媒体制作⽽打造的Ubuntu版本，可以编辑和处理⾳频、视频和图形图像等多媒体⽂

件，对于多媒体专业⼈⼠⽽⾔，是⼀个⻥和熊掌兼得的好选择。

Xubuntu采⽤了⼩巧和⾼效的Xfce作为桌⾯环境，界⾯简约，类似于GNOME2，功能全⾯，系统资源消耗较⼩，是追求

速度和低配置计算机⽤户的福⾳，同时也为⽼旧计算机提供了发挥余热的机会

1.4.1.5：基于debain的deepin：

deepin：基于debain的发⾏版，Deepin原名Linux Deepin、deepin os、深度系统、深度操作系统，在2014年4⽉改名Deepin，由武汉深之度科技有限公司开发，被誉为“最美国产操作系统”，其以桌⾯应⽤为主的开源GNU/Linux 操作系统，⽀持笔记本、台式机和⼀体机。

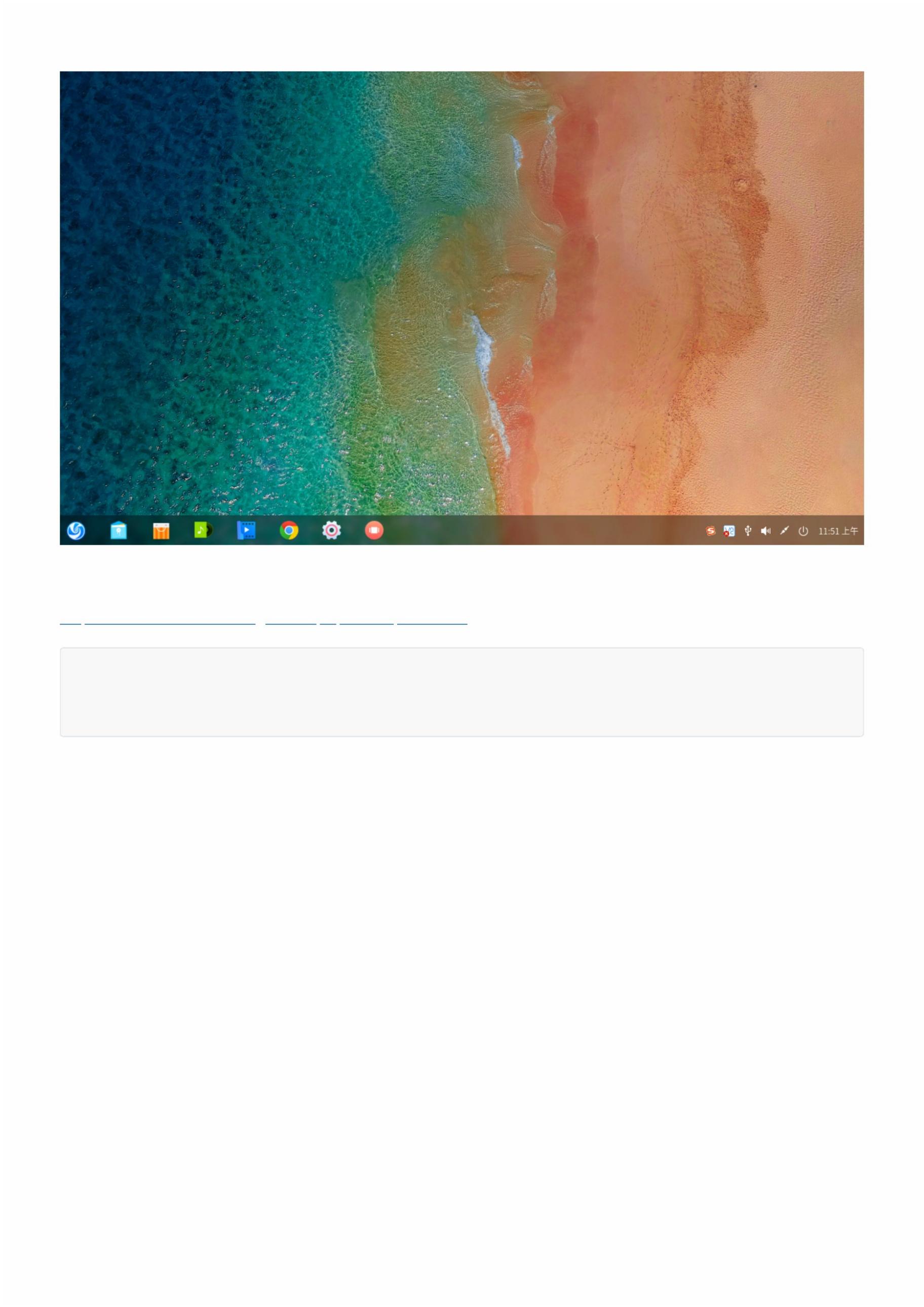
深度操作系统（deepin）是中国第⼀个具备国际影响⼒的Linux发⾏版本，截⽌⾄2019年7⽉25⽇，深度操作系统⽀

持33种语⾔，⽤户遍布除了南极洲的其它六⼤洲，深度桌⾯环境（deepinDDE）和⼤量的应⽤软件被移植到了包括

Fedora、Ubuntu、Arch等⼗余个国际Linux发⾏版和社区，在开源操作系统统计⽹站DistroWatch上，deepin⻓

期位于世界前⼗。

1.4.1.6：其他Linux发⾏版：



<http://www.distrowatch.org/index.php?dataspan=2018>

Manjaro是⼀款基于Arch Linux对⽤户友好的Linux发⾏版，使⽤pacman管理安装包。

MX Linux是基于Debian稳定分⽀的⾯向桌⾯的Linux发⾏版本。

elementary OS是⼀个基于ubuntu的衍⽣版本，早期是⼀个ubuntu主题，后来衍⽣为⼀个专⻔的发⾏版。

1.4.2：Ubuntu与云环境：



https://cn.ubuntu.com/cloud

1.4.2.1：Ubuntu与私有云：

ubuntu在私有云环境⼤量在openstack、容器与kubernetes环境使⽤。

openstack解决⽅案：

openstack客户：



ubuntu与kubernetes：

1.4.2.2：Ubuntu与公有云：



是各种公有云上最受欢迎的U操b作un系t u统，这是因为我们与各⼤云运营商密切合作，针对他们的基础架构优化

内核和系统，加快安全更新速度，并最⼤限度降低⽤户默认需要的⽹络或存储成本。

1.4.3：Ubuntu server系统：

Ubuntu Server 可让您的公共或私有数据中⼼在经济和技术上都具有出⾊的可扩展性。⽆论是部署 OpenStack云、Hadoop 集群还是 50,000 个节点的⼤型渲染场，Ubuntu Server 都能提供性价⽐最佳的横向扩展能⼒。

1.4.3.1：Ubuntu Server 18.04 LTS 新增功能：

<https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes/>

由 Canonical 提供到 2023 年的⻓期⽀持

可在所有主流架构上运⾏ - x86、x86-64、ARM v7、ARM64、POWER8 和 IBM s390x (LinuxONE)

占⽤空间更⼩、启动速度更快的全新 ubuntu-minimal 映像

通过 chrony 实现快速准确的时间同步 新的默认服务器安装程序 ISO 带来全新界⾯和更快的安装速度

⽀持 ZFS（⾮常适合服务器和容器的新⼀代卷管理/⽂件系统）

LXD 3.0 - 提供包括集群、Qos 和资源控制（CPU、内存、块 I/O/ 图形处理器和存储配额）等在内的多种 Linux

容器

更新了 LXD (v3.0)、DPDK (v17.11.1)、Postgresql (v10.3)、Libvirt (v4.0)、NGINX (v1.13)、

Qemu (v2.11.1)、Docker (v17.03)、Puppet (v4.10)、MySQL (v5.7)、PHP (v7.2) 等

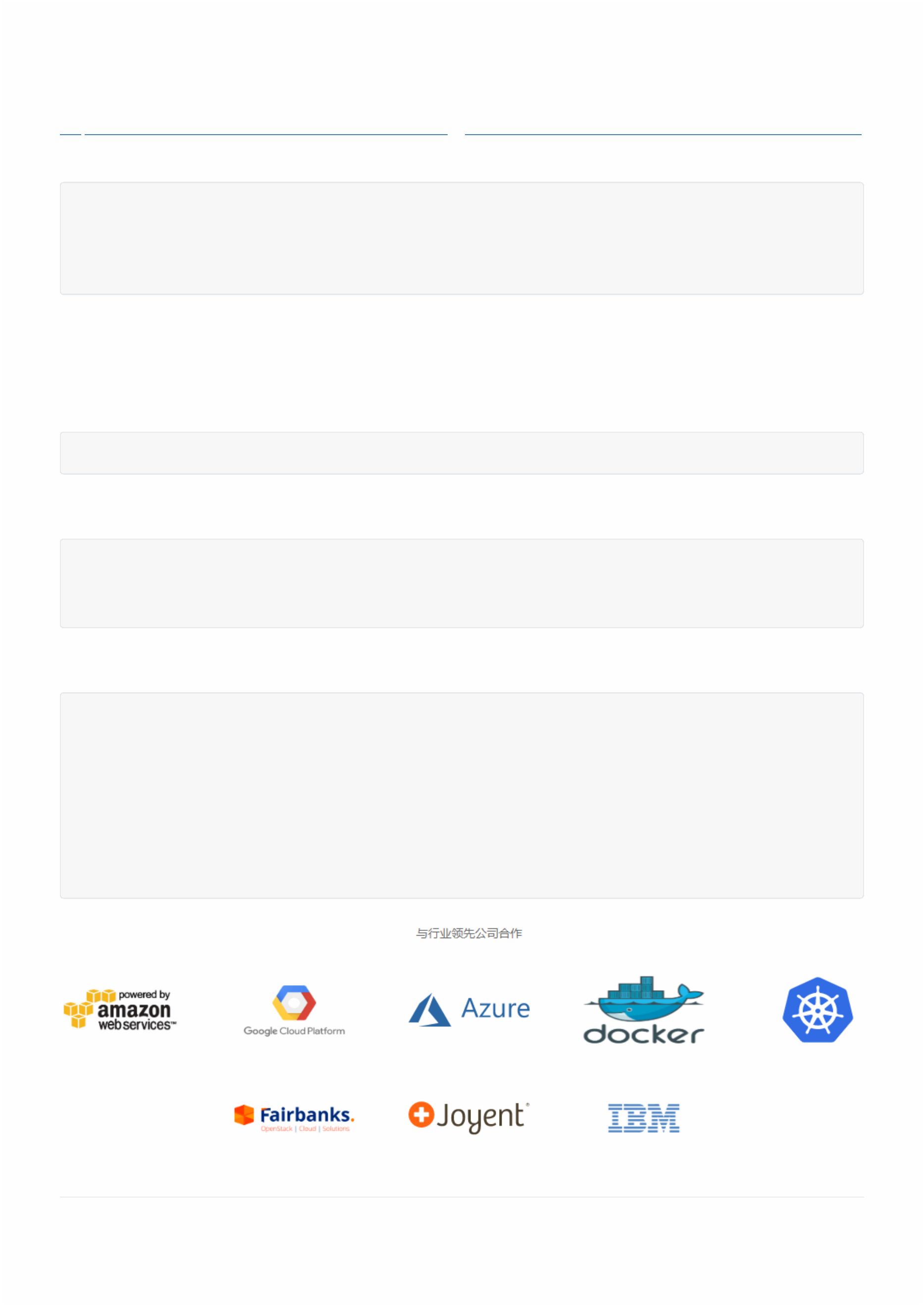
安装 snap 来⽀持简单的应⽤程序安装和发布管理

Linux 4.15 内核

获得 AWS、Microsoft Azure、Joyent、IBM、Google Cloud Platform 和 Rackspace 客户机认证

通过 netplan.io 轻松配置⽹络连接

1.4.3.2：Ubuntu Server 19.10 新增功能：



[htt](https://wiki.ubuntu.com/EoanErmine/ReleaseNotes?_ga=2.36184999.620886555.1573809180-329374192.1565688951)~~[ps://wiki.ubuntu.com/EoanErmine/ReleaseNotes](https://wiki.ubuntu.com/EoanErmine/ReleaseNotes?_ga=2.36184999.620886555.1573809180-329374192.1565688951)~~[?\_g](https://wiki.ubuntu.com/EoanErmine/ReleaseNotes?_ga=2.36184999.620886555.1573809180-329374192.1565688951)~~[a=2.36184999.620886555.1573809180-329374192.1](https://wiki.ubuntu.com/EoanErmine/ReleaseNotes?_ga=2.36184999.620886555.1573809180-329374192.1565688951)~~[565688951](https://wiki.ubuntu.com/EoanErmine/ReleaseNotes?_ga=2.36184999.620886555.1573809180-329374192.1565688951)

由Canonical提供9个⽉的⽀持 Linux 5.3内核

更新为qemu（v4.0），libvirt（v5.4），mysql（v8.0），postgresql（v11）等

新的修复程序集和对Ubuntu Server安装程序的刷新

新的Ubuntu优势体验

1.4.4：Ubuntu与容器环境：

Ubuntu是容器的⾸选平台，从LXD到Kubernetes再到Docker，Canonical与⾏业领先的公司合作，提供全⽅位的

技术和服务，帮助⽤户在公有云、私有云和混合云或裸机云上⼤规模运⾏容器。

https://github.com/moby/moby/blob/master/CHANGELOG.md #docker的历史版本更新记录

1.4.4.1：企业级Docker：

与Docker Inc.密切合作，在Ubuntu上提供商业⽀持的集成式Docker引擎。

Docker引擎的映像以snap包的形式发布，并会⾃动进⾏事务性更新。

Canonical为CS Docker引擎提供1级和2级技术⽀持，Docker Inc.则负责3级⽀持。

1.4.4.2：Canonical的Kubernetes发⾏版：

内置弹性扩展能⼒：

Canonical的Kubernetes简单易⽤，⽽且本身具有弹性。

规模化成本效益：

Canonical的Kubernetes使⽤Juju进⾏建模，并拥有管理持久性集群所需的所有操作脚本和⼯具，包括升级和弹

性扩展。

随时可⽤的分析功能：

可轻松与Prometheus集成以进⾏监控，与Ceph集成以进⾏存储，以及与Elastic Stack(包括Kibana)集成以进

⾏分析和可视化。

1.5：Ubuntu镜像下载：

Ubuntu这么好，哪⾥可以下载的到呢？



http://cdimage.ubuntu.com/releases/ #ubuntu server(服务器版)

http://releases.ubuntu.com/ #ubuntu desktop(

桌⾯版)

http://cdimage.ubuntu.com/daily-live/current/ #20.04每天镜像

1.5.1：系统镜像版本区别：

<https://packages.ubuntu.com/search?lang=zh-cn&arch=any&keywords=libfuse-dev>

ubuntu-18.04.3-live-server-amd64.iso #带live，ISO镜像提供不安装就可以试⽤系统的功能

ubuntu-18.04.3-server-amd64.iso #不带live，不可⽤试⽤，但是可以直接进⾏系统安装

1.5.2：不同CPU指令集的ISO镜像：

CPU架构 安装包标识 备注 镜像名称示例

x86---------->i386 #32位，server版已不再⽀持32位 ubuntu-16.04.6-desktop-i386.iso

x86-64------->amd64 #64位 ubuntu-18.04.3-server-amd64.iso

ARM v7------->ARM64 #arm平台 ubuntu-18.04.3-server-arm64.iso

IBM s390x-------->s390x #IBM System z ubuntu-18.04.3-server-s390x.iso

POWER------->PPC64el #PowerPC ubuntu-18.04.3-server-ppc64el.iso

preinstalled-server-arm64+raspi3 #预安装的树莓派系列 ubuntu-18.04.3-preinstalled-server-

arm64+raspi3.img.xz:

