Docker 部署 supernova 测试仪

目录

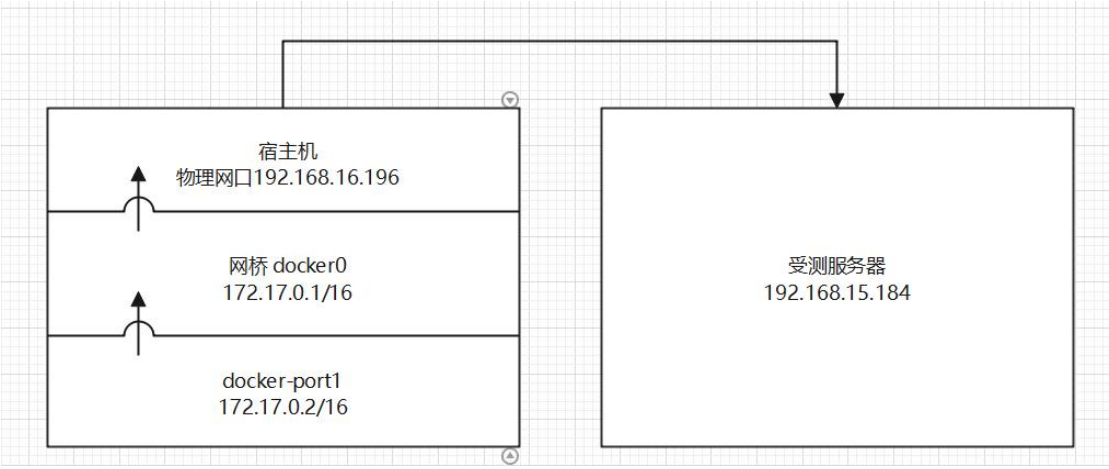
一、 Docker 介绍

Docker 使用 Google 公司推出的 Go 语言 进行开发实现，基于 Linux 内核的 cgroup，namespace，以及 AUFS 类的 Union FS 等技术，对进程进行封装隔离，属于 操作系统层面的虚拟化技术。由于隔离的进程独立于宿主和其它的隔离的进程，因此也称其为容器。最初实现是基于 LXC，从 0.7 版本以后开始去除 LXC，转而使用自行开发的 libcontainer，从 1.11 开始，则进一步演进为使用 runC 和 containerd。

Docker 在容器的基础上，进行了进一步的封装，从文件系统、网络互联到进程隔离等等，极大的简化了容器的创建和维护。使得 Docker 技术比虚拟机技术更为轻便、快捷。

二、 Docker 部署 supernova 测试仪拓扑

192.168.16.196 是宿主机 CentOS 的一个物理网口，这个网口负责跟外部通信，docker 创建的默认网桥 docker0 就桥接在这个网卡上，同时 docker 里面的 Supernova 测试仪的 Port1 也桥接在 docker0 这个网桥上，外部通过此网桥访问 docker 里面的 Supernova 虚拟测试仪。



