部署及注意事项

# Kiloview NDI 矩阵系统部署指导 V1

(2022-02 2.0 版)

# 内容：



**注意**

部署 NDI 矩阵，需在本地加载 NDI 矩阵镜像，获取镜像请联系千视销售人员，或发邮件到 info@kiloview.com

# 1 Kiloview NDI 矩阵系统部署

## 1.1 服务器环境准备

### 1.1.1 硬件环境

处理器：高主频 CPU，比如 E2288G、12 代酷睿 i5 12600K

硬盘：64G 硬盘或更高

内存：4GB RAM 或更高 16G 以上高频内存

网卡：一块或者多块万兆或更高速网卡

### 1.1.2 软件环境

操作系统：Linux64 位操作系统（Ubuntu 18.04、20.04）

### 1.1.3 网络环境

Internet 应用工具和镜像文件

局域网带宽：万兆网络

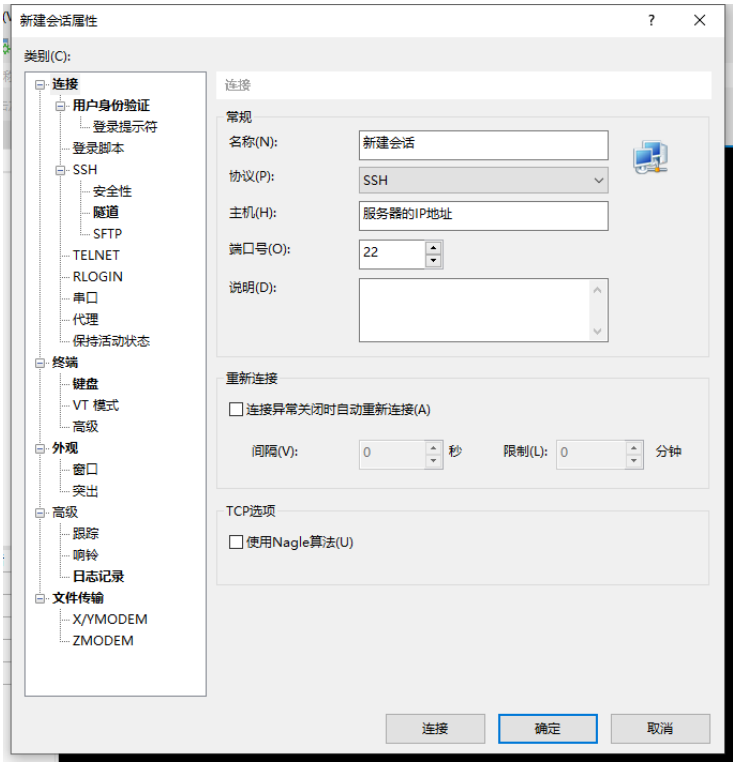
## 1.2 登录服务器

采用远程终端软件登录服务器，建议采用 Xshell 或者 PuTTY 工具.

Xshell 下载链接：https://www.netsarang.com/zh/xshell-download/

PuTTy 下载地址：https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

1.2.1 下载并安装后，在新建会话属性中输入服务器 IP 地址，采用 SSH 协议通信，端口号默认为 22，输入完成后，点击[确定]按钮;




1.2.2 在弹出的对话框中输入用户名和密码，普通用户需要 sudo 获取管理权限，或以 root 身份登录，下面的部署过程是以 root 用户完成。

在命令窗口中输入以下命令，矩阵到 root 用户：

sudo su -

## 1.3 部署指导

### 1.3.1 安装容器。在终端窗口中输入：

curl -fsSL https://get.docker.com | bash

```
root@kiloview:/# curl -fsSL https://get.docker.com | sh

# Executing docker install script, commit: 93d2499759296ac1f9c510605fef85052a2c32be

+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /usr/share/keyrings/docker-archive-
keyring.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu
focal stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends  docker-ce-cli docker-scan-plugin docker-
ce >/dev/null
+ version_gte 20.10
+ [ -z  ]
+ return 0
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce-rootless-extras >/dev/null
+ sh -c docker version
  Client: Docker Engine - Community
   Version:           20.10.11
   API version:       1.41
   Go version:        go1.16.9
   Git commit:        dea9396
   Built:             Thu Nov 18 00:37:06 2021
   OS/Arch:           linux/amd64
   Context:           default
   Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.11
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.9
  Git commit:       847da18
  Built:            Thu Nov 18 00:35:15 2021
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.12
  GitCommit:        7b11cfaabd73bb80907dd23182b9347b4245eb5d
 runc:
  Version:          1.0.2
  GitCommit:        v1.0.2-0-g52b36a2
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

1.3.2 安装 NDI 发现工具。在终端窗口中输入：

apt install avahi-daemon