⼆：Ubuntu Server版安装及使⽤：

ubuntu 18.04服务器版本和桌⾯版的安装介绍

<https://help.ubuntu.com/>#官⽅⽂档

2.1：Ubuntu Server版本安装：

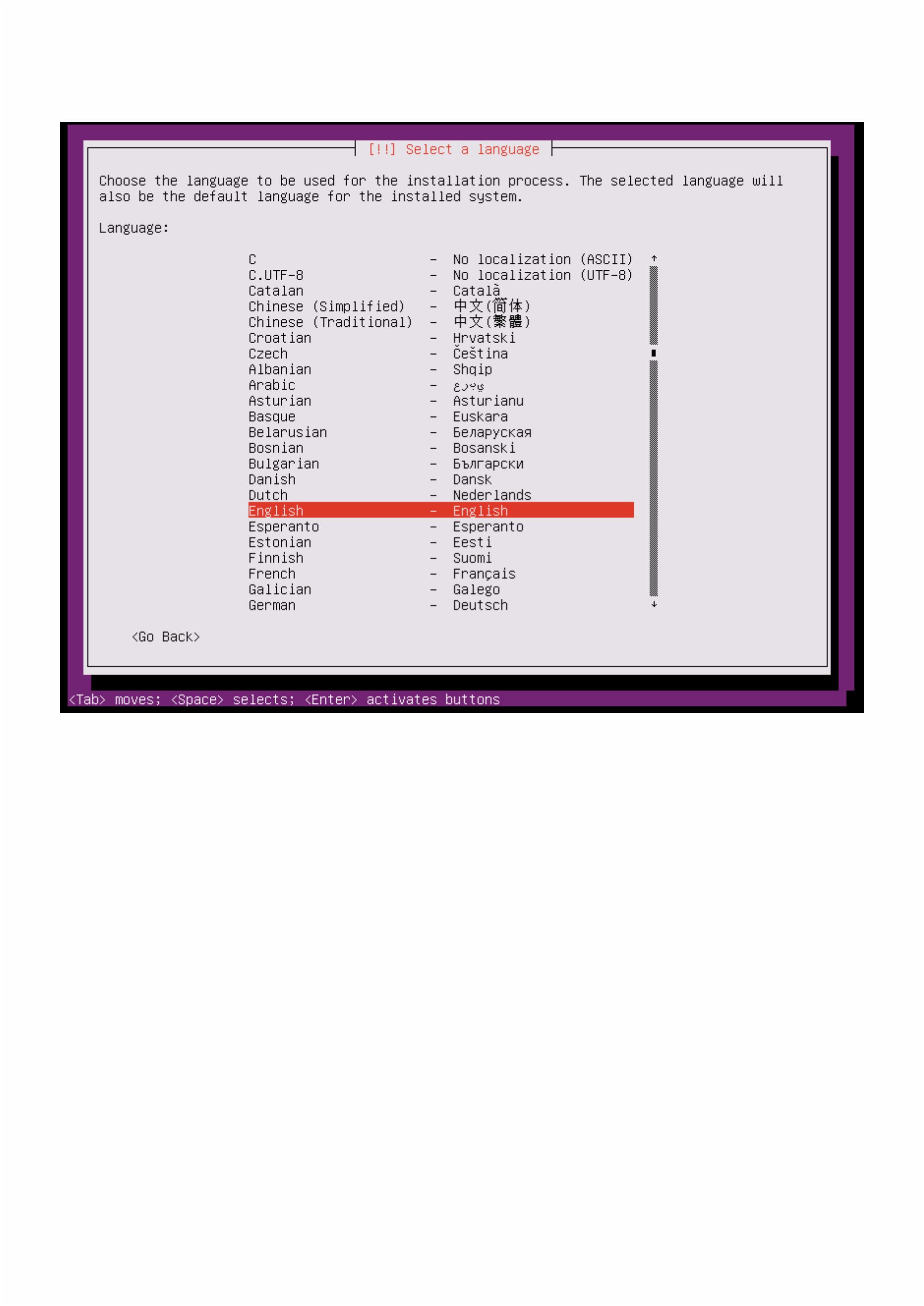
ubuntu 18.04服务器版本安装

2.1.1：选择安装语⾔：

2.1.1：安装界⾯传递内核参数：



2.1.3：选择安装过程语⾔：



2.1.4：地区设置：

2.1.5：键盘选择：



不进⾏探测，⼿动设置

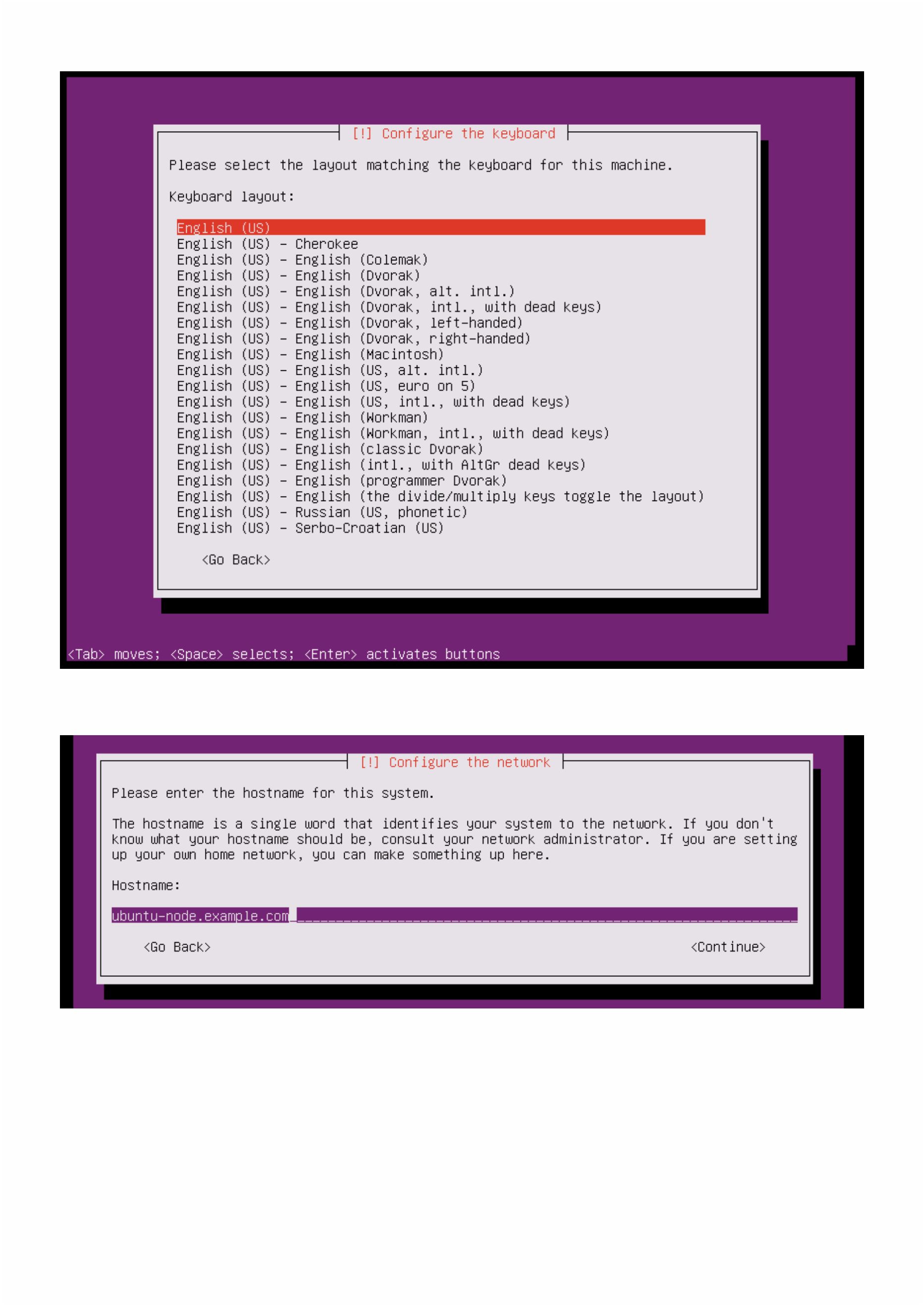
2.1.6：选择键盘：

选美式键盘

2.1.7：键盘确认：



2.1.8：配置主机名：



2.1.9：创建新的普通系统⽤户：

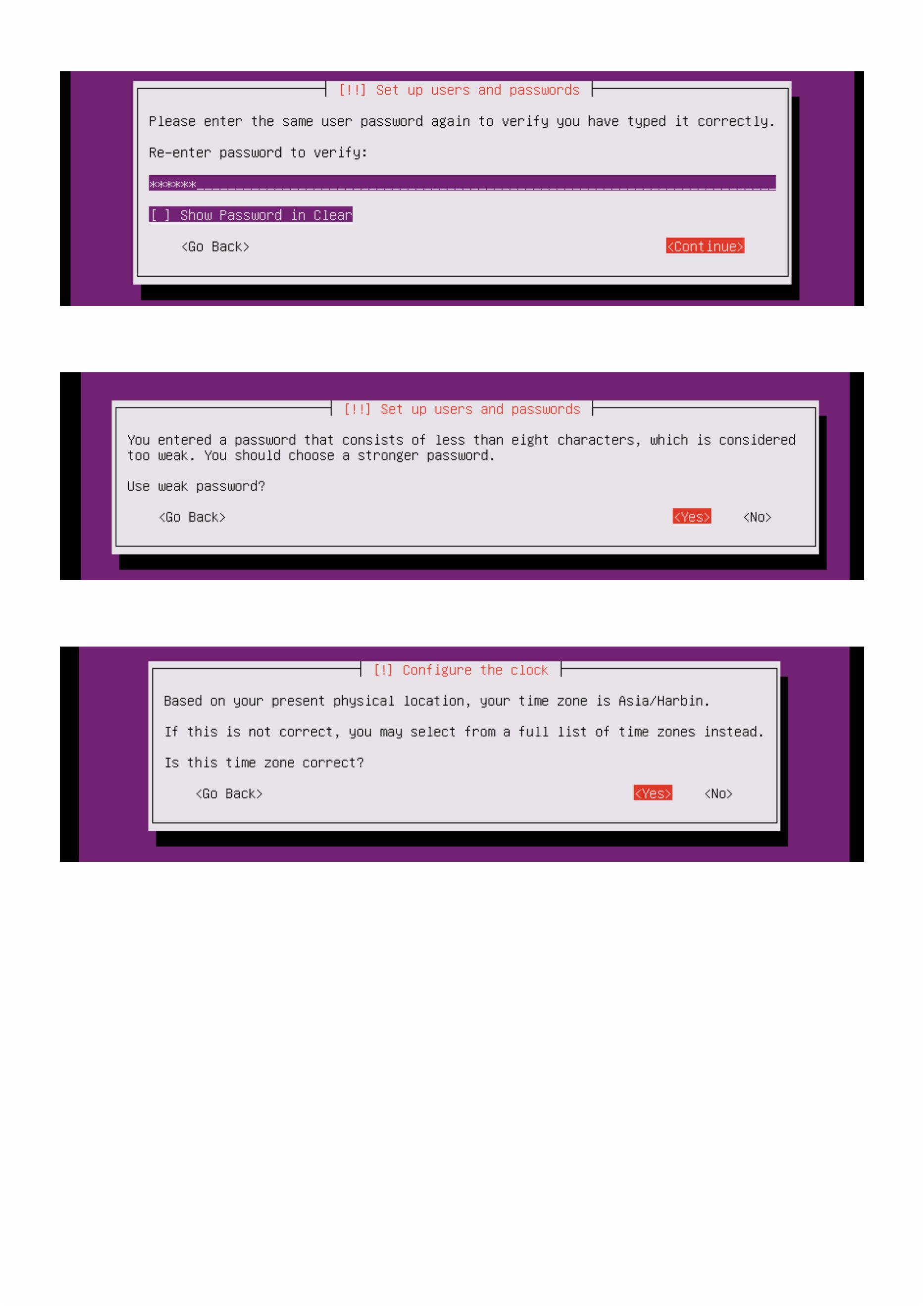
2.1.10：确认新账户名称：



2.1.11：设置⽤户密码：

2.1.12：确认账户密码：

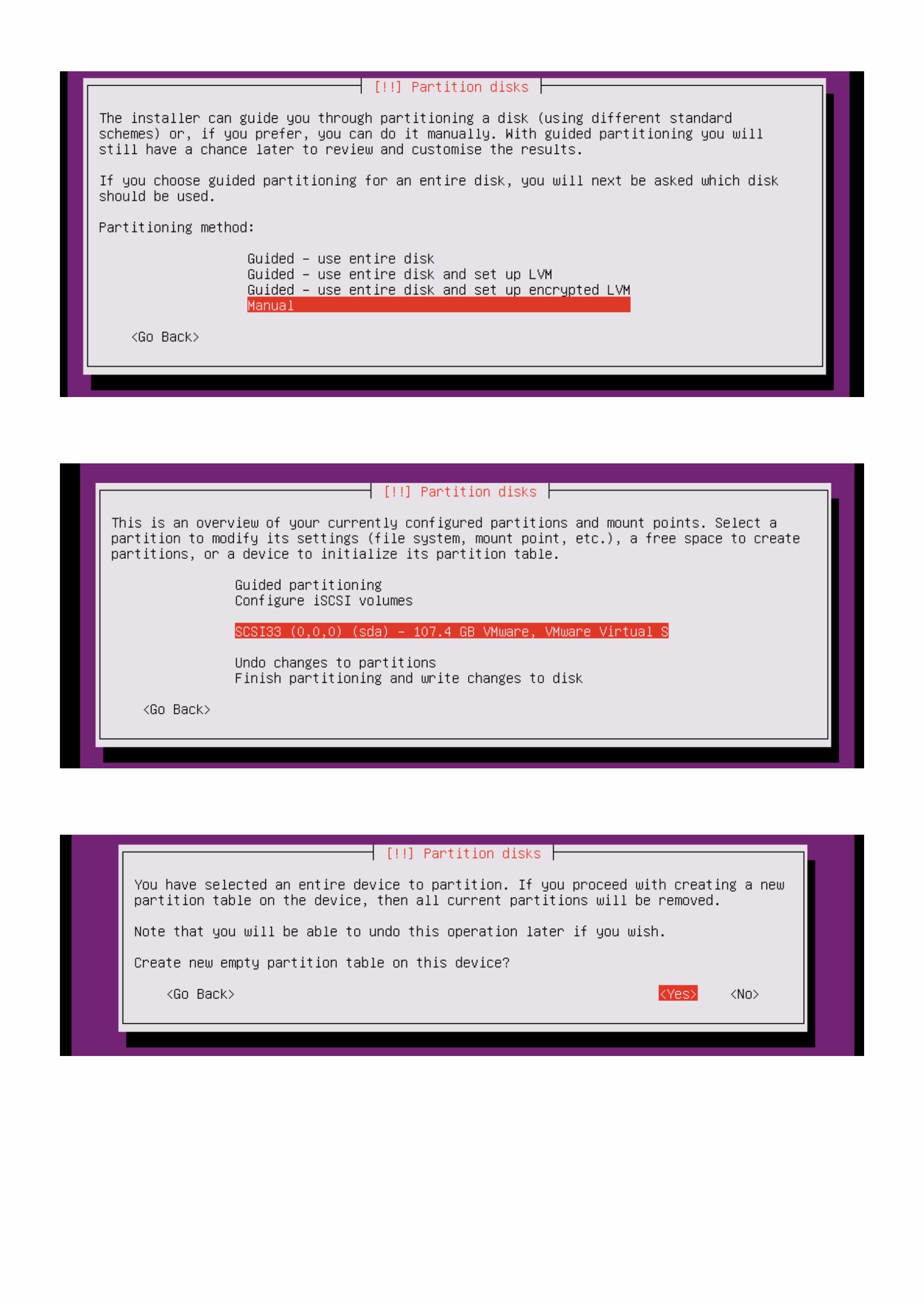
2.1.13：确认使⽤密码：



2.1.14：确认时区：

2.1.15：⼿动分区：

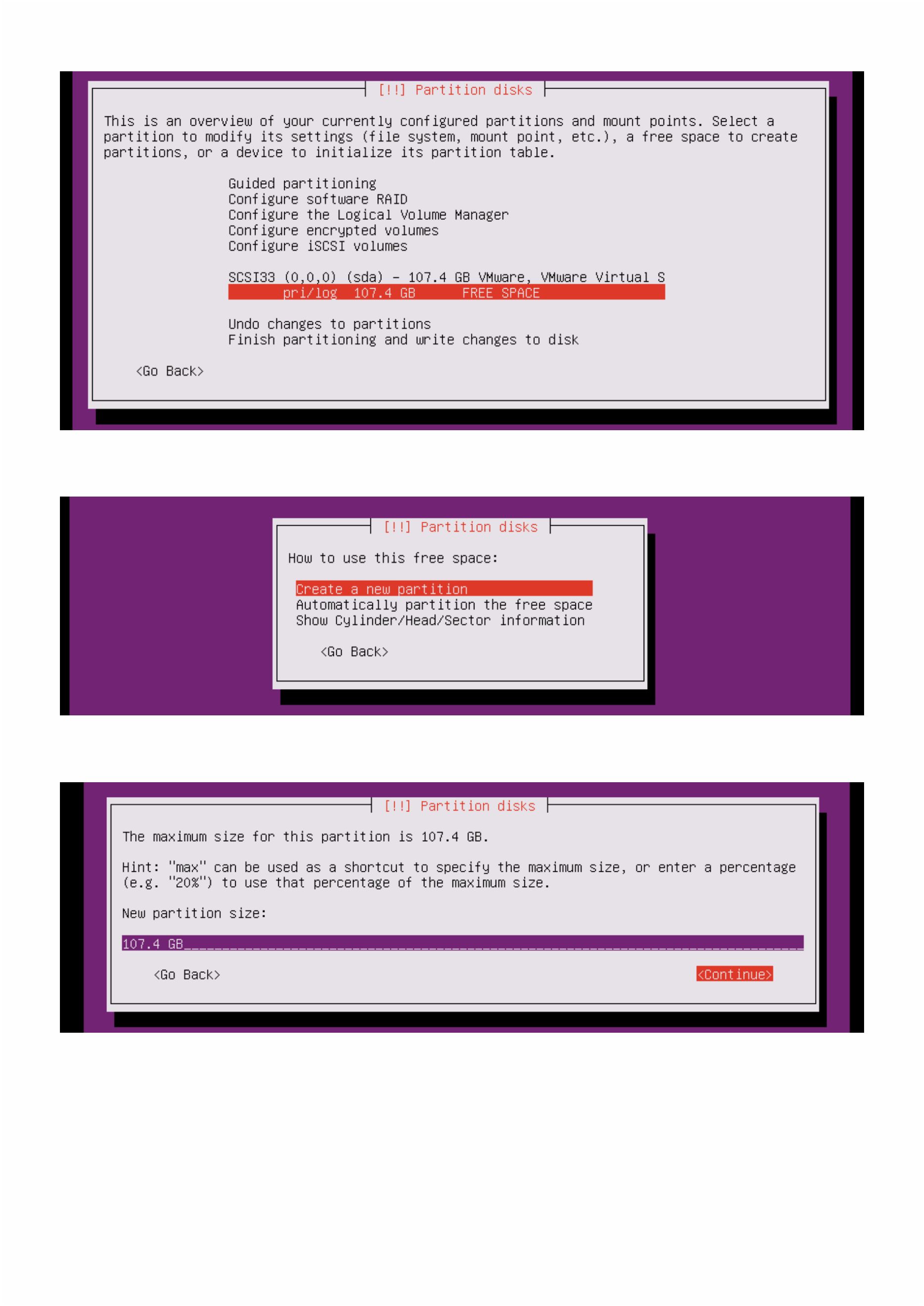
2.1.16：选择磁盘：



2.1.17：确认使⽤磁盘：

2.1.18：对磁盘分区：

2.1.19： 创建新分区：

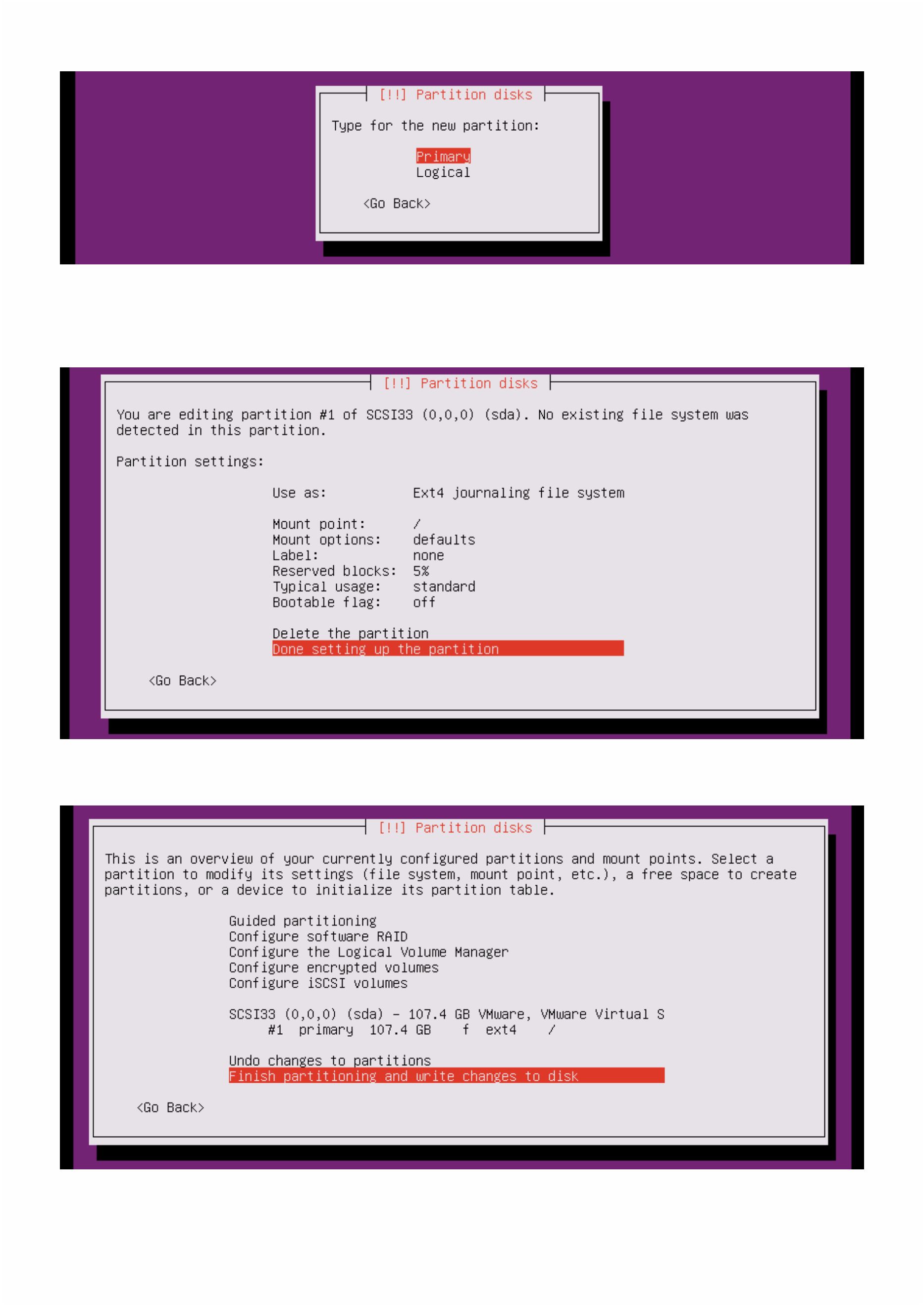


2.1.20：指定分区⼤⼩：

2.1.21：选择分区类型：

分区类型为主分区

2.1.22：分区完成：



⽂件系统类型可以选择ext4/xfs/等

2.1.23：完成分区并写⼊数据：

2.1.24：确认写⼊磁盘：

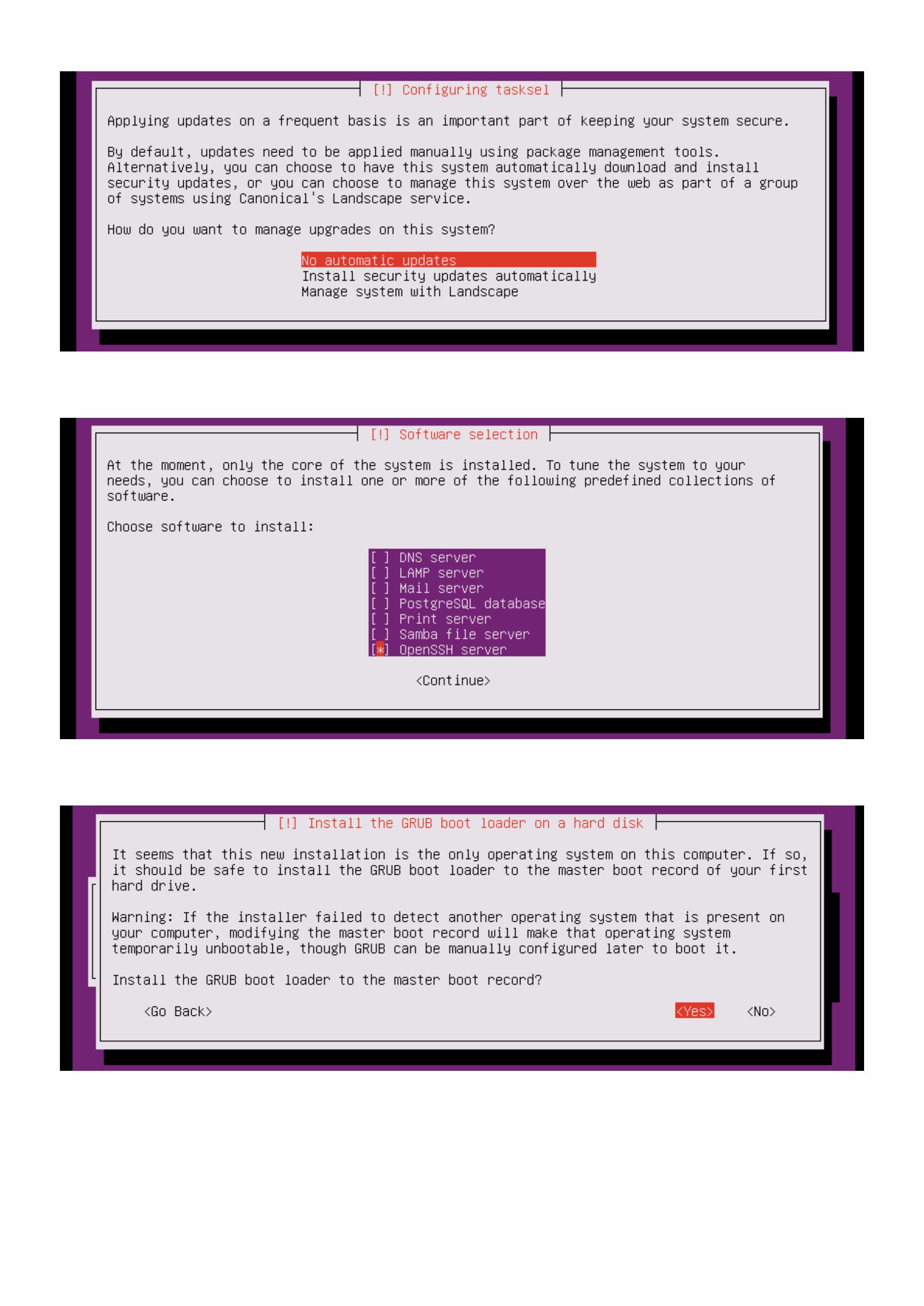


2.1.15：是否使⽤代理：

2.1.26：系统安装过程中：

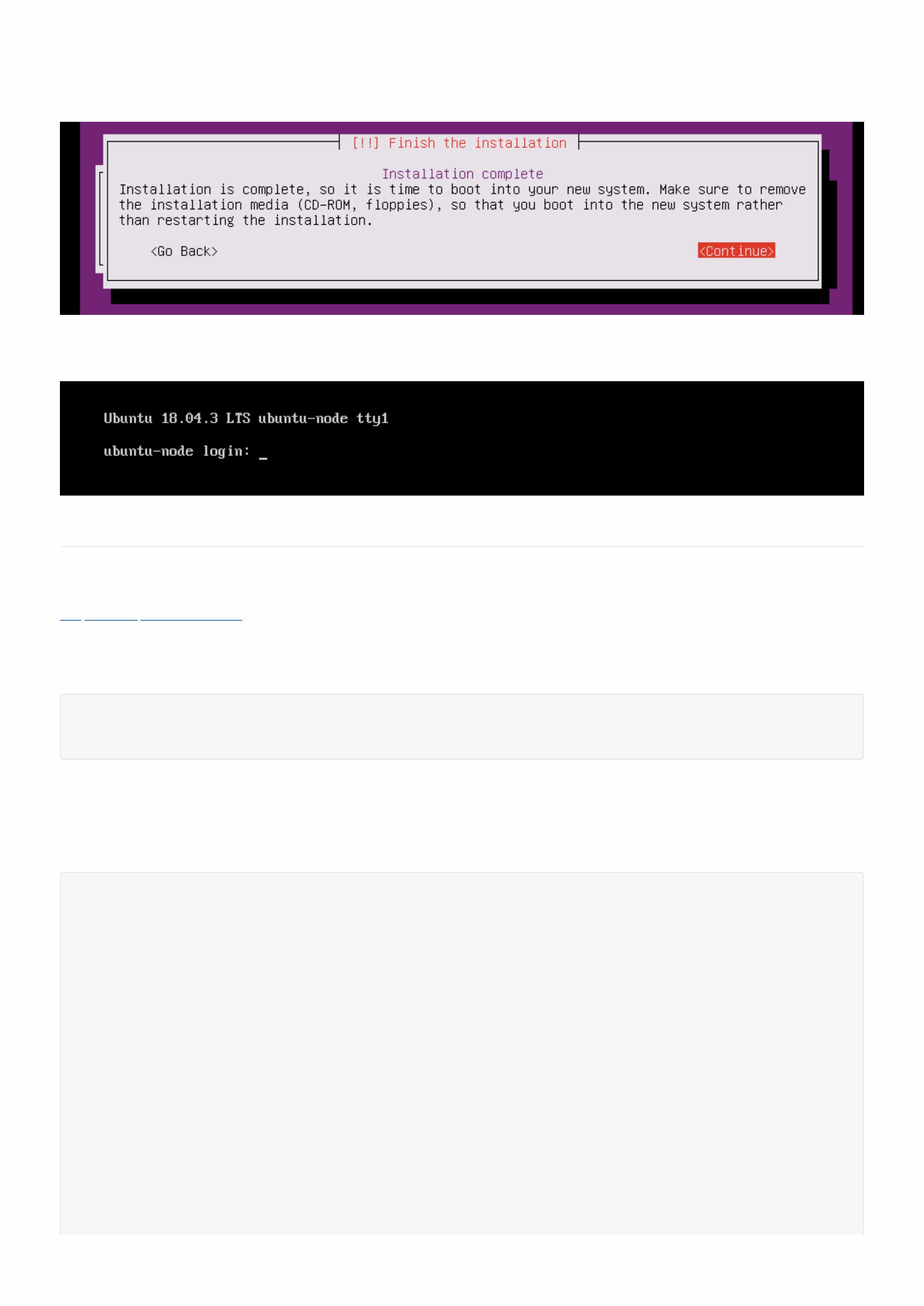
2.1.27：是否⾃动更新：

2.1.28：选择安装组件：



2.1.29：写⼊grub引导：

2.1.30：确认重启服务器：



2.1.31：系统登录界⾯：

2.2：Ubuntu Server系统基础配置：

官⽅使⽤⽂档：

[https://help.ubuntu.com](https://help.ubuntu.com/)

