Kiloview intercom server (KIS) deployment

1 Preparations

1.1 Hardware environment

CPU:Intel Core i3 CPU or higher

Disk: 32G hard disk or higher

Memory: 2GB RAM or higher

1.2 Software

Ubuntu 18.04+ / Debian 9+

1.3 Network

1.3.1 Requirements of IP address and Bandwidth

IP address: The intercom server does not need a public network IP if all the devices are in the local network, otherwise public network IP address will be needed.

Bandwidth: According to your users number, we list some bandwidth requirements as below:

Users at the same time	Requirments of Bandwidth （Mbps)
10	7
20	30
30	70

Calculation: 64kbps×(Square of the quantity of users-the quantity of users)/0.8

> Note: Considering the hardware and maintenance costs of the server, and the subsequent update of the cloud platform, it is recommended to rent the server of the cloud-based service provider, such as: AWS, Alibaba Cloud, etc.

1.3.2 Requirements of ports

The devices need to establish calling with other devices through the port of intercom server, so all the ports of the intercom server should be enabled.

Below you can see the example of opening all the ports of Tecnet cloud server, click "firewall">"add", input “0.0.0.0/0” in the address of the source, click "OK", then all the ports of the server will be enable.

2 The process of deployment

2.1 Login to the server

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

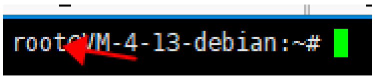
Xshell download link: https://www.netsarang.com/zh/xshell-download/

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

(2) Enter the username and password in the pop-up dialog box, and login as the root user.

If not root user, switching to root environment before operating.

```
sudo su
```



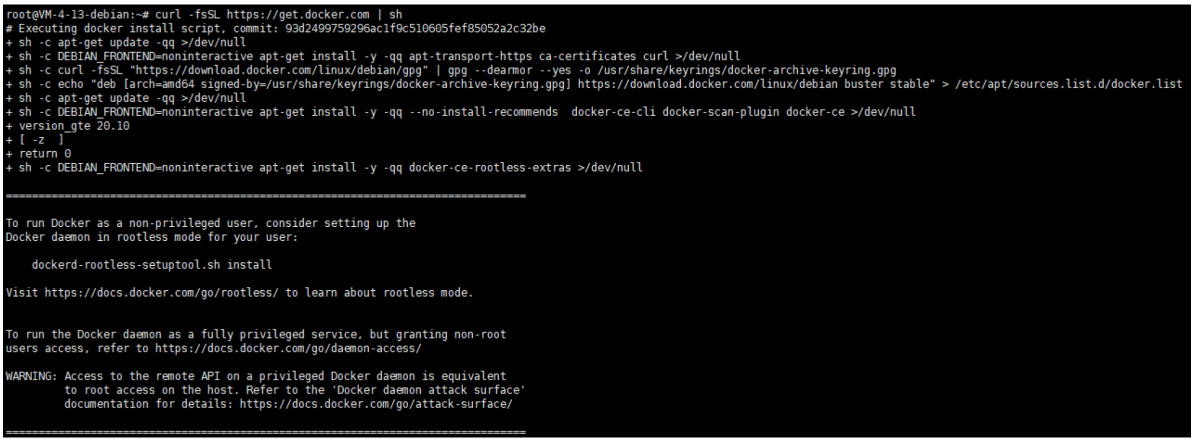
```
root@VM-4-13-debian:~#
```

2.1 Docker installation environment

2.1.1 Online installation

> If your docker version is 17.06 or above, please skip this step , you can view the current Docker version number by entering "docker version" in the terminal.

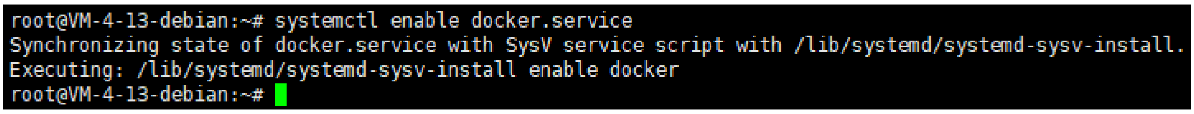
```
curl -fsSL https://get.docker.com | bash
```



```
root@VM-4-13-debian:~# 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/debian/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/debian buster 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

================================================================================

To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

    dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.


To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
         to root access on the host. Refer to the 'Docker daemon attack surface'
         documentation for details: https://docs.docker.com/go/attack-surface/

================================================================================
```

