KVM_SR-IOV 上部署 Supernova

(界面方式和命令行方式)

网测科技

2023.03.01

KVM 介:	绍	5
主题内容	<u> </u>	5
1.	安装 KVM 前准备	5
	1.1 安装 CentOS 系统时注意:	6
	1.2 禁用它如下:	6
	1.3 验证 cpu 是否支持 KVM,结果有 vmx(INTEL)或 svm(AMD)说明支持	÷6
	1.4 在 BIOS 中开机虚拟化功能	7
	1.5 关闭 XELinux	8
	1.6 查看物理网卡是否支持 SR-IOV(单根虚拟化),说明支持 SR-IOV	8
	1.7 支持 CPU 虚拟化透传	8
2. 5	安装 KVM	8
	2.1 安装 KVM 包	8
	2.2 安装 KVM 核心包——虚拟操作系统模拟器加速模块	8
	2.3 重启宿主机,加载 KVM 相关模块	9
	2.4 查看 KVM 模块是否被正确加载	9
	2.5 开启 KVM 服务,并设置开机启动	9
	2.6 查看操作结果,出现 Active: active (running) 字样则说明运行情况良好.	9
	2.7 如果执行"开启 KVM 服务"报错请更新系统	10
3.	需要在内核中启用,在打开的文件中添加 intel_iommu=on 参数 和 pci=as	sign-
bus	sses 和 pci=realloc 参数	10
	3.1 更新 GRUB 后重启使之生效	10
4.硝	甬认网卡设备的驱动类型	10
	4.1 可以查看某个物理网卡支持多少 VF (虚拟网卡)	11
	4.2 若需要重新加载驱动,先移除驱动,再加载	11
	4.3 举例每个网卡创建一个 VF	11
	4.4 分离虚拟出来的网卡	12
5.界	科面方式部署 KVM	13
	5.1 添加存储池	13
	5.2 创建虚拟机	16
	5.3 再增加一块 Data 盘(SuperNova 默认要用到两块硬盘)	19
	6.4 把从宿主机上分离出来的 PCI 网卡添加到虚拟机上来	21
	5.5 调整 CPU 和内存大小	23
6.这	≤行 KVM 启动 Supernova	23
	6.1 运行 KVM	24
	6.2 成功运行后设置 ip 地址	27
	6.3 登陆界面	27
	6.4 配置用例并成功运行	28
7 伎	^{使用命令行方式部署虚拟机}	28
	7.1.创建 sriov 文件夹	28
	8.2 进入该目录上传镜像	29

目录

8.3.创建 sriov 的池并启动	29
8.4.分离网卡	
8.5.上传虚拟机的 xml 文件	31
8.6 启动虚拟机	
8.7 登陆虚拟机后配置虚拟机 ip 和网关	
9.附加: VNC 配置方法	35
9.1 安装软件包	35
9.2 关闭防火墙	
9.3 复制配置文件	
9.4 编辑复制出来的配置文件	
9.5 重新加载配置文件	
9.6 设置 VNC 密码	
9.7 开启 VNC 并设置成开机启动	
9.8 启动	
9.9 如果启动成功端口是监听状态(VNC 端口号默认 5900+1)	
9.10 vnc 客户端连接	

版本	说明	时间
V1.0	初版拟制	2019/01/15
V1.1	增加"1.7: 支持 CPU 虚拟化透传"	2019/09/10
V1.2	增加8使用命令行方式部署虚拟机	2022/09/09

KVM 介绍

KVM 是 Kernel-based Virtual Machine 的简称,是一个开源的系统虚拟化模块,自 Linux 2.6.20 之后集成在 Linux 的各个主要发行版本中。它使用 Linux 自身的调度器进行管理。 KVM 目前已成为学术界的主流 VMM 之一。

KVM 的虚拟化需要硬件支持(如 Intel VT 技术或 AMD V 技术)。是基于硬件的完全虚拟化。 网卡是一对 Intel 82599-10G 网卡

VT-d 的性能非常好,但是它的物理设备只能分配给一个客户机使用。为了实现多个虚机 共享一个物理设备,并且达到直接分配的目的,PCI-SIG 组织发布了 SR-IOV (Single Root I/O Virtualization and sharing) 规范,它定义了一个标准化的机制用以原生地支持实 现多个客户机共享一个设备。不过,目前 SR-IOV (单根 I/O 虚拟化)最广泛地应用还 是网卡上。

SR-IOV 使得一个单一的功能单元(比如,一个以太网端口)能看起来像多个独立的物理 设备。一个带有 SR-IOV 功能的物理设备能被配置为多个功能单元。SR-IOV 使用两种功 能 (function):

- 物理功能 (Physical Functions, PF): 这是完整的带有 SR-IOV 能力的 PCIe 设备。PF 能像普通 PCI 设备那样被发现、管理和配置。
- 虚拟功能 (Virtual Functions, VF): 简单的 PCle 功能, 它只能处理 I/O。每个 VF 都是从 PF 中分离出来的。每个物理硬件都有一个 VF 数目的限制。一个 PF, 能 被虚拟成多个 VF 用于分配给多个虚拟机。

从 SRIOV 的中文字面不难理解,它属于 VT-d 技术的一个分支,要实现 SRIOV 功能,前提 条件就是你的网卡首先要支持 SRIOV,你的主板要支持 VT-d 技术(支持 VT-d 自然也就支 持 SRIOV)

Hypervisor 能将一个或者多个 VF 分配给一个虚机。在某一时刻,一个 VF 只能被分配给 一个虚机。一个虚机可以拥有多个 VF。在虚机的操作系统看来,一个 VF 网卡看起来和 一个普通网卡没有区别。SR-IOV 驱动是在内核中实现的。

宿主机:就是实体机

主题内容

1. 安装 KVM 前准备

1.1 安装 CentOS 系统时注意:



1.2 禁用它如下:

- 命令: chkconfig NetworkManager off
- 命令: chkconfig network on
- 命令: service NetworkManager stop

1.3 验证 cpu 是否支持 KVM,结果有 vmx (INTEL) 或 svm (AMD) 说明支持

命令: cat /proc/cpuinfo | egrep 'vmx|svm' 效果:

[root@localhost ~]# egrep '(vmx|svm)' /proc/cpuinfo flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss httm pbe syscall n pdpelgb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_c 1 vmx smx est tm2 sse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowpre etch epb intel_pt tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmil hle avx2 smep bmi2 erms invpcid rtm mpx rdseed adx smap clflushopt ssaveopt xsavec xgetbv1 dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall n pdpelgb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_c 1 vmx smx est tm2 sse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowpre etch epb intel_pt tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmil hle avx2 smep bmi2 erms invpcid rtm mpx rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall n pdpelgb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_c 1 vmx smx est tm2 sse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowpre etch epb intel_pt tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmil hle avx2 smep bmi2 erms invpcid rtm mpx rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_e

1.4 在 BIOS 中开机虚拟化功能

方法:开启按 delete 键进入 BIOS 中--用键盘方向键选中"Advanced"菜单--选中"CPU Configuration"—找到"Intel Virtualization Technology"开启 找到"VT-D"开启 效果:



1.5 关闭 XELinux

命令: vi /etc/sysconfig/selinux 效果:

This file controls the state of SELinux on the system.
SELINUX= can take one of these three values:
enforcing - SELinux security policy is enforced.
permissive - SELinux prints warnings instead of enforcing.
disabled – No SELinux policy is loaded.
SELINUX=disabled
SELINUXTYPE= can take one of three two values:
targeted - Targeted processes are protected,
minimum - Modification of targeted policy. Only selected processes are protected.
mls - Multi Level Security protection.
SELINUXTYPE=targeted

1.6 查看物理网卡是否支持 SR-IOV(单根虚拟化), 说明支持 SR-IOV

命令: Ispci -v |grep SR-IOV

效果:

[root@localhost ~]# lspci -v grep	SR-IOV
Capabilities: [160] Single	Root I/O Virtualization (SR-IOV)
Capabilities: [160] Single	Root I/O Virtualization (SR-IOV)
[root@localhost ~]#	

1.7 支持 CPU 虚拟化透传

命令: echo 'options kvm_intel nested=1' >/etc/modprobe.d/kvm-nested.conf

2. 安装 KVM

2.1 安装 KVM 包

命令: yum -y install kvm

2.2 安装 KVM 核心包——虚拟操作系统模拟器加速模块

命令: yum -y install qemu-kvm qemu-kvm-tools 命令: yum -y install libvirt python-virtinst libvirt-python virt-manager libguests-tools bridge-utils virt-install 说明: libvirt: 必须要装的核心工具 pyhon-virtinst: 包含 python 模块和工具 (virt-install, virt-clone 和 virt-image) virt-manager: 虚拟机图形管理工具 (宿主机有桌面环境时可以考虑安装, 命令操作或者远 程控制则不需要) bridge-utils: 虚拟机与外界通信的命令管理工具 virt-install: 虚拟机安装工具

2.3 重启宿主机,加载 KVM 相关模块

命令: reboot

2.4 查看 KVM 模块是否被正确加载

命令: lsmod | grep kvm 出现以下信息则表示正确加载。

[root@localhost	~]#	lsmod g	rep kvm
kvm_intel		183621	0
kvm		586948	1 kvm_intel
irqbypass		13503	1 kvm
[root@localhost	~]#		

2.5 开启 KVM 服务,并设置开机启动

- 命令: systemctl start libvirtd.service (开启) (如果报错请看 2.7)
- 命令: systemctl enable libvirtd.service (开机启动)

2.6 查看操作结果,出现 Active: active (running) 字样则说明运行情况良好

命令: systemctl status libvirtd (启动状态)

enabled

[root@localhost ~]#

命令: systemctl is-enabled libvirtd(是否开机自动启动) 效果:

[root@localhost ~]# systemct] status libvirtd
Iibvirtd.service - Virtualization daemon
<u>Loaded: loaded (/usr/lib/</u> systemd/system/libvirtd.service; enabled; vendor preset: enabled)
Active: active (running) since — 2019-01-14 18:04:47 CST; 18min ago
Docs: man:libvirtd(8)
https://libvirt.org
Main PID: 6050 (libvirtd)
Tasks: 19 (limit: 32768)
CGroup: /system.slice/libvirtd.service
—6050 /usr/sbin/libvirtd
⊣8500 /usr/sbin/dnsmasqconf-file=/var/lib/libvirt/dnsmasq/default.confleasefile-rodhcp-scr
∽8501 /usr/sbin/dnsmasqcont-file=/var/lib/libvirt/dnsmasq/default.contleasefile-rodhcp-scr
1 1 14 18.04.47 Jacobart Jacobaroin system/[1], started virtualization deamon
1 + 14 10:04:47 localhost.localdomain systemorr[]: started virtualization daemon.
1 + 14 10:04:52 localhost.localdomain drismasq[0500]: started, version 2.76 cachesize 130
1 + 14 10.04.32 local host. local domain dursmasci dom [2000]. Compile time options. 1700 dwo-getopt bass no-fisin taw bars
$1 \neq 14$ 18:04:52 localhost.localdomain dhsmasq-dhcp[8500]: bHCP, 1P range 192.106.122.2 - 192.106.122.234, leas
1 H 18:04:52 localhost. localdomain dismasgreence [Stol], modified (as a full stole) conf
1 + 14 10:04:52 localhost. localdomain drismasq [0500]: reading /etc/resolv.com
1 + 14 10:04:52 localhost. localdomain drismasq[0500]; using nameserver 100.76.76.76.76.55
$1 \neq 14$ 10:04:52 localhost. localdomain drismasq[0500]; read /etc/hosts - 2 addresses
$1 \neq 14$ 18:04:52 localhost. localdomain drismasq[disv0]: read /var/lib/lib/ir/drismasq/delault.addintosts - 0 address
1)- 14 10.04.32 Tocarnost. Tocartomann unsmass-pointe fostog. Teau /var/The/The/The/The/The/The/The/The/The/The
HINC. Some times were ethipsized, use -i to show in turi.
[noot@localhact]# cyctamet] ic anahlad libyintd

2.7 如果执行"开启 KVM 服务"报错请更新系统

[root@localhost ~]# systemctl start libvirtd.service
Job for libvirtd.service failed because the control process exited with
error code. See "systemctl status libvirtd.service" and "journalctl -xe"
for details.
[root@localhost ~]#

命令: yum -y update

3. 需要在内核中启用,在打开的文件中添加 intel_iommu=on 参数

和 pci=assign-busses 和 pci=realloc 参数

命令: vi /etc/default/grub 效果:

="\$(sed 's, release .*\$,,g' /etc/system-release) aved JBMENU=true DUTPUT="console" INUX="crashkernel=auto ECOVERY="true" centos/root rd.lvm.lv=centos/swap rhgb quiet intel_iommu=on pci=assign-busses pci=realloc