三、 Docker 部署 supernova 测试仪

1. 部署 docker

   #部署可参考 https://www.jb51.net/article/255132.htm

   1.1 卸载以前安装的 docker

   #yum remove docker*

```
[root@localhost ~]# yum remove docker*
已加载插件：fastestmirror
正在解决依赖关系
--> 正在检查事务
---> 软件包 docker-buildx-plugin.x86_64.0.0.10.2-1.el7 将被 删除
---> 软件包 docker-ce.x86_64.3.23.0.1-1.el7 将被 删除
---> 软件包 docker-ce-cli.x86_64.1.23.0.1-1.el7 将被 删除
---> 软件包 docker-ce-rootless-extras.x86_64.0.23.0.1-1.el7 将被 删除
---> 软件包 docker-compose-plugin.x86_64.0.2.16.0-1.el7 将被 删除
---> 软件包 docker-scan-plugin.x86_64.0.0.23.0-3.el7 将被 删除
--> 解决依赖关系完成

依赖关系解决

================================================================================
 Package                    架构     版本                 源                 大小
================================================================================
正在删除:
 docker-buildx-plugin       x86_64   0.10.2-1.el7         @docker-ce-stable   53 M
 docker-ce                  x86_64   3:23.0.1-1.el7       @docker-ce-stable   94 M
 docker-ce-cli              x86_64   1:23.0.1-1.el7       @docker-ce-stable   34 M
 docker-ce-rootless-extras  x86_64   23.0.1-1.el7         @docker-ce-stable   19 M
 docker-compose-plugin      x86_64   2.16.0-1.el7         @docker-ce-stable   45 M
 docker-scan-plugin         x86_64   0.23.0-3.el7         @docker-ce-stable   12 M

事务概要
================================================================================
移除  6 软件包

安装大小：258 M
是否继续？[y/N]：y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  正在删除    : 3:docker-ce-23.0.1-1.el7.x86_64                              1/6
  正在删除    : docker-ce-rootless-extras-23.0.1-1.el7.x86_64                2/6
  正在删除    : 1:docker-ce-cli-23.0.1-1.el7.x86_64                          3/6
  正在删除    : docker-buildx-plugin-0.10.2-1.el7.x86_64                     4/6
  正在删除    : docker-compose-plugin-2.16.0-1.el7.x86_64                    5/6
  正在删除    : docker-scan-plugin-0.23.0-3.el7.x86_64                       6/6
  验证中      : docker-compose-plugin-2.16.0-1.el7.x86_64                    1/6
  验证中      : docker-scan-plugin-0.23.0-3.el7.x86_64                       2/6
  验证中      : docker-ce-rootless-extras-23.0.1-1.el7.x86_64                3/6
  验证中      : 3:docker-ce-23.0.1-1.el7.x86_64                              4/6
  验证中      : docker-buildx-plugin-0.10.2-1.el7.x86_64                     5/6
  验证中      : 1:docker-ce-cli-23.0.1-1.el7.x86_64                          6/6

删除:
  docker-buildx-plugin.x86_64 0:0.10.2-1.el7
  docker-ce.x86_64 3:23.0.1-1.el7
  docker-ce-cli.x86_64 1:23.0.1-1.el7
  docker-ce-rootless-extras.x86_64 0:23.0.1-1.el7
  docker-compose-plugin.x86_64 0:2.16.0-1.el7
  docker-scan-plugin.x86_64 0:0.23.0-3.el7

完毕！
[root@localhost ~]#
```

1.2 配置 yun 源

下载 yum 工具类所需要的依赖

yum install -y yum-utils

添加 docker 安装源地址

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

```
[root@localhost ~]# yum install -y yum-utils
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.bupt.edu.cn
 * extras: mirrors.bupt.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
软件包 yum-utils-1.1.31-54.el7_8.noarch 已安装并且是最新版本
无须任何处理
[root@localhost ~]# yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux
/centos/docker-ce.repo
已加载插件：fastestmirror
adding repo from: http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
grabbing file http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo to /etc/yum.r
epos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[root@localhost ~]#
```

1.3 安装 docker

安装最新版本得 docker

docker-ce---docker 社区版

docker-ce-cli---操作 docker 服务器的命令行程序

containerd.io---docker 容器化的运行环境

yum install -y docker-ce docker-ce-cli containerd.io

```
[root@localhost ~]# yum install -y docker-ce docker-ce-cli containerd.io
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.bupt.edu.cn
 * extras: mirrors.bupt.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
软件包 containerd.io-1.6.18-3.1.el7.x86_64 已安装并且是最新版本
正在解决依赖关系
--> 正在检查事务
---> 软件包 docker-ce.x86_64.3.23.0.1-1.el7 将被 安装
--> 正在处理依赖关系 docker-ce-rootless-extras，它被软件包 3:docker-ce-23.0.1-1.el7.x86_64
需要
---> 软件包 docker-ce-cli.x86_64.1.23.0.1-1.el7 将被 安装
--> 正在处理依赖关系 docker-buildx-plugin，它被软件包 1:docker-ce-cli-23.0.1-1.el7.x86_64
需要
--> 正在处理依赖关系 docker-compose-plugin，它被软件包 1:docker-ce-cli-23.0.1-1.el7.x86_64
需要
--> 正在处理依赖关系 docker-scan-plugin(x86-64)，它被软件包 1:docker-ce-cli-23.0.1-1.el7.x8
6_64 需要
--> 正在检查事务
---> 软件包 docker-buildx-plugin.x86_64.0.0.10.2-1.el7 将被 安装
---> 软件包 docker-ce-rootless-extras.x86_64.0.23.0.1-1.el7 将被 安装
---> 软件包 docker-compose-plugin.x86_64.0.2.16.0-1.el7 将被 安装
---> 软件包 docker-scan-plugin.x86_64.0.0.23.0-3.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

================================================================================
 Package                  架构       版本                  源                大小
================================================================================
正在安装:
 docker-ce                x86_64     3:23.0.1-1.el7        docker-ce-stable   23 M
 docker-ce-cli            x86_64     1:23.0.1-1.el7        docker-ce-stable   13 M
为依赖而安装:
 docker-buildx-plugin     x86_64     0.10.2-1.el7          docker-ce-stable   12 M
 docker-ce-rootless-extras x86_64    23.0.1-1.el7          docker-ce-stable  8.8 M
 docker-compose-plugin    x86_64     2.16.0-1.el7          docker-ce-stable   11 M
 docker-scan-plugin       x86_64     0.23.0-3.el7          docker-ce-stable  3.8 M

事务概要
================================================================================
安装  2 软件包 (+4 依赖软件包)

总下载量：72 M
安装大小：258 M
Downloading packages:
(1/6): docker-buildx-plugin-0.10.2-1.el7.x86_64.rpm                        17% |
            17% [================-
(1/6): docker-buildx-plugin-0.10.2-1.el7.x86_64.rpm
            18% [=================-
(1/6): docker-buildx-plugin-0.10.2-1.el7.x86_64.rpm
            18% [==================
(1/6): docker-buildx-plugin-0.10.2-1.el7.x86_64.rpm
            18% [==================
(1/6): docker-buildx-plugin-0.10.2-1.el 29% [========-                  ] 646 kB/s |  21 MB  00:01:20 ETA
```