2.1.2 Set auto start service of the Docker

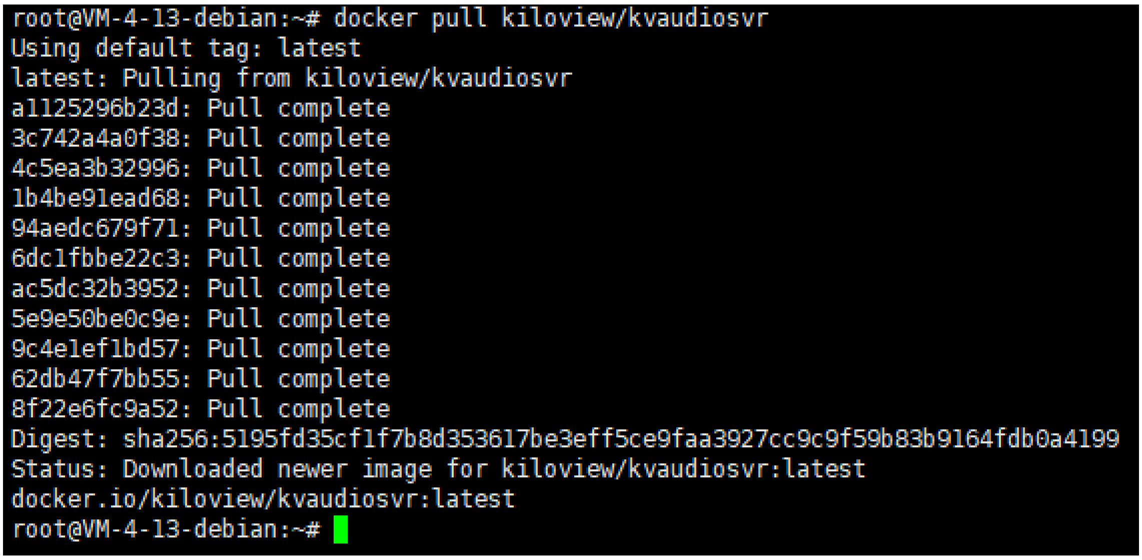
```
systemctl enable docker.service
```



```
root@VM-4-13-debian:~# systemctl enable docker.service
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
root@VM-4-13-debian:~#
```

2.2 Pull the intercom server mirror

```
docker pull kiloview/kvaudiosvr
```



```
root@VM-4-13-debian:~# docker pull kiloview/kvaudiosvr
Using default tag: latest
latest: Pulling from kiloview/kvaudiosvr
a1125296b23d: Pull complete
3c742a4a0f38: Pull complete
4c5ea3b32996: Pull complete
1b4be91ead68: Pull complete
94aedc679f71: Pull complete
6dc1fbbe22c3: Pull complete
ac5dc32b3952: Pull complete
5e9e50be0c9e: Pull complete
9c4e1ef1bd57: Pull complete
62db47f7bb55: Pull complete
8f22e6fc9a52: Pull complete
Digest: sha256:5195fd35cf1f7b8d353617be3eff5ce9faa3927cc9c9f59b83b9164fdb0a4199
Status: Downloaded newer image for kiloview/kvaudiosvr:latest
docker.io/kiloview/kvaudiosvr:latest
root@VM-4-13-debian:~# 
```

If visit "hub.cocker.com" is slow, the following solution is recommended.

1. Download the mirror image file

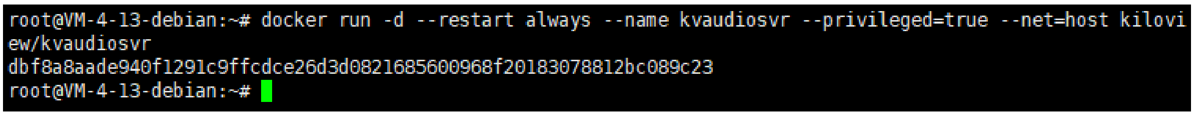
   ```
   wget https://www.kiloview.com/downloads/Tools/.server/kvaudiosvr.tar
   ```

2. Docker/2. Unzip and load the intercom mirror image to Docker

   ```
   docker load < kvaudiosvr.tar
   ```

2.3 Creat and run the container

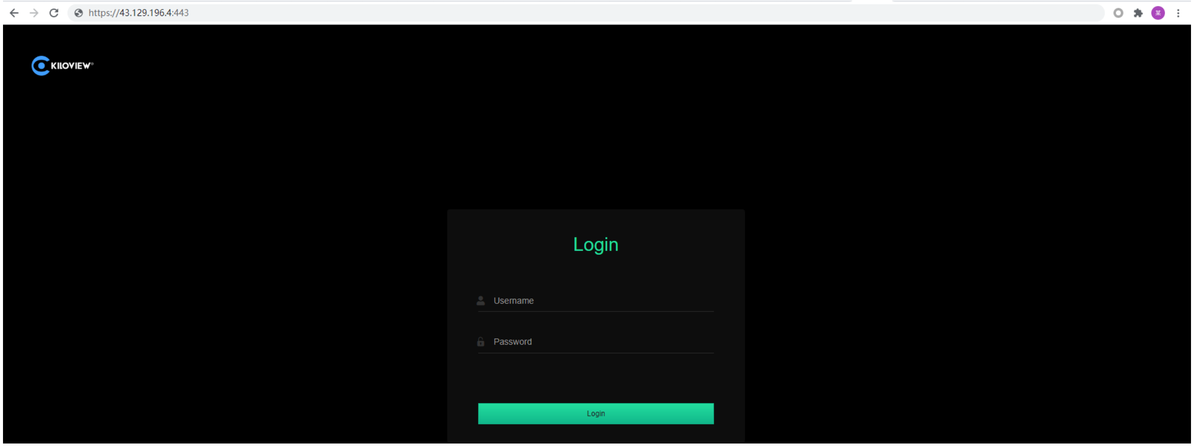
```
docker run -d --restart always --name kvaudiosvr --privileged=true --net=host
kiloview/kvaudiosvr
```



```
root@VM-4-13-debian:~# docker run -d --restart always --name kvaudiosvr --privileged=true --net=host kilovi
ew/kvaudiosvr
dbf8a8aade940f1291c9ffcdce26d3d0821685600968f20183078812bc089c23
root@VM-4-13-debian:~# 
```

3. Login to the server

```
https://server IP:443
```



The username and password are admin by default, please modify them after your first login.