3.1 更新 GRUB 后重启使之生效

命令: grub2-mkconfig > /boot/grub2/grub.cfg 重启 reboot

4. 确认网卡设备的驱动类型

命令: ethtool -i enp1s0f0 效果: [root@localhost ~]# ethtool -i enp1s0f0
driver: ixgbe
version: 5.1.0-k-rh7.6
firmware-version: 0x00011bab
expansion-rom-version:
bus-info: 0000:01:00.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: yes
[root@localhost ~]#

4.1 可以查看某个物理网卡支持多少 VF (虚拟网卡)

命令: cat /sys/class/net/enp1s0f0/device/sriov_totalvfs 效果:

[root@localhost ~]# cat /sys/class/net/enp1s0f0/device/sriov_totalvfs

4.2 若需要重新加载驱动,先移除驱动,再加载

命令: modprobe -r ixgbe (正常安装, <mark>不需要执行</mark>) 效果:

[root@localhost ~]# ethtool -i enpls0f0 Cannot get driver information: No such device [root@localhost ~]#

已经找不到这个网卡了 因为已经成功将驱动移除 注意:在 virsh 中 ":"和 "." 变为 "_"编号都是一样的支持显示格式不一样

4.3 举例每个网卡创建一个 VF

命令: echo 1 > /sys/class/net/enp1s0f0/device/sriov_numvfs echo 1 > /sys/class/net/enp1s0f1/device/sriov_numvfs

[root@localhost ~]# echo 1 > /sys/class/net/enp1s0f0/device/sriov_numvfs
[root@localhost ~]# echo 1 > /sys/class/net/enp1s0f1/device/sriov_numvfs

命令含义为将实体网卡 enp1s0f0 虚拟出一个 VF; echo 2 就是虚拟出 2 个 VF;

4.4 分离虚拟出来的网卡

[root@]	[root@localhost ~]# lspci grep Ethernet								
00:1f.6	Ethernet	controller:	Intel	Corporation	Ethernet Connection (2) I219-V				
01:00.0	Ethernet	controller:	Intel	Corporation	82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)				
01:00.1	Ethernet	controller:	Intel	Corporation	82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)				
02:10.0	Ethernet	controller:	Intel	Corporation	82599 Ethernet Controller Virtual Function (rev 01)				
02.10 1	Ethernet	controller:	Intel	Corporation	82599 Ethernet Controller Virtual Eunction (rev 01)				

命令: virsh nodedev-detach pci_0000_02_10_0 virsh nodedev-detach pci_0000_02_10_1

效果:



[root@localhost ~]# virsh nodedev-detach pci_0000_02_10_1 已分离设备 pci_0000_02_10_1

[root@localhost ~]#

这里是可以选择添加的(直到6以前):

命令: vi /etc/rc.d/rc.local

效果:



命令: vi /etc/rc.d/rc.local 效果:

It is highly advisable to create own systemd services or udev rules # to run scripts during boot instead of using this file. # # In contrast to previous versions due to parallel execution during boot # this script will NOT be run after all other services. # # Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure # that this script will be executed during boot. touch /var/lock/subsys/local echo 1 > /sys/class/net/enp1s0f0/device/sriov_numvfs echo 1 > /sys/class/net/enp1s0f1/device/sriov_numvfs ip link set enp1s0f0 vf 0 mac aa:bb:cc:dd:ee:00 ip link set enp1s0f1 vf 0 mac aa:bb:cc:dd:ee:01

5. 界面方式部署 KVM

5.1 添加存储池

	roo	t@localhost	t:~				-
文件(F) 编辑(E) 查看(V) 搜索(S	5) 终端(T)	帮助(H)					
[root®localhost ~] # virt-man [root®localhost ~] # []	nager						
I	虚拟系统管 环	里器		-		×	
文件(F) 编辑(E) 查看(V) 帮	助(H)						
正≦ 连接详情(C)							
名称 虚拟机详情(V) 思 OEMU/K 删除(D)			Ŧ	CPU 使用	率		
● d21000 首选项(P) 运行中						1	

虚拟系统管理器	_ ¤ ×			
文件(F) 编辑(E) 查看(V) 帮助(H) 连接详情(C)				QEMU/KVM 连接详情 _ □ ×
	▼ CPU 使用率	文件(F)		
◆ GEMUA 首选项(P) 注行中		 機还 虚拟网络(V) ⁸⁶⁵ 文件系统目录 ⁸⁶⁵ 文件系统目录 ⁸⁶⁶ 次件系统目录 	存储(S) 名称: 大小: 位置: 状态: 自动启动	阿鉛茲ロ KVM_Pass-through 5.55 GIB Free / 44.43 GIB In Use /var/lib/lib/it/images/KVM_Pass-through 교话跃 U): ♥ 引导时
			卷 boot.qco	 ・ 大小 格式 用于 W2 396.58 MiB qcow2 KVM_Pass-through w2 30.00 GiB qcow2
		● Þ ● ⊗		应用(A)

		虚拟系统	管理器		>	¢					SR-IOV	- QEM
文件(F) 编	辑(E) 查看(V)	帮助(H)				文件(F) 虚	ē拟机(M)	查看(V)	发送按	躍(K)		
	打开 ⊳		Ŧ					00	-			
名称				- CPU	使用率							
- QEMU/KVM				_								
SR-IO 运行中	V			_								
	-). (d. 1991			QEMU/K	VM 连接详情	E. 1		-		c		
	文件(F)			添加	新存储池	×						
	概述 虚	题初网络(V)		创建存储池								
	90% default 文件系统目	큪	Ľ	2 的步骤 1								
	90% sriov	_	选择您希	自己的存住	赭 池类型。							
	文件系统日:	录	名称(N)	: sriov								
			类型(T)	: dir: 文件	系统目录	-						
			I	取消(C)	后退(B)	前进(F)						
			-									
									应用(A)			
		虚拟	系统管理器	Ŧ	-	o x						
文件(F) 编	辑(E) 查看(V) 帮助(H)			ź.	5件(F) .	虚拟机()	4) 查看	(V) 2	过送按键	(K)
	打开 🖻	> 00 (00		-	
				~	CPU使用	率						÷
- QEMU/KVM												
SR-IO	V											
· 运行9				QI	EMU/KVM i	车接详情				_	0 X	
	→仕(F)		_			CL3X FT IN				_		
						添加新存储液	ŧ.			×		
	概述	虚拟网络()	V)	入 创建石	存储池							
	90% default			9 2 的步								
	文件系统 Sriov	目录			[Addition of the second			
	90% 文件系统	目录	日月	际路径(T):	/var/lib/lil	bvirt/images/si	riov	-	浏览(R)	_		
					I	反消(C)	后退(B)		完成(F)			
			L									
											H (A)	
		• •										