```
root@VM-0-9-ubuntu:~# apt install avahi-daemon
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  dblatex dblatex-doc dh-strip-nondeterminism docbook-dsssl docbook-utils docbook-xml docbook-xsl dvisvgm dwz eatmydata fonts-droid-fallback fonts-gfs-baskerville fonts-gfs-porson fonts-lmodern fonts-noto-mono fonts-texgyre fonts-urw-base35 ghostscript
  libalgorithm-c3-perl libapache-pom-java libarchive-cpio-perl libarchive-zip-perl libauthen-sasl-perl libb-hooks-endofscope-perl libb-hooks-op-check-perl libc-dev-bin libcairo2 libclass-c3-perl libclass-c3-xs-perl libclass-data-inheritable-perl
  libclass-method-modifiers-perl libclass-xsaccessor-perl libcommons-logging-java libcommons-parent-java libcrypt-dev libcups2 libdata-dump-perl libdata-optlist-perl libdatrie1 libdebhelper-perl libdevel-callchecker-perl libdevel-caller-perl
  libdevel-globaldestruction-perl libdevel-lexalias-perl libdevel-stacktrace-perl libdist-checkconflicts-perl libdynaloader-functions-perl libeatmydata1 libemail-date-format-perl libencode-locale-perl libeval-closure-perl libexception-class-perl libfile-basedir-perl
  libfile-desktopentry-perl libfile-homedir-perl libfile-listing-perl libfile-mimeinfo-perl libfile-stripnondeterminism-perl libfile-which-perl libfont-afm-perl libfontbox-java libgraphite2-3 libgs9 libgs9-common libharfbuzz-icu0 libharfbuzz0b libhtml-form-perl
  libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libijs-0.35 libio-html-perl libio-socket-ssl-perl libio-stringy-perl
  libipc-shareable-perl libipc-system-simple-perl libjbig2dec0 libkpathsea6 liblcms2-2 libllvm10 liblog-dispatch-perl liblog-log4perl-perl libltdl-dev liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl libmime-charset-perl libmime-lite-perl
  libmime-types-perl libmodule-implementation-perl libmodule-runtime-perl libmro-compat-perl libnamespace-autoclean-perl libnamespace-clean-perl libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libosp5 libostyle1c2 libpackage-stash-perl
  libpackage-stash-xs-perl libpadwalker-perl libpaper-utils libpaper1 libparams-classify-perl libparams-util-perl libparams-validationcompiler-perl libpdfbox-java libpixman-1-0 libptexenc1 libreadonly-perl libref-util-perl libref-util-xs-perl librole-tiny-perl
  libsgmls-perl libsombok3 libspecio-perl libsub-exporter-perl libsub-exporter-progressive-perl libsub-identify-perl libsub-install-perl libsub-name-perl libsub-override-perl libsub-quote-perl libsynctex2 libtcl8.6 libteckit0 libtexlua53 libtexluajit2 libthai-data
  libthai0 libtie-ixhash-perl libtimedate-perl libtk8.6 libtry-tiny-perl libunicode-linebreak-perl liburi-perl libvariable-magic-perl libwoff1 libwww-perl libwww-robotrules-perl libx11-protocol-perl libxcb-render0 libxcursor1 libxdamage1 libxml-parser-perl
  libxml-twig-perl libxml-xpathengine-perl libxss1 libxstring-perl libyaml-tiny-perl libzzip-0-13 lmodern openjade opensp perl-openssl-defaults poppler-data preview-latex-style python3-importlib-metadata python3-jinja2 python3-json-pointer python3-jsonpatch
  python3-jsonschema python3-markupsafe python3-more-itertools python3-pyrsistent python3-zipp sgml-base sgml-data sgmlspl squashfs-tools t1utils tcl tcl8.6 teckit tex-common tex-gyre texlive texlive-base texlive-bibtex-extra texlive-binaries texlive-extra-utils
  texlive-fonts-recommended texlive-formats-extra texlive-lang-greek texlive-latex-base texlive-latex-extra texlive-latex-recommended texlive-pictures texlive-plain-generic texlive-science texlive-xetex tipa tk tk8.6 x11-xserver-utils xdg-utils xfonts-encodings
  xfonts-utils xml-core xmlto xsltproc
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libavahi-core7 libdaemon0 libnss-mdns
Suggested packages:
  avahi-autoipd avahi-autoipd | zeroconf
The following NEW packages will be installed:
  avahi-daemon libavahi-core7 libdaemon0 libnss-mdns
0 upgraded, 4 newly installed, 0 to remove and 43 not upgraded.
Need to get 180 kB of archives.
After this operation, 765 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