如果想要安装其他版本，查看一下 docker 资源的安装列表

yum list | grep docker

1.4 启动 docker 并设置为开机自启

systemctl start docker

systemctl enable docker

```
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/d
ocker.service.
[root@localhost ~]#
```

1.5　查看一下当前的 docker 运行状态

systemctl status docker

```
[root@localhost ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2023-03-09 11:22:38 CST; 47s ago
     Docs: https://docs.docker.com
 Main PID: 1964 (dockerd)
   CGroup: /system.slice/docker.service
           └─1964 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.159732185+08:00" level...ng"
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.186253863+08:00" level...ng"
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.413758931+08:00" level...ss"
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.526041961+08:00" level...ng"
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.640876732+08:00" level...e."
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.657086350+08:00" level...0.1
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.657196990+08:00" level...on"
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.675466860+08:00" level...rpc
3月 09 11:22:38 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
3月 09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.680874492+08:00" level...ck"
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]#
```

1.6　验证 docker 安装是否完成

打印 docker 的详细信息

docker info

```
[root@localhost ~]# docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.10.2
    Path:     /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.16.0
    Path:     /usr/libexec/docker/cli-plugins/docker-compose
  scan: Docker Scan (Docker Inc.)
    Version:  v0.23.0
    Path:     /usr/libexec/docker/cli-plugins/docker-scan
```

2.　获取 supernova 测试仪镜像

方式一：从 dockerhub 下载镜像，自动加载至 docker

docker pull netitest/supernova:Supernova-23.03.04-build3076-X86_64

```
[root@localhost ~]# docker pull netitest/supernova:Supernova-23.03.04-build3076-
X86_64
Supernova-23.03.04-build3076-X86_64: Pulling from netitest/supernova
2d473b07cdd5: Pull complete
9fc8273c4d95: Pull complete
70c3491e3992: Pull complete
Digest: sha256:dd0f9357c3e229e4a8bebf7467e9f4620196e7ddc473c0cdf4ae9f7f003a778f
Status: Downloaded newer image for netitest/supernova:Supernova-23.03.04-build30
76-X86_64
docker.io/netitest/supernova:Supernova-23.03.04-build3076-X86_64
[root@localhost ~]# docker images
REPOSITORY           TAG                                   IMAGE ID       CREATE
D        SIZE
netitest/supernova   Supernova-23.03.04-build3076-X86_64   b9092e94422a   23 hou
rs ago   3.08GB
[root@localhost ~]#
```