第14页

< > < 🗨 var	lib libvirt images sriov >	Q == = _
 ○ 最近使用 ① 主目录 ■ 视频 ③ 图片 	boot.qcow2 data.qcow2	提前导入到此目录的image选中boot
运行中	QEMU/KVM 连接详情	×
文件(F) 概述 虚拟网络 90% default 文件系统目录 90% Sflov 文件系统目录	A称: sriov 大小: 4.82 GiB Free / 45.1 位置: /var/lib/libvirt/image 状态: ■ 活跃 自动启动(U): ▼引导时 卷 ▼ 大小 格式 boot.qcow2 396.58 MiB qcow2 data.qcow2 30.00 GiB qcow2	L5 GiB In Use es/sriov 第一一章
• • • •		应用(A)

注意:从 23.03 版本开始需要手动创建数据盘 用 23.03 以后版本解压出的文件为 README 和系统盘 boot.qcow2

提取 + NOVA_V	M_CLD-I	HW01-v23.03.0)5-build3083	• Q = ×
< > 企 位置(L):	i /			
名称	~	大小	类型	已修改
boot.qcow2		606.0 MB	未知	2023年3月13日 21:36
README		112 字节	未知	2023年1月5日 09:12

打开终端执行创建数据盘命令

打开(0)	• <u>A</u>			~/,	README cache/.fr-9r3zvk		保存(S)		=	×
reate 40G empty disk by following command:										
# qemu-im	≠ qemu-img create -f qcow2 -o preallocation=full ./data.qcow2 40G									
	root@localhost:~ _							×		
文件(F)	编辑(E)	查看(V)	搜索(S)	终端(T)	帮助(H)					

[root@localhost ~]# qemu-img create -f qcow2 -o preallocation=full ./data.qcow2 40G Formatting './data.qcow2', fmt=qcow2 size=42949672960 encryption=off cluster_siz e=65536 preallocation='full' lazy_refcounts=off [root@localhost ~]# ls anaconda-ks.cfg [data.qcow2] README 公共 视频 文档 音乐 boot.qcow2 initial-setup-ks.cfg root 模板 图片 下載 桌面 [root@localhost ~]# ■

5.2 创建虚拟机

			虚	拟系统	管理器	ł			-	•	×
文件(F)	编辑(E)	查看(V)	帮助	(H)							
E	📃 打开	• >	00		~						
名称		创建虚	拟机				-	CPU包	用	×	
V QEMU/k	WM										

新建虚拟机	×
生成新虚拟机 4 的步骤 1	
连接(<u>O</u>): QEMU/KVM	
 选择如何安装操作系统 ▲地安装介质(ISO 映像或者光驱)(L) 网络安装(L),可为 HTTP、FTP 或者 NFS。 网络引导(PXE)(B) ●导入现有磁盘映像(E) 	
取消(<u>C</u>) 后退(B) 前进(E)

新建虚拟机	×
生成新虚拟机 4 的步骤 2	用率
提供现有存储路径(G):	
浏览(R))

选择存储卷

	^{92%} default 文件系统目录 ^{50%} Pass-through 文件系统目录	大小: 位置: 卷	24.98 /var/lil	GiB Free / 25 b/libvirt/imag	5.00 GiB In es/SR-IO\	Use /	
Γ	50% SR-IOV 立件系统目录	卷	-	大小	格式	用于	
۲		boot.c	qcow2	396.58 MiB	qcow2	SR-IOV	
	test		•				
	_{93%} test 文件系统目录	data.c	cow2	30.00 GiB	qcow2	SR-IOV	

初 注土 15 15 15 15 1
■ 生成新虚拟机 4 的步骤 3
选择内存和 CPU 设置
内存(M), RAM: 8192 - +
主机中最多有 32607 MiB 可用
CPU: 4 – +
最多有 6 可用
取消(C) 后退(B) 前进(F)
新建虚拟机 ×
■ 生成新虚拟机 4 的步骤 4
住成新虚拟机 4 的步骤 4 的步骤 4 准备开始安装
生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称
生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称 操作系统: Generic
生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称 操作系统: Generic 安装: 导入现有 OS 映像
生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称 操作系统: Generic 安装: 导入现有 OS 映像 内存: 8192 MiB
生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称 操作系统: Generic 安装: 导入现有 OS 映像 内存: 8192 MiB CPU: 4 左妹: (and the initial formation for the initial formation formation formation formation formation formation forma

生成新虚拟机 4 的步骤 4 准备开始安装 名称(N) 虚拟机名称 操作系统: Generic 安装: 导入现有 OS 映像 内存: 8192 MiB CPU: 4 存储: /var/lib/lib/virt/images/test/boot.qcow2
 ▶ 选择网络(E)

在创建虚拟机的过程中第一块 Boot 盘就有了

虚拟系统管理器 _ 0 × 文件(F) 编辑(E) 查看(V) 帮助(H) 连接详情(C) **E** -虚拟机详情(V) 名称 ▼ CPU 使用率 删除(D) QEMU/K 首选项(P) s 运行中 添加新虚拟硬件 × 存储 存储 ■ 控制器 ○ 为虚拟机创建磁盘镜像(R) 🗐 网络 20.0 – + GiB 输入 ۲ 💻 图形 25.0 GiB available in the default location ■『 声音 ● 选择或创建自定义存储(S) 🗐 串口 /var/lib/libvirt/images/SR-IOV/d 管理(M)... 剩 并口 <table-cell-columns> 控制台 设备类型(D): ■ 磁盘设备 • <table-cell-columns> 信道 💩 USB 主机设备 总线类型: IDE 🔻 💩 PCI 主机设备 ▼高级选项 🔜 视频 缓存模式(h): 管理程序默认 ▼ 🛒 Watchdog 🛅 文件系统 ≧ 智能卡 🜒 USB 重新定向 TPM 🖓 随机数生成器 🎄 Panic 通知器 取消(C) 完成(F)

5.3 再增加一块 Data 盘 (SuperNova 默认要用到两块硬盘)