在“Do you want to continue?[Y/n]”后面输入：Y。等待安装完成。

1.3.3 安装 netdata，以获取 CPU、网络等数据。在终端窗口中输入：

docker run -d --name status --restart=always -v /var/run/docker.sock:/var/run/docker.sock:ro --pid host --network host -e GLANCES_OPT="-w" nicolargo/glances

```
root@VM-0-9-ubuntu:~# docker run -d --name status --restart=always -v /var/run/docker.sock:/var/run/docker.sock:ro --pid host --network host -e GLANCES_OPT="-w" nicolargo/glances
Unable to find image 'nicolargo/glances:latest' locally
latest: Pulling from nicolargo/glances
a10c77af2613: Pull complete
eab08a61c249: Pull complete
93533675153b: Pull complete
bfa1edacf570: Pull complete
88cf10da884f: Pull complete
9e671e821651: Pull complete
b62c8aa4ba07: Pull complete
df54e77b2043: Pull complete
Digest: sha256:76a921619e799f8eea2544e1555d80da214fdae9c31c8c29d75882b4b233a81a
Status: Downloaded newer image for nicolargo/glances:latest
7768b11b7d5a885bbf7ece8aa563f5f5310f801da462c6db3c9ea2bd1a0838f8
root@VM-0-9-ubuntu:~#
```

