Preparations

Hardware

Processor： Intel Core i3 CPU or higher

Hard disk: 64G hard disk or higher

RAM: 4GB RAM or higher

Software

Operating system: Linux64-bit operating system (Ubuntu 18.04+ / Debian 9+)

Network

IP address: one public IP address

Bandwidth: related to the video encoding rate, e.g.: encoding rate 4Mbps, bandwidth will be 8Mbps

Server Login

Login to the server by remote terminal software, Xshell is recommended.

https://www.netsarang.com/zh/xshell-download/

Download link of Xshell: https://www.netsarang.com/zh/xshell-download/

(1) After downloading and installing, enter the server IP address in the new session, and choose"SSH" protocol. The port number is 22 by default. Click "OK" when finished.

Properties of America

Category:

- **Connection**
  - **Authentication**
    - Login Prompts
  - Login Scripts
  - SSH
    - Security
    - **Tunneling**
    - SFTP
  - TELNET
  - RLOGIN
  - SERIAL
  - Proxy
  - Keep Alive
- **Terminal**
  - **Keyboard**
  - VT Modes
  - Advanced
- **Appearance**
  - Window
  - Highlight
- Advanced
  - Trace
  - Bell
  - **Logging**
- **File Transfer**
  - X/YMODEM
  - ZMODEM

Connection

General

Name: kilolink platform

Protocol: SSH

Host: 43.130.44.152

Port Number: 22

Description:

Reconnect

☑ Reconnect automatically if connection is terminated unexpectedly

Interval: 0 sec   Limit: 0 min

TCP Options

☐ Use Nagle's algorithm

Connect   OK   Cancel

(2) Enter the username and password in the pop-up dialog box, and login as the root user.
 Enter the command in the Xshell terminal: sudo su

```
Xshell 7 (Build 0077)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 43.130.44.152:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-77-generic x86_64)

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

  System information as of Wed 18 Aug 2021 07:49:00 PM CST

  System load:  0.0               Processes:          122
  Usage of /:   4.8% of 78.69GB   Users logged in:    1
  Memory usage: 6%                IPv4 address for eth0: 10.0.4.5
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

Last login: Wed Aug 18 19:48:50 2021 from 175.0.228.228
ubuntu@VM-4-5-ubuntu:~$ sudo su
root@VM-4-5-ubuntu:/home/ubuntu#
```

Coturn Server Installation

(1) Install docker to create a working directory, and enter the command in the Xshell terminal: mkdir /data.

(2) Create and open the turnserver.conf file in the data directory, enter the command in the Xshell terminal: vi /data/ turnserver.conf.

(3) Enter the following content in the turnserver.conf file, the fields in the form should be set according to the actual situation, and other fields can be just by default.

```
listening-port=3478

listening-ip=" 43.130.44.152"

alt-listening-port=0

relay-ip=" 43.130.44.152"

external-ip=" 10.0.4.5"

realm=demo

no-tls

no-dtls

mobility

no-cli

verbose
```

```
fingerprint

lt-cred-mech

stale-nonce=3600

min-port=5000

max-port=9500

user="admin:kiloview"
```

Field	Description
listening-port	Listening port is 3478 by default, no need to modify it generally.
listening-ip	Server Public IP address
relay-ip	Server Public IP address
external-ip	Enter ifconfig in the terminal to check the network card of eth0

```
root@VM-4-5-ubuntu:/data# vi turnserver.conf
root@VM-4-5-ubuntu:/data# cat turnserver.conf
listening-port=3478
listening-ip=" 43.130.44.152"
alt-listening-port=0
relay-ip=" 43.130.44.152"
external-ip=" 10.0.4.5"
realm=demo
no-tls
no-dtls
mobility
no-cli
verbose
fingerprint
lt-cred-mech
stale-nonce=3600
min-port=5000
max-port=9500
user="admin:kiloview"
```

(4) Install and run the docker

<1> Install the docker, enter the command in the Xshell terminal command line:

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

```
root@VM-4-5-ubuntu:/data# curl -fsSL https://get.docker.com | bash
# Executing docker install script, commit: 0e685c6ac0bddd7b2ba7bcaaeb519746ad249a29
+ 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 -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'
```

