新支点虚拟化管理软件 uVirt™ 用户手册

这是新支点虚拟化管理软件 uVirt™ 的用户手册，适合于开发人员、测试人员和运维人员阅读。本手册共分为产品说明、安装指南和使用手册三部分。

# 产品说明

## 简介

新支点虚拟化管理软件uVirt™ 是企业级虚拟化平台管理软件，它是行业云技术解决方案中 IaaS 层的关键产品。

uVirt 支持主流 CPU 虚拟化技术，高度适配龙芯、鲲鹏、飞腾、海光、兆芯和申威等国产 CPU，特别适合构建信创虚拟化平台。uVirt 深度集成操作系统虚拟化，支持轻量虚拟机的创建和管理，其性能表现近似于物理机。uVirt 深度融合软件定义存储和软件定义网络，统一管理计算、存储和网络等虚拟化云基础设施资源。uVirt 提供图形用户界面、命令行工具集和应用编程接口（API）三种使用方式，满足不同层次的用户需求。

## 术语

|  |  |
| --- | --- |
| **术语** | **说明** |
| 物理服务器 | 用户购买的服务器机器。uVirt 支持主流 CPU 架构，包括龙芯、海光、鲲鹏、飞腾、兆芯、申威等国产 CPU 。uVirt 将在物理服务器上安装 Debian GNU/Linux 操作系统。 |
| 传统虚拟机 | 基于 QEMU + KVM 创建的虚拟机，有时也简称为虚拟机。 |
| 轻量虚拟机 | 基于 LXC 创建的轻量虚拟机，简称为「轻虚机」或者「系统容器」。轻虚机的性能表现近似于物理机。 |
| uVirt 节点 | 安装了 uVirt 的单台物理服务器，用户按需管理该服务器上的所有虚拟机或轻量虚拟机。 |
| uVirt 节点集群 | 由多个 uVirt 节点构成的对等集群。用户访问任何一个节点，都可以管理所有节点的虚拟机或者轻量虚拟机。 |
| 本地存储 | 直接安装在服务器上的存储资源 |
| 外部存储 | 从外部提供给服务器使用的存储资源 |
| 系统存储 | 保存 uVirt 系统的存储资源 |
| 虚拟机存储 | 保存虚拟机或者轻虚机的存储资源 |
| 管理网络 | uVirt 系统组件之间通信、用户访问 uVirt 管理界面使用的网络 |
| 虚拟机网络 | 虚拟机通信使用的网络 |

备注：本地存储和外部存储的区分依据是存储来源，而系统存储和虚拟机存储的区分依据是存储用途。

## 部署架构

### 单节点架构

单节点架构部署（见图 1）是指在一台物理服务器上安装和运行 uVirt。成功部署后，这台服务器就成为一个 uVirt 节点。

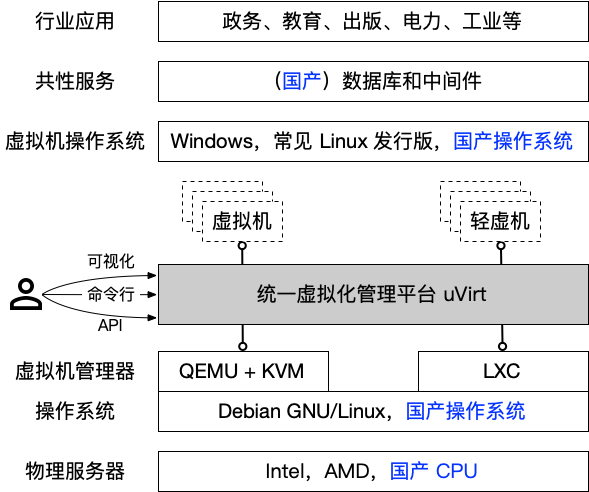
这是最简单的 uVirt 部署方式，也是部署 uVirt 集群的前提和基础。单节点架构部署的使用场景主要包括 uVirt 的试用、学习、培训和开发等。

### 对等集群架构

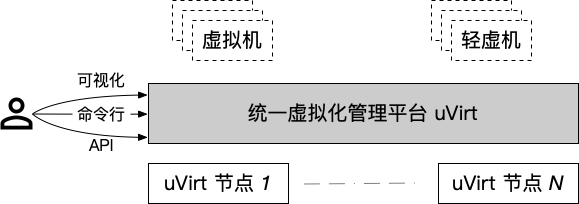
对等集群架构（见图 2）是指多个 uVirt 节点共同构成一个地位对等的 uVirt 集群。用户访问其中任意一个节点，都能够管理所有节点的虚拟机或者轻虚机。

用户首先在每台服务器上按照单节点架构方式部署 uVirt；然后调用 uVirt 的集群管理命令，创建一个集群，并将所有 uVirt 节点添加到集群中。

建议在生产环境中采用对等集群架构部署方式部署，保证整个系统的高可用性。



**图1 单节点架构部署 uVirt**



**图2 部署 uVirt 集群**

## 主要优势

### 支持国产 CPU 虚拟化

uVirt 支持主流 CPU 虚拟化技术，完美适配国产 CPU（见表 1）和服务器，支持统信 UOS、中标麒麟、银河麒麟等国产操作系统；支持武汉达梦、人大金仓、神通等国产数据库；支持东方通、金蝶天燕、中创软件、宝兰德、华宇软件等国产中间件。uVirt 和上述国产软硬件产品共同构成安全可控的信创整体解决方案。

