Docker 部署 supernova 测试仪

目录

Doc	ker 部署 supernova 测试仪	1
—、	Docker 介绍	2
_`	Docker 部署 supernova 测试仪拓扑	2
三、	Docker 部署 supernova 测试仪	2
1.	部署 docker	2
1.1	卸载以前安装的 docker	2
1.2	配置 yun 源	3
1.3	安装 docker	4
1.4	启动 docker 并设置为开机自启	4
1.5	查看一下当前的 docker 运行状态	5
1.6	验证 docker 安装是否完成	5
2.	获取 supernova 测试仪镜像	5
3.	查看加载完成的 docker 镜像	6
4.	启动 docker 容器	6
5.	查看正在运行的容器状态	6
6.	进入 docker 容器的内部 shell 终端窗口,验证网络联通性。	6
7.	在容器内安装部署 supernova-cos 版本	6
8.	登陆测试仪 web 界面	7
四、	配置测试用例	8
1.	创建用例,选择应用服务测试,由网关转发	8
2.	设置测试口、网关以及受测设备 ip 地址	8
3.	设置 cpu 核数	9
4.	运行用例,点击后自动跳转至监控界面	10
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	10

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-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 删除 =--> 解决依 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 3:23.0.1-1.el7 @docker-ce-stable 53 M docker-ce x86_64 @docker-ce-stable 94 M docker-ce-cli x86_64 docker-ce-rootless-extras x86_64 1:23.0.1-1.el7 23.0.1-1.el7 @docker-ce-stable 34 M @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-正在删除 : docker-ce-正在删除 : docker-ce-正在删除 : docker-ce-正在删除 : docker-comp 正在删除 : docker-comp 正在删除 : docker-scar 验证中 : docker-scar 验证中 : docker-ce-验证中 : docker-ce-验证中 : docker-ce-验证中 : 3:docker-ce-Downloading packages: : 3:docker-ce-23.0.1-1.el7.x86_64 1/6 : 3:docker-ce-23.0.1-1.el7.x86_64 : docker-ce-rootless-extras-23.0.1-1.el7.x86_64 : 1:docker-ce-cli-23.0.1-1.el7.x86_64 : docker-buildx-plugin-0.10.2-1.el7.x86_64 : docker-compose-plugin-2.16.0-1.el7.x86_64 : docker-scan-plugin-0.23.0-3.el7.x86_64 : docker-compose-plugin-2.16.0-1.el7.x86_64 : docker-ce-rootless-extras-23.0.1-1.el7.x86_64 : 3:docker-ce-23.0.1-1.el7.x86_64 2/6 3/6 4/6 5/6 6/6 1/6 2/6 3/6 : 3:docker-ce-23.0.1-1.el7.x86_64 : docker-buildx-plugin-0.10.2-1.el7.x86_64 : 1:docker-ce-cli-23.0.1-1.el7.x86_64 4/6 验证中 5/6 验证中 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 ~]# ■ 配置 yun 源 1.2 # 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 /cmtos/docker-ce.repo 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

 Iroot@localhost ~]# yum install -y docker-ce docker-ce-cli containerd.io

 E加载插件: fastestmirror

 Loading mirror speeds from cached hostfile

 * base: mirrors.bupt.edu.cn

 * extras: 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-li.x86 64.1.23.0 1-1 el7 将被 安装

 ™素 ---> 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 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.el7 将被 安装
--> 软件包 docker-scan-plugin.x86_64.0.0.23.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
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.e17	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%

] 646 kB/s | 21 MB 00:01:20 ETA

17%

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

启动 docker 并设置为开机自启 1.4

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 ~]#

查看一下当前的 docker 运行状态 1.5

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 (runnig) since 四 2023-03-09 11:22:38 CST; 47s ago
Docs: https://docs.docker.com
           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
                09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.159732185+08:00" level...ng"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.186253863+08:00" level...ng"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.413758931+08:00" level...s"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.526041961+08:00" level...e"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.640876732+08:00" level...e"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.657086350+08:00" level...e"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.657086350+08:00" level...on"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.657466860+08:00" level...on"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.67466860+08:00" level...on"

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.67466860+08:00" level...rpc

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.67468680+08:00" level...rpc

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.680874492+08:00" level...rpc

