P2V/V2V 用户手册

杭州数梦工场科技有限公司

★ 内部使用文档 ★

目　录

目　录 2
1 综述 4
2 .迁移 5
2.1 源主机为 CentOS 系统 5
2.1.1 迁移准备 5
2.1.2 迁移操作步骤 5
2.2 源主机为 Ubuntu 系统 9
2.2.1 迁移准备 9
2.2.2 迁移操作步骤 9
2.3 源主机为 Windows Server 9
2.3.1 迁移准备 9
2.3.2 迁移操作步骤 10
3. AWS 平台操作过程 10
3.1 支持 AWS EC2 实例迁移的操作系统类型 10
3.2 迁移机的创建步骤 11
3.3 迁移操作过程 18
3.3.1 windows 源主机 18
3.3.2 Linux 源主机 24
4. AZure 平台操作过程 33
4.1 支持 AZure 实例迁移的操作系统类型 33
4.2 迁移机的创建步骤 33
4.3 迁移操作 36
5. CentOS 制作 nbd.ko 37
5.1 前言 37
5.1.1 特殊处理说明 37
5.2 配置依赖环境 38
5.3 编译加载 nbd 模块 41

修订历史

版本	日期	修订人	备注
0.1	2017-7-28	庄首	初稿
0.2	2017-8-10	怡天	修改综述
0.3	2017-8-11	登风	修改 2.1.2 迁移操作步骤
0.4	2017-8-11	王鑫	修改 2.3.1 windows 迁移准备
0.5	2017-8-15	韦巧曼	修改 AWS 及 Azure 平台操作过程

1 综述

- P2V 迁移

P2V 指迁移物理服务器上的操作系统及其上的应用软件和数据到阿里云平台管理的 ECS 服务器中。这种迁移方式，主要是使用各种工具软件，把物理服务器上的系统状态和数据镜像到一个虚拟磁盘文件中，并上传至阿里云平台中，确保运行的 ECS 和原有物理服务器运行的应用及数据一致。

- V2V 迁移

V2V 是指从其他云平台或传统虚拟化平台的虚拟主机迁移到阿里云的 ECS 虚拟主机，比如 VMware 迁移到阿里云，AWS 迁移到阿里云等。

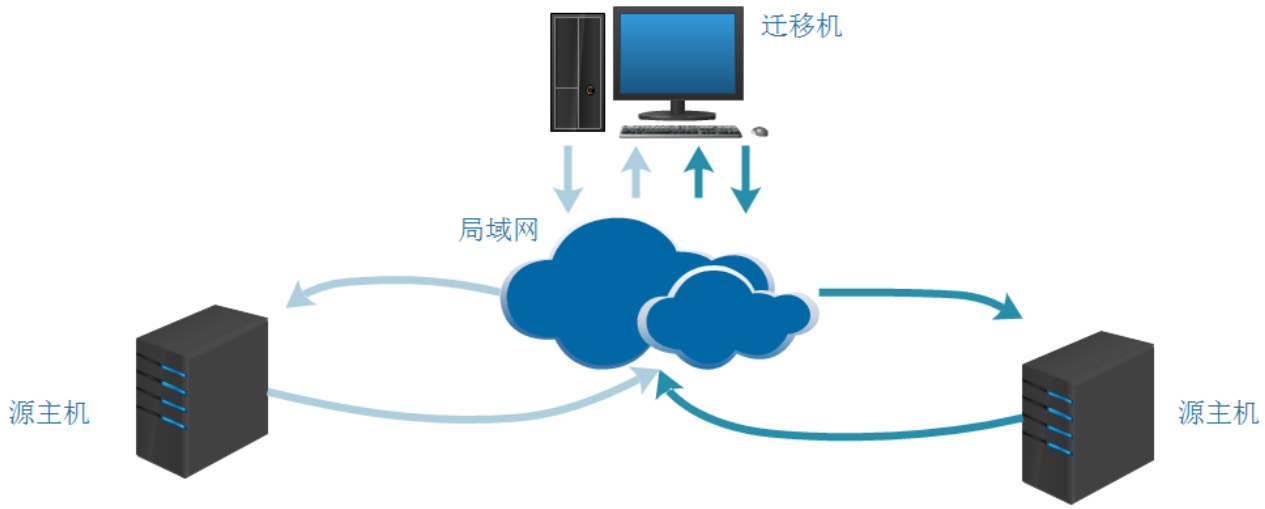


图 1-1 系统迁移流程图

其　中：

迁移机：安装有 P2V/V2V 工具的主机。

源主机：用户待迁移主机。

P2V/V2V 主要流程为：安装有 P2V/V2V 工具的迁移机对源主机环境环境进行校验，校验满足后对源主机发起迁移请求，同时迁移机在本地开启端口监听；源主机将本地的文件通过网络传送到迁移机上并保存到对应的磁盘文件中。

依赖条件：

1、 P2V/V2V 迁移 windows 源主机依赖 WMI、RPC 等服务；

2、 P2V/V2V 迁移 Linux 源主机，依赖 ssh(22 端口)；

3、 P2V/V2V 迁移 Linux 源主机因涉及到对镜像文件的磁盘操作，依赖 Linux 网络块设备 NBD；

4、 镜像拷贝，因涉及整个系统的拷贝，数据流量较大；

综上 P2V/V2V 约束条件为：

➢ 迁移机和源主机必须是同一局域网内，并确保迁移机和源主机可以互相 PING 通；

➢ 迁移机

1）保证安装有.NET Framework 4.0 以上版本

2）关闭防火墙

➢ windows 源主机

1）保证开启 RPC 和 Windows Management Instrumentation 服务；

2）分区（卷）必须是 NTFS 格式；

3）关闭防火墙；

4）不支持 GPT 磁盘格式；

5）保证安装有.NET Framework 4.0 以上版本；

6）工具将提供环境准备批处理脚本，请根据 windows 迁移步骤下载脚本执行即可准备好源主机环境；

➢ linux 源系统

1）对于 CentOS 和 Ubuntu 系统没有默认加载 nbd.ko，P2V/V2V 会对官方标准的 CentOS 和 Ubuntu 默认加载 nbd.ko 文件。目前工具支持全系列 Ubuntu 版本，支持标准的 CentOS 操作系统为：CentOS5.4/5.6/5.8/5.11 -64 位、CentOS 6.3/6.5~6.9/ -64 位、CentOS 6.8~6.9/ -32 位、CentOS 7.0~7.3 -64 位，其它内核版本请用户自行编译并加载 nbd.ko，详细步骤见第五章 nbd 编译加载。

2）执行 P2V/V2V 工具提供的 auto_check_dependence.sh 脚本，确保所有条件均被满足。

2 .迁移

2.1 源主机为 CentOS 系统

2.1.1 迁移准备

1. 源主机执行脚本 auto_check_dependence.sh，检查限制条件是否已经满足。

2. 手动编译加载 nbd 内核模块，参考第五章 nbd 内核编译加载章节（可选、操作系统不在工具默认加载 nbd 模块之列，默认支持的系统版本请参见综述中 linux 源系统说明）。

3. 系统重新加载 nbd 模块（可选、步骤 1、2 执行以后，系统出现重启情况）。

1）工具默认加载 nbd 模块，终端执行命令： **insmod /root/nbd.ko max_part=16。**

2）用户手动编译加载 nbd 模块，终端执行命令：**insmod /lib/modules/内核版本 /kernel/drivers/block/nbd.ko max_part=16。**

2.1.2 迁移操作步骤

2.1.2.1 填写原系统信息

以 Centos7.2 为例，使用 P2V 软件进行迁移，配置样例见图 2-1 所示。

1. 设置源机器信息。

1) 源系统类型：选择运行中的物理机

2) 源机器位置：远程

3) 操作系统：linux

4) 源机器 IP：根据实际情况填写。(查看 ip 命令：ip ad）

5) 源机器密码：根据实际情况填写

2. 点击下一步。

图 2-1

2.1.2.2 环境检查

1. 界面弹出警示框提醒用户是否已经执行了环境检查脚本，同时提供下载脚本功能。一般用户第一次迁移时， 需要通过此功能下载环境检查脚本，如图 2-2 所示。

2. 若未执行脚本，用户可单击“下载脚本”链接下载脚本，并到源主机上执行此脚本检查环境。

3. 若脚本执行后，无提示有失败项，则说明源主机依赖环境初步符合迁移条件。

4. 脚本执行成功后，点击按钮“是”：

1）源主机的操作系统在支持自动加载 nbd 模块操作系统范围之内，则直接进入 2.1.2.3。

2）源主机的操作系统不在支持自动加载 nbd 模块操作系统范围之内，则会跳出消息提示框提示用户：没有匹配内核的 nbd. ko， 请到源系统手动编译加载此模块。此时用户需根据第五章编译加载 nbd 模块。成功以后，点击确定及下一步， 重新进入环境检查。

图 2.2

2.1.2.3 填写目标信息

1）. 上面一栏是源系统现有的所有逻辑卷，下面一栏是用户需要选择的逻辑卷，用户可以根据需求选择需要转换的逻辑卷（对于需要上云的逻辑卷勾选复选框）。

2）. 镜像文件格式可以选择 RAW 或者 VHD 格式（区别在 RAW 格式拷贝全部磁盘空间，不进行压缩。VHD 格式会进行压缩）。建议存储目录不够大或者需要上云的逻辑卷过大的使用 VHD 格式。

3）. 选择存储目录（注意：存储目录所在磁盘大小要大于转换出来的镜像文件的大小）。

4）. 点击下一步。

配置样例见图 2-3.

图 2-3

2.1.2.4 转换摘要

1）确定转换信息是否正确，是否需要导入镜像。如果不需要导入镜像，选择否。点击提交，进入转换阶段。配置样例见图 2-4。

2）如果需要导入镜像，进入导入镜像阶段。

图 2-4

2.1.2.5 导入镜像

如果需要导入镜像，上一步骤选择是，点击下一步。AccessID 和 Secrect Key 填写用户阿里云账号的 AccessID 和 Secrect Key（获取 AccessID 和 Secrect Key 参考阿里云相关文档）。 点击登陆 OSS，选择需要存放镜像的 Bucket ，配置样例如图 2-5。

图 2-5

2.2 源主机为 Ubuntu 系统

2.2.1 迁移准备

1. 在源主机执行脚本 auto_check_dependence.sh。检查限制条件是否已经满足。

2.2.2 迁移操作步骤

参考 2.1.2 。

2.3 源主机为 Windows Server

2.3.1 迁移准备

1. 在输入源主机 ip、账号和密码后，点击下一步界面弹出警示框提醒用户是否已经执行了环境检查脚本，同时提供下载脚本功能。一般用户第一次迁移时， 需要通过此功能下载环境检查脚本，如图 2-3 所示。

2. 若未执行脚本，用户可单击“下载脚本”链接下载脚本，将会得到一个 dependencescript.zip 压缩文件，并到源主机上解压后执行 WinP2Vcheck.bat 脚本检查环境。如果源主机未装.NET 4.0 执行过程可能会比较长，因为脚本会自动安装.NET，请耐心等待。

3. 若脚本执行后，无提示有失败项，则说明源主机依赖环境初步符合迁移条件。

4. 脚本执行成功后，点击按钮“是”：

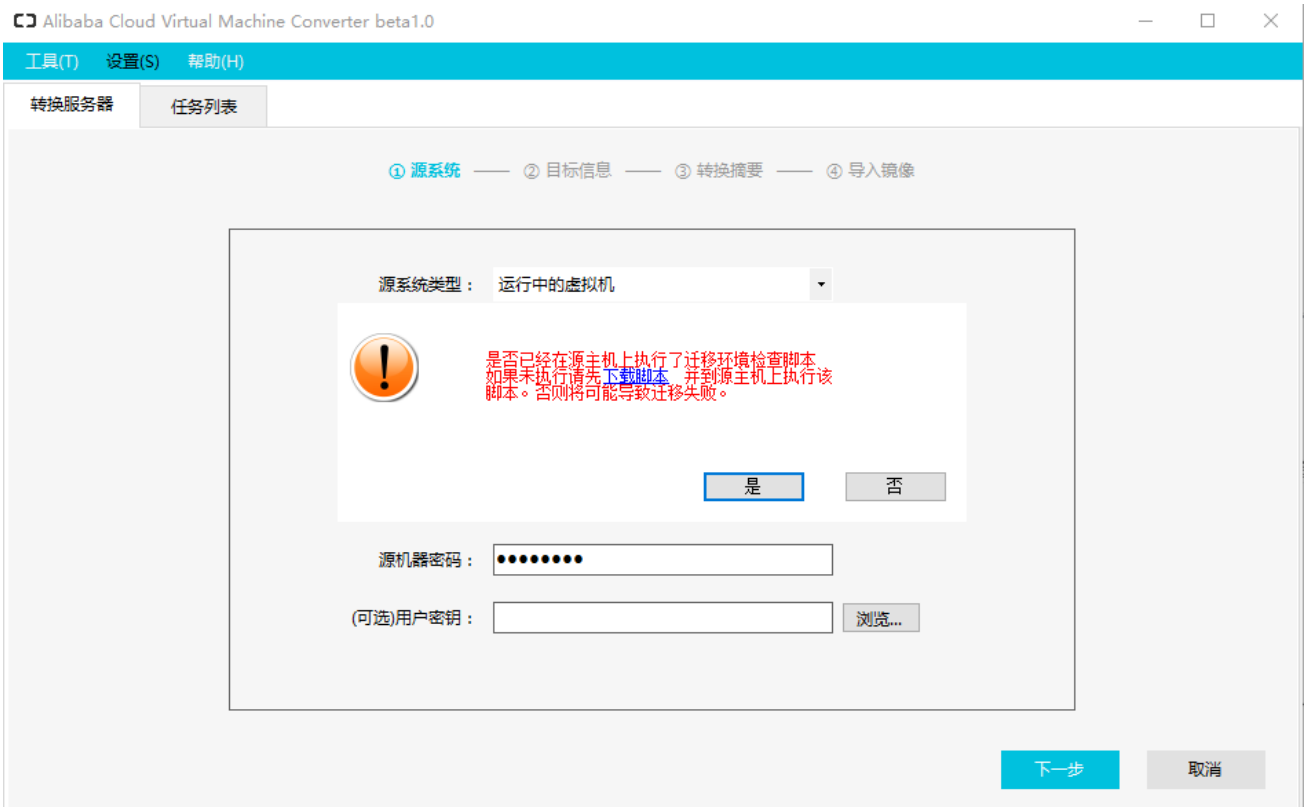


图 2-6

2.3.2 迁移操作步骤

1. 设置源机器信息。

   1）源系统类型：选择运行中的物理机

   2）源机器位置：远程

   3） 操作系统： Windows

   4）源机器 IP：根据实际情况填写。(查看 ip 命令：ipconfig）

   6) 源机器密码：根据实际情况填写

其余参考 2.1.2 。

3. AWS 平台操作过程

3.1 支持 AWS EC2 实例迁移的操作系统类型

➢ Red Hat Enterprise Linux 7.4 (HVM)