Wait for Docker Engine's Client and Server to complete the installation.

```
Client: Docker Engine - Community
 Version:              20.10.8
 API version:          1.41
 Go version:           go1.16.6
 Git commit:           3967b7d
 Built:                Fri Jul 30 19:54:27 2021
 OS/Arch:              linux/amd64
 Context:              default
 Experimental:         true

Server: Docker Engine - Community
 Engine:
  Version:             20.10.8
  API version:         1.41 (minimum version 1.12)
  Go version:          go1.16.6
  Git commit:          75249d8
  Built:               Fri Jul 30 19:52:33 2021
  OS/Arch:             linux/amd64
  Experimental:        false
 containerd:
  Version:             1.4.9
  GitCommit:           e25210fe30a0a703442421b0f60afac609f950a3
 runc:
  Version:             1.0.1
  GitCommit:           v1.0.1-0-g4144b63
 docker-init:
  Version:             0.19.0
  GitCommit:           de40ad0
```

<2> Run the docker, enter the command in the Xshell terminal command line:

```
docker run -d --name coturn --restart=always --network=host \

 -v /data/turnserver.conf:/etc/coturn/turnserver.conf \

 instrumentisto/coturn
```

```
root@VM-4-5-ubuntu:/data# docker run -d --name coturn --restart=always --network=host \
> -v /data/turnserver.conf:/etc/coturn/turnserver.conf \
> > instrumentisto/coturn
Unable to find image 'instrumentisto/coturn:latest' locally
latest: Pulling from instrumentisto/coturn
801bfaa63ef2: Pull complete
6632f8d51ce2: Pull complete
0790e8d398d5: Pull complete
e1f1f91d740d: Pull complete
Digest: sha256:6c25f154177c9e9fccc3ba6f28455ad3b0a869f00be0562853a327b5f7b6069a
Status: Downloaded newer image for instrumentisto/coturn:latest
07319b3d296f8f6dfe01df2af331e4b287ee8e74c5f5a9d48b1fde936a9dbdf7
```

(5) Get the version image

<1> Get the management container version image, enter the command in the Xshell terminal command line: docker pull kiloview/polymaster

```
root@VM-4-5-ubuntu:/data# docker pull kiloview/polymaster
Using default tag: latest
latest: Pulling from kiloview/polymaster
a1125296b23d: Pull complete
3c742a4a0f38: Pull complete
4c5ea3b32996: Pull complete
1b4be91ead68: Pull complete
94aedc679f71: Pull complete
6dc1fbbe22c3: Pull complete
f7549a5c15da: Pull complete
6ab7654f2ad6: Pull complete
0a4863586013: Pull complete
bbe78b70d686: Pull complete
11677d749cec: Pull complete
591973d08f34: Pull complete
Digest: sha256:25be0f1c2c537aad419c02bc84ebd2d8b48b22bdab1a71d79547507ca807c6fa
Status: Downloaded newer image for kiloview/polymaster:latest
docker.io/kiloview/polymaster:latest
root@VM-4-5-ubuntu:/data#
```

<2> Get the worker container version image, enter the command in the Xshell terminal command line: docker pull kiloview/polyworker

```
root@VM-4-5-ubuntu:/data# docker pull kiloview/polyworker
Using default tag: latest
latest: Pulling from kiloview/polyworker
a1125296b23d: Already exists
3c742a4a0f38: Already exists
4c5ea3b32996: Already exists
1b4be91ead68: Already exists
70d24a33f756: Pull complete
bef07455c3e2: Pull complete
a920fc4b7308: Pull complete
0018af2afe89: Pull complete
17f549731699: Pull complete
752c839f9b80: Pull complete
3fa398f7b559: Pull complete
Digest: sha256:809edbbbf86f7eb1fe7bc9c31bfdb698e9922f14ec20abb2627fe301a81dd3f5
Status: Downloaded newer image for kiloview/polyworker:latest
docker.io/kiloview/polyworker:latest
root@VM-4-5-ubuntu:/data#
```

(6) Run the master container, enter the command in the Xshell terminal command line, the fields in the form should be set according to the actual situation, and other fields can be just by default.