2.2.1：更改主机名：

~$ cat /etc/hostname

ubuntu-node.example.com

2.2.2：更改⽹卡名称为eth\*：

如果没有在安装系统之前传递内核参数将⽹卡名称更改为eth\*，则可以在安装系统之后使⽤以下⽅式修改：

~$ sudo vim /etc/default/grub

GRUB\_DEFAULT=0

GRUB\_TIMEOUT\_STYLE=hidden

GRUB\_TIMEOUT=2

GRUB\_DISTRIBUTOR=`lsb\_release -i -s 2> /dev/null || echo Debian`

GRUB\_CMDLINE\_LINUX\_DEFAULT=""

GRUB\_CMDLINE\_LINUX="net.ifnames=0 biosdevname=0"

~$ sudo update-grub

Sourcing file `/etc/default/grub'

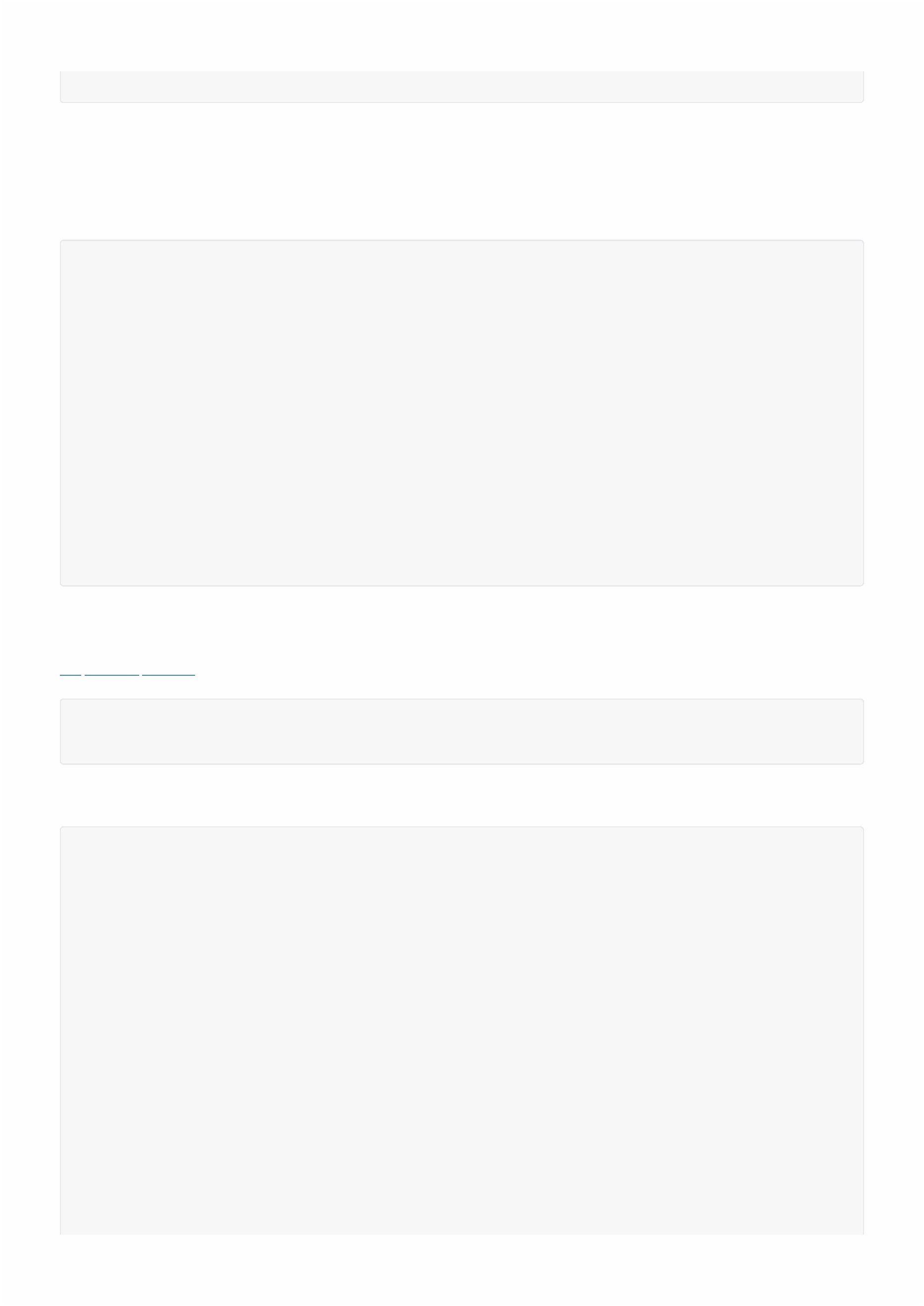
Generating grub configuration file ...

Found linux image: /boot/vmlinuz-4.15.0-55-generic

Found initrd image: /boot/initrd.img-4.15.0-55-generic

done

~$ sudo reboot



2.2.3：配置root远程登录：

默认情况下，ubuntu不允许root⽤户远程ssh，如果有实际场景需要允许root⽤户远程ssh，则需要设置root密码，并且编辑/etc/ssh/sshd\_config⽂件修改如下：

~$ sudo vim /etc/ssh/sshd\_config

32 #PermitRootLogin prohibit-password #默认为禁⽌登录

33 PermitRootLogin yes #改为允许登录

57 #PasswordAuthentication yes

58 PasswordAuthentication yes #打开密码认证，其实默认就是允许通过密码认证登录

~$ sudo su - root #切换到root⽤户环境

~# passwd #设置密码

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