- SUSE Linux Enterprise Server 12 SP2 (HVM),
- Ubuntu Server 14.04 LTS (HVM)
- Ubuntu Server 16.04 LTS (HVM)
- Microsoft Windows Server 2008 R2 Base
- Microsoft Windows Server 2012 Base
- Microsoft Windows Server 2016 Base Nano

3.2 迁移机的创建步骤

目标：创建一台和源主机在同一子网下的 windows 操作系统的迁移机。

1. 在平台中选择 EC2->进入创建实例的控制面板，如图 3-1。

图 3-1

2. 点击“启动实例”，迁移机要选择一个 windows 操作系统，本操作选择“Microsoft Windows Server 2016 Base”为例，如图 3-2。

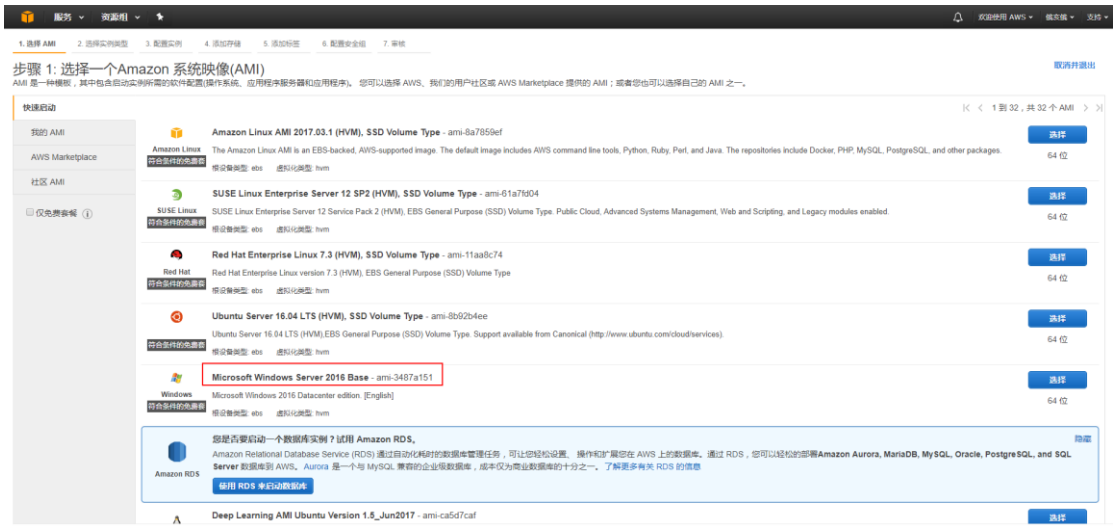


图 3-2

3. 根据业务需要选择实例类型，样例如图 3-3 所示。

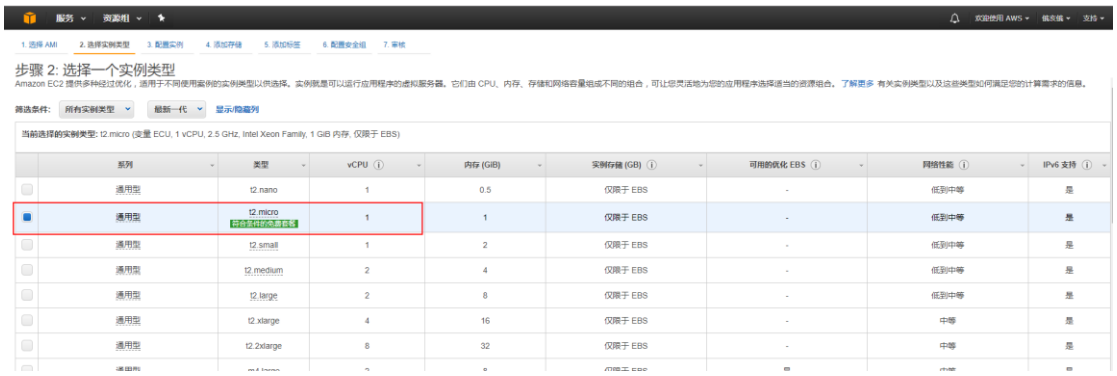


图 3-3

4. 迁移机和源主机都在同一个子网下。本实例选择的同一个子网是 subnet-08f88b61|默认范围 us-east-2a（源主机在该子网下），配置样例如图 3-4 所示。

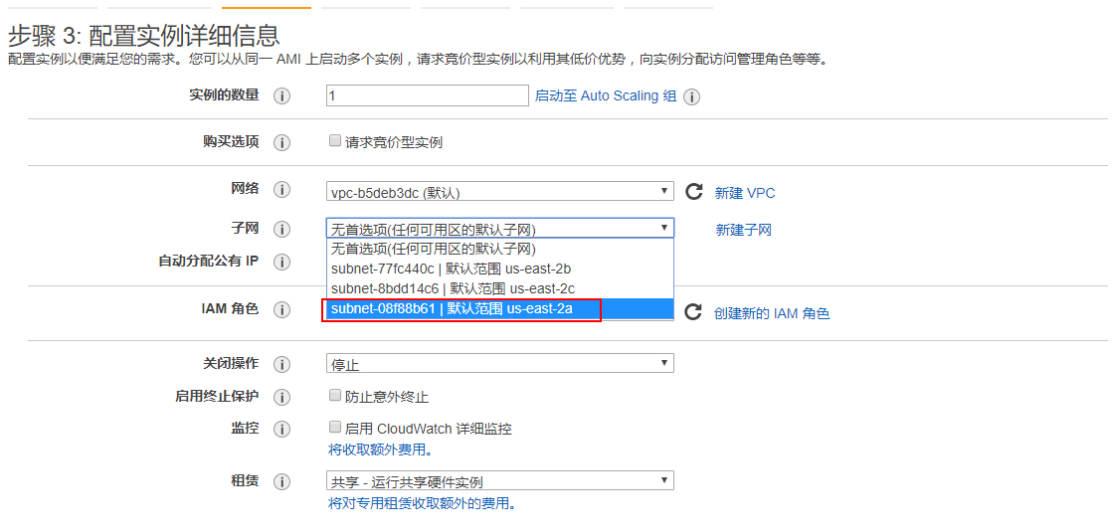


图 3-4

5. 添加存储，根据业务需求选择，实例中选择默认设置。配置样例如图 3-5 所示。

图 3-5

6. 配置安全组，默认入站规则只开放了 3389 端口，但是工具会依赖其他端口，便于迁移顺利，在迁移时可暂时放开所有端口。添加一条“所有流量”的入站规则。迁移完成后再删除即可。配置样例如图 3-6 所示。

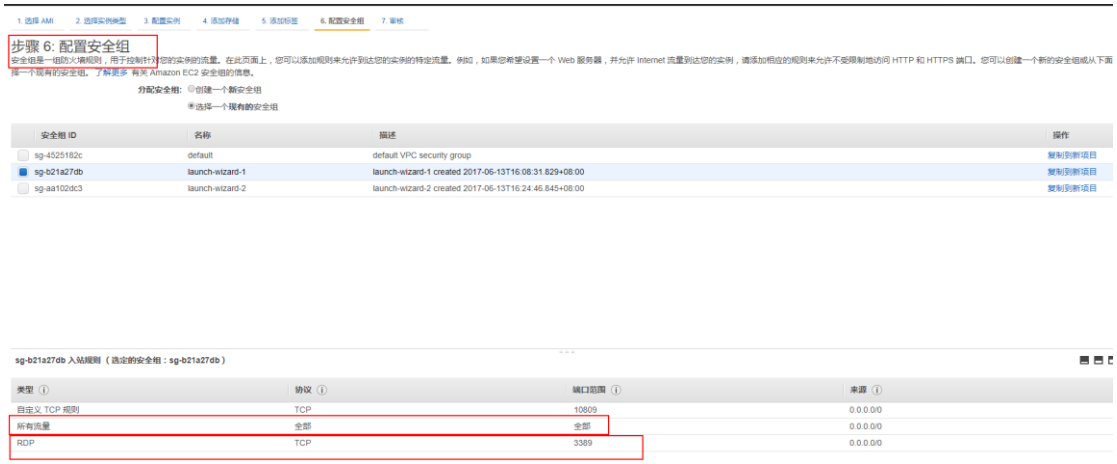


图 3-6

备注：

编辑入站规则的方式：安全组->点击需要编辑的安全组->编辑入站规则，如图 3-7 和 3-8 所示。

图 3-7

图 3-8

7.核查实例启动，配置样例如图 3-9 所示。

图 3-9

8.创建一个密钥对，并下载保存到本地，配置样例如图 3-10 所示。

图 3-10

9.连接实例，配置样例如图 3-11，3-12，3-13 和 3-14 所示。

图 3-11

连接到您的实例 > 获取密码
下列密钥对会在被创建时与此实例关联。
密钥名称 AWSWin2016Key.pem
要检索您的密码，您需要在本地计算机上指定此密钥对的路径:
密钥对路径 选择文件 未选择任何文件
或者您可以复制和粘贴以下密钥对的内容:
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAokyIGjkUB1RqmpmB94OpWo8riF+kgRPOFJ94WWEQNpmCFi4YPf9LcwRrELy
Q
L2dCb7+thP4/oqGvCorV4LVhcJliCGrIYmgrV/j2FH9F0eSbmsEw/9iYIeUWqWoDHY3IQV94+jaZ
KcOAOvUXRHBYdRuha/Our0CkIvWINrxEJP+YZHLeb5+H/w1Jr7G77QKv8fAy9gbhFfhk5NPL1RQ0
解密密码
返回
关闭

图 3-12

连接到您的实例
要连接至 Windows 实例，您可以使用自选的远程桌面客户端，并下载和运行以下 RDP 快捷方式文件：
下载远程桌面文件
收到系统提示时，请使用以下详细信息连接至您的实例：
公有 DNS ec2-13-58-36-255.us-east-2.compute.amazonaws.com
用户名 Administrator
密码 -8RJaf;x?TUU7sR)q=-G75Q?vM3mczI3
如果您已将自己的实例加入目录，便可使用目录证书连接到您的实例。
如果您在连接到您的实例方面需要任何帮助，请参阅我们的 链接文档。
关闭