1.3.4 导入 NDI 矩阵系统镜像至服务器

（1）从本地电脑上传 NDI 矩阵镜像至服务器

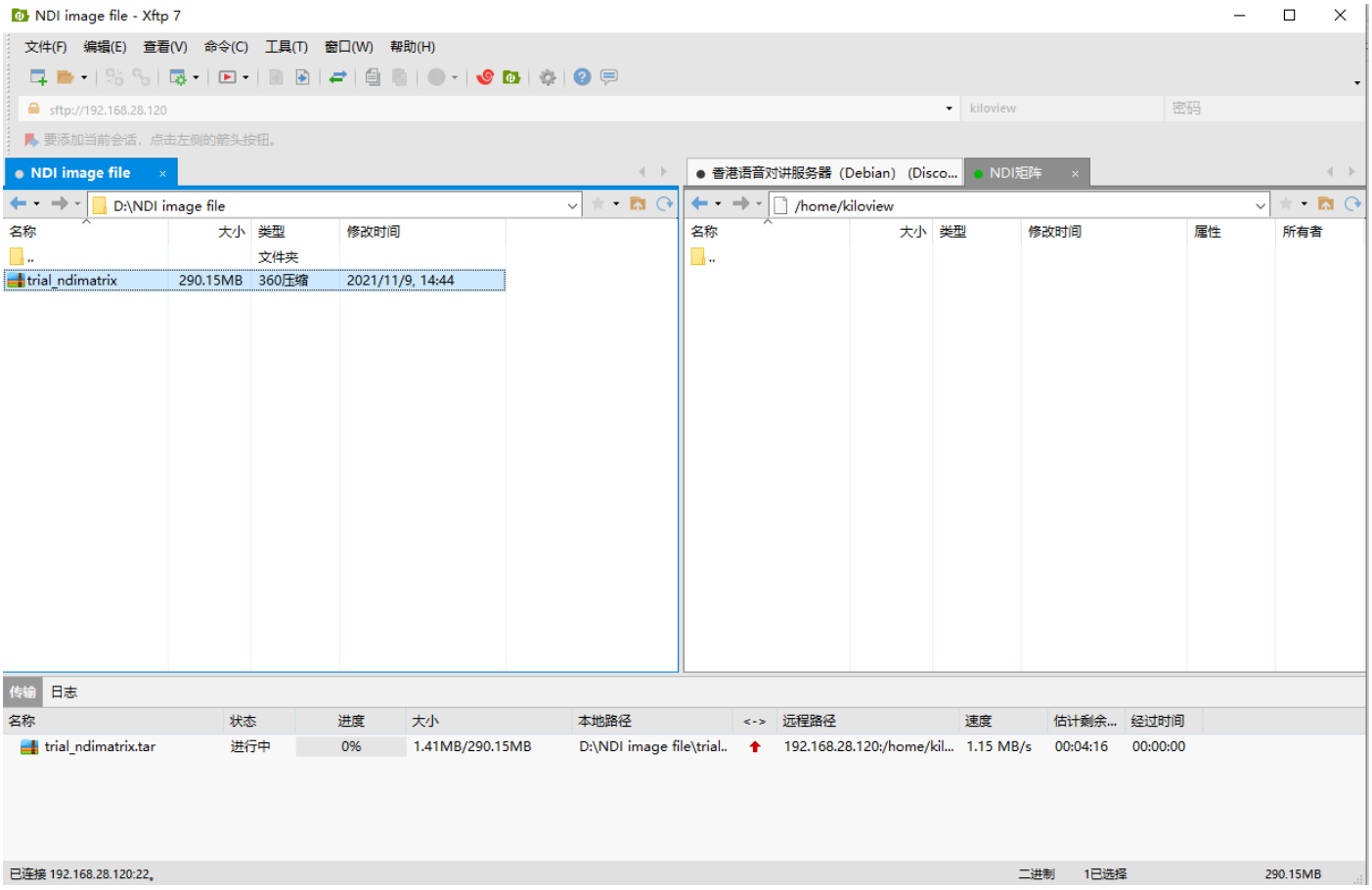
**说明**

NDI 矩阵系统镜像需要从本地 PC 上传至 liunx 系统，因此需要通过文件传输工具传输文件，部署人员可使用 Xhell

工具自带的 xftp 文件传输工具，或其它文件传输工具，如：SecureCRT。

<1>点击 Xhell 工具中的文件传输图标


<2>将本地电脑中的 NDI 矩阵系统镜像文件从左边窗口拖拽到右边窗口，完成文件从本地电脑传输到服务器。(NDI 矩阵镜像文件请联系千视销售人员，或发邮件到 info@kiloview.com 获取)



(2)在服务器中加载 NDI 矩阵系统镜像

docker load -i kv_ndicore_pro_011801.tar

```
root@VM-0-9-ubuntu:/home/ubuntu# docker load -i kv_ndicore_pro_011801.tar
cc967c529ced: Loading layer [==================================================>]  65.57MB/65.57MB
2c6ac8e5063e: Loading layer [==================================================>]  991.2kB/991.2kB
6c01b5a53aac: Loading layer [==================================================>]  15.87kB/15.87kB
e0b3afb09dc3: Loading layer [==================================================>]  3.072kB/3.072kB
9d3cadc3d17e: Loading layer [==================================================>]  27.69MB/27.69MB
db6e2a638ab1: Loading layer [==================================================>]  114.3MB/114.3MB
9f2fd036a4a9: Loading layer [==================================================>]  3.584kB/3.584kB
20b2e7d325d4: Loading layer [==================================================>]   2.56kB/2.56kB
758a8236e8b9: Loading layer [==================================================>]  2.048kB/2.048kB
8abff239dac8: Loading layer [==================================================>]  1.536kB/1.536kB
ac9c6590408b: Loading layer [==================================================>]  4.643MB/4.643MB
bda72356d77c: Loading layer [==================================================>]  9.728kB/9.728kB
75810feecf4b: Loading layer [==================================================>]  39.62MB/39.62MB
7e47cfdd3260: Loading layer [==================================================>]    556kB/556kB
bc90e555b3c8: Loading layer [==================================================>]  3.072kB/3.072kB
87bf4592c061: Loading layer [==================================================>]  36.35kB/36.35kB
8017a395ddee: Loading layer [==================================================>]  16.38kB/16.38kB
f275a7a61f6f: Loading layer [==================================================>]  10.46MB/10.46MB
e98830ac8618: Loading layer [==================================================>]  14.39MB/14.39MB
a10c5d895642: Loading layer [==================================================>]  927.7kB/927.7kB
87e801439ab8: Loading layer [==================================================>]  9.403MB/9.403MB
1488b496f22c: Loading layer [==================================================>]  879.6kB/879.6kB
4ee4f7e0dd9a: Loading layer [==================================================>]  894.5kB/894.5kB
b0439f642b95: Loading layer [==================================================>]  1.386MB/1.386MB
f0af0a693169: Loading layer [==================================================>]  891.9kB/891.9kB
5168f669e526: Loading layer [==================================================>]  1.038MB/1.038MB
8687bcb79567: Loading layer [==================================================>]  1.086MB/1.086MB
2360bb09cc5c: Loading layer [==================================================>]  2.048kB/2.048kB
addf2d3eb25e: Loading layer [==================================================>]  10.31MB/10.31MB
19bd1a2d4a1c: Loading layer [==================================================>]  52.75MB/52.75MB
c2e76cee58fa: Loading layer [==================================================>]  3.584kB/3.584kB
df304ad5eca3: Loading layer [==================================================>]  636.9kB/636.9kB
3396c01c88c1: Loading layer [==================================================>]  242.7kB/242.7kB
Loaded image: kiloview/kv_ndicore_senior_011801:latest
```