	选择存储卷	×
50% default 文件系统目录 50% Pass-through 文件系统目录 50% SR-IOV 文件系统目录 93% test 93% test 50% Vswich 50% 文件系统目录	大小: 24.98 GiB Free / 25.00 GiB In Use 位置: /var/lib/libvirt/images/SR-IOV 巻 ご (2) 巻 マ 大小 格式 用于 boot.qcow2 396.58 MiB qcow2 SR-IOV data.qcow2 30.00 GiB qcow2	
+ 0 0	本地浏览(B) 取消(C) 选	择卷
 ○ 概况 ○ 性能 ○ 内存 ○ 引导选项 ○ IDE 磁盘 1 ○ IDE 磁盘 2 ○ NIC : 2a:56:5b ○ 鼠标 ○ 鍵盘 ○ 認示协议 Spice ○ Sound ich6 ○ 串口 1 ○ 信道 spice ○ PCI 0000:04:10.0 ○ PCI 0000:04:10.1 ○ 显卡 QXL ○ Controller USB 0 ○ Controller IDE 0 ○ Controller VietIO Secial 0 ○ 添加硬件(D) 	虚拟磁盘 源路径: /var/lib/libvirt/images/SR-IOV/boot.qcow2 设备类型: IDE 磁盘 1 存储大小: 396.58 MIB 只读(E): □ 可共享(B): □ *高级选项(o) * 磁盘总线(u): IDE 序列号(L): * 存储格式(t): qcow2 *性能选项(P) * 援存模式(h): writethrough IO 模式: 管理程序默认	

💻 概况	虚拟磁盘
	源路径: /var/lib/libvirt/images/SR-IOV/data.qcow2
CPUs	设备类型: IDE 磁盘 2
■ 内存	存储大小: 30.00 GiB
🔊 引导选项	只读(E): 🗌
IDE 磁盘 1	可共享(B): 🗌
🛋 IDE 磁盘 2	▼高级选项(o)
🚅 NIC :2a:56:5b	磁盘总线(u): IDE
👌 鼠标	序列号(1):
🛋 键盘	//// (L) ·
🛒 显示协议 Spice	存储格式(t): qcow2
📑 Sound ich6	▼ 性能选项(P)
🚵 串口 1	缓存模式(h): writethrough ▼
🚵 信道 spice	
FCI 0000:04:10.0	
FCI 0000:04:10.1	
🕎 显卡 QXL	
Controller USB 0	
Controller PCI 0	
Controller IDE 0	
Controllor Virtl@ Social @	
添加硬件(D)	

6.4 把从宿主机上分离出来的 PCI 网卡添加到虚拟机上来

虚拟系统管理器 _ □ ×			Wild Date-through OEMU/W/M P V
文件(F) 编辑(E) 資看(V) 帮助(H)	文件(F) 虚		盜加新塵扨硬件 ×
正 连接详情 📜 🔹		三 存储	PCI 设备
名称 GPU 使用率 CPU 使用率 #lk(D)	🖳 概况	 控制器 回 网络 	主机设备(D):
▼ QEMU/K	▶ 性能	• 输入	0000:00:1B:4 Intel Corporation 200 Series PCH PCI Express Root Port #21
▶ K = 22 × K(*) 通行中	CPUs	- 图形	0000:00:1C:0 Intel Corporation 200 Series PCH PCI Express Root Port #1
	■ 内存	■ 声音	0000:00:1C:1 Intel Corporation 200 Series PCH PCI Express Root Port #2
	33 引导选	-≪ 串口	0000:00:1C:2 Intel Corporation 200 Series PCH PCI Express Root Port #3
	画 IDE 磁线	≪] 并口	0000:00:1C:4 Intel Corporation 200 Series PCH PCI Express Root Port #5
$\mathbf{\lambda}$	🔜 SCSI 確	✓ 控制台	0000:00:1C:5 Intel Corporation 200 Series PCH PCI Express Root Port #6
	🛒 NIC :c1	<table-cell-columns> 信道</table-cell-columns>	0000:00:1C:6 Intel Corporation 200 Series PCH PCI Express Root Port #7
	鼠标	ىڨ USB 主机设备	0000:00:1C:7 Intel Corporation 200 Series PCH PCI Express Root Port #8
	🛋 键盘	🝰 PCI 主机设备	0000:00:1D:0 Intel Corporation 200 Series PCH PCI Express Root Port #9
	🐖 显示协	📮 视频	0000:00:1F:0 Intel Corporation Z370 Chipset LPC/eSPI Controller
	📑 Sound	Watchdog	0000:00:1F:2 Intel Corporation 200 Series/Z370 Chipset Family Power Manage
	🚵 串口 1	📄 文件系统	0000:00:1F:3 Intel Corporation 200 Series PCH HD Audio
	🚵 信道 sj	🚵 智能卡	0000:00:1F:4 Intel Corporation 200 Series/Z370 Chipset Family SMBus Contro
	PCI 000	USB 重新定向	0000:00:1F:6 Intel Corporation Ethernet Connection (2) I219-V (接口 enp0s3
	PCI 000	TPM	0000:01:00:0 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
	🕎 显卡 C	@ 随机数生成器	0000:01:00:1 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
	Control	- ⑧ Panic 通知器	0000:03:00:0 ASMedia Technology Inc. ASM2142 USB 3.1 Host Controller
	Contro		取造(C) 完成(E)
	Cantra		AXIB(C) JUAN(F)
	添加在	硬件(D)	



5.5 调整 CPU 和内存大小



6. 运行 KVM 启动 Supernova

6.1 运行 KVM

	虛拟系统管理器	_ = ×
文件(F) 编辑(E) 查看(V)	帮助(H)	
📔 📃 打开 🕞		
名称		▼ CPU使用率
▼ QEMU/KVM		
KVM_Pass-through		
运行(<u>R</u>)		
暂停(P)		
关机(S)	•	
克隆		
迁移		
删除(<u>D</u>)		
打开(<u>O</u>)		

注: 如果启动虚拟机失败,报错为: pci…请更换宿主机网卡插槽位置;

请在 vi /etc/fstab 下增加 devpts /dev/pts devpts gid=5,mode=620 0 0 如下图:

[root@localhost centos_chroot]# cat /etc/fstab				
# ///fatab				
# /etc/istab # Created by anaconda on Tue Jan 8 15:57:37 20	19			
# # Accessible filesystems, by reference, are main	ntained	under '/dev/o	disk'	
# 5% man pages istab(5), iindis(8), mount(8) al # 	nd/or b.	IKId(8) Ior mo	ore info	
/dev/mapper/centos-root / UUID=52cd09d9-03f7-405f-85f0-ed0eec22ee8a /boot	xfs	defaults xi	00 fs defaults	0 0
/dev/mapper/centos-home /home	xfs	defaults	0 0	
devpts /dev/pts devpts gid=5, mode=620	o 0	ueraults		
[root@localhost_centos_chroot]#				



注:如果在启动过程中报 port1/port2 错误,进入到页面后发现刚报出错误的 port 没有显示 出来那么请在 cd /etc/sysconfig/network-scripts 中加入相对应的接口。 先查看我 4 个 10G 口的网卡名称:

[root@localhost network-scripts]# ifconfig -a br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.16.95 netmask 255.255.255.0 broadcast 192.168.16.255 inet6 fe80::8ad7:f6ff:fec4:216f prefixlen 64 scopeid 0x20<link> ether 88:d7:f6:c4:21:6f txqueuelen 1000 (Ethernet) RX packets 2707 bytes 189894 (185.4 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 532 bytes 91021 (88.8 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet6 fe80::8ad7:f6ff:fec4:216f prefixlen 64 scopeid 0x20<link> ether 88:d7:f6:c4:21:6f txqueuelen 1000 (Ethernet) RX packets 3452 bytes 462497 (451.6 KiB) RX errors 0 dropped 10 overruns 0 frame 0 TX packets 547 bytes 95489 (93.2 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 device interrupt 16 memory 0xdf400000-df420000 enp1s0f0: flags=4098<BROADCAST,MULTICAST> mtu 1500 ether 68:91:d0:61:be:cc txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) 2 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 enp1s0f1: flags=4098<BROADCAST,MULTICAST> mtu 1500 ether 68:91:d0:61:be:cd txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) 3 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 enp4s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet6 fe80::216:31ff:fef2:4942 prefixlen 64 scopeid 0x20<link>
 ether 00:16:31:f2:49:42 txqueuelen 1000 (Ethernet)
 RX packets 1 bytes 78 (78.0 B)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 0 bytes 0 (0.0 B)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 enp4s0f1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet6 fe80::216:31ff:fef2:4943 prefixlen 64 scopeid 0x20<link> ether 00:16:31:f2:49:43 txqueuelen 1000 (Ethernet) RX packets 1 bytes 78 (78.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) 在 cd /etc/sysconfig/network-scripts 中要一一对应;如果没有对应需要手动添加: llocalhost ~ Alocalhost r -br0 -enp0s31f6 -enp1s0f0 -enp1s0f1 -enp4s0f0 st ~]# cd /etc/sysconfi st network-scripts]# 1s ifcfg-enp4s0f1 ifdd ifdown ifddown ifdown-bnep ifdd o ifdown-eth ifdd st network-scripts]# ■ a/network-scripts/ own-ib own-ippp wn-ppp wn-routes wn-sit wn-Team wn-TeamPort p-Team p-TeamPort p-tunnel network-functions network-functions-ipv6 ifup-aliases ifup-bnep ifup-eth ifup-tunnel ifup-wireless init.ipv6-global ifup-isdn

这是我的四块 10G 网卡, 我用到的是 ifcfg-enp4s0f0 和 ifcfg-enp4s0f1 如果这个表中没有这 两个是需要添加的添加的内容为:



6.2 成功运行后设置 ip 地址



6.3 登陆界面

€ → ሮ ŵ	(i) 🔏 192.168.16.93/index.html#navigate/Login	靈 … ☆	¥ IN 🖸 🌣 🌩 🗏
文記記形SuperNova影列网	9495600		帮助 关于
	〇 阿淵科技 SuperNova-KVM visition		
	用户登录		
	🛓 admin		
		_	

6.4 配置用例并成功运行

Notitest.com	例 对象 🖄 报告 系统	admin		V19.03.0	SuperNova-
除: <u>HttpCps_TP_admin_20190115-14:2</u>	29.34 用例类型: HTTP新建	运行结果: ●成功 [查看历史]			0 显示张目 才 再次
· 有户选 服务器	报文捕获 用例取速				
TCP 关键结果					
	客户端	日起集计		服务器	日经停止
应用层			应用层	OK75 HH	Links of all
	砂鐵	总数			
TP_新建数量	0	221,765,336			
TP_请求数量	0	221,765,336			
TP_国应码2xx	0	221,765,336			
TP_请求超时	0	0			
TP_财延微粉		195			
转输层			传输层		
	砂值	总数		砂痕	总额
Pv4_并发数量	0		UDPv4_并获数量	0	
Pv4_主动新建连接	0	221,765,336	TCPv4_主动新建连接	0	(
Pv4_主动建成连接	0	221,765,336	TCPv4_主动建成连接	0	(
Pv4_被动建成连接	0	0	TCPv4_被动建成连接	0	221,765,336
Pv4_FIN关闭成功		221,765,336	TCPv4_FIN关闭成功		221,765,336
Pv4_主动Reset关闭		0	TCPv4_主动Reset关闭		0
Pv4_被动Reset关闭		0	TCPv4_被动Reset关闭		0
Pv4_发送超时次数		0	TCPv4_发送超时次数		C
网络层			网络层		
	砂值	80 B		砂塘	6B
文发送	0	887,061,747	报文发送	0	665,296,411
文接收	0	665,296,411	报文接收	0	887,061,747
國政 選	0	887,061,745	单播发送	0	665,296,011
墨接收	0	665,296,011	单播接收	0	887,061,745
唐 秋送	0	0	多播发送	0	0
	0		ALTERNA IN		

7 使用命令行方式部署虚拟机

7.1.创建 sriov 文件夹

[root@localhost home]# cd /home [root@localhost home]# mkdir sriov 进入到 home 目录下 创建 sriov 文件夹

[root@localhost ~]# cd /home [root@localhost home]# mkdir sriov [root@localhost home]# ls admin CLD sriov [root@localhost home]#

8.2 讲入该目录上传镜像

NOVA_VM_CLD-HW01-v22.12.11-build2907-20230225.img.cloud.zip

[root@localhost ~]# cd /home/sriov

进入到/home/sriov 文件夹下

[root@localhost sriov]# rz

上传镜像文件

[root@localhost sriov]# rz rz waiting to receive. Starting zmodem transfer. Press Ctrl+C to cancel. Transferring NOVA_VM_CLD-HW01-v22.12.11-build2907-20230225.img.cloud.zip... 100% 604313 KB 10242 KB/sec 00:00:59 0 Errors