docker run -d --restart=always --name polymaster \

```
-p 60000:81 \

-v /var/run/docker.sock:/var/run/docker.sock \

-v /usr/bin/docker:/usr/bin/docker \

-e PLATFORMIP=43.130.44.152 \

-e STARTPORT=60000 \

-e FWDPORT=10000 \

-e FWDNUM=10 \

-e TUNIP=43.130.44.152 \

-e TUNPORT=3478 \
```

```
-e TUNUSER=admin \

-e TUNPASS=kiloview \

-e TUNREALM=demo \

-v /data:/data \

--privileged --user root \

kiloview/polymaster
```

| Field	Description
-p 60000	60000 is the server management/API Http port, which can be modified
PLATFORMIP	Server Public IP address
TUNIP	Server Public IP address
TUNPORT	The default port is 3478, which should be the same as the listening-port in the turnserver.conf file

```
root@VM-4-5-ubuntu:/home/ubuntu# docker run -d --restart=always --name polymaster \
> -p 60000:81 \
> -v /var/run/docker.sock:/var/run/docker.sock \
> -v /usr/bin/docker:/usr/bin/docker \
> -e PLATFORMIP=43.130.44.152 \
> -e STARTPORT=60000 \
> -e FWDPORT=10000 \
> -e FWDNUM=10 \
> -e TUNIP=43.130.44.152 \
> -e TUNPORT=3478 \
> -e TUNUSER=admin \
> -e TUNPASS=kiloview \
> -e TUNREALM=demo \
> -v /data:/data \
> --privileged --user root \
> kiloview/polymaster
436cf4442c074215b86957fb3fd253842de0c1f294daf6311943be384305fb1c
root@VM-4-5-ubuntu:/home/ubuntu#
```

Note : The 4G bonding server will use many ports. If there is a firewall during deployment, the related ports need to be opened in the firewall. Kiloview uses AES256 encryption to ensure security in live streaming. If there are no special requirements, it is recommended to open all the ports of the 4G bonding server. If port restrictions are necessary, please contact kiloview support team for assistance at: support@kiloview.com.

Login the 4G bonding platform

Enter the server IP and the port number in the browser to enter the login webpage of the bonding platform.
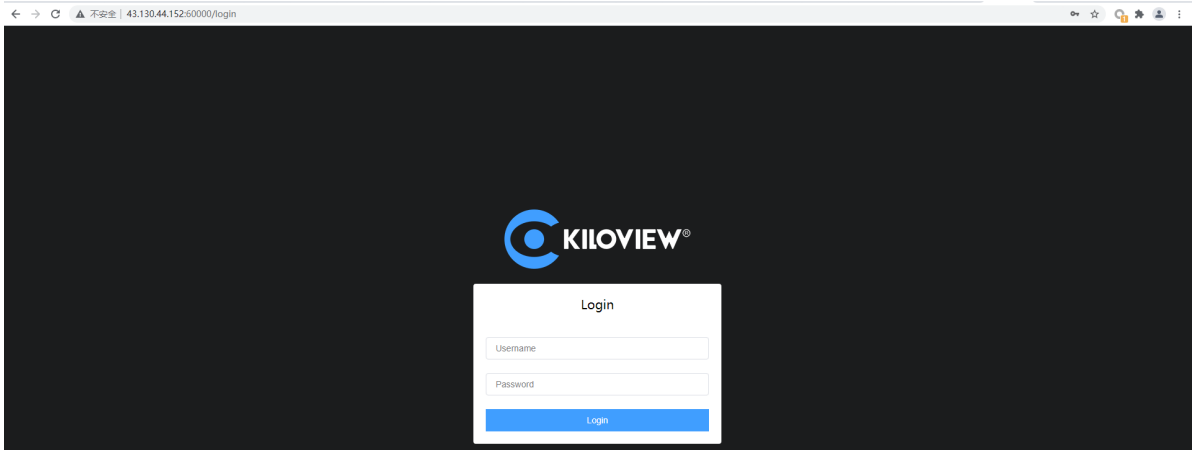
The username and the password both are admin by default, please modify them after your first login.