图 3-13

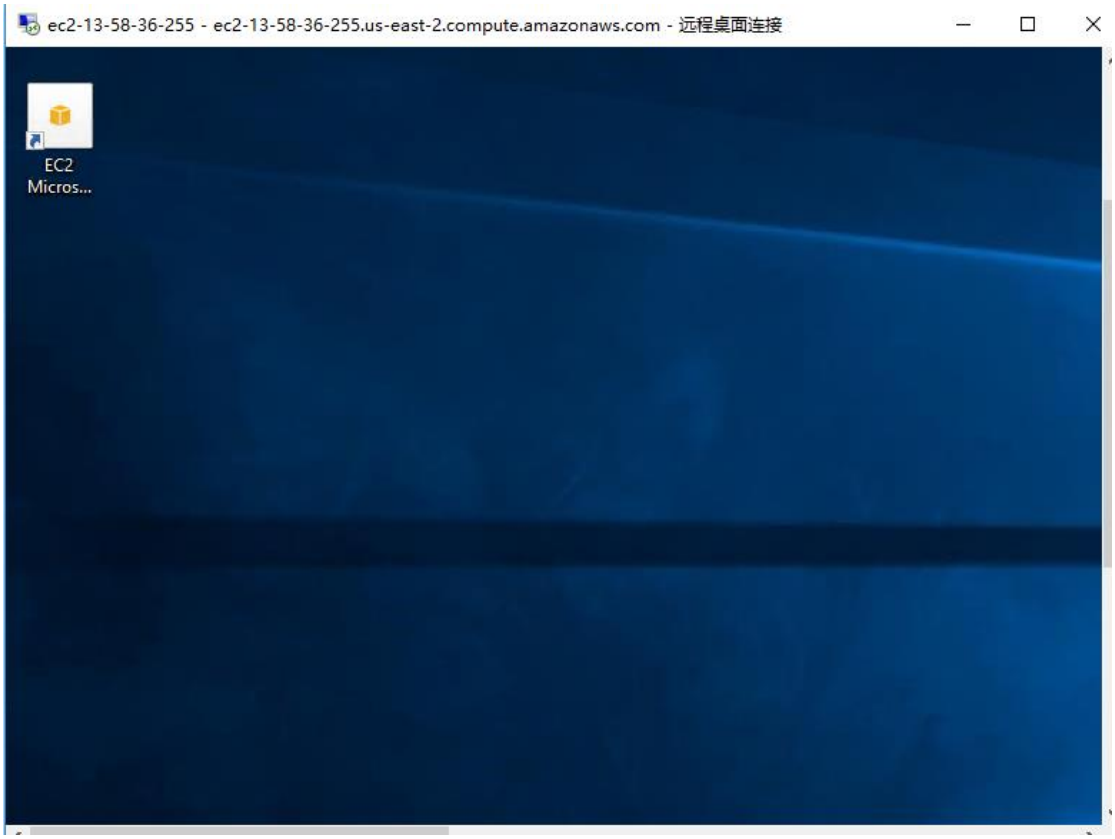


图 3-14

10.关闭迁移机防火墙状态，配置样例如图 3-15 所示。

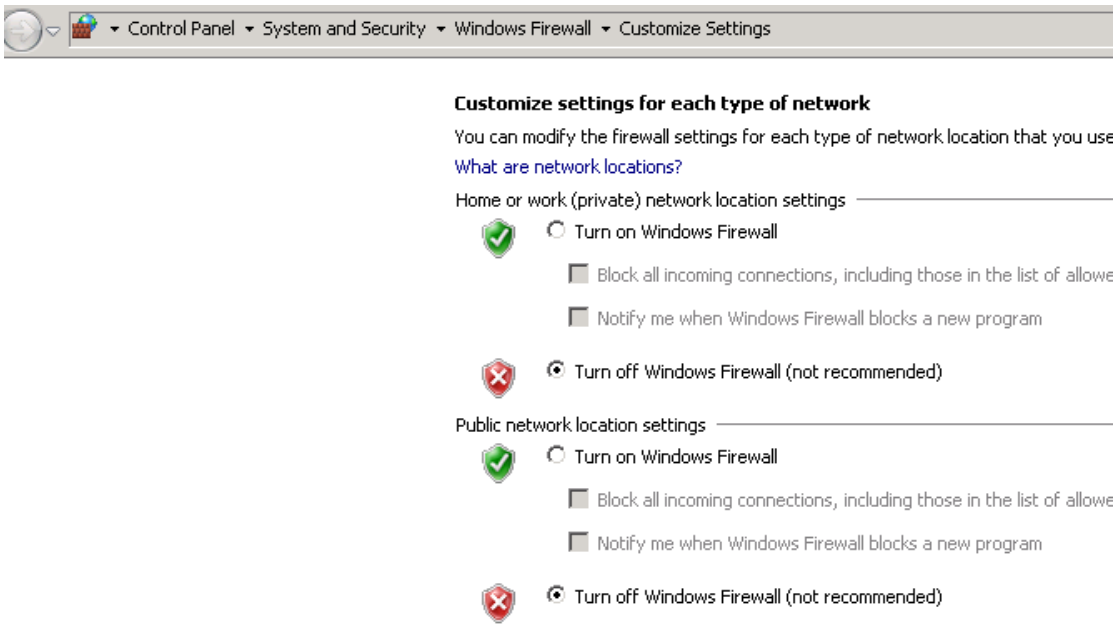


图 3-15

至此，迁移机创建完成。

3.3 迁移操作过程

明确源主机的操作系统类型，然后参考本文档中第 2 部分-“迁移”，对应模块下的操作指示。

- 若源主机是 Redhat 操作系统，可参考 2.1 下 CentOS 系统的操作步骤；
- 若源主机是 Suse 操作系统，可参考 2.2 下 Ubuntu 系统的操作步骤；
- 若源主机是 Windows 操作系统，可参考 2.3 下 Windows Server 系统的操作步骤；

下面是两个实例参考，此参考是在源主机尚未建立时开始举例，若源主机已经存在，可省略 3.3.1.1 和 3.3.2.1 创建源主机的步骤。

3.3.1 windows 源主机

3. 3. 1. 1 源主机的创建步骤

以 windows 2008 为例：

1. 选择一个系统映像，如图 3-16 所示。

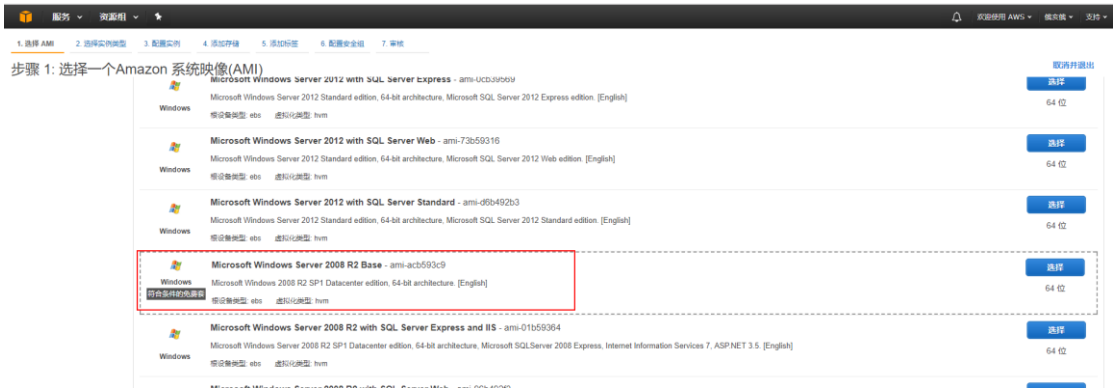


图 3-16

2. 选择一个实例类型，如图 3-17 所示。

图 3-17

3.选择和 Server 端主机在同一个子网，配置样例如图 3-18 所示。

图 3-18

4.添加存储，根据业务需要选择，测试时使用的是默认套餐大小。配置样例如图 3-19 所示。

图 3-19

5. 配置安全组，默认入站规则只开放了 3389 端口，但是工具会依赖其他端口，便于迁移顺利，在迁移时可暂时放开所有端口。添加一条“所有流量”的入站规则。迁移完成后再删除即可。配置样例如图 3-20 所示。

图 3-20

备注：

编辑入站规则的方式：安全组->点击需要编辑的安全组->编辑入站规则，如图 3-21 和 3-22 所示。

图 3-21

图 3-22

6.启动实例和连接实例，配置样例如图 3-23、3-24、3-25 和 3-26 所示。

连接到您的实例
要连接至 Windows 实例，您可以使用自选的远程桌面客户端，并下载和运行以下 RDP 快捷方式文件：
下载远程桌面文件
收到系统提示时，请使用以下详细信息连接至您的实例：
公有 DNS
ec2-13-58-36-255.us-east-2.compute.amazonaws.com
用户名
Administrator
密码
获取密码
如果您已将自己的实例加入目录，便可使用目录证书连接到您的实例。
如果您在连接到您的实例方面需要任何帮助，请参阅我们的 链接文档。
关闭

图 3-23

连接到您的实例 > 获取密码
下列密钥对会在被创建时与此实例关联。
密钥名称
AWSWin2016Key.pem
要检索您的密码，您需要在本地计算机上指定此密钥对的路径:
密钥对路径
选择文件
未选择任何文件
或者您可以复制和粘贴以下密钥对的内容:
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAokyIGjkUB1RqmpmB94OpWo8riF+kgRPOFJ94WWEQNpmCFi4YPf9LcwRrELy
Q
L2dCb7+thP4/oqGvCorV4LVhcJliCGrlYmgrV/j2FH9F0eSbmsEw/9iYIeUWqWoDHY3IQV94+jaZ
KcOAOvUXRHBYdRuha/Our0CklvWlNrxEJP+YZHLeb5+H/w1Jr7G77QKv8fAy9gbhFfhk5NPL1RO0
解密密码
返回
关闭

图 3-24

连接到您的实例
要连接至 Windows 实例，您可以使用自选的远程桌面客户端，并下载和运行以下 RDP 快捷方式文件：
下载远程桌面文件
收到系统提示时，请使用以下详细信息连接至您的实例：
公有 DNS ec2-13-58-36-255.us-east-2.compute.amazonaws.com
用户名 Administrator
密码 -8RJaf;x?TUU7sR)q=-G75Q?vM3mczI3
如果您已将自己的实例加入目录，便可使用目录证书连接到您的实例。
如果您在连接到您的实例方面需要任何帮助，请参阅我们的 链接文档。
关闭

图 3-25

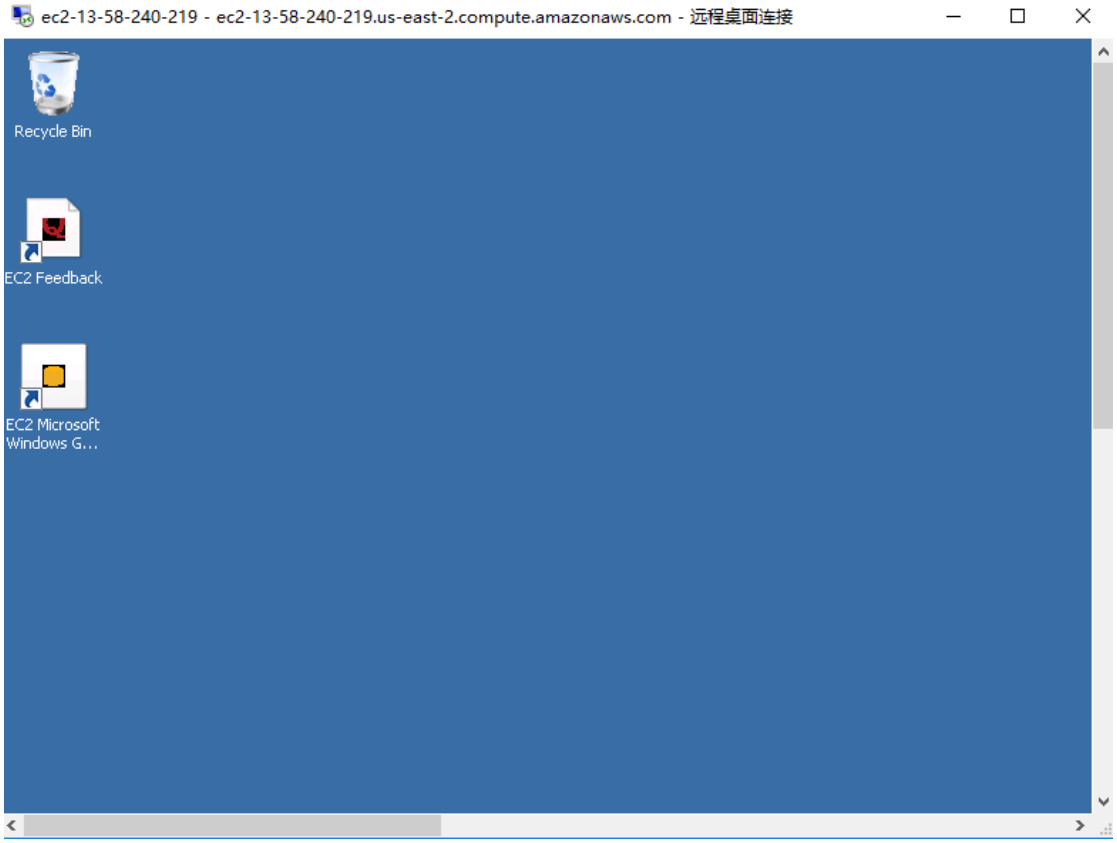

ec2-13-58-240-219 - ec2-13-58-240-219.us-east-2.compute.amazonaws.com - 远程桌面连接
Recycle Bin
EC2 Feedback
EC2 Microsoft Windows G...