方式二：本地上传镜像，并手动加载镜像至 docker

#wget

http://192.168.10.3/nova_images/23.03/23.03.07/23.03.07_build3107/NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar

```
[root@localhost ~]# wget http://192.168.10.3/nova_images/23.03/23.03.07/23.03.07_build3107/NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar
--2023-04-04 14:43:06--  http://192.168.10.3/nova_images/23.03/23.03.07/23.03.07_build3107/NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar
正在连接 192.168.10.3:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：1100492800 (1.0G) [application/x-tar]
正在保存至："NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar"

100%[==========================================================================================================================================>] 1,100,492,800 11.2MB/s 用时 94s

2023-04-04 14:44:40 (11.2 MB/s) - 已保存 "NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar" [1100492800/1100492800])
```

加载本地镜像

#docker load < NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar

```
[root@localhost ~]# docker load < NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img.docker.tar
174f56854903: Loading layer [==================================================>]  211.7MB/211.7MB
1d7e535d095f: Loading layer [==================================================>]  888.8MB/888.8MB
Loaded image: supernova:Supernova-v23.03.07-build3107-X86_64
```

3. 查看加载完成的 docker 镜像

docker images

```
[root@localhost ~]# docker images
REPOSITORY   TAG                                  IMAGE ID       CREATED       SIZE
supernova    Supernova-v23.03.07-build3107-X86_64   f3930a4b9af5   3 weeks ago   1.09GB
```

4. 启动 docker 容器

#docker run --privileged=true -e LANG=zh_CN.UTF-8 -v /dev/hugepages:/devhugepages -v /:/host_root -p 8080:80 -p 2222:22 -itd supernova:Supernova-v23.03.07-build3107-X86_64 /usr/sbin/init

启动导入的 docker 镜像以指定的登陆端口访问测试仪·

```
[root@localhost ~]# docker run  --privileged=true -e LANG=zh_CN.UTF-8 -v /dev/hugepages
:/devhugepages -v /:/host_root -p 8080:80 -p 2222:22 -itd supernova:Supernova-v23.03.07
-build3107-X86_64 /usr/sbin/init
bc4e9f997f97ac59802ee3523e0a2c14215e2234f0099c190393d6743ba9d13f
[root@localhost ~]#
```

5. 查看正在运行的容器状态

docker container ls

```
[root@localhost ~]# docker container ls
CONTAINER ID   IMAGE                                                COMMAND             CREA
TED          STATUS          PORTS
                    NAMES
bc4e9f997f97   supernova:Supernova-v23.03.07-build3107-X86_64   "/usr/sbin/init"   28 s
econds ago   Up 27 seconds   0.0.0.0:2222->22/tcp, :::2222->22/tcp, 0.0.0.0:8080->80/tc
p, :::8080->80/tcp   dreamy_jones
[root@localhost ~]#
```

6. 进入 docker 容器的内部 shell 终端窗口，验证网络联通性。

docker exec -it bc4e9f997f97 /bin/bash

```
[root@localhost ~]# docker exec -it bc4e9f997f97 /bin/bash
[root@bc4e9f997f97 /]# ping baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=48 time=34.3 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=48 time=34.3 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=48 time=33.8 ms
^C
--- baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 33.865/34.189/34.380/0.230 ms
[root@bc4e9f997f97 /]#
```

7. 在容器内安装部署 supernova-cos 版本

#cd /opt

tar zxvf NOVA_VM_COS-HW01-v23.03.06-build3091-20230318-X86_64.img.linux.tgz

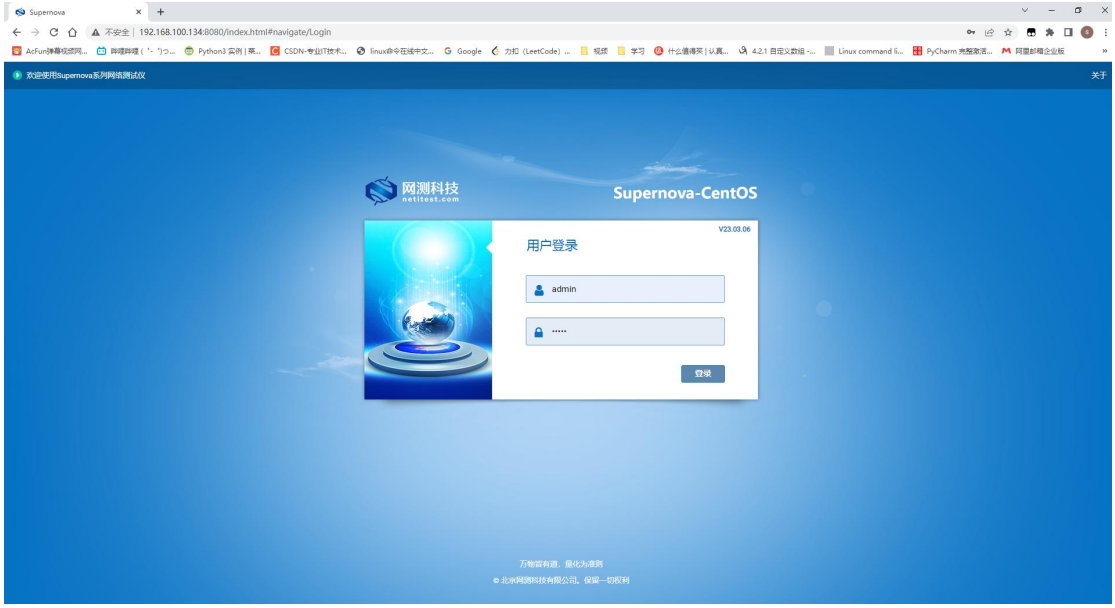
./nova_install -i NOVA_VM_COS-HW01-v23.03.06-build3091-20230318-X86_64.img