09 11:22:38 localhost.localdomain dockerd[1964]: time="2023-03-09T11:22:38.680874492+08:00" level...ck"

t: Some lines were ellipsized, use -l to show in full.
          3333333333333333
          3月 09 11:22:38 localhost
3月 09 11:22:38 localhost
Hint: Some lines were ell
[root@localhost ~]# ■
                      验证 docker 安装是否完成
         16
         # 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
                                               /usr/libexec/docker/cli-plugins/docker-compose
                    Path:
               scan: Docker Scan (Docker Inc.)
                    Version: v0.23.0
                                               /usr/libexec/docker/cli-plugins/docker-scan
                    Path:
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
Supernova-23.03.04-build30/6-x86_64: Putting 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
                     STZE
netitest/supernova
                                                 Supernova-23.03.04-build3076-x86_64
                                                                                                                                         b9092e94422a
                                                                                                                                                                           23 hou
                     3.08GB
 rs ago
 [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-HW0
1-v23.03.07-build3107-20230404-X86_64.img.docker.tar
     --2023-04-04 14:43:06-- http://132.168.10.3/nova_images/23.03/23.03.07/23.03.07
正在注意 1921.168.10.3:80... 已法意
已发出 11709-1800.1.1000年[10]10.1000 (K)
已发出 11709-1800.1.1000年[10]10.1000 (K)
卫星任师师室: 1900A.W.(2008-1901)-03.03.07-bu1143107-20230404-x86_64.1mg.docker.tar
```

->] 1,100,492,800 11.2MB/s 用时 94s

D

加载本地镜像

202-04-04 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])

#docker

3. 查看加载完成的 docker 镜像

docker images

[root@localhost ~]# docker images supernova Supernova-v23.03.07-build3107-x86_64 REPOSITORY TAG TMAGE TD CREATED STZE f3930a4b9af5 1.09GB 3 weeks ado 4. 启动 docker 容器 #docker -е LANG=zh_CN.UTF-8 run -V

#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 Is

[root@localhost ~]# docker container ls CONTAINER ID IMAGE 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 tt]=48 time=34.3 ms 64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 tt]=48 time=34.3 ms 64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 tt]=48 time=33.8 ms 64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 tt]=48 time=33.8 ms 64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 