图 3-26

至此，源主机也创建完毕，可进行使用工具迁移步骤操作。

3.3.1.2 迁移操作

根据源主机的操作系统类型，参考本文第 2 部分对应模块下的操作步骤。本源主机为 Windows Server，即可参考 2.3 下的相关步骤。

运行的结果如下图所示：

1. 运行 CloudVMC，输入源主机的私网 IP（与迁移机在同一子网下的 IP 地址），用户名：administrator ，密码。配置样例如图 3-27 所示。

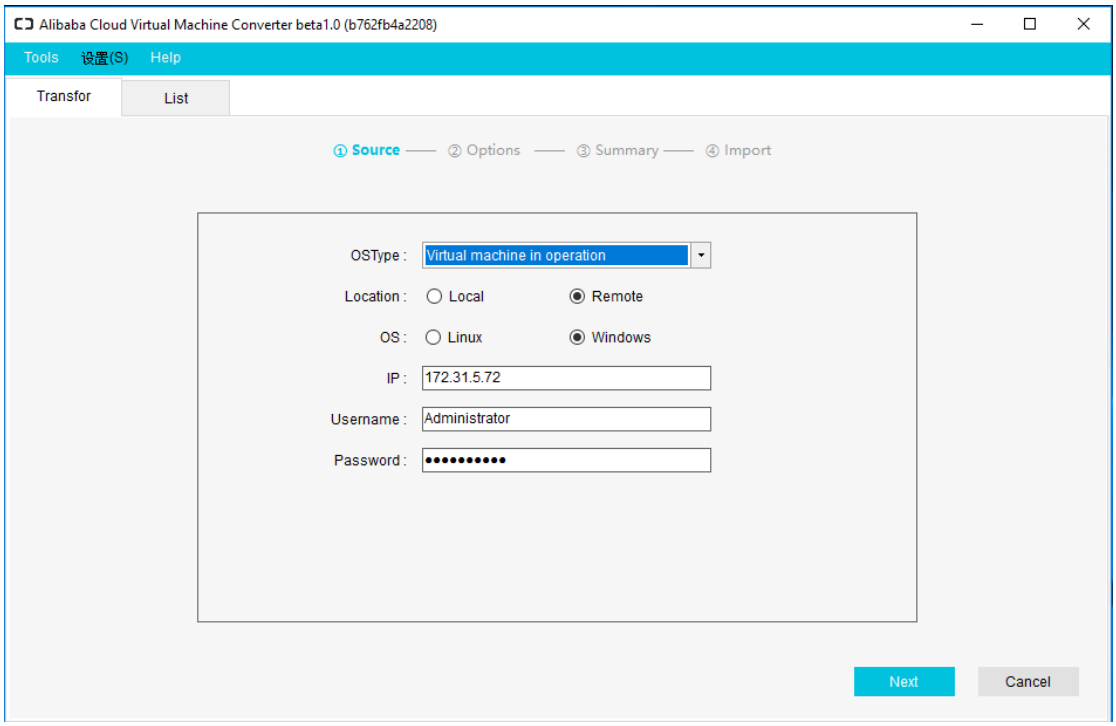


图 3-27

2.勾选需要迁移的磁盘，选择导出的格式 VHD，存放 VHD 的目录。配置样例如图 3-31 所示。

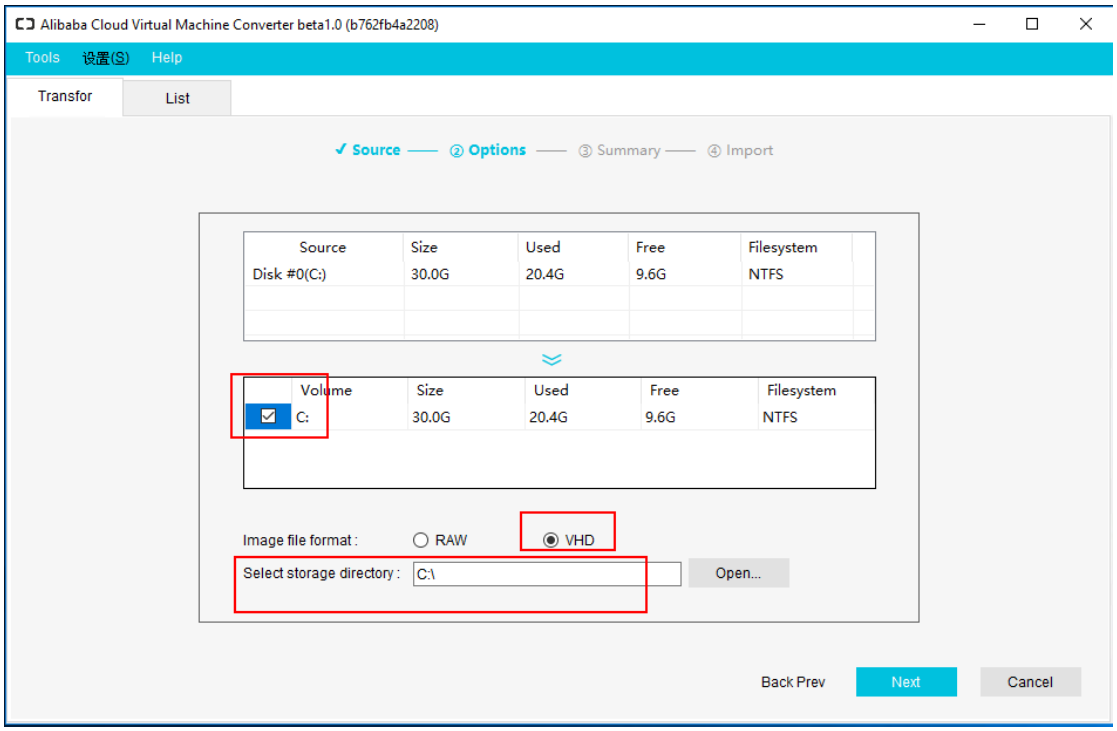


图 3-28

3.3.2 Linux 源主机

3. 3. 2. 1 源主机的创建步骤

这里以 Ubuntu 为例

1.启动实例->选择一个 Amazon 系统映像 Ubuntu Server 16.04 LTS。配置样例如图 3-29 所示。

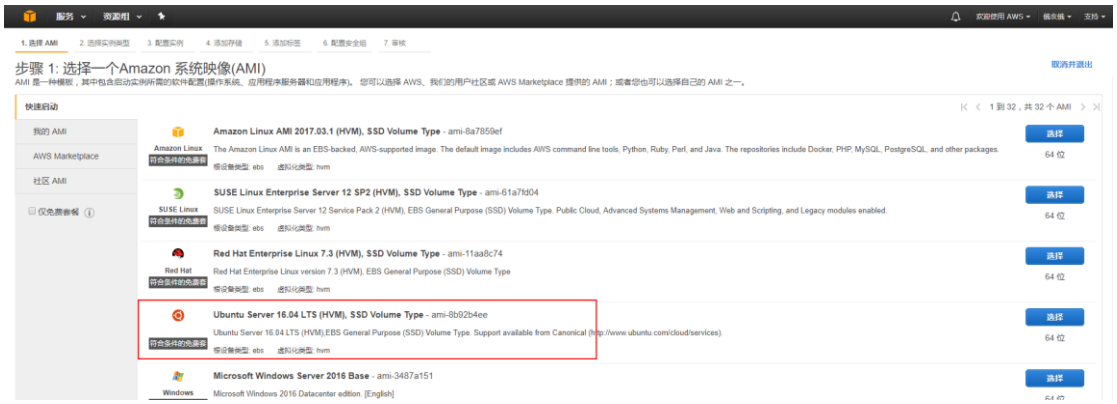


图 3-29

2.选择默认的实例类型，配置样例如图 3-30 所示。

图 3-30

3.配置实例详细信息，此步骤切记将源主机子网选择和迁移机子网相同，配置样例如图 3-31 所示。

图 3-31

4.添加存储，选择默认存储，配置样例如图 3-32 所示。

图 3-32

5.添加标签，实例中未进行操作，配置样例如图 3-33 所示。

图 3-33

6.配置安全组，同理给安全组编辑添加一条“所有流量”的入站规则。配置样例如图 3-34 所示。

图 3-34

7.连接实例，配置样例如图 3-35 和 3-36 所示。

图 3-35

图 3-36

至此，创建源主机的操作步骤完毕，

3.3.2.2 迁移操作

创建 root 用户密码，可用 root 账户及密码方式登录

AWS 上的源主机多是以密钥对的形式登录的，要进行以下 3 步使其可以使用 root 和密码方式登录

1.打开 SSH 客户端，以 xshell 为例。

在迁移机主机安装好 Xshell，Xshell 安装程序可自行在网上搜索。

选择文件->新建会话->输入源主机公网 IP 地址。配置样例如图 3-37 所示。

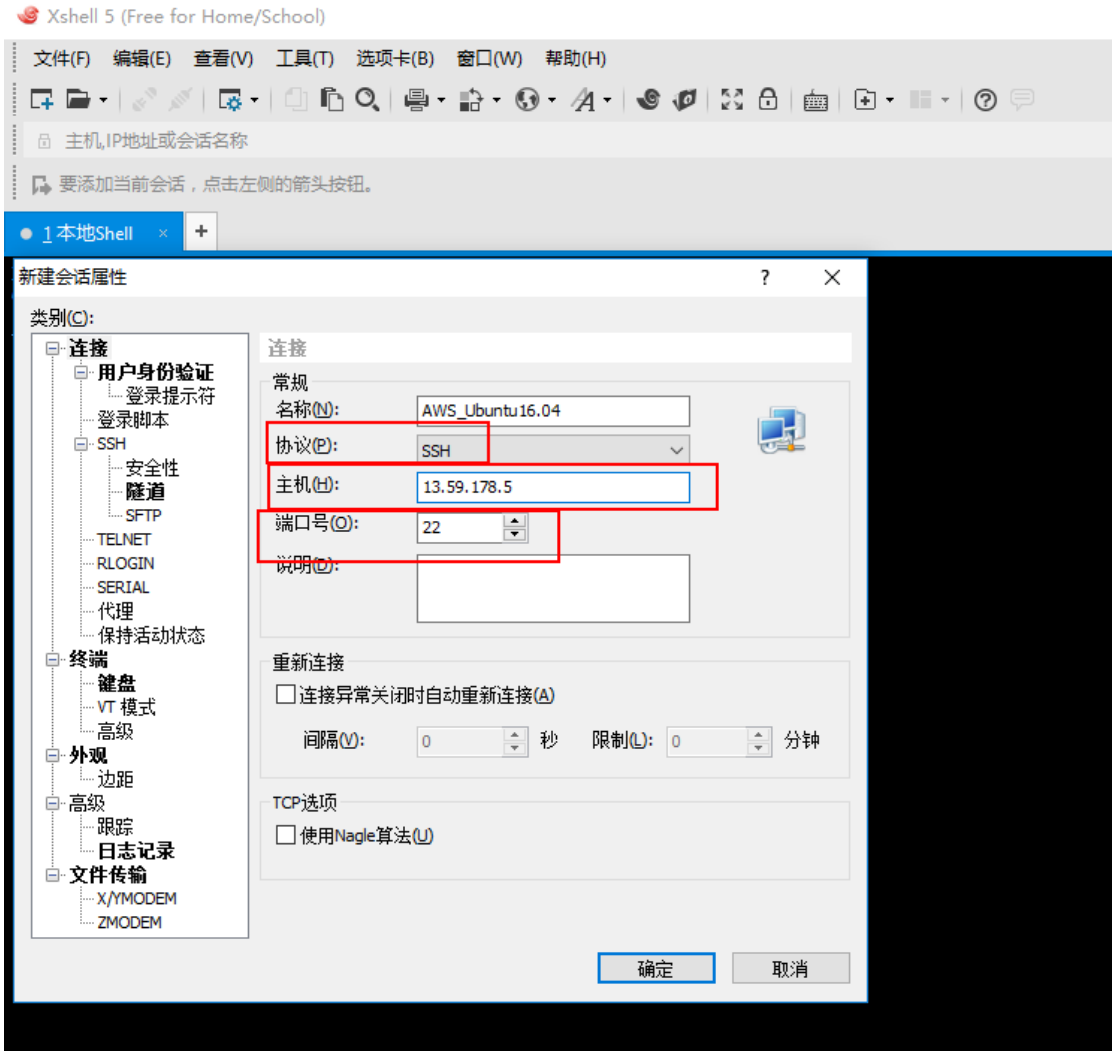
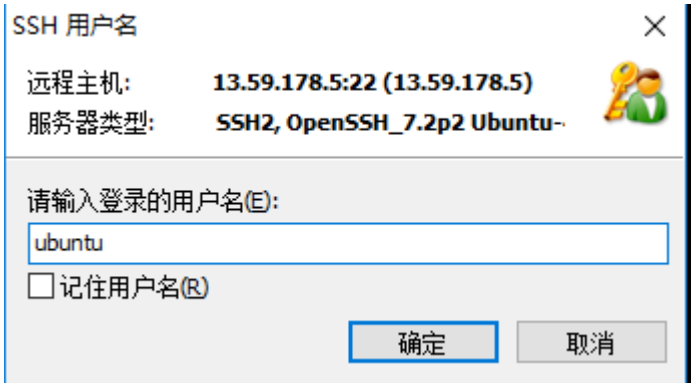
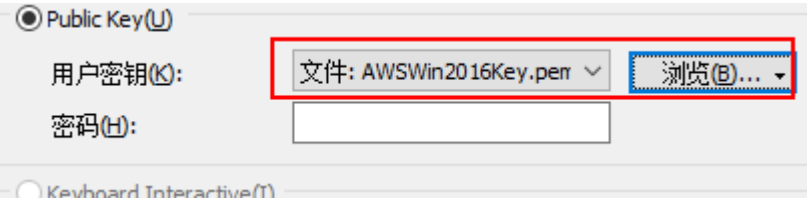


图 3-37

用户名：ubuntu

密钥：选择创建实例时配置的密钥对。





如果出现图 3-38 表示登录成功。

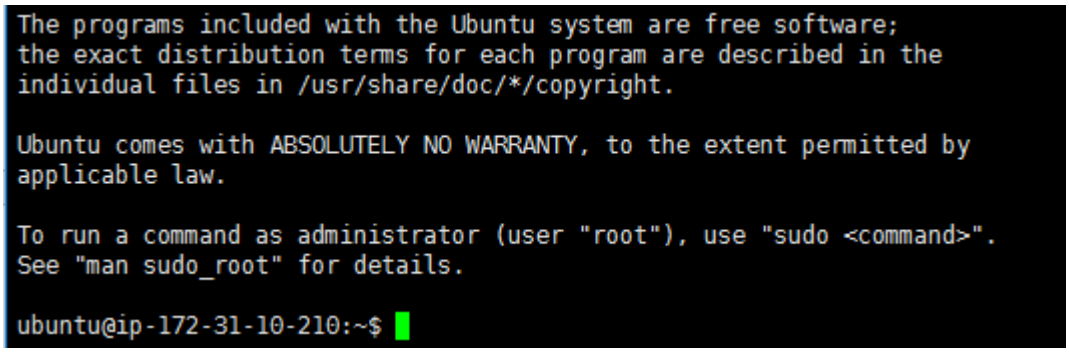


图 3-38

2.给 root 账户设置密码，配置样例如图 3-39 所示。

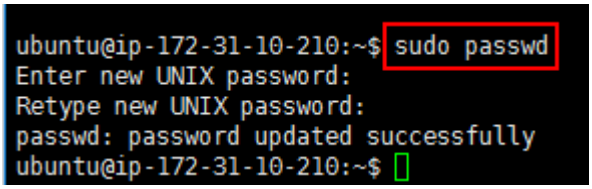


图 3-39

3.编辑 sshd_config，使其能够通过输入 root 账户和密码登录，密码的形式不再是选择密钥文件。

文件路径：/etc/ssh/sshd-config，配置样例如图 3-40 所示。注意配置完成后重启 sshd 服务（重启命令：service ssh restart）。