**注意**

如果 NDI 镜像文件所在的位置，不在当前目录下，需要指定 NDI 镜像文件所在的文件夹。

如：NDI 镜像文件所在目录为/home/kiloview，则 NDI 镜像文件加载命令为：

docker load -i /home/kiloview/-i kv_ndicore_pro_011801.tar

1.3.5 运行容器

docker run -d -v /root/cp_data3:/data/configs -v /etc/timezone:/etc/timezone -v /etc/localtime:/etc/localtime -v /var/run/avahi-daemon:/var/run/avahi-daemon -v /var/run/dbus:/var/run/dbus --restart=always --name kv_ndicore_senior_011801 --network host --privileged=true kiloview/kv_ndicore_senior_011801:latest


**注意**

上面命令中的最后的镜像名称（如图中的 kv_ndicore_senior_011801）需跟命令行上方中 loaded image 后面的名称一致

## 1.4 登录验证

在浏览器（推荐使用 google）地址栏中输入“服务器 IP 地址：81”，回车后显示 NDI 矩阵系统的登录界面，默认用户名和密码均为：admin.

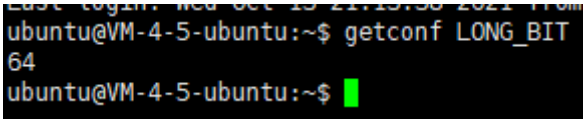
# 2 部署常见问题及解决方法

## 2.1 如果在部署过程中遇到错误提示

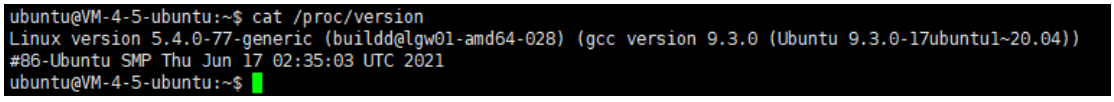
解决方法：

请先检查操作系统版本，目前仅支持 Linux64 位操作系统 Linux64-bit operating system（Ubuntu 18.04+ / Debian 9+）

（1）查询 linux 位数： getconf LONG_BIT




（2）查询 linux 版本号：cat /proc/version




## 2.2 安装 docker 长时间未响应。

解决方法：

docker 安装过程较为缓慢，需要耐心等待。可以使用命令“docker version”查看以及确认是否安装成功 docker。

```
root@ndi:~/cp_data3# docker version
Client: Docker Engine - Community
 Version:           20.10.6
 API version:       1.41
 Go version:        go1.13.15
 Git commit:        370c289
 Built:             Fri Apr  9 22:47:17 2021
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.6
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.13.15
  Git commit:       8728dd2
  Built:            Fri Apr  9 22:45:28 2021
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.4
  GitCommit:        05f951a3781f4f2c1911b05e61c160e9c30eaa8e
 runc:
  Version:          1.0.0-rc93
  GitCommit:        12644e614e25b05da6fd08a38ffa0cfe1903fdec
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
root@ndi:~/cp_data3#
```

## 2.3 拉取镜像失败

```
root@ndi:/# docker run -d --name status --restart=always -v /var/run/docker.sock:/var/run/docker.sock:ro --pid host --network host -e GLANCES_OPT="-w" nicolargo/glances
Unable to find image 'nicolargo/glances:latest' locally
latest: Pulling from nicolargo/glances
e1acddbe380c: Pulling fs layer
ecc7ff4d2622: Pulling fs layer
dae1d9fd74c1: Pulling fs layer
07bc5aa6fc42: Waiting
76f124aca9af: Waiting
9c40be6c51a4: Waiting
c125888d0329: Waiting
342605490b7f: Waiting
docker: error pulling image configuration: Get https://production.cloudflare.docker.com/registry-v2/docker/registry/v2/blobs/sha256/b3/b39a65d9d3bba1f740dd5c3fde71c65ab5f7113448ee923d459547969d65e222/data?verify=1636450334-W9xG%2BdO5e3GWgJ4
08FJ%2FByyQne4%3D: dial tcp 104.18.124.25:443: i/o timeout.
See 'docker run --help'.
root@ndi:/# docker run -d --name status --restart=always -v /var/run/docker.sock:/var/run/docker.sock:ro --pid host --network host -e GLANCES_OPT="-w" nicolargo/glances
Unable to find image 'nicolargo/glances:latest' locally
latest: Pulling from nicolargo/glances
docker: error parsing HTTP 408 response body: invalid character '<' looking for beginning of value: "<html><body><h1>408 Request Time-out</h1>\nYour browser didn't send a complete request in time.\n</body></html>\n\n".
See 'docker run --help'.
root@ndi:/#
```