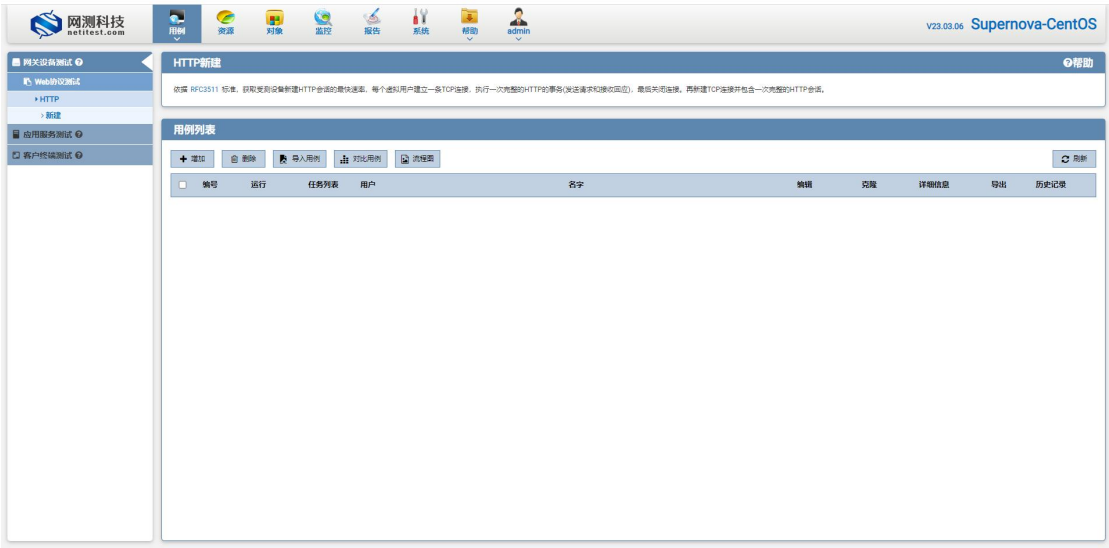
./nova_install -a

```
[root@bc4e9f997f97 /]# cd /opt
[root@bc4e9f997f97 opt]# tar zxvf NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.
img.linux.tgz
NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img
nova_install
README
[root@bc4e9f997f97 opt]# ./nova_install -i NOVA_VM_COS-HW01-v23.03.07-build3107-2023040
4-X86_64.img
linux_os_version: 11101000

Install Supernova image NOVA_VM_COS-HW01-v23.03.07-build3107-20230404-X86_64.img on Cen
tOS7.7 System ...

Next we will rename traffic port by following map:
Interface eth0, MAC address 02:42:ac:11:00:02 will be rename as port1

%20, Decrypt image ... Done
%40, Install ramdisk ... Done
%60, Install dhparam ... Done
%70, Install novacode ... Done
%80, Install toolkit ... Done
%90, Update grub config ... Done
%100, Install booter ... Done

Please run command './nova_install -a' to start Supernova web service
[root@bc4e9f997f97 opt]# ./nova_install -a
linux_os_version: 11101000
Starting ..................................................

2023-04-04 06:52:07, Supernova system is started.
```