tt]=48 time=33.8 ms 64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 tt]=48 time=33.8 ms 7C --- 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-build3107-20230404 -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 %40, Install hovacode ... Done %40, Install hovacode ... Done %80, Install hovacode ... Done %80, Install toolkit ... Done %80, Install boter ... Done %100, Install boter ... Done %20, update grub config ... Done %100, Install boter ... Done %20, update grub config ... Done %100, Install boter ... Done %200, Install

8. 登陆测试仪 web 界面

#http://192.168.16.196:8080

😒 Supernova 🗙 🕂							× -	a a	×
← → C 合 ▲ 不安全 192.168.100.134:8080/index.html#navigate/Login						0-	x 🗃 🗯	• • •	÷
📴 AcFun弹簧视频网 🛅 哔哩哔哩 ('- ')つ 💿 Python3 实例 莱 🥃 CSDN-专业打技术	③ līnux命令在线中文 G Google	🞸 力扣 (LeetCode) 📒 税级	📃 学习 🙆 什么值得买 认真	🖓 4.2.1 自定义数组 📗 Linux e	command li 🔠	PyCharm 完整激活	M 阿里邮箱企业	比板	39
欢迎使用Supernova派列网络测试仪								*	ŧŦ
	 		Supernova-Cent	os					
		用户登录 2 admin 2 ·····	923. 1799	2.06					

 网测科技 netitest.com		目 対象	▲ 送 報告	▲¥ 系統	1580 adm					V23.03.06	Superno	ova-CentOS
■ 网关设有测试 Ø Ny Webb)的23%系 → HTTP	HTTP新建 依据 RFC3511 标准。	RR受到设备新建HTT	IP会适的最快速率,每个盘	約用戶證立—条70299	眼,执行一次完整的	ITTP的事务(发送请求和接收回应),最后关闭	Ti连接,再新建TCP连接并包含一次流图	вонттр е ца,				❷帮助
→新建	用例列表		時期 .1: 対比用例									C RM
	 第号 	运行 (任务列表 用户			87		编辑	克隆	详细信息	导出	历史记录

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

 顾测科技 netitest.com	Пи	الله الله الله الله الله الله الله ال		日本	系统	(1) 一部的	admin				
■ 网关设备测试 @	нттр	网页CPS速率》	则试仑								
◎ 代理设备测试 😡	每个虚拟	用户建立——各TCP连	医 执行一次宗教的	HTTP的事名(发送	建立印度の同応)	最后关闭连接 1	王新建TCP连接共有会-	-次完整的HTTP会	ita imeteri	OHTTPROTTORE	」可以自动報告Cookie共保持会话;
🗐 应用服务测试 😡 🖌		In MEDIC IN THIS		1111 434-33 (44.62	og/2************************************		27/128 1 01 24/24 1 25/24		Jonneten		
🖪 Web协议题试											
+ HTTP	FB1919	衣									
> 网页CPS速率测试	+ 增加	自制除	🛃 导入用例	. 对比用例	流程图						
>网页TPS逐渐测试											
>网页访问吞吐测试		编号 运行	r 任务外	」表 用户	选择用例选项			名字			编辑
> TCP并发连接测试											-
HTTPS											
+ HTTP2						IP#bth版本	● v4 ○ v6 ○	v4v6温合	用例占用	一端口	
WebSocket						1 ADADAG		THIONEL	友白港		
◎ 视频协议测试					测试	仪和应用服务器	○ 在同一子网 💿	由网关转发	H/ 35		
♥ VPN隧道测试						网络拓扑对象	默认模板	~	port1	~ 👿 O	
% UDP/TCP/GPRO											
◎ RFC基准则试											
■ 4-7层协议仿真									油中	BD/H	
⊙ 工控协议仿真									WEAL	-10/19	
由 安全检测评估											
▲ 数据流量模型											
□ 客户终端测试 Ø											
♥ 系统安全测试 ❷											
◆开源WIFI测试 ◎											
ビ 网络流量分析 😡											

2. 设置测试口、网关以及受测设备 ip 地址

基本信息(HTTP网页CPS速率测试)	
用的名称 [HttpCps.IAT_admin_20233321-11:1642) 測试时长 (时分秒) 合法学符映版 A Z / s 2 / o 9 / : / - (不交帅中交符号)	0 : 10 : 0 輸送 期送在建築时に振用的結果、例此时に不能通过100个字符
网络 参数 对象 客户端 服务器 记录	
◎ 禁用跳口	虚拟码卡(MAC 02:42 AC:11:00:02)