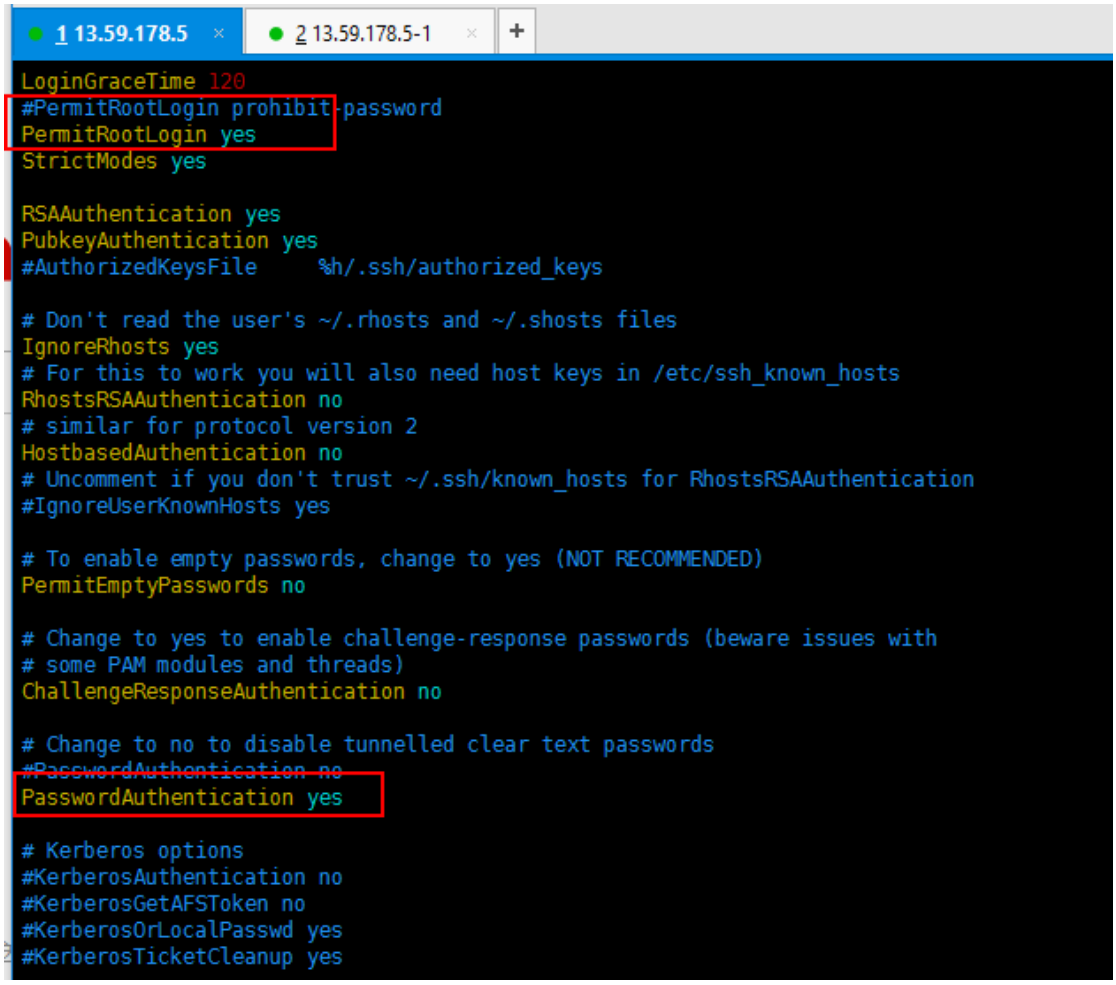


图 3-40

运行工具，导出镜像文件

至此，即可在迁移机运行工具，导出镜像文件。若对工具的使用操作有疑惑，可参考文档中第二部分的 2.2 内容。

以下是运行过程和结果示例：

在迁移机运行工具，输入源主机的内网 IP 地址，配置样例如图 3-41、3-42、3-43 和 3-44 所示。

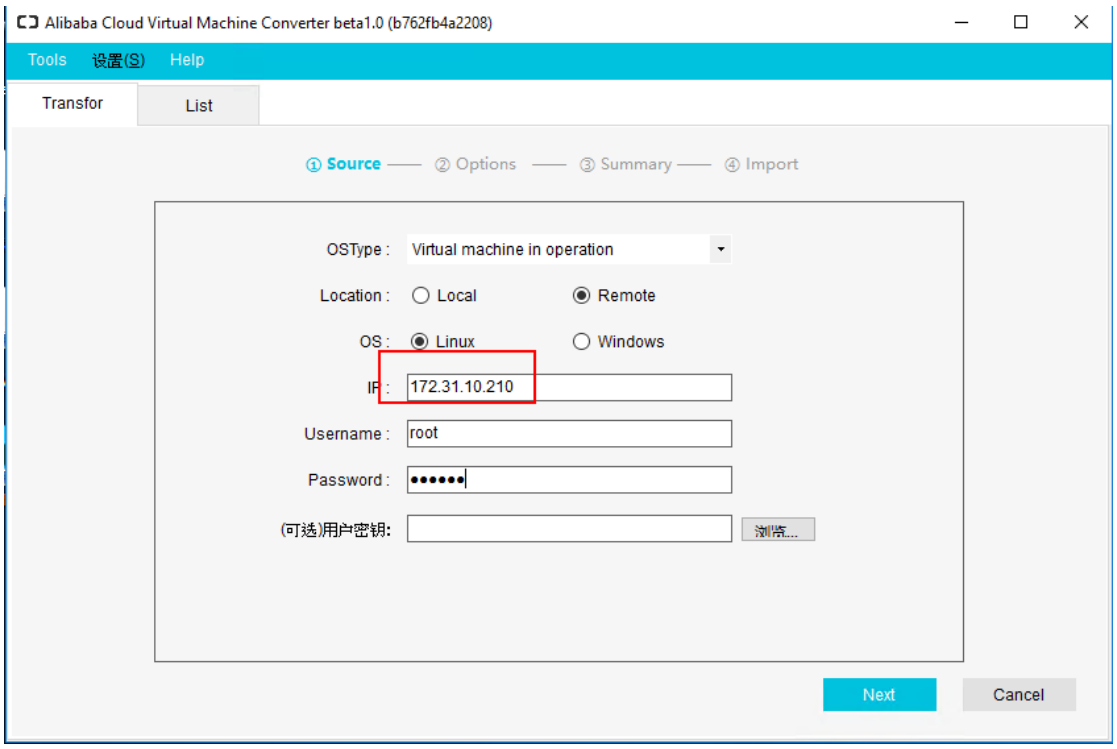

Alibaba Cloud Virtual Machine Converter beta1.0 (b762fb4a2208)
Tools
设置(S)
Help
Transfor
List
① Source
② Options
③ Summary
④ Import
OSType :
Virtual machine in operation
Location :
Local
Remote
OS :
Linux
Windows
IP :
172.31.10.210
Username :
root
Password :
(可选)用户密钥:
浏览...
Next
Cancel

图 3-41

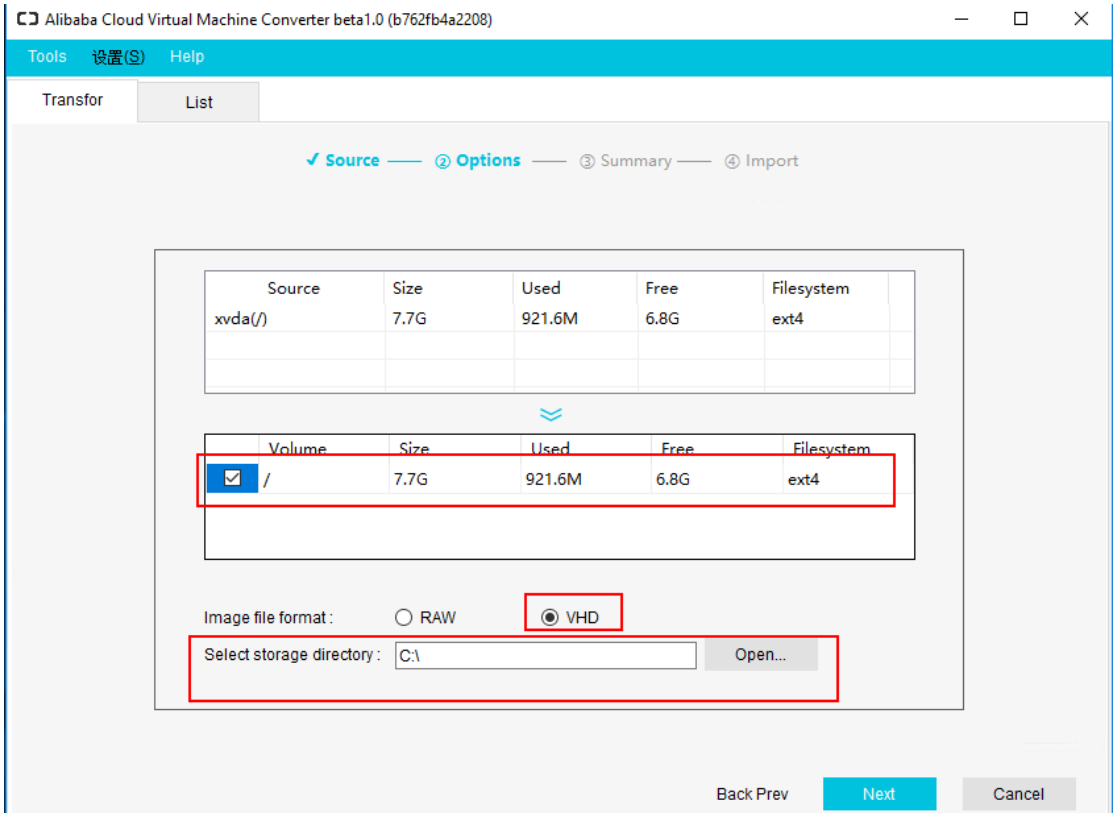

Alibaba Cloud Virtual Machine Converter beta1.0 (b762fb4a2208)
Tools
设置(S)
Help
Transfor
List
✓ Source
② Options
③ Summary
④ Import
Source
Size
Used
Free
Filesystem
xvda(/)
7.7G
921.6M
6.8G
ext4
Volume
Size
Used
Free
Filesystem
/
7.7G
921.6M
6.8G
ext4
Image file format :
RAW
VHD
Select storage directory :
C:\
Open...
Back Prev
Next
Cancel