8. 登陆测试仪 web 界面

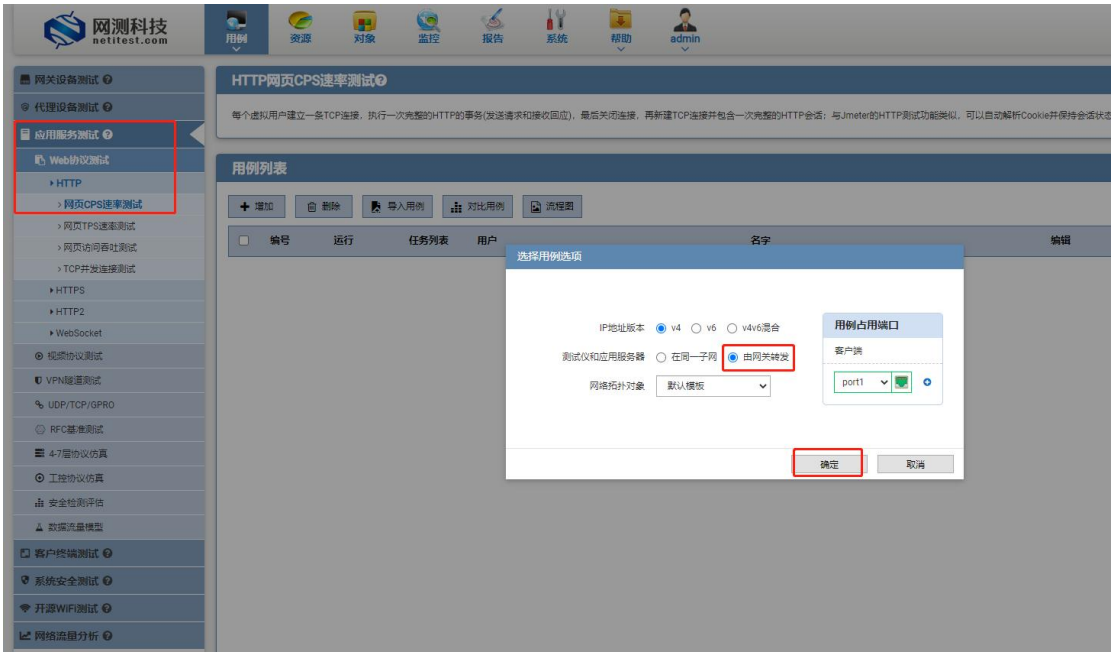
#http://192.168.16.196:8080



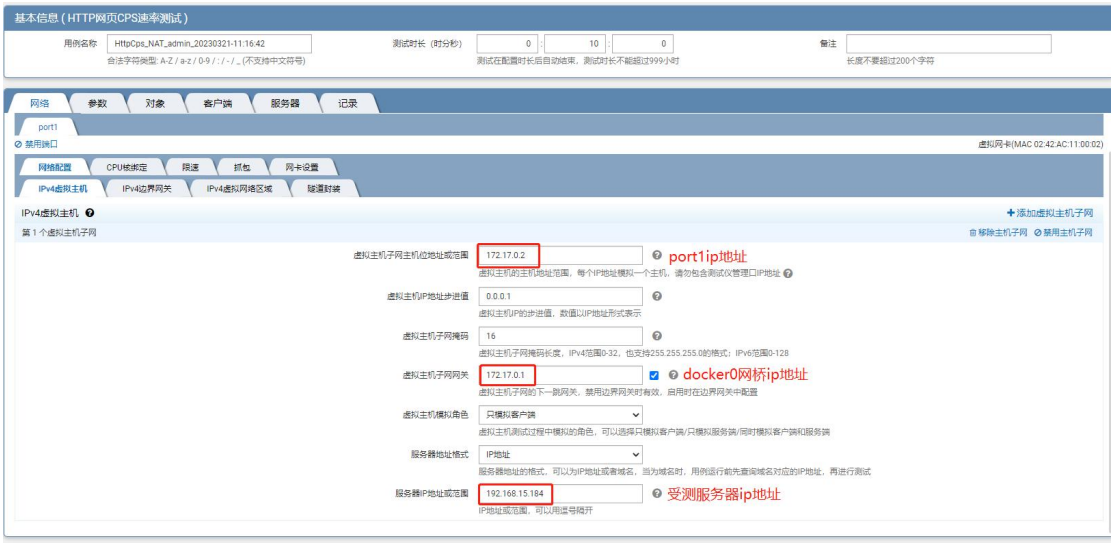


四、 配置测试用例

1. 创建用例，选择应用服务测试，由网关转发



2. 设置测试口、网关以及受测设备 ip 地址