[root@localhost sriov]# unzip NOVA_VM_CLD-HW01-v22.12.11-build2907-

20230225.img.cloud.zip unzip 命令解压镜像

[root@localhost sriov]# unzip NOVA_VM_CLD-HW01-v22.12.11-build2907-20230225.img.cloud.zip Archive: NOVA_VM_CLD-HW01-v22.12.11-build2907-20230225.img.cloud.zip inflating: data.qcow2 inflating: boot.qcow2 [root@localhost sriov]#

注意:从 23.03 版本开始需要手动创建数据盘

用 23.03 以后版本解压出的文件为 README 和系统盘 boot. qcow2

[root@localhost sriov]# unzip NOVA_VM_CLD-HW01-v23.03.06-build3094-20230321-x86_64.img.cloud.zip Archive: NOVA_VM_CLD-HW01-v23.03.06-build3094-20230321-x86_64.img.cloud.zip inflating: README inflating: boot.gcow2 [root@localhost sriov]#

README 内容为创建数据盘命令

[[root@localhost sriov]# cat README [Create 40G empty disk by following command:

#[qemu-img create -f qcow2 -o preallocation=full ./data.qcow2 40G]执行创建数据盘命令

[root@localhost sriov]# gemu-img create -f gcow2 -o preallocation=full ./data.gcow2 40G
Formatting './data.gcow2 , fmt=qcow2 size=42949672960 encryption=off cluster_size=65536 preallocation='full' lazy_refcounts=off
[root@localhost sriov]#
[root@localhost sriov]# 1s
boot.gcow2 [data.gcow2] MOVA_VM_cLD-Hw01-v23.03.06-build3094-20230321-x86_64.img.cloud.zip README
[root@localhost sriov]#

8.3. 创建 sriov 的池并启动

[root@localhost sriov]# virsh----进入 virsh 工具 virsh # pool-define-as sriov dir --target '/home/sriov/' -----定义池 sriov virsh # pool-build sriov-----构建池 sriov virsh # pool-start sriov------ 启动池 sriov virsh #pool-autostart sriov-----池 sriov 标记为自动启动 virsh # pool-list --all ------查看池运行状态 virsh #pool-refresh sriov------刷新池

virsh # virsh # pool-define-as sriov dir --target '/home/sriov/' 定义池 sriov virsh # pool-build sriov 构建池 sriov virsh # pool-start sriov 池 sriov 己启动 virsh # pool-autostart sriov 池 sriov 标记为自动启动 virsh # pool-list --all 状态 自动开始 名称 _____ cld-1 活动 否 日是是是是 novavm 活动 pass-through 活动 sriov 活动 win02 活动 virsh # pool-refresh sriov 池 sriov 被刷新

8.4.分离网卡

8.4.1 ifconfig 查看可用网卡名称

enp101s0f0: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500 ether 0c:42:a1:49:94:44 txqueuelen 1000 (Ethernet) RX packets 579 bytes 71796 (70.1 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 590 bytes 72824 (71.1 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 enp101s0f1: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500 ether 0c:42:a1:49:94:44 txqueuelen 1000 (Ethernet) RX packets 579 bytes 71796 (70.1 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 581 bytes 72044 (70.3 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

8.4.2 命令含义为将实体网卡 enp101s0f0 虚拟出一个 VF; echo 2 就是虚拟出 2 个 VF; echo 1 > /sys/class/net/enp101s0f0/device/sriov_numvfs echo 1 > /sys/class/net/enp101s0f0/device/sriov_numvfs

[root@localhost~]# lspci |grep Ethernet-----查看虚拟出来的网卡

[[root@localhost ~]# lspci grep Ethernet	
19:00.0 Ethernet controller: Intel Corporation Ethernet Connection X722 for 10GBASE-T (rev	09)
19:00.1 Ethernet controller: Intel Corporation Ethernet Connection X722 for 10GBASE-T (rev	09)
65:00.0 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]	
65:00.1 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]	
65:00.2 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Funct	ion]
65:01.2 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Funct	ion]
[b3:00.0 Ethernet controller: Mellanox Technologies MT2/800 Family [ConnectX-5]	
b3:00.1 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]	

8.4.3 分离虚拟网卡

virsh nodedev-detach pci_0000_65_00_2

virsh nodedev-detach pci_0000_65_01_2

[root@localhost ~]# virsh nodedev-detach pci_0000_65_00_2 已分离设备 pci_0000_65_00_2

[root@localhost ~]# virsh nodedev-detach pci_0000_65_01_2 己分离设备 pci_0000_65_01_2

注网卡分离和取消分离的命令为

virsh nodedev-detach pci_0000_网卡	分离网卡
virsh nodedev-reattach pci_0000_网卡	取消分离

8.5.上传虚拟机的 xml 文件



[root@localhost ~]# cd /etc/libvirt/qemu/

rz 上传 sr-iov.xml

```
[root@localhost qemu]# rz
rz waiting to receive.
Starting zmodem transfer. Press Ctrl+C to cancel.
Transferring sr-iov.xml...
100% 4 KB 4 KB/sec 00:00:01 0 Errors
修改文件内容调整虚拟机配置
```

[root@localhost ~]# vim /etc/libvirt/gemu/sr-iov.xml



8.6 启动虚拟机

[root@localhost qemu]# virsh define sr-iov.xml ----导入虚拟机配置 [root@localhost qemu]# virsh start sr-iov-----启动虚拟机 [root@localhost ~]# virsh list -----查看虚拟机运行状态 [root@localhost qemu]# virsh console sr-iov----连接到虚拟机,登陆用 admin/admin.

```
      [root@localhost ~]# virsh define novavm.xml]

      定义域 novavm (从 novavm.xml)

      [root@localhost ~]# virsh start novavm

      域 novavm 已开始

      [root@localhost ~]# virsh list

      Id
      名称

      6
      novavm

      6
      novavm

      定oot@localhost ~]# virsh console novavm

      连接到域 novavm

      按码行为 ^]

      kvm: no hardware support

      Loading, please wait...

      mount: no /proc/mounts

      Detected USB Boot Flash /dev/sda, Log Disk /dev/sdb

      Supernova is formatting log disk ...

      Error: /dev/sdb: unrecognised disk label

      sh: 32.2: bad number

      Make msdos lable on /dev/sdb ... Done

      Delete all partition on /dev/sdb ... Done

      Create Swap partition on /dev/sdb ... Done

      Format log disk /dev/sdb2 ... Done

      mount --make-shared /mnt/huge/ ... Done

      mount --make-shared /mnt/huge/ ... Done

      mount --t hugetlbfs nugetlbfs /dev/hugepages ... Done

      Model: supernova-KVM

      Serial Number: UnknownSN

      Database root user is not exists, creating users!

      Datebase successfully.

      Initialize database successfully.

      Initialize database successfully.
```