图 3-42

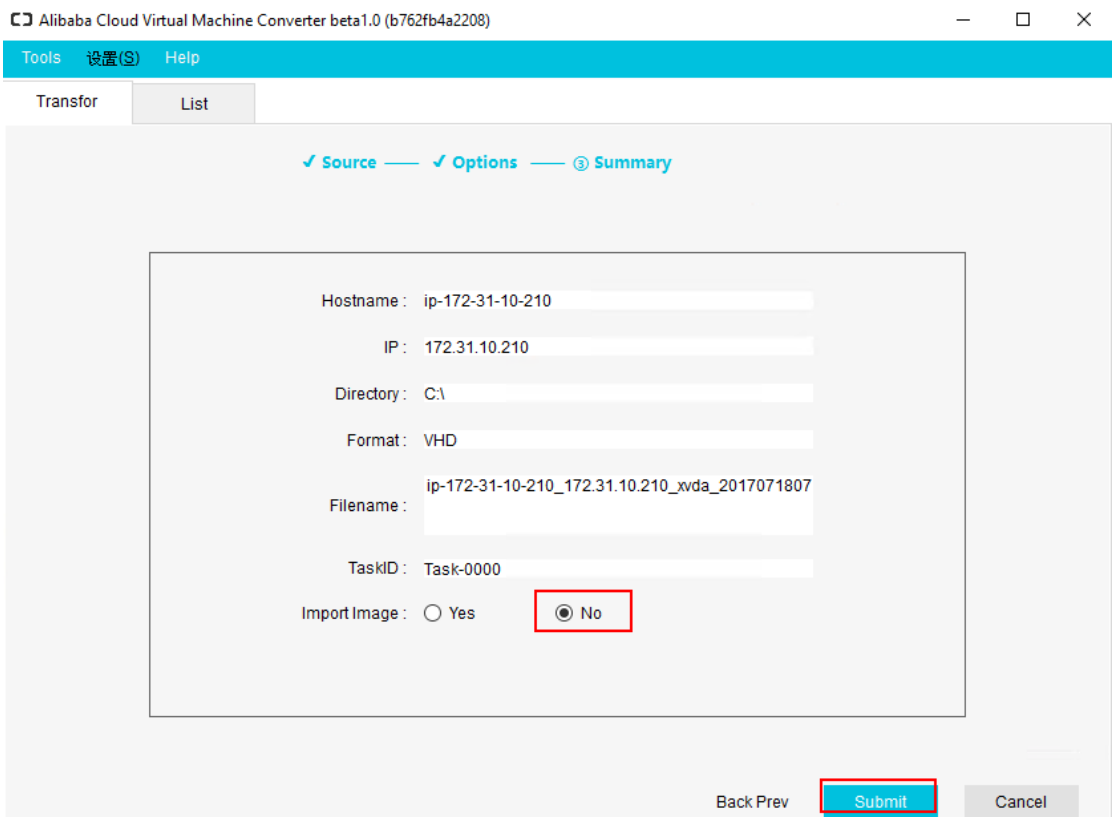


图 3-43

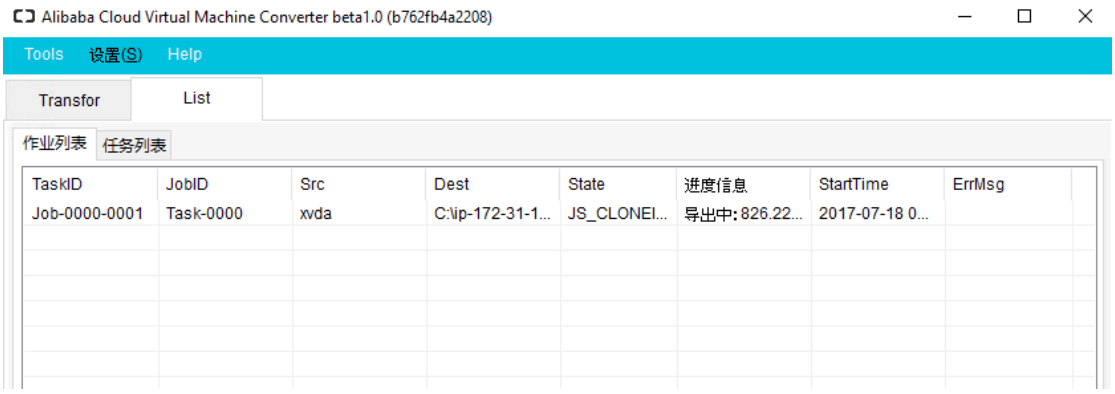


图 3-44

导出完成，如图 3-45 所示：

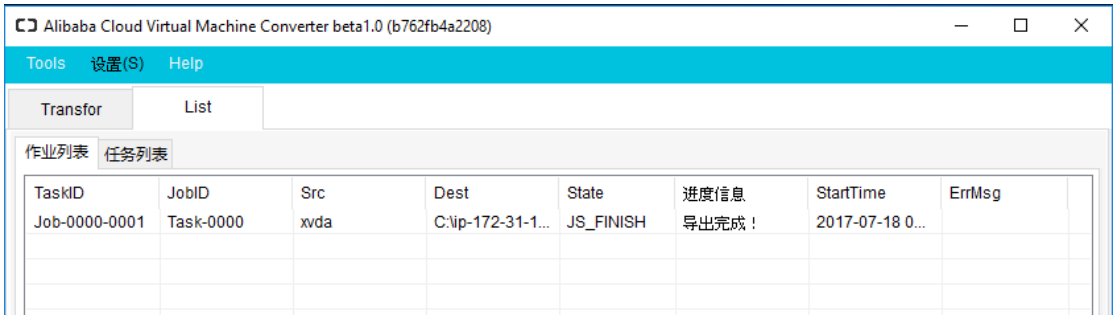


图 3-45

生成的 VHD：

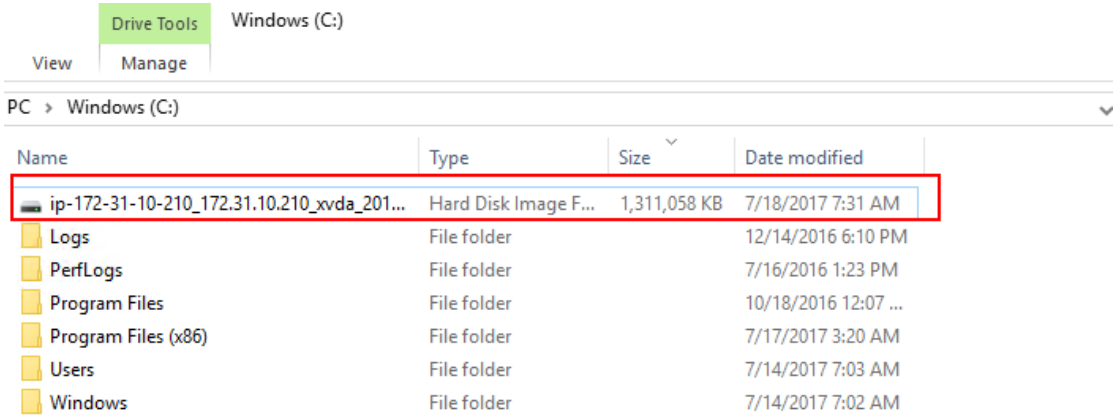


图 3-46

4. AZure 平台操作过程

4.1 支持 AZure 实例迁移的操作系统类型

- Windows server 2016 数据中心版 64 位
- Windows server 2012 数据中心版 64 位
- Windows server 2008 R2 企业版 64 位
- Windows server 2008 标准版 sp2 32 位
- Ubuntu 14.04 64 位
- Ubuntu 16.04 64 位
- CentOS 6.8 32 及 64 位
- CentOS 7.2 64 位
- CentOS 7.3 64 位
- SUSE 12 SP1 64 位