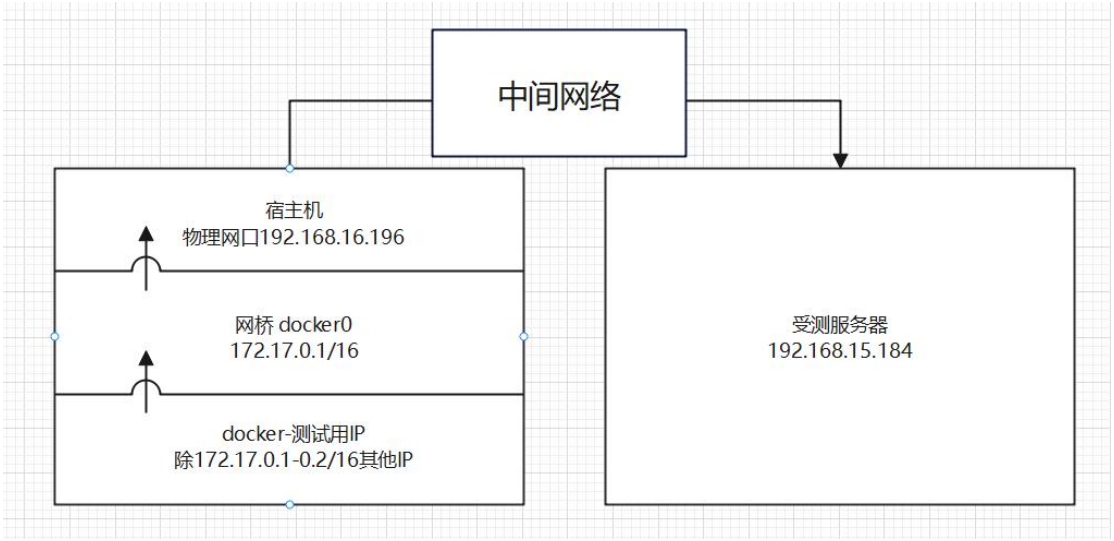


配置用例时，网络配置注意事项，可参考如下测试拓扑

1.配置测试 ip 地址范围为 docker 网桥地址范围默认是 172.17.0.0/16

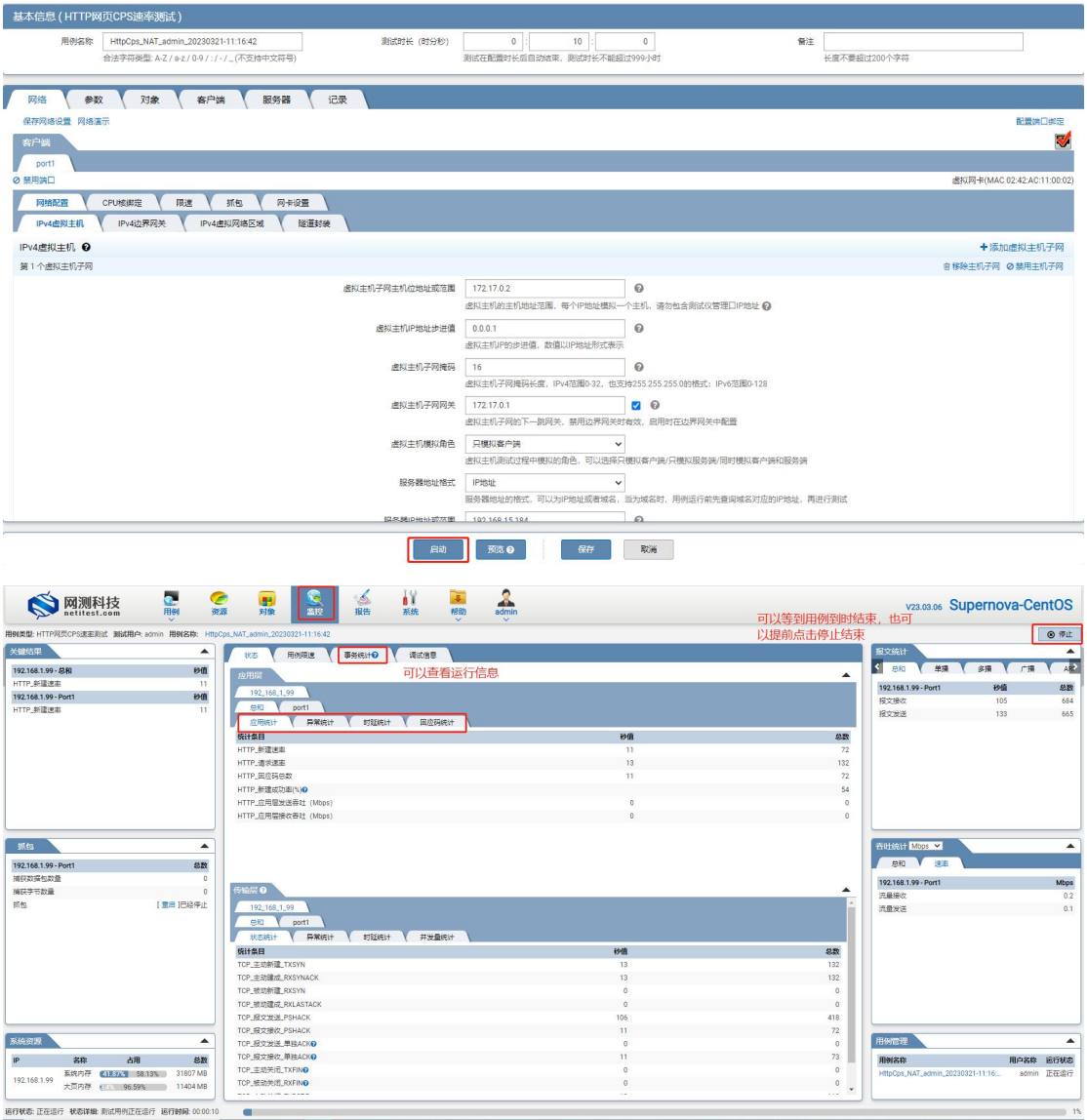
2.配置测试 ip 地址需避免使用 172.17.0.2（port1 的 IP 地址）172.17.0.1（docker0 的 IP 地址）