8.7 登陆虚拟机后配置虚拟机 ip 和网关

Supernova # interface mgmt1 Supernova interface mgmt1 # ip address 192.168.18.205 255.255.255.0 Supernova interface mgmt1 # end Supernova # unset ip route 0.0.0.0/0 192.168.1.1 mgmt1 Supernova # ip route 0.0.0.0/0 192.168.18.1 mgmt1 default

Supernova login: admin Password: welcome ! For interactive help, Please type "show running-config". For interactive help, Please typ Supernova # show running-config show running-config show system status show interface firmware show hardware info show system interface show network route show system setting show system memsize show timezone help interface mgmt1 interface mgmt2 ip route unset ip route hostname ping date <mm/dd/yyyy hh:mm:ss Zone> reboot shutdown factoryreset formatlogdisk exit show running-config. System status. Interface firmware version. Hardware information. Show network interfaces and configurations. Show wetwork interfaces and configurations. Show system setting. Show memory total size. Display support time zone options. Config mgmt1 Config mgmt1 Config mgmt2 Config route Delete route Config hostname PING command. Set date. Set date include time zone. (Use: show timezone help) Reboot SuperNova. Factoryreset SuperNova. Factoryreset SuperNova. Format storage. Exit the CLI. show running-config. exit network config usage: interface mgmt1 ip address 192.168.1.99 255.255.255.0 end interface mgmt2 ip address 10.1.1.99 255.255.255.0 end route config usage: ip route 0.0.0.0/0 192.168.1.1 mgmt1 default ip route 10.1.0.0/16 10.1.1.1 mgmt2 unset ip route x.x.x.0/24 x.x.x.1 mgmt1 hostname config usage: hostname <string> 配置ip和网关后,可以在浏览器直接登 web server port usage: http port 80 https port 443 陆测试仪了 Supernova # interface mgmt1 Supernova interface mgmt1 # ip address 192.168.18.205 255.255.0 Supernova interface mgmt1 # end Supernova # unset ip route 0.0.0.0/0 192.168.1.1 mgmt1 Supernova # ip route 0.0.0.0/0 192.168.18.1 mgmt1 default Supernova #

8.8 通过 web 登录测试仪

1 0 - 1	92.168.1	8.205/in	ndex.html#	navigate/Lo	gin					
\$用网址	⊕天猫	⊕ 微博	⊕ 爱淘宝	⊕ 携程旅行	⊕ Index of /nova_imag	📈 网址大全	JD 京东商城	🕀 京东商城		
						31+5				
					MJ/KJ4 netites	41X t.com			Supernova-KVM	
									V22 09 08	
							2	用户登录		
							Í	admin		
						>				
									登录	

9. 附加: VNC 配置方法

9.1 安装软件包

命令: yum check-update

yum install -y tigervnc-server

yum groupinstall "X Window System"

yum groupinstall "fonts"

yum install gnome-classic-session gnome-terminal nautilus-open-terminal control-center liberation-mono-fonts

unlink /etc/systemd/system/default.target

In -sf /lib/systemd/system/graphical.target /etc/systemd/system/default.target

9.2 关闭防火墙

命令: systemctl stop firewalld systemctl disable firewalld systemctl status firewalld ----查看防火墙状态

9.3 复制配置文件

命令: cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:1.service

9.4 编辑复制出来的配置文件

命令: vi /etc/systemd/system/vncserver@:1.service



9.5 重新加载配置文件

命令: systemctl daemon-reload

9.6 设置 VNC 密码

命令: vncpasswd root

9.7 开启 VNC 并设置成开机启动

命令: systemctl start vncserver@:1.service systemctl enable vncserver@:1.service

9.8 启动

命令: vncserver

9.9 如果启动成功端口是监听状态(VNC 端口号默认 5900+1)

命令: netstat -an |grep 5901

[root@localh	ost ~j#	netstat -an	grep 5901		
tcp 0	0 0	0.0.0.0:5901		0.0.0.0:*	LISTEN
tcp6 0	0	:::5901		*	LISTEN
[root@localh	ost ~]#				

9.10 vnc 客户端连接

V2 VNC Viewer	– 🗆 X
VNC® Viewer	NS
VNC Server: 192.168.16.95:1	~
Encryption: Let VNC Server choose	~
<u>A</u> bout <u>O</u> ptions	Connect
VNC Viewer	– 🗆 X
V2 VNC Viewer - Authentication	
VNC Server: 192.168.16.95::5901	C
VN Username:	~
Enc Password:	~
ОК	Cancel
344	

🛿 localhost.localdomain:1 (root) - VNC Viewer 🦳 🗆	×
	() ()
이 이 것은 것은 이렇게 있는 Nation 이 있는 것은 것은 것이 있는 것을 했다.	
「「「」」「「」」」「「」」」「」」」「「」」」「「」」」」「「」」」」」「「」」」」	
以另一个用户身份登录	

查看运行状态

systemctl status vncserver@:.service
systemctl is-enabled vncserver@.service

vncserver 启动失败的一种情况

#vncserver

Warning: localhost.localdomain:1 is taken because of /tmp/.X11unix/X1 Remove this file if there is no X server localhost.localdomain:1

Warning: localhost.localdomain:2 is taken because of /tmp/.X11unix/X2 Remove this file if there is no X server localhost.localdomain:2

New 'localhost.localdomain:3 (root) ' desktop is localhost.localdomain:3

Starting applications specified in /root/.vnc/xstartup Log file is /root/.vnc/localhost.localdomain:3.log 解决办法:

删除 / tmp/.X11-unix/的内容

重启 vnvserver

systemctl restart vncserver@:2.service

Linux 删除文件夹和文件的命令

-r 就是向下递归,不管有多少级目录,一并删除 -f 就是直接强行删除,不作任何提示的意思

删除文件夹实例:

rm -rf /var/log/httpd/access

将会删除/var/log/httpd/access 目录以及其下所有文件、文件夹

删除文件使用实例: rm -f /var/log/httpd/access.log 将会强制删除/var/log/httpd/access.log 这个文件