解决方法：

镜像拉取需要通过 internet 获取镜像文件，如果网络延迟较高或者无法连接 internet,请检查网络是否通畅，通过 ping 一个外部网站（如：百度）检查.

## 2.4 NDI 矩阵系统无法正常登录

解决方法：

检查服务器是否正常启动。“win+R”打开命令提示符窗口 -> “cmd” -> “ping 服务器 IP”，

（1）如果能够 ping 通服务器 IP，请通过以下命令检查：

<1>检查 docker 运行状态：systemctl status docker

```
root@ndi:~/cp_data3# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2021-11-09 09:00:05 UTC; 1 day 1h ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 53134 (dockerd)
      Tasks: 25
     Memory: 640.7M
     CGroup: /system.slice/docker.service
             └─53134 /usr/bin/dockerd -H fd:// -H 0.0.0.0:2375 --containerd=/run/containerd/containerd.sock
```

如果 docker 没有正常启动，启动 docker：systemctl start docker

<2>检查容器运行状况：docker ps -a

```
root@ndi:~/cp_data3# docker ps -a
CONTAINER ID   IMAGE                              COMMAND                  CREATED        STATUS        PORTS     NAMES
563282ea0ee8   kiloview/trial_ndimatrix:latest    "/start_server.sh"       24 hours ago   Up 24 hours             trial_ndimatric
2594ce003d0d   nicolargo/glances                  "/bin/sh -c 'python3…"   24 hours ago   Up 24 hours             status
root@ndi:~/cp_data3#
```

如果容器没有正常运行，执行以下命令删除容器和镜像，再根据部署指导重新部署。

停止所有容器：docker stop $(docker ps -aq)

删除所有容器：docker rm $(docker ps -aq)

删除所有镜像：docker rmi $(docker images -q)

（2）如果无法 ping 通服务器 IP，请按照以下步骤检查：

<1>检查维护 PC 和服务器是否正常运行，网口灯是否正常闪烁,；

<2>服务器 IP 可能已经变化，通过直连服务器，“win+R” -> “cmd” -> “ifconfig”，采用变更后的 IP 地址：81 访问

## 2.5 怎样给 NDI 矩阵系统设置静态 IP 地址

解决方法：

不同版本的 linux 操作系统设置静态 IP 地址的方式可能不同，以下配置方式适用于 ubuntu 20.04 版本。

sudo vi /etc/netplan/00-installer-config.yaml

**注意**

(1) 不同小版本的网络配置文件名可能不同，即“00-installer-config.yaml“文件名不同，先通过 cd /etc/netplan 进入 netplan 文件夹，查找网络配置文件，再通过 vi 命令进入对应的网络配置文件；

(2) 按下“i”键进入文件编辑模式；

(3) 输入完成后，按下“ESC”键，再输入”:wq “，保存文件并退出；

(4) 如果输入错误，不想保存文件，按下“ESC”键，再输入”:q!”，不保存文件并退出。

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enol:
      addresses:
      - 192.168.28.120/24
      gateway4: 192.168.28.254
      nameservers:
        addresses:
        - 8.8.8.8
    enp3s0f0:
      addresses:
      - 192.168.0.114/24
      gateway4: 192.168.0.1
    enp3s0f1:
      addresses:
      - 192.168.2.115/24
      gateway4: 192.168.2.1
  version: 2
~
```

## 2.6 命令执行过程中报“no such file or directory”错误


解决方法：

命令从文本文件复制到命令窗口时，带入了格式符，导致命令发生了变化。遇到这样的情况，请手动输入对应命令。



长沙千视电子科技有限公司

https://www.kiloview.com/

湖南省长沙市雨花区汇金路 877 号嘉华智谷产业园（长沙屿）B4-106 栋/109 栋

邮箱：support@kiloview.com　联系电话：18573192787