网络配置 CPU核绑定 膜速 抓包 网卡设置	
IPv4虚拟主机 IPv4边界网关 IPv4虚拟网络区域 隧道时装	
IPv4虚拟主机	+ 添加虚拟主机子网
篇1个虚拟主机子网	會 解除主机子网 ◎ 疑用主机子网
虚拟主机子网主机位地址或范围	17217.0.2
虚拟主的UP地址步进值	0.0.1 虚拟主WP的步进值、数值以P他址形式表示
遗拟主机子网通码	16 直线主約子何機弱於歲, IPv4范围0-32, 也支持255.255.0的例题で; IPv6范围0-128
虚拟主机子网网关	172.17.0.1 2 O docker0网桥ip地址 曲以主机子例的下一個网块、備用边界网头对海线、自用时在边界网头中都重
虚拟主机模拟角色	只機制客户購 本社工和原则过程中構制的角色、可以提導目標制等户與/目標制度分為/同时標制客户時和服务等
服务器地址拖式	IP#stit v
服务器户性址或范围	総合著物地が形式、可以为iP2地址式準確点、当次地名FI、制作語行物主要特徴な71位30P3地址、利由行期は 192.168.15.184 ● 受測服分器ip1地址 P2地域空気型、可以加速中49开

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

基本信息(HTTP网页CPS速率测试)	
用除高額 HttpCps_NAT_admin_20230321-11:16.42 奥ば封长(由法字研制版 AZ/az/04/:/-/_(不支持中文符号)	28596) 0 10 0 0 第注 第述在記題が长周囲が伝承、第述が未不能描述2000-243
网络 参数 对象 客户端 服务器 记录	
保存网络设置 网络演示	配置通口绑定
香户端	2
port1	
⊘ 蒸用跳口	虚拟网卡(MAC 02:42 AC:11:00 02)
网络配置 V CPU核規定 V 限速 V 抓包 V 网卡设置	
CPU核绑定	
c	23 12日1日2月10日10日10日10日5-0。推荐##11460123.4567.8310.11 13.8349-41451 (通用用的20-1145, 0.123.4567.8310.11 13.8349-41451 (通用用的20-1145, 0.123.456.78.910.11 14.76381G2489881989999944, 一个地球, 一个地球, 一个地球, 24.8549 25.7754849551038292480420, 49.448.849882 7.788949551038292480420, 49.448.849882 27.88949551038292480420, 49.448.849882 27.88949551038292480420, 19.4548402489882 27.88949551038292480420, 19.4548402489882 27.88949541038292480420, 19.4548402489882 27.88949541038292480420, 19.4548402489882 27.889495492484840, 19.4548484234988 27.8894954103849440, 19.4548442349882 27.8894954103842480420, 19.4548442349882 27.8894954103842480420, 19.454844234984 27.8894954103842480420, 19.454844234988 27.8894954103842480420, 19.454844234988 27.8894954103842480420, 19.454844234988 27.8894954103842480420, 19.454844234988 27.8894954103842480420, 19.454844234988 27.8894954103842484042, 19.45484444444444444444444444444444444444

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

A Herrescould		v v				
型:HTTP网页CPS速率测试 测试用户:admin 用例名约	t: HttpCps_NAT_admin_20230321-11:16:42 运行描果: ● 成功 查看历史 [语武文件]					大 再次运
結果	▲ · · · · · · · · · · · · · · · · · · ·			报文统计		
168.1.99 - 总和	and the second s		•	▲ 総和 単微	家職 广	HE A
新建速率	0			192 168 1 99 - Port1	ines.	2
68.1.99 - Port1	90 192_168_1_99			报文编辑	32	45.8
新疆遠志	0 BR port1			授文发送	3	25.8
	应用统计 V 异常统计 V 时延続计 V 回应码统计 V					
	统计条目	砂值	总数			
	HTTP_新建速率	D	7,086			
	HTTP_请求速率	0	7,691			
	HTTP_回应码总数	0	7,086			
	HTTP_新建成功率(%)●		92			
	HTTP_应用层发送香吐 (Mbps)	D	4			
	HTTP_应用温操收吞吐 (Mbps)	0	15			
8				AUISEIT Mops V		
168.1.99 - Port1	BAR			284 1 286	<u>)</u>	
	(伊納田の		•	192.168.1.99 - Port1		Mb
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			×	流量接收		
	192_168_1_99			流量发送		
	统计集目	砂体	8 R			
	TCP_主助新建_TXSYN	0	7,691			
	TCP_主动建成_RXSYNACK	0	7,691			