~# systemctl restart sshd #重启ssh服务并测试root⽤户远程ssh连接

2.2.4：Ubuntu 18.04⽹络配置：

<https://netplan.io/>#官⽅⽂档

Ubuntu 从 17.10 开始，已放弃在 /etc/network/interfaces ⾥固定IP的配置，⽽是改成 netplan ⽅式，

配置⽂件是：/etc/netplan/01-netcfg.yaml

ubuntu 17.04及之前的静态IP配置⽅式：

~# cat /etc/network/interfaces

root@magedu:~# cat /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)

auto lo

iface lo inet loopback

auto eth0 #⽹卡⾃启动，写⾃⼰要配置IP的实际⽹卡名称

iface eth0 inet static #配置静态IP，写⾃⼰要配置IP的实际⽹卡名称

address 172.18.3.12 #IP地址

netmask 255.255.0.0 #掩码

gateway 172.18.0.1 #⽹关

dns-nameservers 223.6.6.6 #DNS

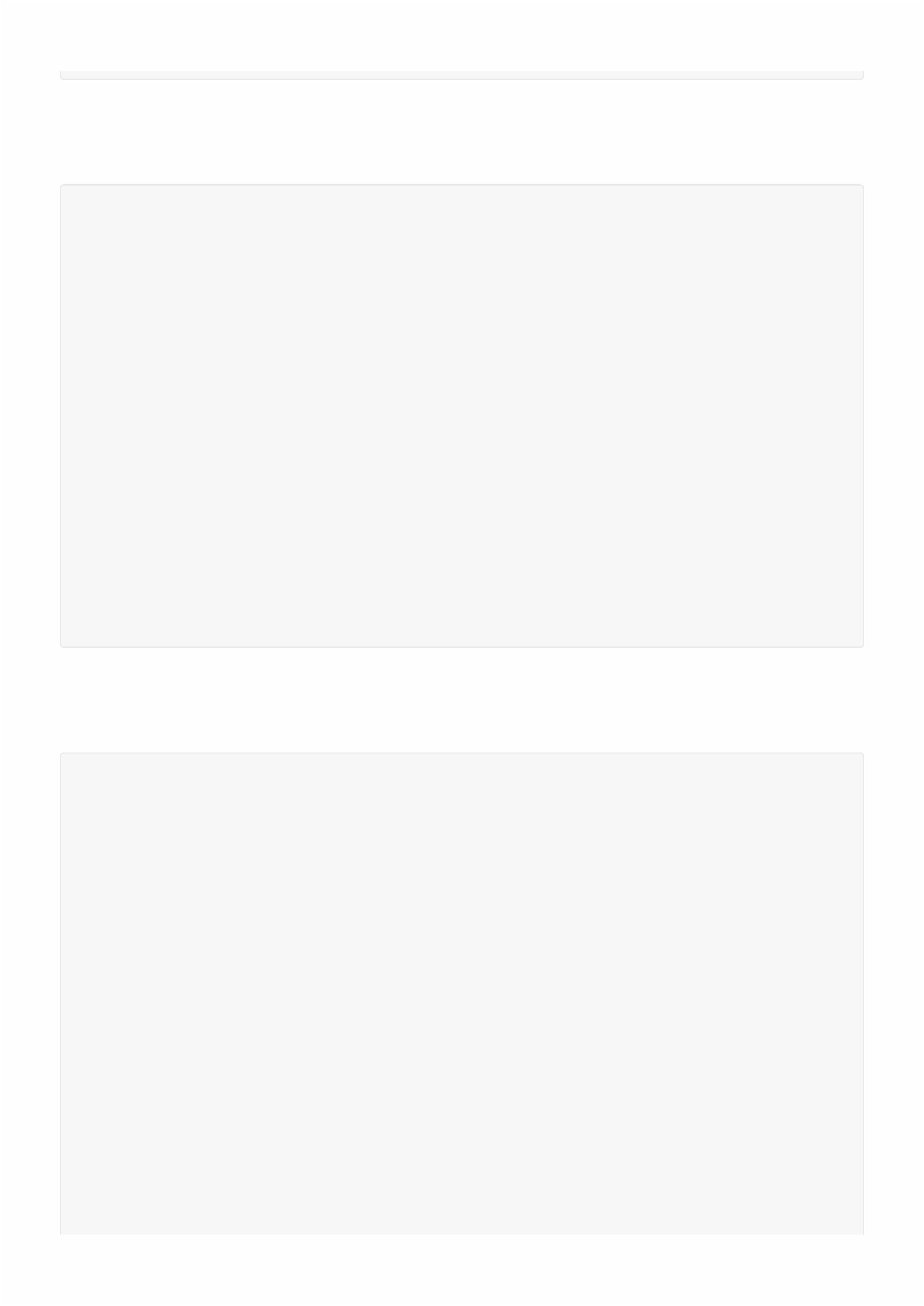
dns-nameservers 223.5.5.5

#重启⽹络服务

~# /etc/init.d/networking restart

~# systemctl restart networking.service

2.2.4.1 IP地址：



：单⽹卡静态

ubuntu 18.04配置静态IP地址

~#sudo vim /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6]

# sudo netplan apply

# vim ~/.vimrc #取消vim ⾃动缩进功能

set paste

2.2.4.2：配置多⽹卡静态IP：

ubuntu 18.04配置多⽹卡静态IP以及静态路由

~# sudo vim /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6]

eth1:

dhcp4: no

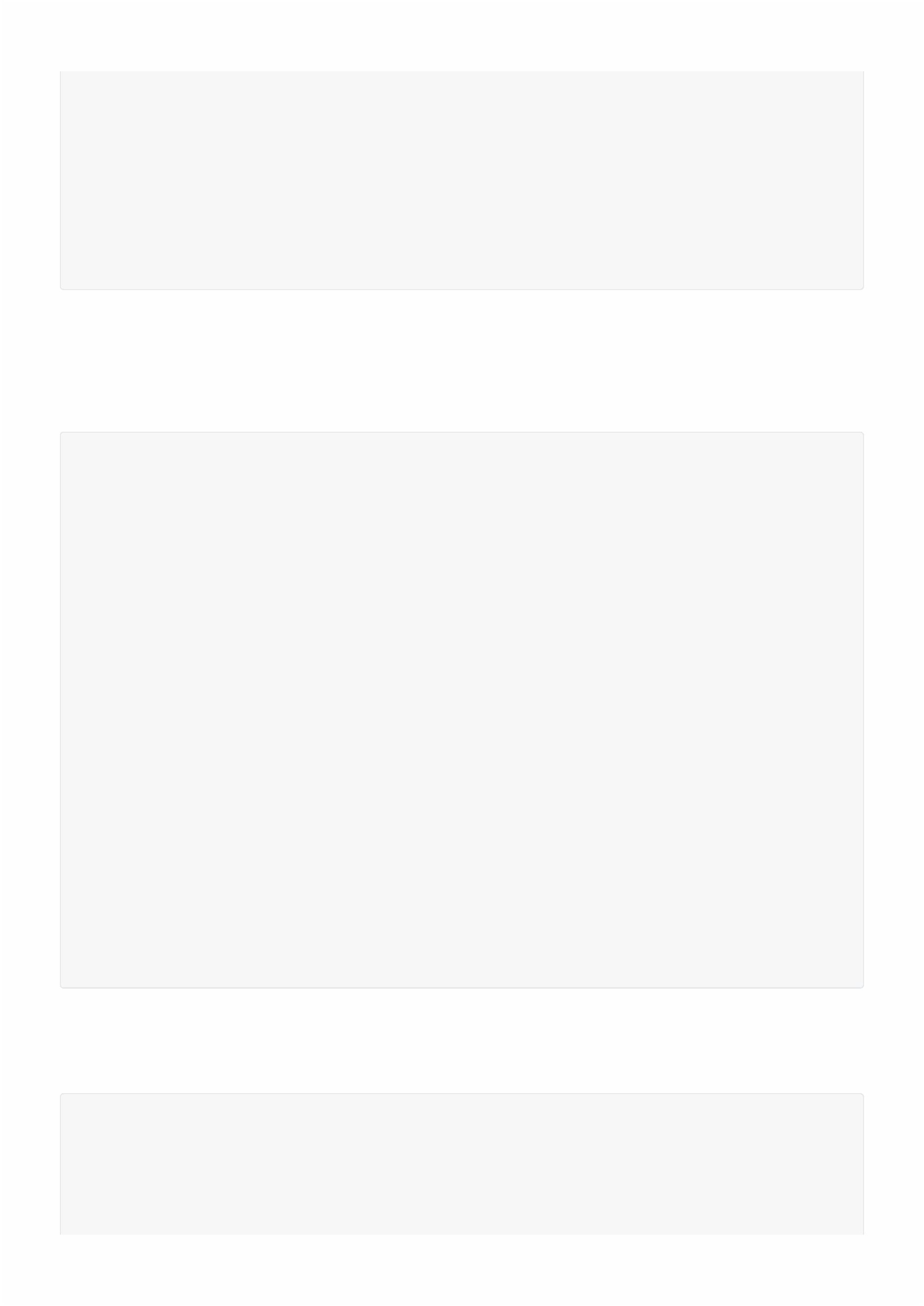
dhcp6: no

addresses: [10.20.3.18/16]

routes:

- to: 172.20.0.0/16

via: 10.20.0.1



- to: 10.20.0.0/16

via: 10.20.0.1

- to: 10.2.0.0/16

via: 10.20.0.1

- to: 10.8.0.0/16

via: 10.20.0.1

~# sudo netplan apply

2.2.4.3：单⽹卡桥接：

ubuntu 18.04的单⽹卡桥接配置

root@s1:~# cat /etc/netplan/01-netcfg.yaml

~# sudo cat /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

bridges:

br0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6]

interfaces:

- eth0

~# sudo netplan apply

2.2.4.4：多⽹卡桥接：

ubuntu 18.04多⽹卡的桥接配置，将br0和br1分别桥接到eth0和eth1。

~# sudo cat /etc/netplan/01-netcfg.yaml

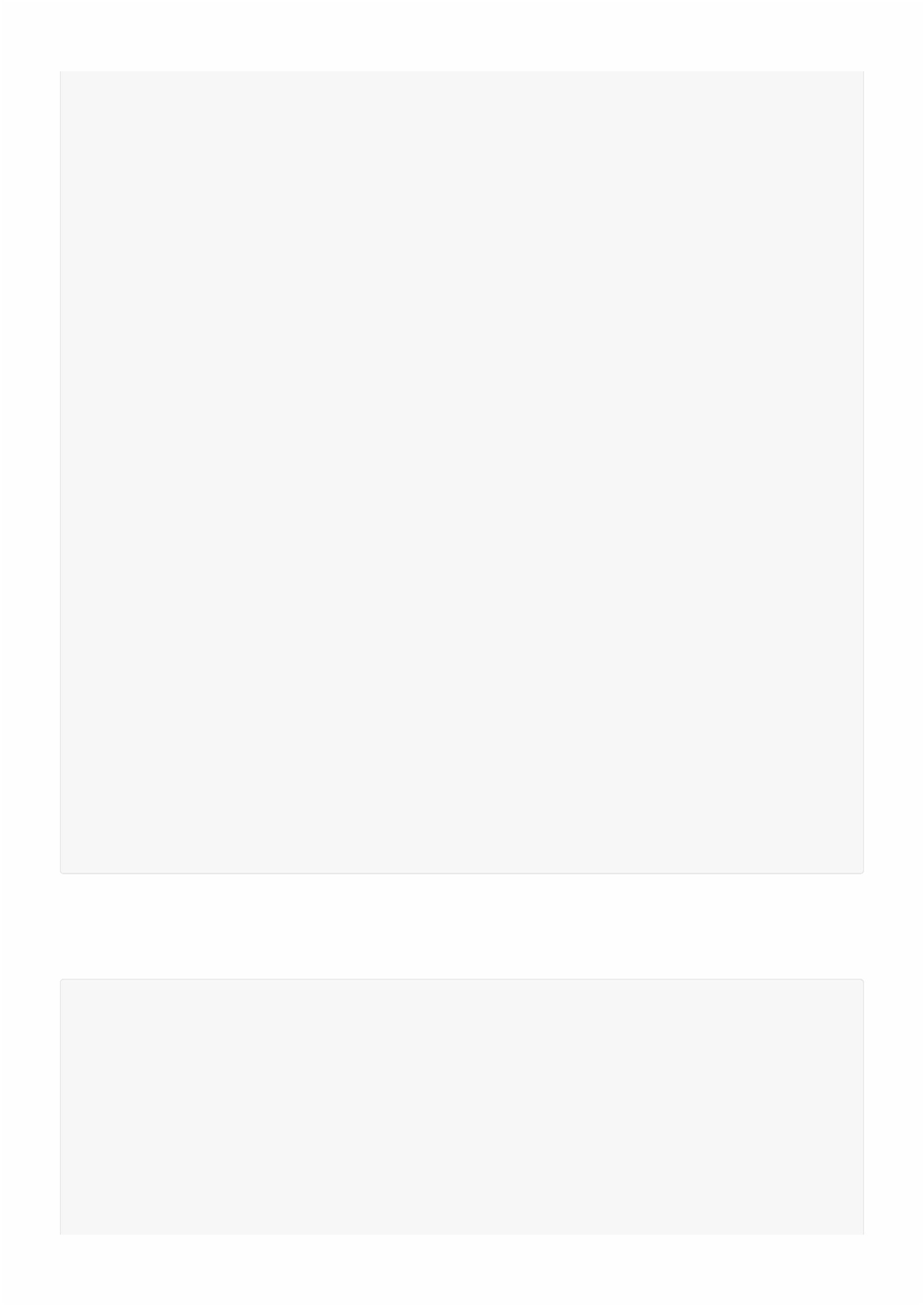
# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd



ethernets:

eth0:

dhcp4: no

dhcp6: no

eth1:

dhcp4: no

dhcp6: no

bridges:

br0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6]

interfaces:

- eth0

br1:

dhcp4: no

dhcp6: no

addresses: [10.20.3.18/16]

routes:

- to: 172.20.0.0/16

via: 10.20.0.1

- to: 10.20.0.0/16

via: 10.20.0.1

- to: 10.2.0.0/16

via: 10.20.0.1

- to: 10.8.0.0/16

via: 10.20.0.1

interfaces:

- eth1

2.2.4.4：双⽹卡绑定：

七种bond模式说明：

第⼀种模式：mod=0，即：(balance-rr) Round-robin policy（平衡抡循环策略）

特点：传输数据包顺序是依次传输（即：第1个包⾛eth0，下⼀个包就⾛eth1….⼀直循环下去，直到最后⼀个传输完

毕），此模式提供负载平衡和容错能⼒。

第⼆种模式：mod=1，即： (active-backup) Active-backup policy（主-备份策略）

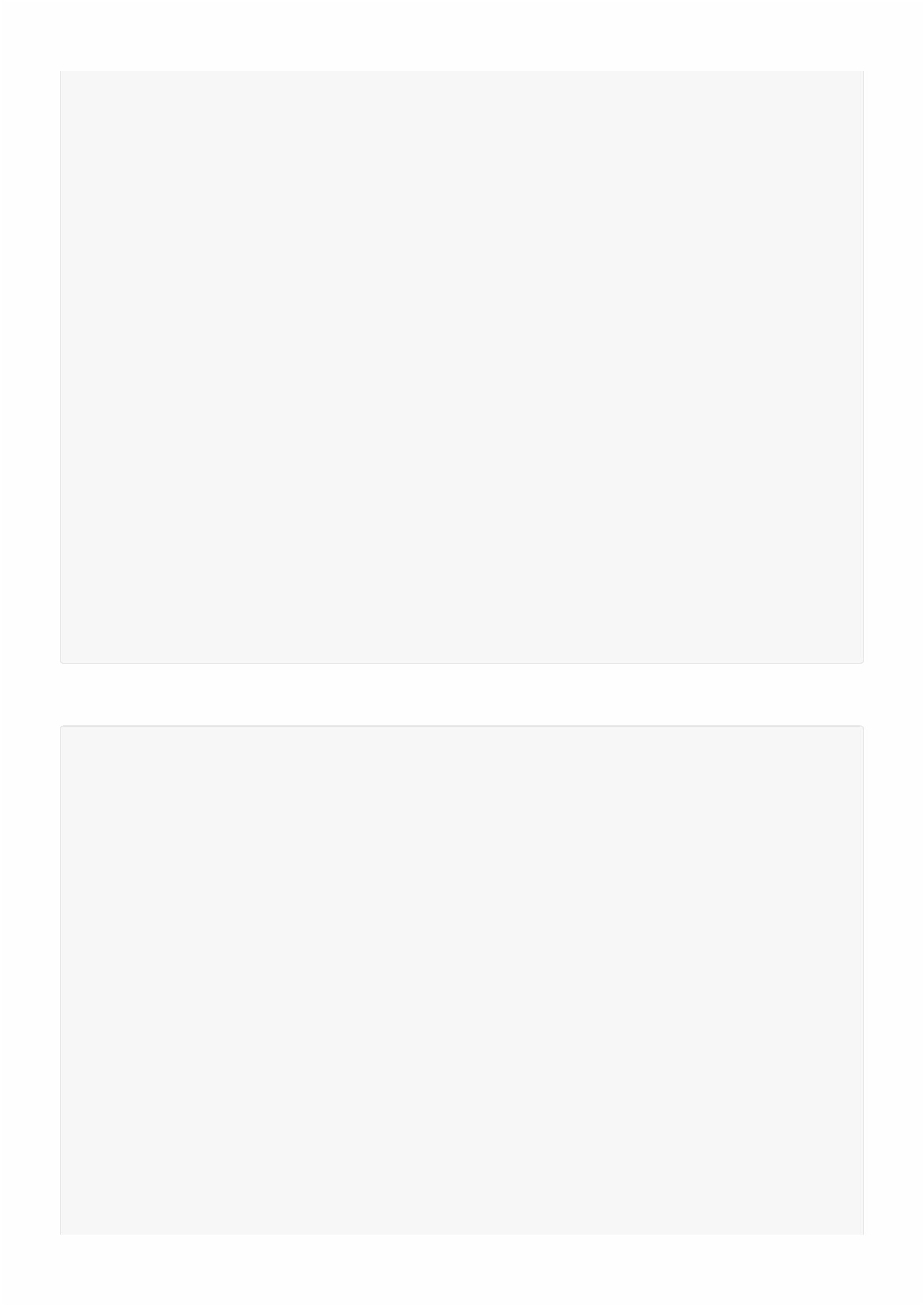
特点：只有⼀个设备处于活动状态，当⼀个宕掉另⼀个⻢上由备份转换为主设备。mac地址是外部可⻅得，从外⾯看

来，bond的MAC地址是唯⼀的，以避免switch(交换机)发⽣混乱。此模式只提供了容错能⼒；由此可⻅此算法的优点

是可以提供⾼⽹络连接的可⽤性，但是它的资源利⽤率较低，只有⼀个接⼝处于⼯作状态，在有 N 个⽹络接⼝的情况

下，资源利⽤率为1/N。

第三种模式：mod=2，即：(balance-xor) XOR policy（平衡策略）



特点：基于指定的传输HASH策略传输数据包。缺省的策略是：(源MAC地址 XOR ⽬标MAC地址) % slave数量。其他

的传输策略可以通过xmit\_hash\_policy选项指定，此模式提供负载平衡和容错能⼒。

第四种模式：mod=3，即：broadcast（⼴播策略）

特点：在每个slave接⼝上传输每个数据包，此模式提供了容错能⼒。

第五种模式：mod=4，即：(802.3ad) IEEE 802.3adDynamic link aggregation（IEEE 802.3ad 动态链接

聚合）

特点：创建⼀个聚合组，它们共享同样的速率和双⼯设定。根据802.3ad规范将多个slave⼯作在同⼀个激活的聚合体

下。

必要条件：

条件1：ethtool⽀持获取每个slave的速率和双⼯设定。

条件2：switch(交换机)⽀持IEEE 802.3ad Dynamic link aggregation。

条件3：⼤多数switch(交换机)需要经过特定配置才能⽀持802.3ad模式。

第六种模式：mod=5，即：(balance-tlb) Adaptive transmit load balancing（适配器传输负载均衡）

特点：不需要任何特别的switch(交换机)⽀持的通道bonding。在每个slave上根据当前的负载（根据速度计算）分

配外出流量。如果正在接受数据的slave出故障了，另⼀个slave接管失败的slave的MAC地址。

该模式的必要条件：

ethtool⽀持获取每个slave的速率

第七种模式：mod=6，即：(balance-alb) Adaptive load balancing（适配器适应性负载均衡）

特点：该模式包含了balance-tlb模式，同时加上针对IPV4流量的接收负载均衡(receive load balance,

rlb)，⽽且不需要任何switch(交换机)的⽀持。

ubuntu 18.04的双⽹卡绑定配置。

需要提前安装好bridge命令

~# sudo vim /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

eth1:

dhcp4: no

dhcp6: no

bonds:

bond0:

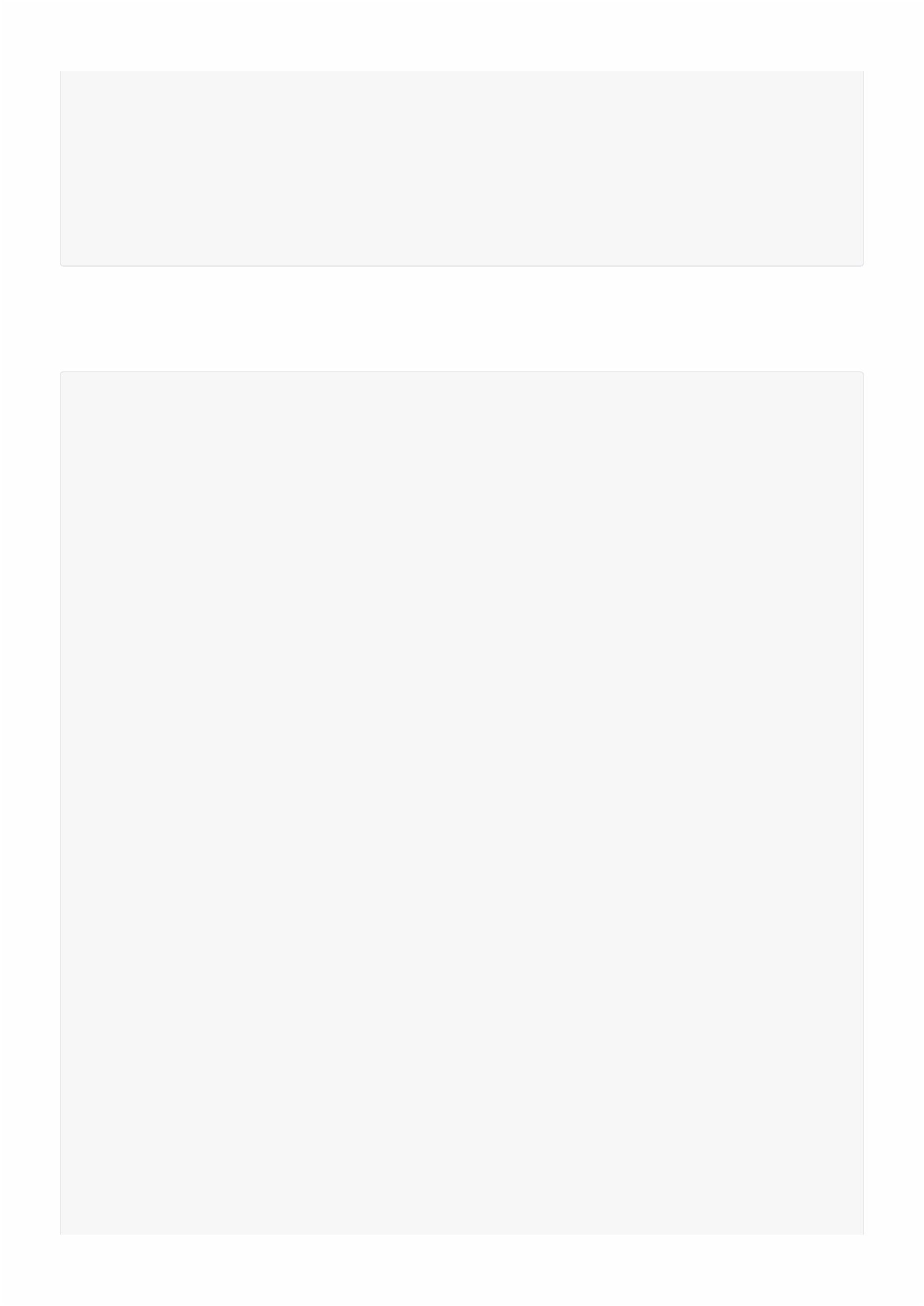
interfaces:

- eth0

- eth1

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1



nameservers:

addresses: [223.6.6.6,223.5.5.5]

parameters:

mode: active-backup

mii-monitor-interval: 100

~# sudo netplan apply

2.2.4.5：双⽹卡绑定+桥接：

⽹卡绑定⽤于提供⽹卡接⼝冗余以及⾼可⽤和端⼝聚合功能，桥接⽹卡再给需要桥接设备的服务使⽤。

~# sudo cat /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

eth1:

dhcp4: no

dhcp6: no

bonds:

bond0:

interfaces:

- eth0

- eth1

#addresses: [172.18.3.18/16]

#gateway4: 172.18.0.1

#nameservers:

# addresses: [223.6.6.6,223.5.5.5]

parameters:

mode: active-backup

mii-monitor-interval: 100

bridges:

br0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

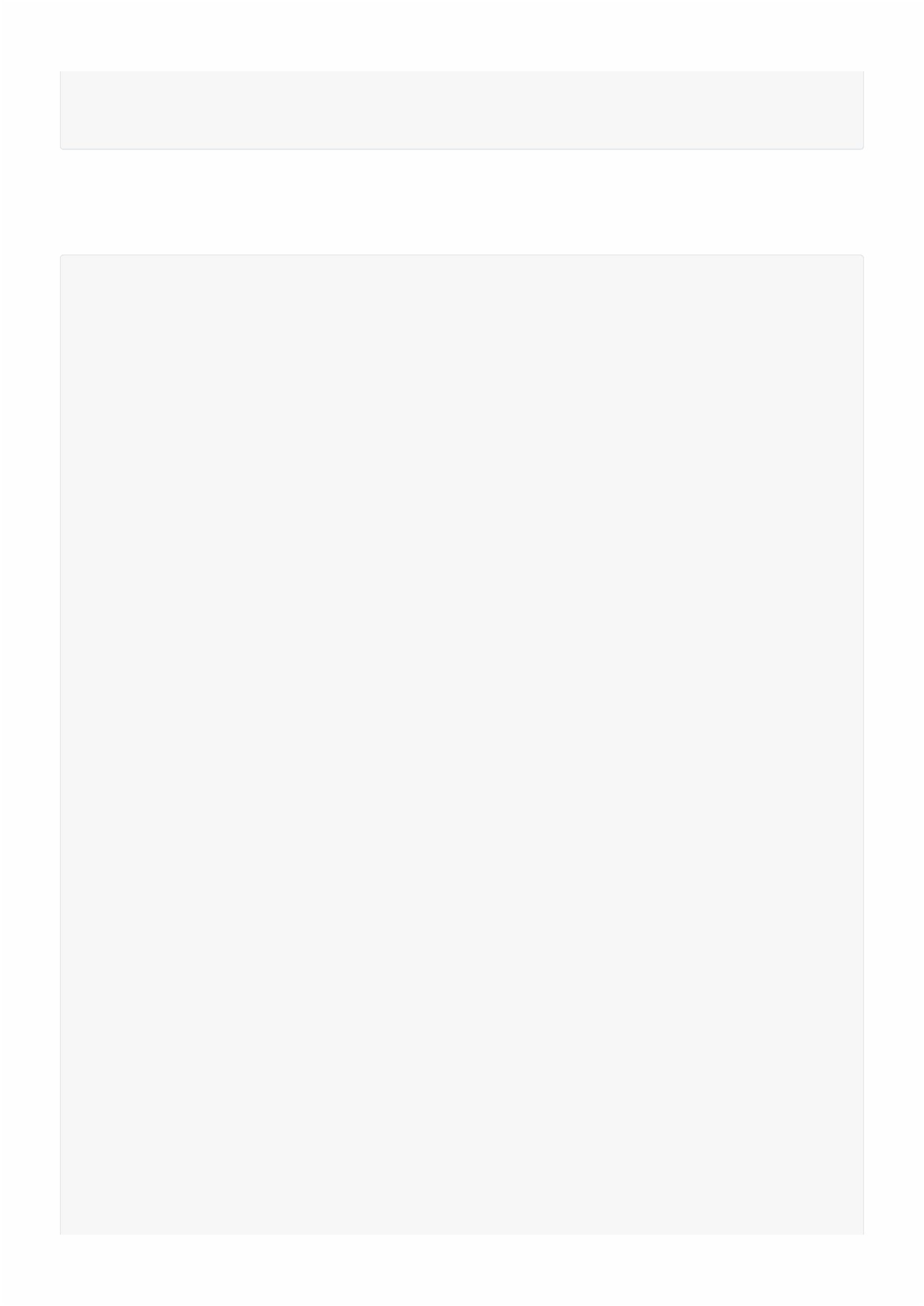
gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6,223.5.5.5]

interfaces:

- bond0



~#reboot

2.2.4.6：内外多⽹卡绑定：

多⽹络情况下实现⽹卡绑定。

~# sudo cat /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

eth1:

dhcp4: no

dhcp6: no

eth2:

dhcp4: no

dhcp6: no

eth3:

dhcp4: no

dhcp6: no

bonds:

bond0:

interfaces:

- eth0

- eth1

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6,223.5.5.5]

parameters:

mode: active-backup

mii-monitor-interval: 100

bond1:

interfaces:

- eth2

- eth3

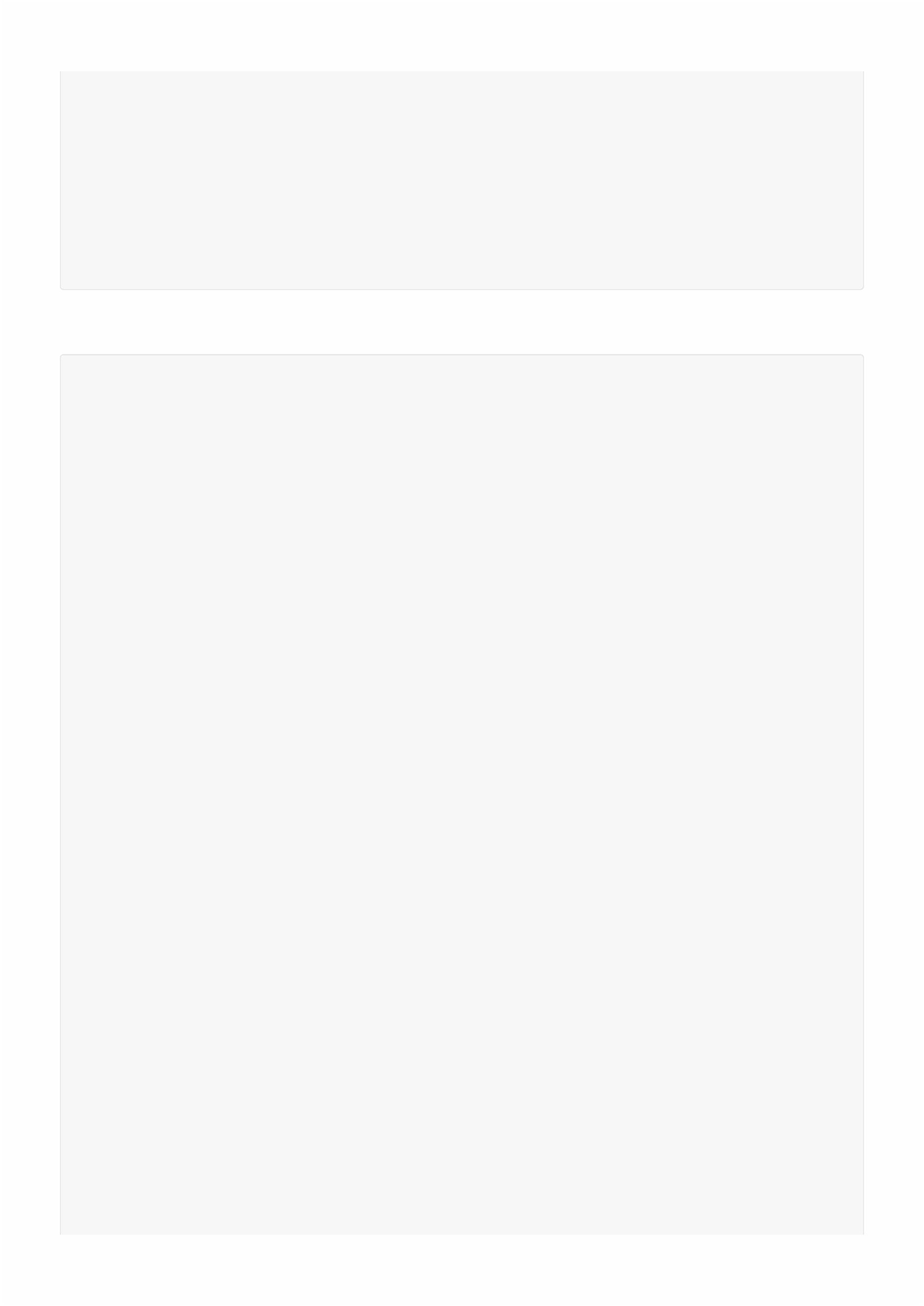
addresses: [10.20.3.18/16]

parameters:

mode: active-backup

mii-monitor-interval: 100

routes:



- to: 172.20.0.0/16

via: 10.20.0.1

- to: 10.20.0.0/16

via: 10.20.0.1

- to: 10.2.0.0/16

via: 10.20.0.1

- to: 10.8.0.0/16

via: 10.20.0.1

2.2.4.7：内⽹多⽹卡绑定+桥接：

~# sudo cat /etc/netplan/01-netcfg.yaml

# This file describes the network interfaces available on your system

# For more information, see netplan(5).

network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: no

dhcp6: no

eth1:

dhcp4: no

dhcp6: no

eth2:

dhcp4: no

dhcp6: no

eth3:

dhcp4: no

dhcp6: no

bonds:

bond0:

interfaces:

- eth0

- eth1

parameters:

mode: active-backup

mii-monitor-interval: 100

bond1:

interfaces:

- eth2

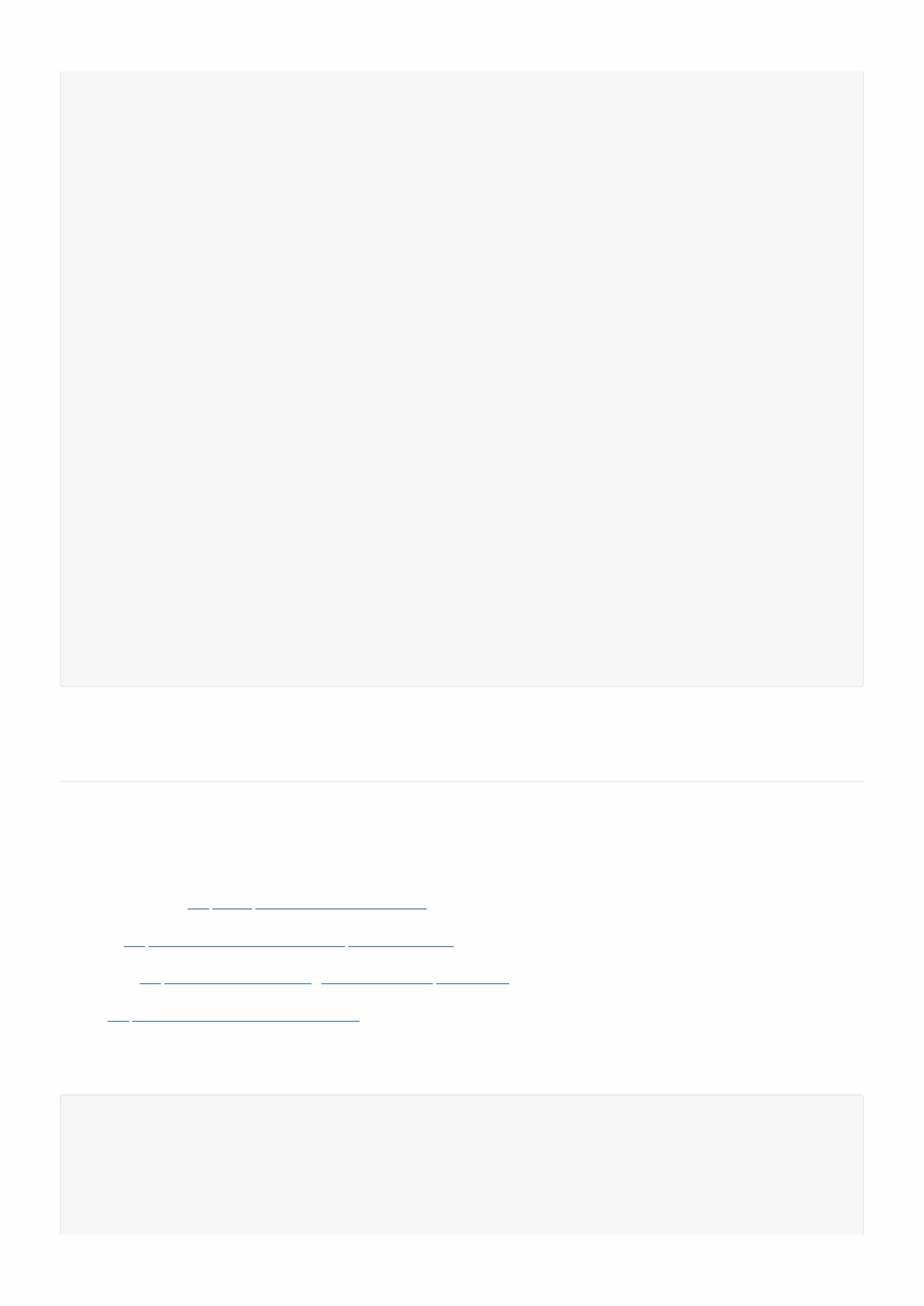
- eth3

parameters:

mode: active-backup

mii-monitor-interval: 100

bridges:



br0:

dhcp4: no

dhcp6: no

addresses: [172.18.3.18/16]

gateway4: 172.18.0.1

nameservers:

addresses: [223.6.6.6,223.5.5.5]

interfaces:

- bond0

br1:

dhcp4: no

dhcp6: no

interfaces:

- bond1

addresses: [10.20.3.18/16]

routes:

- to: 172.20.0.0/16

via: 10.20.0.1

- to: 10.20.0.0/16

via: 10.20.0.1

- to: 10.2.0.0/16

via: 10.20.0.1

- to: 10.8.0.0/16

via: 10.20.0.1

2.3：Ubuntu软件包管理：

在ubuntu安装、升级、卸载软件包等常规操作。

2.3.1：修改软件仓库：

阿⾥云仓库地址：<https://opsx.alibaba.com/mirror>

中科⼤：<http://mirrors.ustc.edu.cn/help/ubuntu.html>

清华⼤学：<https://mirror.tuna.tsinghua.edu.cn/help/ubuntu/>

华为：<https://mirrors.huaweicloud.com/>

2.3.2：apt/apt-get：

# apt list #apt列出仓库软件包，等于yum list

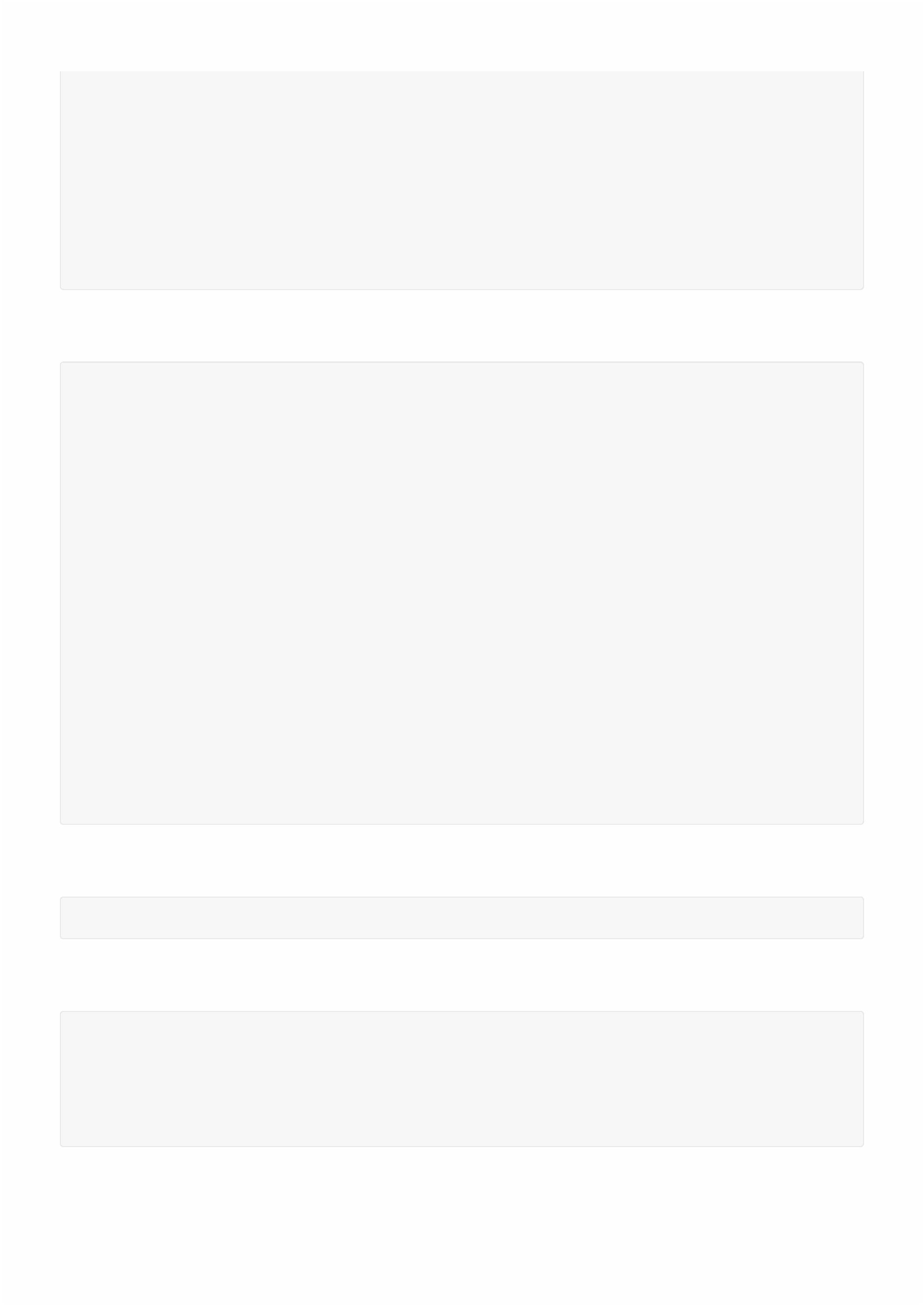
# apt search NAME #搜索安装包

# apt show apache2 #查看某个安装包的详细信息

# apt install apache2 #在线安装软件包

# apt remove apache2 #卸载单个软件包但是保留配置⽂件

# apt autoremove apache2 #删除安装包并解决依赖关系



# apt update #更新本地软件包列表索引，修改了apt仓库后必须执⾏

# apt purge apache2 #卸载单个软件包删除配置⽂件

# apt upgrade #升级所有已安装且可升级到新版本的软件包

# apt full-upgrade #升级整个系统，必要时可以移除旧软件包。

# apt edit-sources #编辑source源⽂件

# apt-cache madison nginx #查看仓库中软件包有哪些版本可以安装

# apt install nginx=1.14.0-0ubuntu1.6 #安装软件包的时候指定安装具体的版本

2.3.3：设置oracle JDK环境：

# pwd

/usr/local/src

解压⼆进制⽂件并设置软连接：

# tar xf jdk-8u212-linux-x64.tar.gz

# ln -sv /usr/local/src/jdk1.8.0\_212 /usr/local/jdk

配置环境变量：

# vim /etc/profile

export JAVA\_HOME=/usr/local/jdk

export PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH

export CLASSPATH=.$CLASSPATH:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib:$JAVA\_HOME/lib/tools.jar

重新导⼊环境变量并验证：

# source /etc/profile

# java -version

java version "1.8.0\_212"

Java(TM) SE Runtime Environment (build 1.8.0\_212-b10)

Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)

2.3.4：安装OpenJDK：

# apt install openjdk-8-jdk

2.3.5：安装常⽤系统命令：

# apt purge ufw lxd lxd-client lxcfs lxc-common

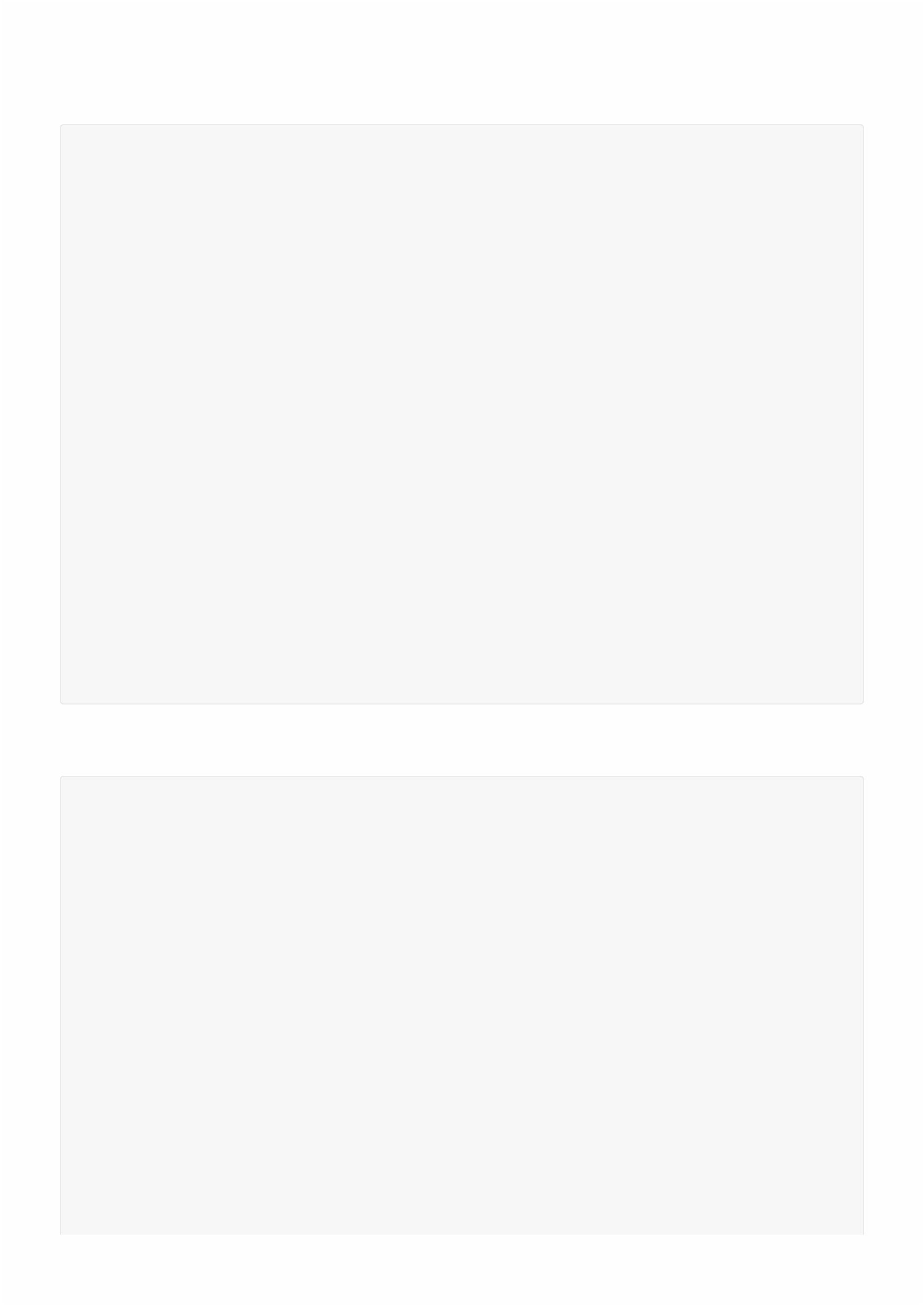
# apt install iproute2 ntpdate tcpdump telnet traceroute nfs-kernel-server nfs-

common lrzsz tree openssl libssl-dev libpcre3 libpcre3-dev zlib1g-dev ntpdate tcpdump

telnet traceroute gcc openssh-server lrzsz tree openssl libssl-dev libpcre3 libpcre3-

dev zlib1g-dev ntpdate tcpdump telnet traceroute iotop unzip zip

2.3.6：系统资源限制优化：



#cat /etc/security/limits.conf

#root账户的资源软限制和硬限制

root soft core unlimited

root hard core unlimited

root soft nproc 1000000

root hard nproc 1000000

root soft nofile 1000000

root hard nofile 1000000

root soft memlock 32000

root hard memlock 32000

root soft msgqueue 8192000

root hard msgqueue 8192000

#其他账户的资源软限制和硬限制

\* soft core unlimited

\* hard core unlimited

\* soft nproc 1000000

\* hard nproc 1000000

\* soft nofile 1000000

\* hard nofile 1000000

\* soft memlock 32000

\* hard memlock 32000

\* soft msgqueue 8192000

\* hard msgqueue 8192000

2.3.7：内核参数优化：

# Controls source route verification

net.ipv4.conf.default.rp\_filter = 1

net.ipv4.ip\_nonlocal\_bind = 1

net.ipv4.ip\_forward = 1

# Do not accept source routing

net.ipv4.conf.default.accept\_source\_route = 0

# Controls the System Request debugging functionality of the kernel

kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.