4.2 迁移机的创建步骤

1. 登录 Microsoft Azure 平台，创建一个虚拟网络，若虚拟网络已存在，可省略此步骤。配置样例如图 4-1 所示。

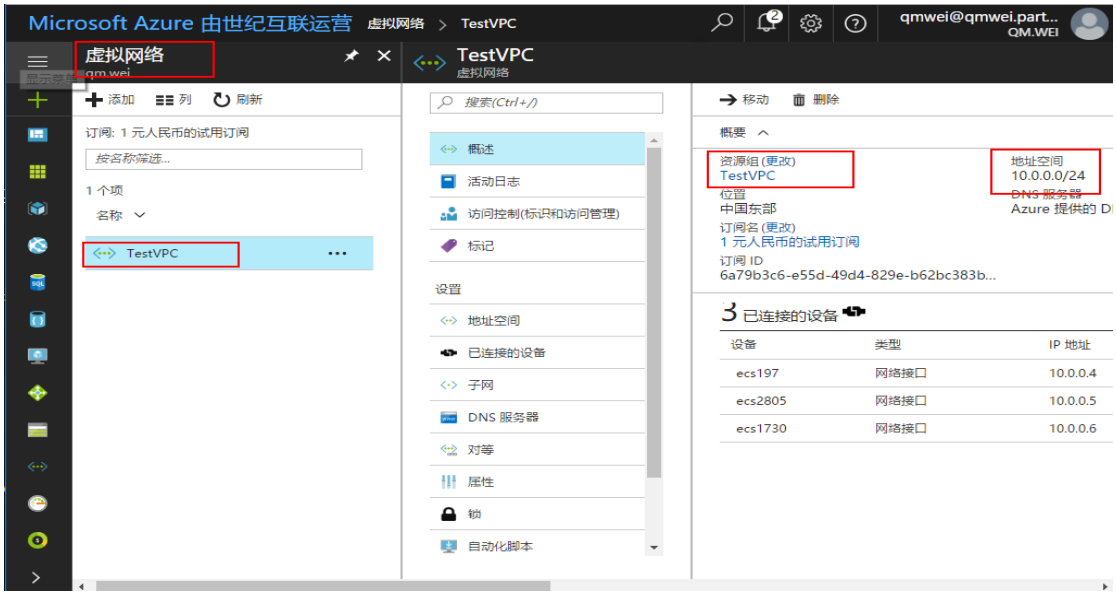


图 4-1

2. 创建一台虚拟机作为迁移机，要注意保证迁移机和源主机在同一个资源组下，配置样例如图 4-2 所示。

图 4-2

3.迁移机以创建 Windows Server 2008 R2 SP1 为例。配置样例如图 4-3 所示。

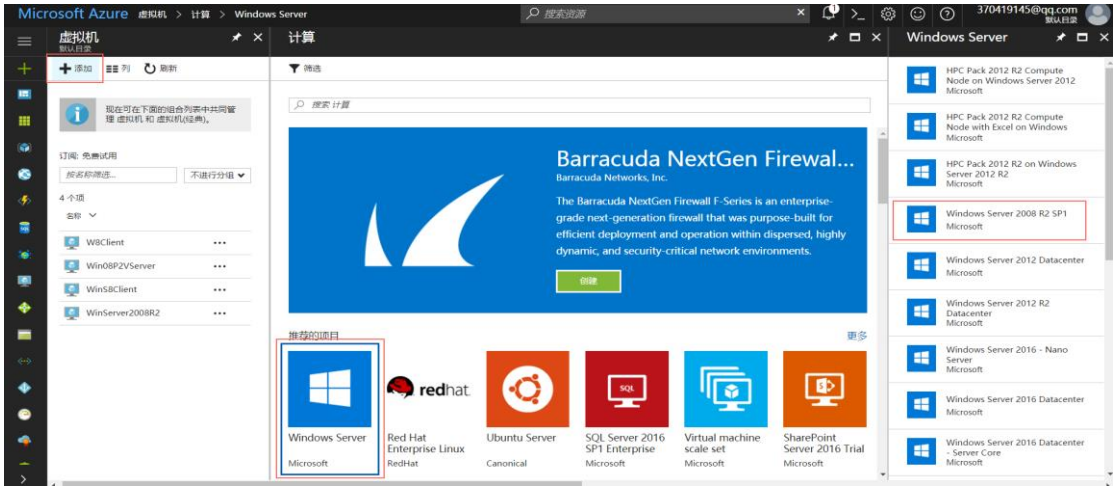


图 4-3

4.“Resource group” 和“位置”迁移机和源主机保持一致，配置样例如图 4-4 所示。

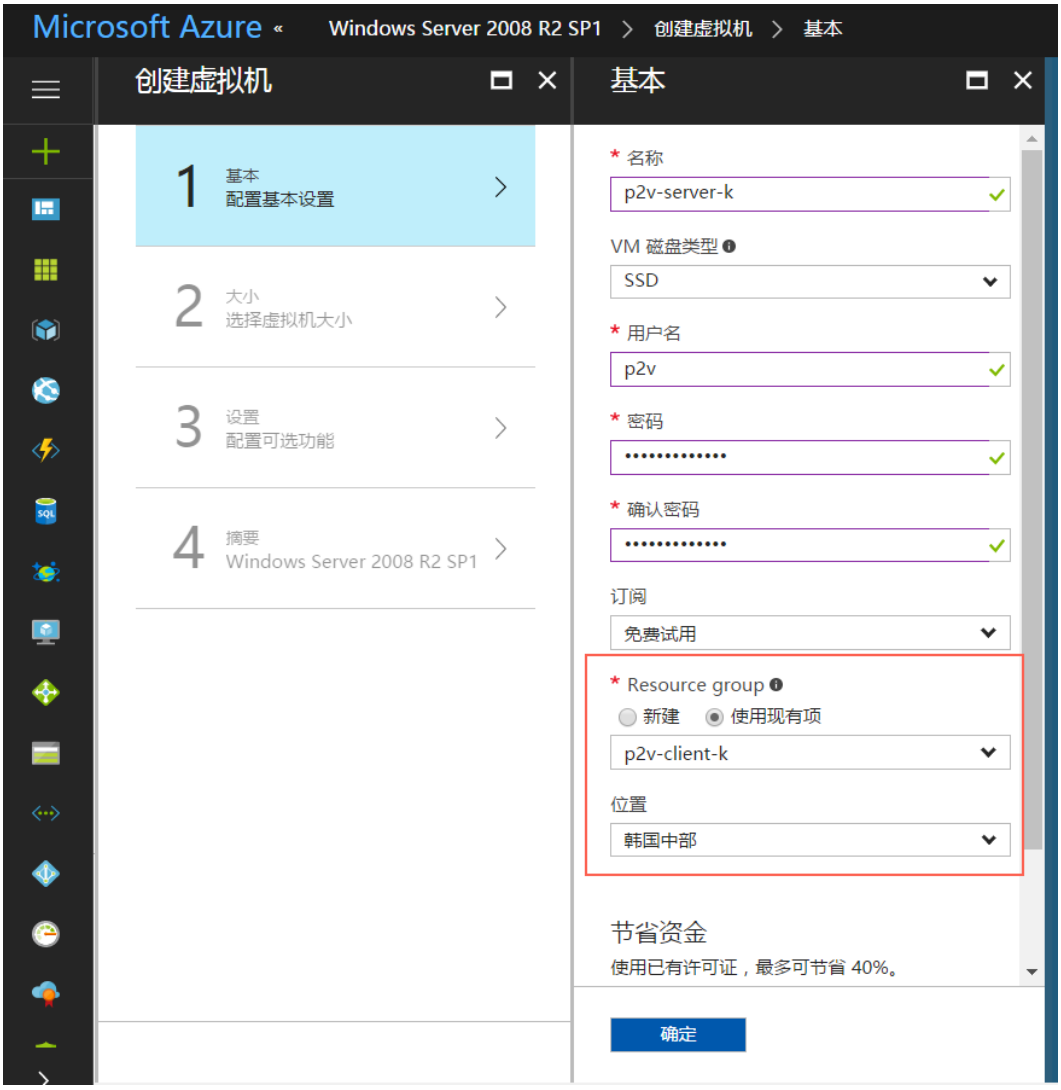


图 4-4

5.虚拟网络，源主机和迁移机在同一虚拟网络，配置样例如图 4-5 所示。

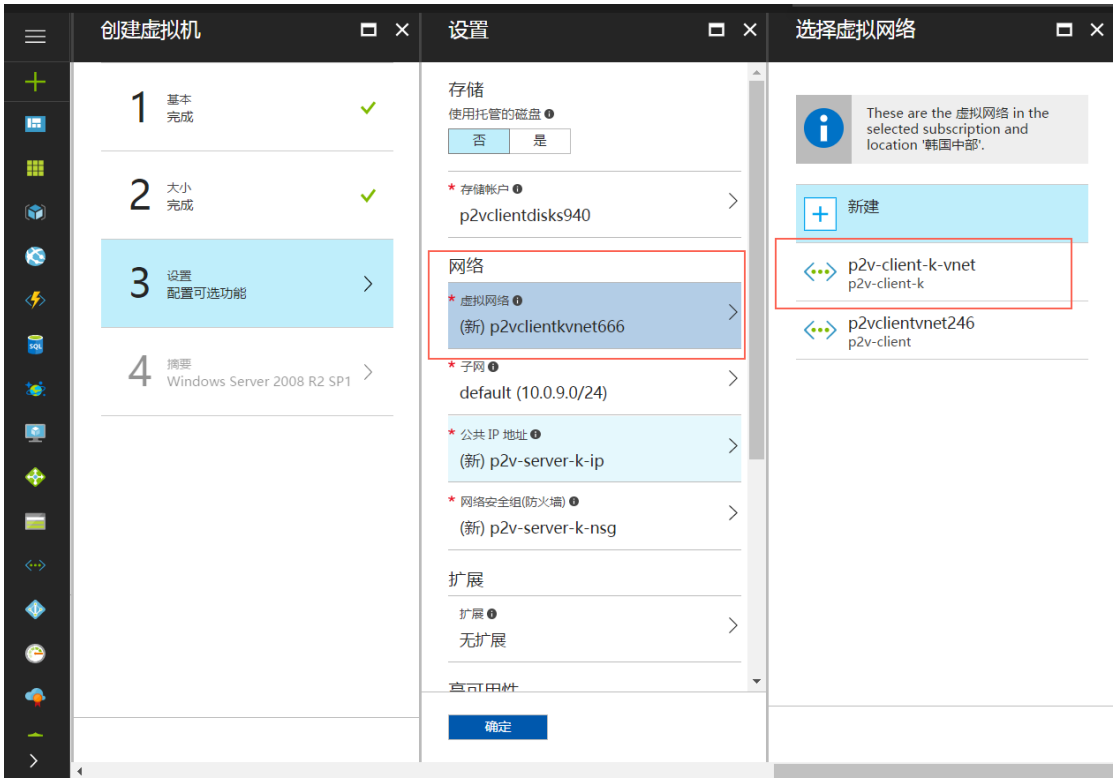


图 4-5

6.至此，和源主机在同一虚拟网络下的迁移机创建完成，如图 4-6。

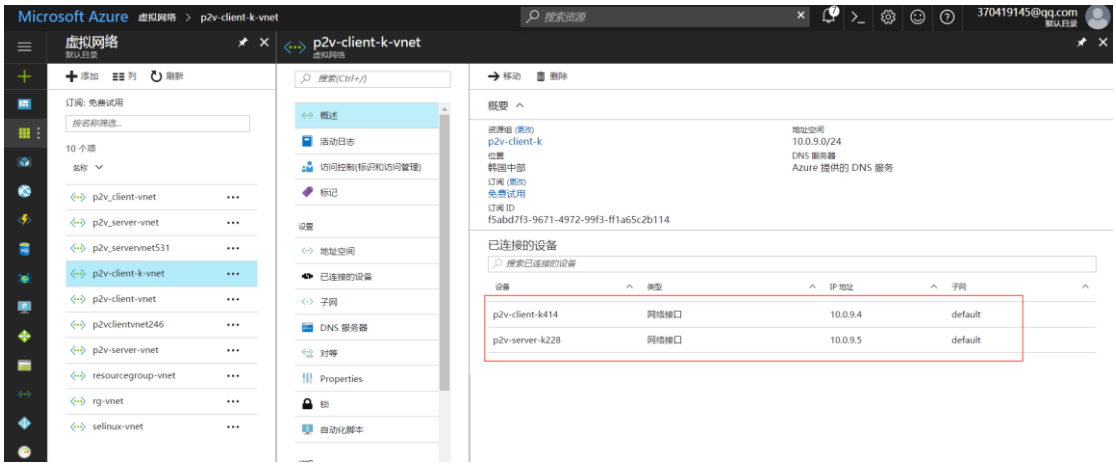


图 4-6

4.3 迁移操作

使用迁移工具的步骤相同，若有疑惑，可根据源主机的操作系统类型，参考第 2 部分对应模块下迁移步骤。

- 若源主机是 CentOS 操作系统，可参考 2.1 下 CentOS 系统的操作步骤；
- 若源主机是 Suse/Ubuntu 操作系统，可参考 2.2 下 Ubuntu 系统的操作步骤；
- 若源主机是 Windows 操作系统，可参考 2.3 下 Windows Server 系统的操作步骤。

5. CentOS 制作 nbd.ko

5.1 前言

本文档以 CentOS5.8 编译加载 nbd 模块为例，说明 CentOS 制作 nbd 模块及加载流程。其它 CentOS 版本编译加载流程与此基本相同，如有特殊情况请尝试自行网上解决或联系阿里云技术人员。

CentOS 发行版没有内置 nbd 模块需要手动编译加载，同时加载的 nbd 模块如果与内核版本不一致，则在迁移过程中存在宕机的风险。

制作 nbd 模块之前请先查看特殊处理说明，理解为什么需要源码安装及不同版本的一些差异。

本文档适用人员为：迁云开发工程师、迁云运维工程师、迁云测试工程师。

5.1.1 特殊处理说明

1、 CentOS 官网地址，制作模块相关源码请在此网站下载

http://vault.centos.org

2、 核查 kernel-devel，kernel-heads 版本。

kernel-devel 版本与内核版本必须一致，否则编译模块不能通过。在开始制作 nbd 模块之前先检查 kernel-devel 是否安装，若有安装则再检查与内核版本是否一致

1） 使用命令 rpm -qa | grep kernel-devel 检查 kernel-devel 是否安装

如出现如下图 5-1 所示内容则说明系统已经安装 kernel-devel。若执行命令后，终端无内容显示则说明系统未安装 kernel-devel。

```
[root@localhost kernels]# rpm  -qa | grep kernel-devel
kernel-devel-2.6.18-308.el5
[root@localhost kernels]#
```

图 5-1

2） 使用命令 uname -r 命令查看内核版本，如下图 5-2 所示：

```
[root@localhost kernels]# uname -r
2.6.18-308.el5
[root@localhost kernels]#
```

图 5-2

如上图所示 kernel-devel-2.6.18-308.el5 为 2.6.18-308.el5，与内核版本一致，符合编译要求。

如发现与内核版本不一致使用如下命令卸载：

yum remove kernel-devel kernel-headers。

3） kernel-devel 源码下载及安装请见**配置依赖环境**，CentOS5/6 发行版禁止使用命令

yum install kernel-devel kernel-headers 自动安装，因为自动安装一般会出现 kernel-devel 版本比内核版本

高。

3、 CentOS6/7 发行版内核拷贝路径

在制作 CentOS5.8 版本 nbd 模块中发现解压内核源码路径与其他版本不同，特说明下（CentOS6/7）的解压拷贝(内核 rpm 包下载过程见配置依赖环境 5)。