	TCP_根助新建_RXSYN	0	0			
	TCP_根动建成_RXLASTACK	0	0			
	TCP_报文发送_PSHACK	0	10,432			
	TCP_HEXHACK	0	7,086			
38	▲ TCP_报文发送_单独ACK❷	0	0	用例配理		
名称 占用 :	TCP_报文授収_单独ACKO	0	7,143	用例名称	用户名称	运行状
系统内存 40% 60% 31807	MB TCP_主助关闭_TXFINO	0	0			
68.1.99	TCP_报助关闭_RXFINO	0	0			

点击生成报告可以查看运行信息

ø	网测科技 netitest.com	開墾	《 》 资源	利泉	送 出注	警 服告	家族	藤助	admin				V23.03	.06 Supernova-	CentOS
报告类型															
测试目	标 全部		~			ş	6019421 (全部		~	用例案称	按用例名查找			
用例题	果 全部		~					开始时间	· 編末8)明		血斑			
报告列表		_		_	_	_	-		_			_	_		
Q #58	运行结果														C 84
	用例信息														
	用例名称 HttpCps_I	NAT_admin_20230	1321-11:16:42							用例类型 HTTP网页CPS速率表试 开始期间 2023-03-21 11-26-54					
	运行续果● 测试运行	于正常结束								結束財间 2023-03-21 11:28:01					
	测试时长0时01分	0715													
	运行数据														
显示行数[🖈 再次运行 🗋	生成报告) 生成文档	⑦ 下號HTML	T THEP	DF	Nord	🕑 下航Excel	由 报酬对比					自 删除此记录	< 1 >
L	点击 生成报告 按钮,就	会生成测试报告。	主要包括摘要信	18.											_
	息息 王成文档 按钮,就	学生成例试报告,	同时包会生成H	TML/PDF/Word/	EXCEPTEIC服告。	包括水平和信息和数	38.92198, DJ	下戰到本効用服	5						
														关闭	1

制名称 HttpCps_NAT_admin_20230321-11:16:42			用例类型 HTTP网页CPS速率测试		
a式用户 admin			开始时间 2023-03-21 11:26:54		
3行結果● 測试运行正常結束			结束时间 2023-03-21 11:28:01		
赋时长 0时01分07秒					
数据					
再次运行 🗋 生成报告 🖸 生成文档 🔝 下数HTML	合下就PDF 🗋 下就Word 🗈	下數Excel 最告对比			自制除此记
状态 () 請口 (关键结果 (系统资源) 事务组	戌計● 【 満試信息 】				
内田田 (内田谷山)			(Etel Providence)		
STATE	10/10	0.85	ACCONTRA A MANAGERI (ADAMATERI (ATAXAMETERI	24.65	0.80
	12.00	10483	现计第日	6710	49-93
TTP_新建速率	118	7,086	Set 亲目 TCP_主动新疆_TXSYN	128	-694X 7,691
ITTP_新建速率 ITTP_請求速率	118 128	7,086	NGT第日 TCP_主动新建_TXSYN TCP_主动建成_RXSYNACK	128	7,691
HTFL新建連車 ITFL-請交連車 ITFL-面立码总数	118 128 118	7,086 7,691 7,086	Sett 第日 TCP_主約希望_TXSYN TCP_主約確応, EXSYNACK TCP_拒約推定, EXSYN	128 128 0	7,691 7,691 0
TTP_新羅連生 TTP_通気速差 TTP_回互発急数 TTP_新選成功率(%)●	118 128 118	7,086 7,691 7,086 92	MentaBE TCP.主約課題、XXSYN TCP.主約課題、XXSYNACK TCP.県初新題、XXASTACK TCP.県初新題、XXASTACK	128 128 0 0	7,691 7,691 0 0
TTP_魚灌塗車 TTP_通常運動 TTP_通知受き数 TTP-新速域功率(%)の TTP-通用環境波路在(Mbos)	118 128 118 118	7,086 7,691 7,086 92 4	Wart新日 TOP 主动端点 EXSYN TOP 主动端点 EXSYNACK TOP 通动端走 RXLASTACK TOP 通动端走 RXLASTACK TOP 通気法学 PARACK	128 128 0 0 173	7,691 7,691 0 0 10,432
177- 泉水塗生 177- 泉水交差単 177- 泉水交差単 177- 泉池成内金(Mopo) 177- 泉油炭肉交通(Mopo) 177- 皮用炭素が高な(Mopo) 177- 皮用炭素が高な(Mopo)	0 0	7,086 7,086 92 4	Win計算 Top 主切時違, TSOYN Top 主切前違, RSOYNACK Top 退防違点, ROLASTACK Top 退防定加止, PsAcK Top 退防力能, PsAcK	0 173 173	7,691 7,691 0 0 10,432 7,086
1779. 新築地主 1779. 通行装置 1779. 国在研究 1779. 国在研究 1779. 国用党状態社 1779. 互用管状化器社 100cg) 1779. 互用管状化器社 100cg)	0 0	7,086 7,086 7,086 92 4 15	###書単 TOP 主要構造_XSINACK TOP 主要構造_KSINACK TOP 通要構成_RALASTACK TOP 通要構成_PALASTACK TOP 通要型構成のFALASTACK TOP 通要型構成のFALASTACK TOP 通要型構成のFALASTACK TOP 通要型構成のFALASTACK	128 128 0 0 173 118 0	7,691 7,691 0 10,432 7,086 0