# Useful for debugging multi-threaded applications.

kernel.core\_uses\_pid = 1

# Controls the use of TCP syncookies

net.ipv4.tcp\_syncookies = 1

# Disable netfilter on bridges.

net.bridge.bridge-nf-call-ip6tables = 0



net.bridge.bridge-nf-call-iptables = 0

net.bridge.bridge-nf-call-arptables = 0

# Controls the default maxmimum size of a mesage queue

kernel.msgmnb = 65536

# # Controls the maximum size of a message, in bytes

kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes

kernel.shmmax = 68719476736

# # Controls the maximum number of shared memory segments, in pages

kernel.shmall = 4294967296

# TCP kernel paramater

net.ipv4.tcp\_mem = 786432 1048576 1572864

net.ipv4.tcp\_rmem = 4096 87380 4194304

net.ipv4.tcp\_wmem = 4096 16384 4194304

net.ipv4.tcp\_window\_scaling = 1

net.ipv4.tcp\_sack = 1

# socket buffer

net.core.wmem\_default = 8388608

net.core.rmem\_default = 8388608

net.core.rmem\_max = 16777216

net.core.wmem\_max = 16777216

net.core.netdev\_max\_backlog = 262144

net.core.somaxconn = 20480

net.core.optmem\_max = 81920

# TCP conn

net.ipv4.tcp\_max\_syn\_backlog = 262144

net.ipv4.tcp\_syn\_retries = 3

net.ipv4.tcp\_retries1 = 3

net.ipv4.tcp\_retries2 = 15

# tcp conn reuse

net.ipv4.tcp\_timestamps = 0

net.ipv4.tcp\_tw\_reuse = 0

net.ipv4.tcp\_tw\_recycle = 0

net.ipv4.tcp\_fin\_timeout = 1

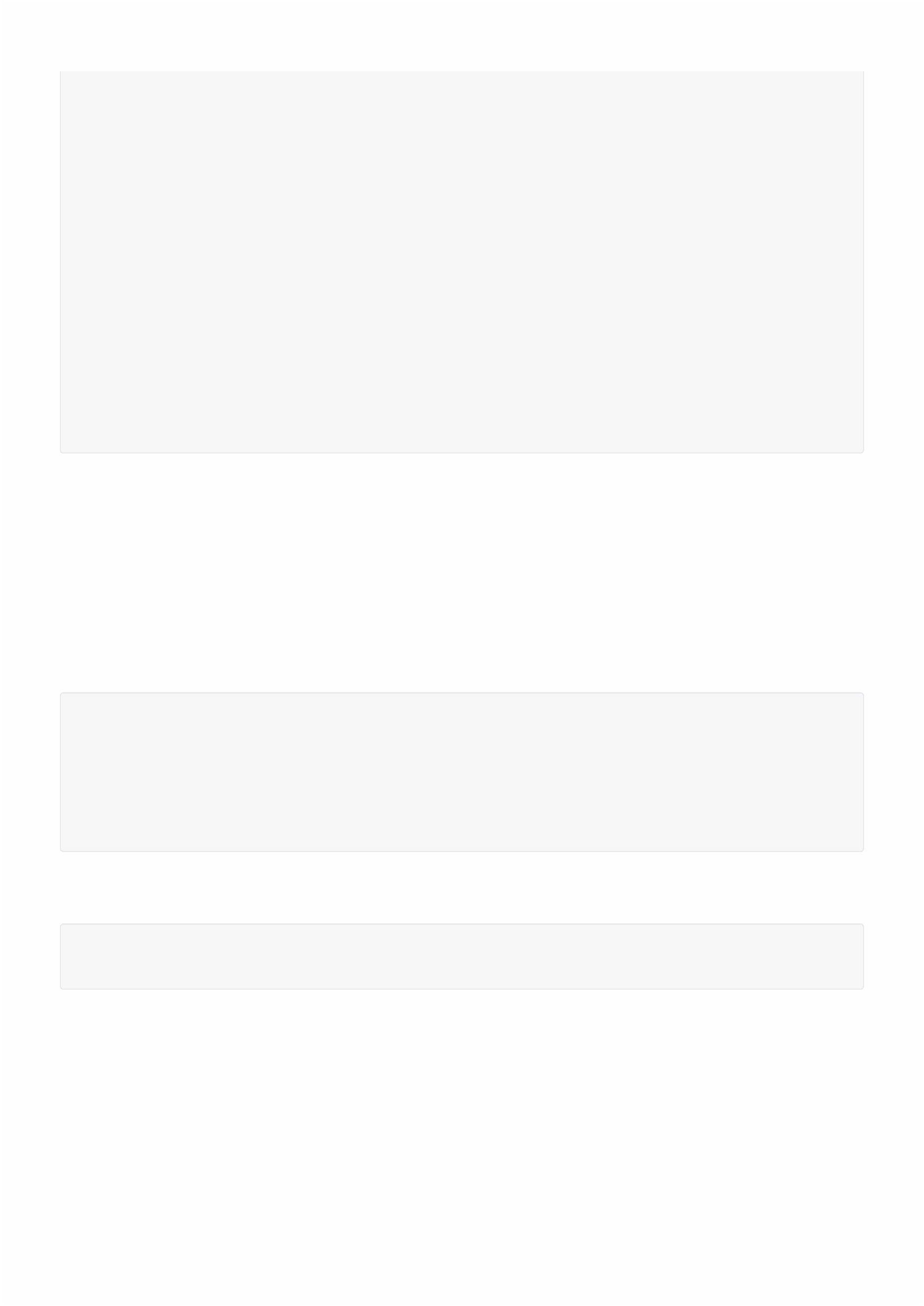
net.ipv4.tcp\_max\_tw\_buckets = 20000

net.ipv4.tcp\_max\_orphans = 3276800

net.ipv4.tcp\_synack\_retries = 1

net.ipv4.tcp\_syncookies = 1

# keepalive conn



net.ipv4.tcp\_keepalive\_time = 300

net.ipv4.tcp\_keepalive\_intvl = 30

net.ipv4.tcp\_keepalive\_probes = 3

net.ipv4.ip\_local\_port\_range = 10001 65000

# swap

vm.overcommit\_memory = 0

vm.swappiness = 10

#net.ipv4.conf.eth1.rp\_filter = 0

#net.ipv4.conf.lo.arp\_ignore = 1

#net.ipv4.conf.lo.arp\_announce = 2

#net.ipv4.conf.all.arp\_ignore = 1

#net.ipv4.conf.all.arp\_announce = 2

2.3.8：dpkg安装包管理：

rpm：RPM(Red Hat Package Manager)，是基于Red hat的Linux Distribution的包管理系统，同时也指rpm包本身，RPM⽤于rpm包的管理（诸如安装、卸载、升级等）

"dpkg "是"Debian Packager "的简写,为 "Debian"专⻔开发的套件管理系统，⽅便软件的安装、更新及移除。所有源⾃“Debian”的“Linux ”发⾏版都使⽤ “dpkg”，例如 “Ubuntu”、“Knoppix ”等。

# dpkg -i gitlab-ce\_11.9.8-ce.0\_amd64.deb #安装某个软件包

# dpkg -r gitlab-ce #删除某个软件包保留配置⽂件

# dpkg -r -P gitlab-ce #删除某个软件包不保留配置⽂件

# dpkg -I gitlab-ce\_11.9.8-ce.0\_amd64.deb #查看软件包信息

# dpkg -c gitlab-ce\_11.9.8-ce.0\_amd64.deb #查看软件包内的⽂件及⽬录内容

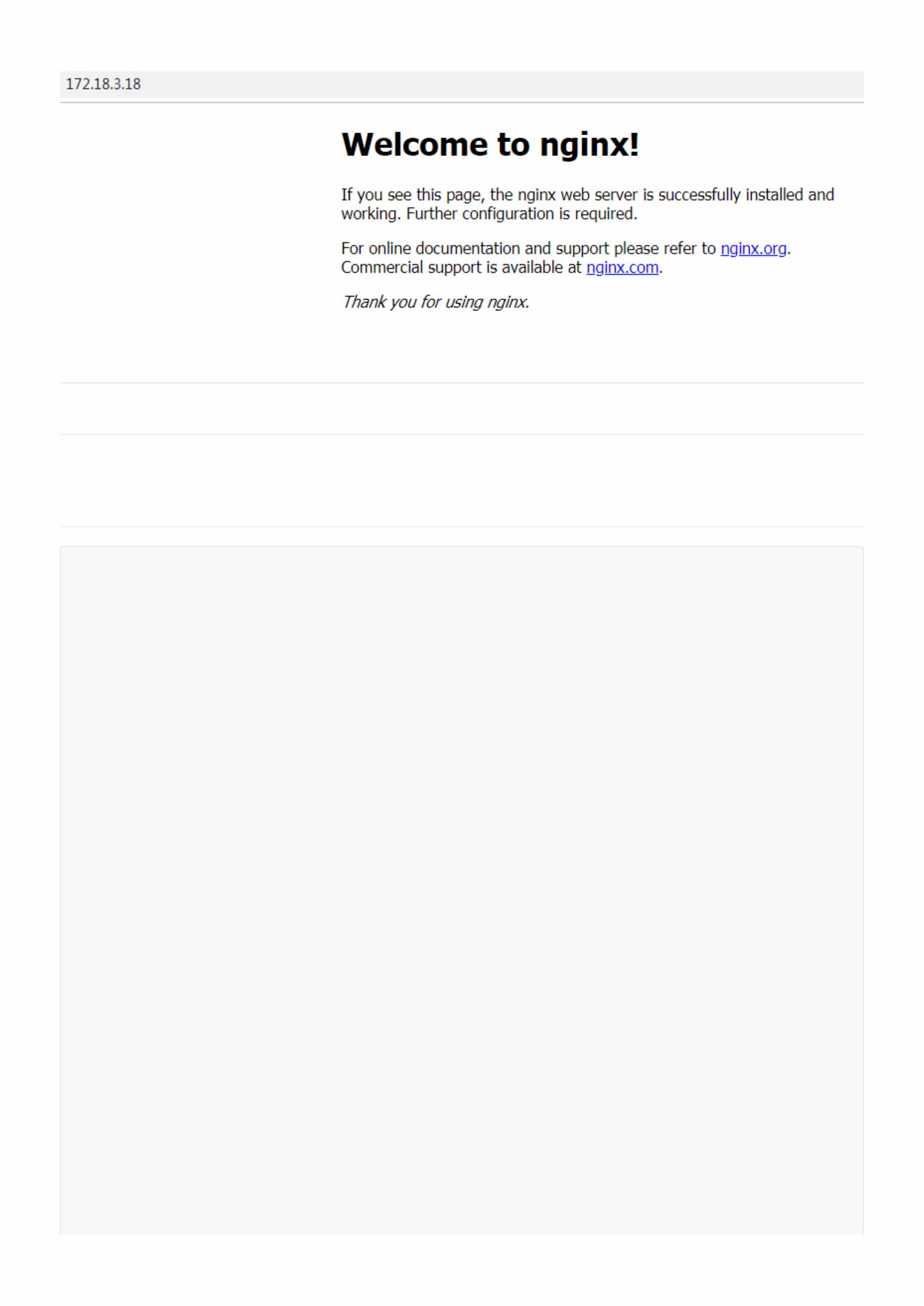
# dpkg -l #列出本机已经安装的所有软件

2.3.9：apt安装Nginx:

~$ sudo apt install nginx=1.14.0-0ubuntu1.6

~$ systemctl start nginx

三：Ubuntu Desktop版本安装及使⽤：



3.1：安装过程：

参考server版安装过程

3.2：环境配置：

1、设置软件源及安装常⽤命令：

https://opsx.alibaba.com/mirror

sudo apt-get install build-essential cmake pkg-config qt4-qmake libqt4-dev desktop-

file-utils \

libavformat-dev libavcodec-dev libavutil-dev libswscale-dev libasound2-dev libpulse-dev

libjack-jackd2-dev \

libgl1-mesa-dev libglu1-mesa-dev libx11-dev libxfixes-dev libxext-dev libxi-dev

libxinerama-dev

2、系统更新及配置中⽂语⾔环境：

3、安装搜狗拼⾳输⼊法

https://pinyin.sogou.com/linux/?r=pinyin

4、安装转码器ffmpeg：

多媒体视频处理⼯具FFmpeg有⾮常强⼤的功能包括视频采集功能、视频格式转换、视频抓图、给视频加⽔印等。

sudo apt-get install ffmpeg

5、安装视频播放器： smplayer https://www.jianshu.com/p/f24252c632d0

sudo apt-get install smplayer

6、办公软件WPS：

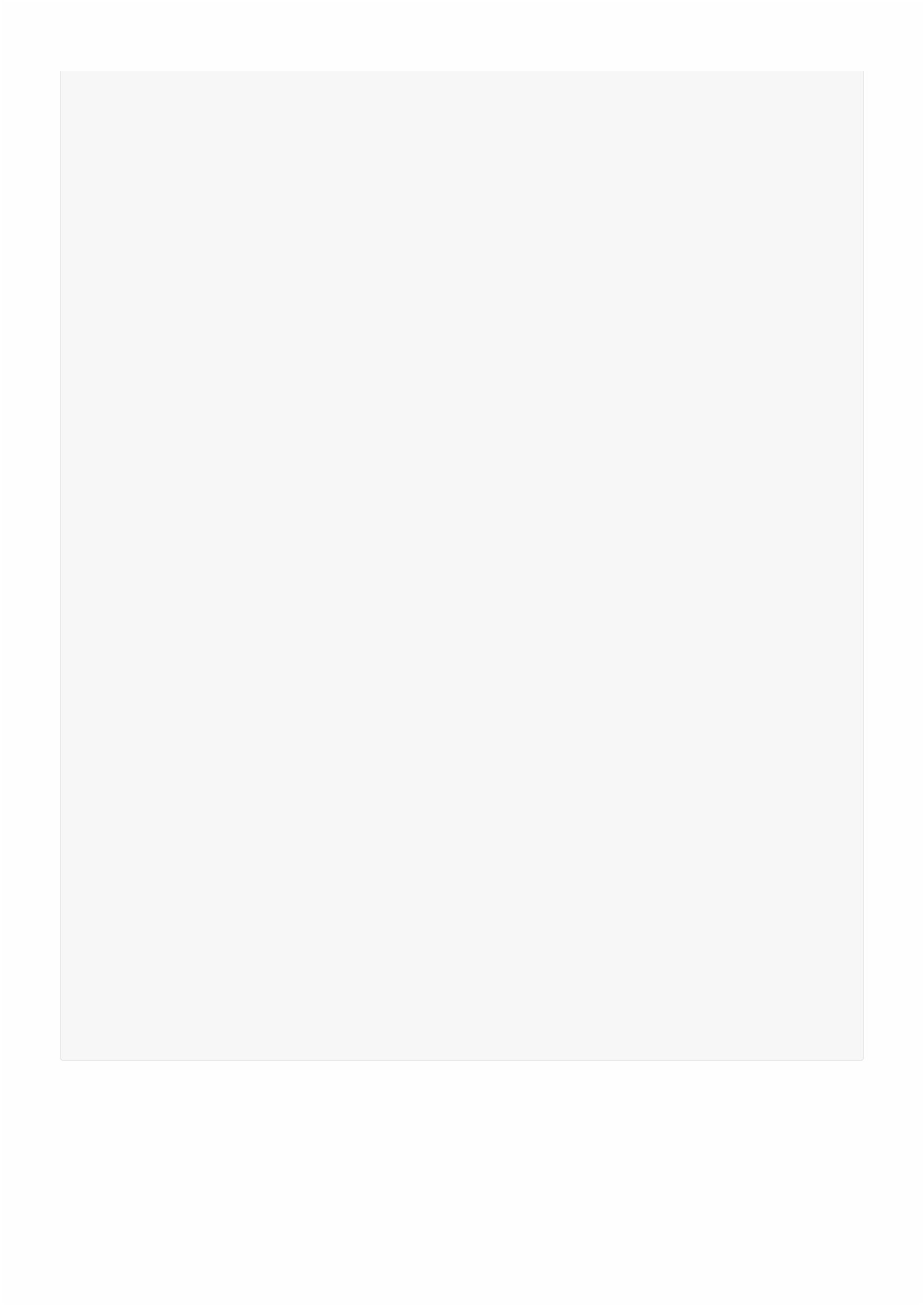
https://www.wps.cn/product/wpslinux

7、单机VNC⼯具:

x11vnc server

8、RealVNC v6.6：

⽀持多个⽤户同时连接



https://www.realvnc.com/en/connect/download/vnc/linux/

9、⽂本编辑器：

visual studio code

10、markdown⼯具：

https://www.typora.io/#linux

sudo apt-get install typora=0.9.60-1

11、Ubuntu 桌⾯3D特效：

sudo apt-get install compiz-plugins compizconfig-settings-manager

12、左侧菜单在底栏显示：

~$ gsettings set com.canonical.Unity.Launcher launcher-position Bottom

13、VMware workstion桥接⽹卡设置

https://kb.vmware.com/s/article/287?lang=zh\_CN

chmod a+rw /dev/vmnet0 #解决⽹卡桥接不通问题

14、远程⼯具:

SecureCRT

15、python开发：

pycharm

16、java开发：

Eclipse

Intellij IDEA

17、vmware workstation：

#⽹卡开启混杂模式 https://kb.vmware.com/s/article/287?lang=zh\_CN

$ sudo vim /etc/init.d/vmware

126 # Start the virtual ethernet kernel service

127 vmwareStartVmnet() {

128 vmwareLoadModule $vnet

129 "$BINDIR"/vmware-networks --start >> $VNETLIB\_LOG 2>&1

130 chgrp magedu /dev/vmnet\*

131 chown magedu /dev/vmnet\*

132 chmod a+rwx /dev/vmnet\*

133 }