3.配置网关地址为 docker0 的 IP 地址 172.17.0.1

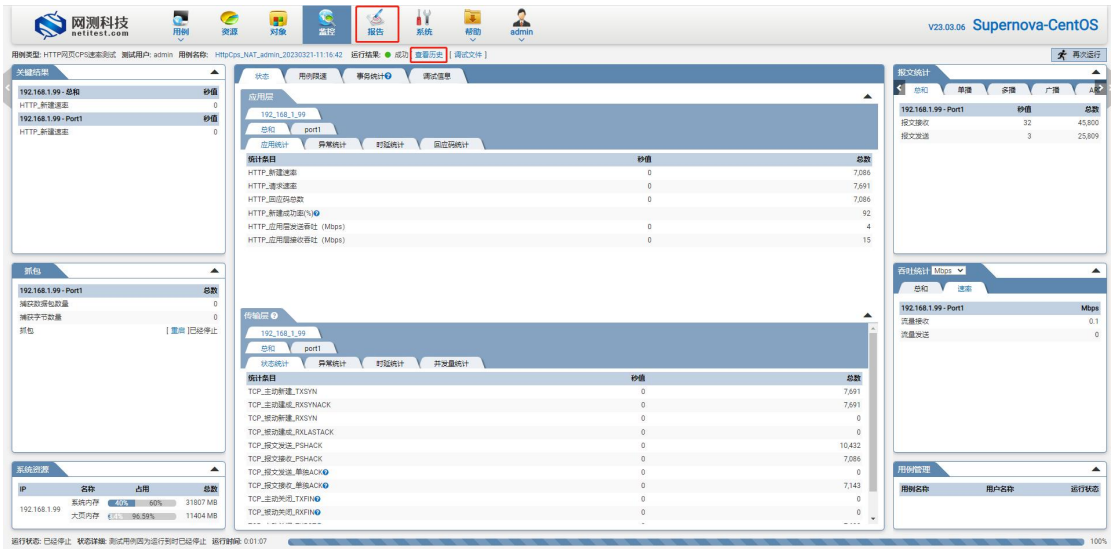


3. 设置 cpu 核数

4. 运行用例，点击后自动跳转至监控界面



5. 用例到时结束或手动结束，点击查看历史跳转至报告界面查看用例运行参数



点击生成报告可以查看运行信息