177年 月 市 市 市 市 市 二 二 二 二 二 二 二 二 二 二 二 二 二	0 0 0	7,086 7,691 7,086 92 4 15	###書題 TOP 主要構成 #SINACK TOP 主要構成 #SINACK TOP 建築成成 #SINA TOP 建築成の #SINACK TOP 建築成の FSIACK TOP 建築成の FSIACK TOP 建築成の FSIACK TOP 建築成の #SIACK	0 ma 128 0 0 173 118 0 119	7,691 7,691 0 10,432 7,086 0 7,143
177. 新型協会 177. 漫戏協会 177. 漫戏協会会 177. 送班的公式会会(Moos) 177. 近期短期使名称(Moos)	0 118 128 118 0 0	7,085 7,691 7,086 92 4 15	With第目 Top 主切時違, XSYN Top 主切能度, XSYNACK Top 建設度, XSYNACK Top 建設度, XSUNATACK Top 建設度, XSUNATACK Top 建設定度, SNACK Top 建設定度, SNACK Top 建設定度, SNACK Top 主動法定(SNAC)	0 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	7,691 7,691 0 10,432 7,086 0 7,143 0 0 7,143
1772 新建生産 1772 漫天漫画 1772 漫風気を受い 1772 漫風気の意いの 1772 漫風気を発行(Micros) 1772 風風気振行者社(Micros)	0 118 128 118 0 0	7,085 7,691 7,086 92 4 15	###書題 TOP 主切構造 FXSYNACK TOP 主切構造 FXSYNACK TOP 生切構造 FXSYNACK TOP 建設造成 FXASTACK TOP 建設造成 FXASTACK TOP 建設立造 FXSACK TOP 建設立造 FXSACK TOP 建設立造 FXSACK TOP 建設立成 FXSACK TOP 建設立成 FXSACK TOP 建設立成 FXSACK TOP 生びあって、 FXSACK FXS	0 44 128 0 0 173 118 0 119 0 0	
175 台湾總統 1772-唐尔选城 1772-唐尔达城 1775 上海道公司(NO 1775 上海道公司(NO 1775 上海道公司)	118 128 118 0 0	7,086 7,091 7,086 92 4 15	##### Top 主動構成, XSON Top 主動構成, XSON Top 主動構成, SSONACK Top 振動構成, SSONA Top 振動構成, SSONA Top 振動化の Top Landval Top Top Top Landval Top Top Landval Top Top Top Landval Top Top Top Landval Top Top Landval Top Top Top Landval Top Top Top Landval Top Landval Top Top Landval Top Top Landval Top Landval Top Top Landval Top	128 128 0 0 173 118 0 119 0 0 0 128	7,691 7,691 0 10,432 7,066 0 7,143 0 0 7,143 0 0 0 0 0 7,143 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
177上解選走差 177上接交通 177上接交通 1771上接用型加速(NO 1771上規模型加速(NOs) 1771上規模型加速(NOs)	118 128 118 0 0	7,086 7,091 7,095 92 4 15	世界書類 「PD 主切希達」XSON IPD 主切希達、XSONACK IPD 主切希達 IPD 正想通道、AKSONAC IPD 正想通道、AKSONA IPD 正想通道、AKSONA IPD 正想通道、AKSONA IPD 正型正規道、AKSONA IPD 正型正規道、AKSONA IPD 正型正規道、AKSONA IPD 正型正規道、AKSONA IPD 正型LEADING IPD EXAMPLE IPD EXAMPLE	228 128 0 0 1773 118 0 119 0 0 128 0	7,691 7,691 0 0 10,432 7,086 7,085 0 7,143 0 0 7,082 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1772 新建生産 1772 - 東京連進 1773 - 東京連載 1775 - 東京連載 1775 - 東京三大学 1775 - 東京三 1775 - 東京三 1775 - 東京	118 128 118 0 0	7,085 7,091 7,095 92 4 15	###書# TOP 主切構造 X507N TOP 主切構造 X507N TOP 主切構造 X507N TOP 生切構造 X507N TOP 建切構点 X507N TOP 建动物点 X507N TOP 建动物 和 X507N TOP 建动物 和 X507N TOP 建动物 和 X507N TOP 建动物 和 X507N TOP 建动物 X507N TOP # X507N T	128 128 0 1773 118 0 119 0 0 128 0	7,691 7,691 0 10,432 7,068 0 7,143 0 0 7,143 0 0 7,743 0 0 0 7,682 0