rpm -ivh kernel-2.6.32-71.el6.src.rpm

cd ~/rpmbuild/SOURces

tar –jxvf linux-2.6.32-71.el6.tar.bz2 –C /usr/src/kernels/

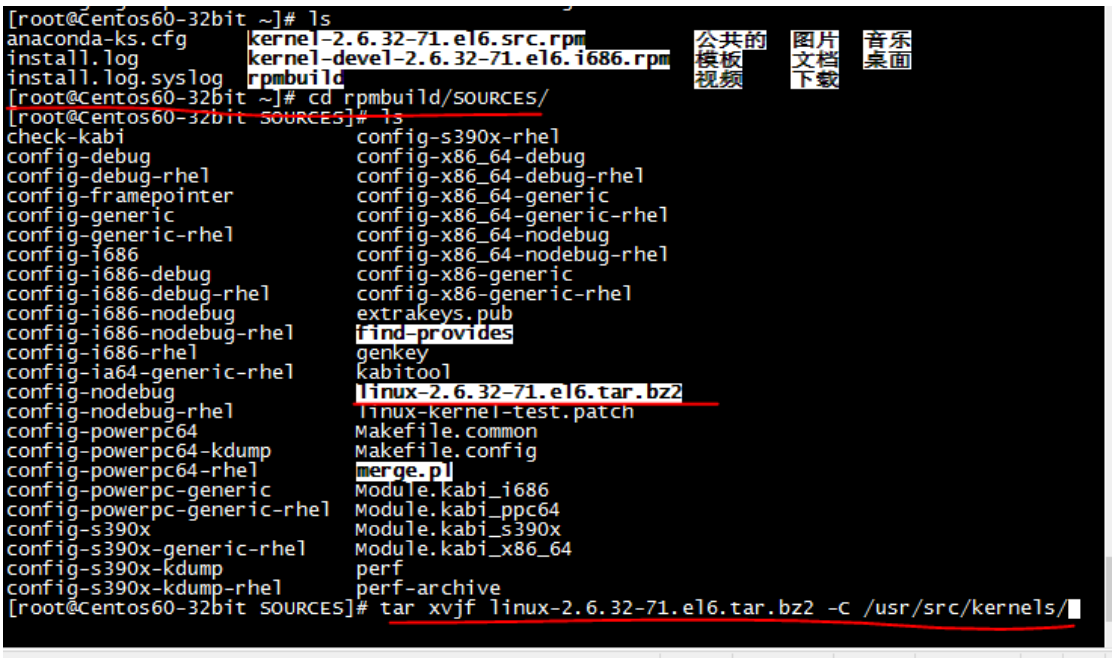


图 5-3

5.2 配置依赖环境

1、安装编译工具

命令：

yum –y install gcc

2、查看 CentOS 内核版本

命令：uname –r 如图 5-4 所示：

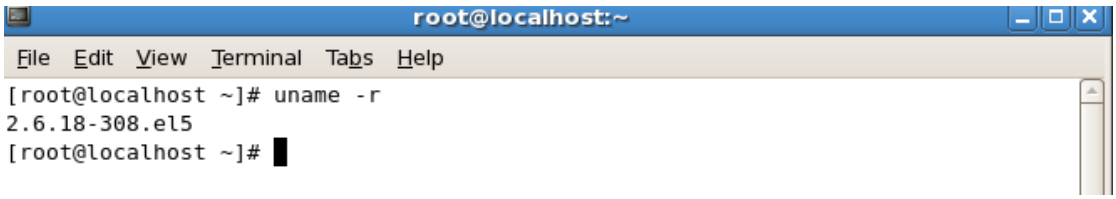


图 5-4

3、下载 kernel-devel 源码并上传到 CentOS 系统

kernel-devel 源码官网内核版本对应的 CentOS 目录下载

官网下载地址：http://vault.centos.org/5.8/os/x86_64/CentOS/，如图 5-5。

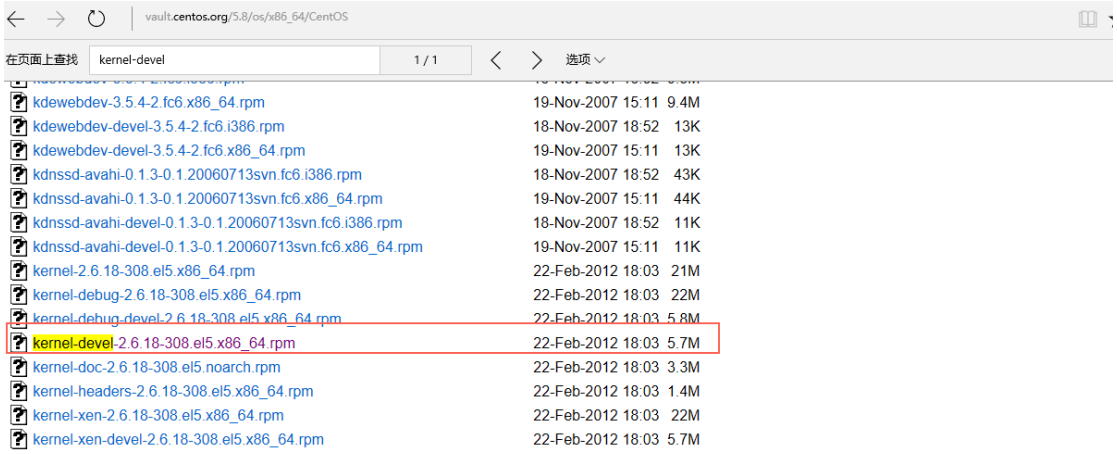


图 5-5

4、解压安装 kernel-devel

命令：rpm -ivh kernel-devel-2.6.18-308.el5.x86_64.rpm

解压安装的文件在：/usr/src/kernels/目录下，如图 5-6 所示：

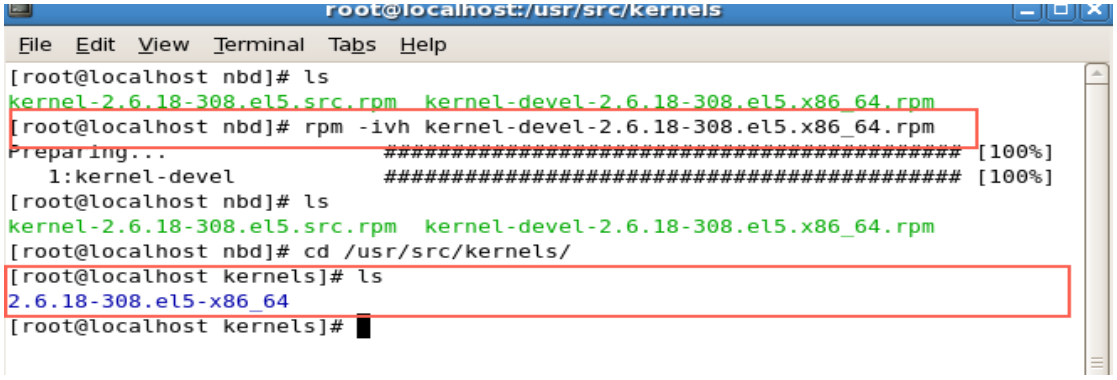


图 5-6

5、根据内核版本下载对应的内核源码

内核源码可到官网 SRPMS 目录下搜索下载 。

官网下载地址：http://vault.centos.org/5.8/os/SRPMS/ 。

2.6.18-308.el5 对应的内核源码是 kernel-2.6.18-308.el5.src.rpm，如图 5-7 所示。

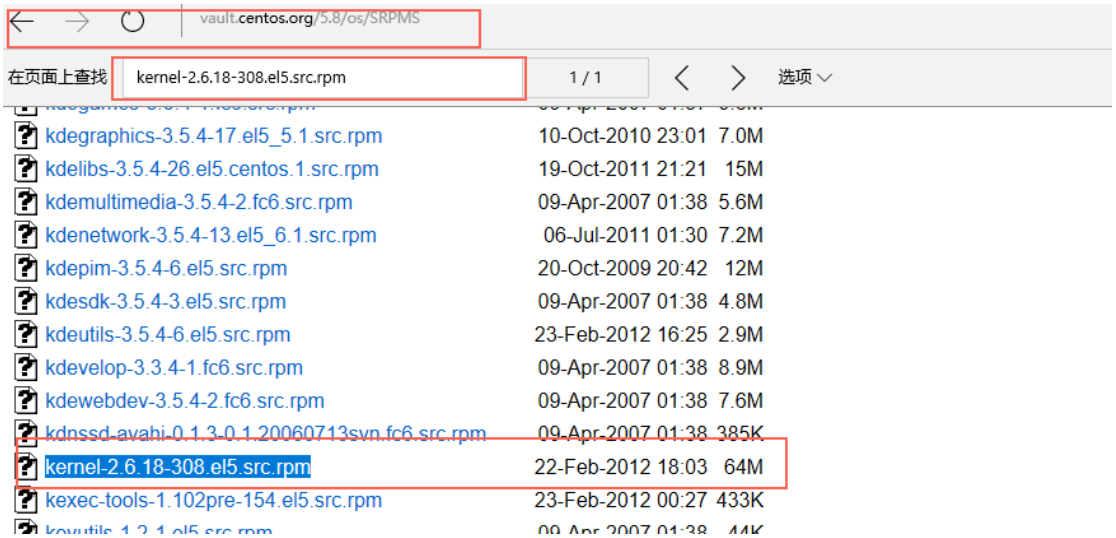


图 5-7

6、解压内核源码

命令：rpm -ivh kernel-2.6.18-308.el5.src.rpm，解压结果如图 5-8 所示。

执行命令时 报 error: cannot create %sourcedir /usr/src/r7edhat/SOURCES

需要手动处理：其他 CentOS 版本解压流程请查看特殊处理说明。

CentOS5.8 解压源码命令流程：

mkdir -p /usr/src/redhat/SOURCES

chmod 777 /usr/src/redhat/SOURCES

cd /usr/src/redhat/ SOURCES

```
[root@localhost SOURCES]# cd cd /usr/src/redhat/SOURCES
bash: cd: cd: No such file or directory
[root@localhost SOURCES]# ls
check-kabi                      kernel-2.6.18-ppc64.config
Config.mk                       kernel-2.6.18-ppc64-debug.config
config-rhel-generic             kernel-2.6.18-ppc64-kdump.config
config-rhel-ppc64-generic       kernel-2.6.18-ppc.config
config-rhel-x86_64-generic      kernel-2.6.18-redhat.patch
COPYING.modules                 kernel-2.6.18-s390.config
find-provides                   kernel-2.6.18-s390x.config
genkey                          kernel-2.6.18-s390x-debug.config
kabitool                        kernel-2.6.18-s390x-kdump.config
kabi_whitelist_i686             kernel-2.6.18-x86_64.config
kabi_whitelist_i686PAE          kernel-2.6.18-x86_64-debug.config
kabi_whitelist_i686xen          kernel-2.6.18-x86_64-xen.config
kabi_whitelist_ia64             linux-2.6.18.4.tar.bz2
kabi_whitelist_ia64xen          linux-kernel-test.patch
kabi_whitelist_ppc64            merge.pl
kabi_whitelist_ppc64kdump       Module.kabi_i686
kabi_whitelist_s390x            Module.kabi_i686PAE
kabi_whitelist_x86_64           Module.kabi_i686xen
kabi_whitelist_x86_64xen        Module.kabi_ia64
kernel-2.6.18-i586.config       Module.kabi_ppc64
kernel-2.6.18-i686.config       Module.kabi_ppc64kdump
kernel-2.6.18-i686-debug.config Module.kabi_s390x
kernel-2.6.18-i686-PAE.config   Module.kabi_x86_64
kernel-2.6.18-i686-xen.config   Module.kabi_x86_64xen
kernel-2.6.18-ia64.config       xen-15502.tar.bz2
kernel-2.6.18-ia64-debug.config xen-2.6.18-redhat.patch
kernel-2.6.18-ia64-xen.config   xen-config-2.6.18-redhat.patch
[root@localhost SOURCES]# tar -xvjf linux-2.6.18.4.tar.bz2 -C /usr/src/kernels/
```

图 5-8

解压后的文件在/usr/src/kernels/ 如下图 5-9 所示：

```
[root@localhost kernels]# ls
2.6.18-308.el5-x86_64  linux-2.6.18.4
[root@localhost kernels]#
```

图 5-9

5.3 编译加载 nbd 模块

1、执行命令：

cd /usr/src/kernels

mv $(uname -r)-x86_64 $(uname -r)-old

mv linux-2.6.32-71.el6 $(uname -r)

cd $(uname -r)

```
[root@localhost kernels]# ls
2.6.18-308.el5-x86_64  linux-2.6.18.4
[root@localhost kernels]# mv $(uname -r)-x86_64 $(uname -r)-old
[root@localhost kernels]# ls
2.6.18-308.el5-old  linux-2.6.18.4
[root@localhost kernels]# mv linux-2.6.18.4 $(uname -r)
[root@localhost kernels]# ls
2.6.18-308.el5  2.6.18-308.el5-old
[root@localhost kernels]# cd $(uname -r)
[root@localhost 2.6.18-308.el5]#
```

2、然后执行如下命令：

make mrproper

cp ../$(uname -r)-old/Module.symvers .

cp /boot/config-$(uname -r) ./.config

执行结果如图 5-10 所示。

```
[root@localhost 2.6.18-308.el5]# make mrproper
[root@localhost 2.6.18-308.el5]# cp ../$(uname -r)-old/Module.symvers .
[root@localhost 2.6.18-308.el5]# cp /boot/config-$(uname -r) ./.config
[root@localhost 2.6.18-308.el5]#
```

图 5-10

3、去除加载模块时内核检查模块签名配置

执行命令：vim .config

将配置项

CONFIG_MODULE_SIG=y

CONFIG_MODULE_SIG_FORCE=y

修改为:

#CONFIG_MODULE_SIG1 is not set

#CONFIG_MODULE_SIG_FORCE is not set

如图 5-11 所示：

```
CONFIG_MODULE_SIG=y
# CONFIG_MODULE_SIG_FORCE is not set
```

图 5-11

修改后如图 5-12 所示：

```
#CONFIG_MODULE_SIG is not set
# CONFIG_MODULE_SIG_FORCE is not set
CONFIG_KMOD=y
```

图 5-12

保存退出。

4、执行命令

执行编译内核命令，如图 5-13 所示。

make oldconfig
make prepare

make scripts

```
[root@localhost 2.6.18-308.el5]# make prepare
scripts/kconfig/conf -s arch/x86_64/Kconfig
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/linux/utsrelease.h
  UPD     include/linux/utsrelease.h
  SYMLINK include/asm -> include/asm-x86_64
  CC      arch/x86_64/kernel/asm-offsets.s
  GEN     include/asm-x86_64/asm-offsets.h
[root@localhost 2.6.18-308.el5]# make scripts
  HOSTCC  scripts/genksyms/genksyms.o
  SHIPPED scripts/genksyms/lex.c
  SHIPPED scripts/genksyms/parse.h
  SHIPPED scripts/genksyms/keywords.c
```

图 5-13

编译 nbd 模块，如图 5-14、5-15 所示。make CONFIG_BLK_DEV_NBD=m M=drivers/block

```
  HOSTCC  scripts/kallsyms
  HOSTCC  scripts/pnmtologo
  HOSTCC  scripts/conmakehash
[root@localhost 2.6.18-308.el5]# make CONFIG_BLK_DEV_NBD=m M=drivers/block
  CC      drivers/block/rd.o
  LD      drivers/block/built-in.o
  CC [M]  drivers/block/floppy.o
  CC [M]  drivers/block/loop.o
  CC [M]  drivers/block/cpqarray.o
  CC [M]  drivers/block/cciss.o
  CC [M]  drivers/block/DAC960.o
  CC [M]  drivers/block/pktcdvd.o
  CC [M]  drivers/block/nbd.o
  CC [M]  drivers/block/cryptoloop.o
  CC [M]  drivers/block/sx8.o
  Building modules, stage 2.
  MODPOST
  CC      drivers/block/DAC960.mod.o
```

图 5-14

```
  LD [M]  drivers/block/cryptoloop.ko
  CC      drivers/block/floppy.mod.o
  LD [M]  drivers/block/floppy.ko
  CC      drivers/block/loop.mod.o
  LD [M]  drivers/block/loop.ko
  CC      drivers/block/nbd.mod.o
  LD [M]  drivers/block/nbd.ko
  CC      drivers/block/pktcdvd.mod.o
  LD [M]  drivers/block/pktcdvd.ko
  CC      drivers/block/sx8.mod.o
  LD [M]  drivers/block/sx8.ko
```

图 5-15

把 nbd 模块的 ko 文件放到指定文件，如图 5-16 所示。

cp drivers/b3lock/nbd.ko /lib4/mod5ules/$(uname -r)/kernel/drivers/block/

```
  CC      drivers/block/sx8.mod.o
  LD [M]  drivers/block/sx8.ko
[root@localhost 2.6.18-308.el5]# cp drivers/block/nbd.ko /lib/modules/$(uname -r)/kernel/drivers/blo
ck/
cp: overwrite `/lib/modules/2.6.18-308.el5/kernel/drivers/block/nbd.ko'? yes
[root@localhost 2.6.18-308.el5]# ls
arch     CREDITS        drivers  init     kernel       Makefile        net             scripts   usr
```

图 5-16

5、确定依赖关系

命令：depmod –a

6、加载模块

命令：

Cd6 drivers/block

insmod nbd.ko max_part=16

```
[root@localhost 2.6.18-308.el5]# cd drivers/block/
[root@localhost block]# ls
acsi.c        cciss.mod.c     cpqarray.o        DAC960.mod.o  loop.c           nbd.mod.o       rd.o         umem.c
acsi_slm.c    cciss.mod.o     cryptoloop.c      DAC960.o      loop.ko          nbd.o           smart1,2.h   viodasd.c
amiflop.c     cciss.o         cryptoloop.ko     floppy.c      loop.mod.c       paride          swim3.c      xd.c
aoe           cciss_scsi.c    cryptoloop.mod.c  floppy.ko     loop.mod.o       pktcdvd.c       swim_iop.c   xd.h
ataflop.c     cciss_scsi.h    cryptoloop.mod.o  floppy.mod.c  loop.o           pktcdvd.ko      sx8.c        z2ram.c
built-in.o    cpqarray.c      cryptoloop.o      floppy.mod.o  Makefile         pktcdvd.mod.c   sx8.ko
cciss.c       cpqarray.h      DAC960.c          floppy.o      Module.symvers   pktcdvd.mod.o   sx8.mod.c
cciss_cmd.h   cpqarray.ko     DAC960.h          ida_cmd.h     nbd.c            pktcdvd.o       sx8.mod.o
cciss.h       cpqarray.mod.c  DAC960.ko         ida_ioctl.h   nbd.ko           ps2esdi.c       sx8.o
cciss.ko      cpqarray.mod.o  DAC960.mod.c      Kconfig       nbd.mod.c        rd.c            ub.c
[root@localhost block]# insmod nbd.ko max_part=16
```

图 5-17

7、核查 nbd 模块是否添加成功

lsmod | grep nbd

如图所示表示已经将 nbd 模块加载到内核。

```
[root@localhost block]# lsmod | grep nbd
nbd                    61088  0
[root@localhost block]#
```

nbd 编译加载命令集：

rpm -ivh kernel-2.6.32-71.el6.src.rpm
cd ~/rpmbuild/SOURCES
tar jxf linux-2.6.32-71.el6.tar.bz2 -C /usr/src/kernels/
cd /usr/src/kernels
mv $(uname -r) $(uname -r)-old
mv linux-2.6.32-71.el6 $(uname -r)
cd $(uname -r)
make mrproper
cp ../$(uname -r)-old/Module.symvers .
cp /boot/config-$(uname -r) ./.config

vim .config
make oldconfig
make prepare
make scripts
make CONFIG_BLK_DEV_NBD=m M=drivers/block

```
cp drivers/block/nbd.ko /lib/modules/$(uname -r)/kernel/drivers/block/

cd drivers/block
depmod -a
insmod nbd.ko max_part=16
```