**表1 国产CPU 支持**

|  |  |  |
| --- | --- | --- |
| **指令集架构** | **厂商** | **CPU 型号** |
| x86 | 海光 | 全部系列 |
| 兆芯 |
| arm | 华为 | 鲲鹏 916/920 |
| 飞腾 | 飞腾 2000+ |
| MIPS / 自研 | 龙芯 | 龙芯 3B4000 |
| 自研 | 申威 | 申威 1621 |

### 集成操作系统虚拟化技术

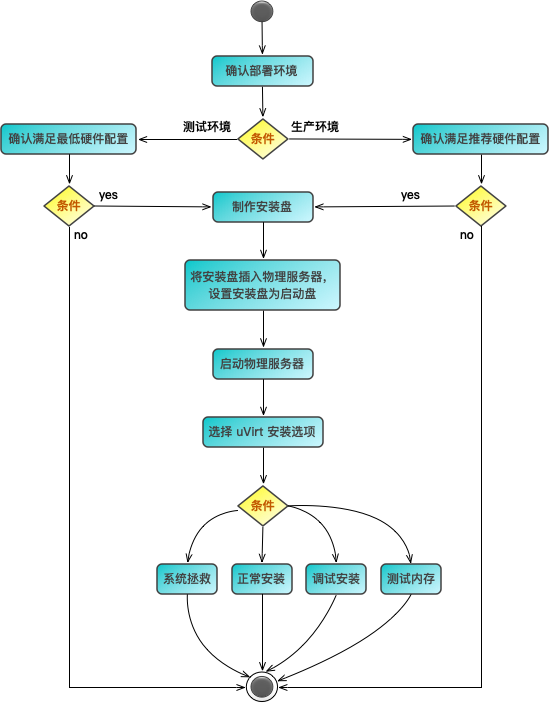
uVirt 集成操作系统虚拟化技术，支持轻量虚拟机的创建和管理。轻量虚拟机的性能表现近似于物理机，适合追求极致性能的应用场景。

### 融合软件定义存储及网络

uVirt 深度融合软件定义存储和软件定义网络技术，统一管理计算、存储和网络等虚拟化云基础设施。

uVirt 具有「开箱即用、操作简单」的特点。即使是一个初学者，也能够在较短时间内学会 uVirt 的安装和使用。普通用户使用浏览器访问基于 Web 的管理界面，可视化完成所有的虚拟化管理操作，完全匹配 VMWare 用户的使用习惯。系统管理员使用 uVirt 命令行工具集，批量完成虚拟化管理操作。开发人员调用 uVirt 应用编程接口，实现 uVirt 功能的定制、集成或扩展。

# 安装指南



**图3 安装流程**

安装 uVirt 的一般流程（如图 3所示）包括下列步骤：

1. 确定物理服务器的硬件配置是否满足安装需求：
   1. 试用 uVirt，至少要满足最低硬件配置；
   2. 在生产环境中使用 uVirt，至少要满足推荐硬件配置。
2. 制作 uVirt 安装盘，包括光盘或者 USB 盘两种形式；
3. 将 uVirt 安装盘插入物理服务器，设置安装盘为系统启动盘；
4. 启动物理服务器，选择 uVirt 安装选项；
5. 按照安装向导的提示，一步一步完成安装；
6. 安装成功后，拔下安装盘，重新启动服务器；
7. 等待系统就绪，用户开始管理或使用 uVirt。

## 硬件配置要求

建议用户在预算许可的前提下，尽可能提高 uVirt 节点的硬件配置。一旦物理服务器发生硬件故障，它运行的所有虚拟机都会宕机，甚至发生无法重启的灾难事故。这需要在单机硬件层面保障服务器的正常运行和灾难恢复。用户还可以利用 uVirt 的集群管理功能，集中管理多台物理服务器，通过备份和恢复，保证系统的高可用性。

**表2 硬件配置要求**

|  |  |  |
| --- | --- | --- |
|  | **最低配置** | **推荐配置** |
| **CPU** | 支持完全虚拟化 64 位 CPU | 支持完全虚拟化的 64 位 CPU；  PCI(e)直通要求支持 VT-d/AMD-d 的 CPU |
| **内存** | 1GB + 运行虚拟机所需内存 | 2GB + 运行虚拟机所需内存 + （可选）运行 ZFS 文件系统所需内存，至少 8GB；  ECC 内存 |
| **存储** | 1 个 32GB 以上容量的硬盘 | 高性能（最好是 SSD 盘）且支持冗余的本地存储：   * 带电池保护写缓存的硬件 RAID   或者   * 带软件 RAID 支持的 ZFS 文件系统 |
| **网卡** | 1 块网卡 | 多块千兆以上网速的网卡。网卡数量取决于存储和 uVirt 集群通信的需求。 |

注意：本地存储的定义，请参见「术语」部分。

如果用户想了解服务器 CPU 和磁盘的大致性能，uVirt 提供了专门的命令行工具 uperf。

## 制作安装盘

uVirt 安装光盘镜像（ ISO 格式）包含下列内容：

* + 一个完整的 64 位 Debian GNU/Linux 操作系统；
  + 安装程序：将本地磁盘分区，格式化为指定的文件系统，安装操作系统；
  + uVirt 所有组件及其依赖：包括管理 uVirt 节点、集群、虚拟机的命令行工具集和 Web 界面。

用户既可以将 ISO 镜像烧录到光盘上，也可以直接复制到闪存介质（ USB 盘）上。推荐使用 USB 盘安装 uVirt。

### 下载 ISO 镜像

从工程中心官方网站下载或者从分销商处获取 uVirt ISO 镜像，假设文件名是 uvirt.iso 。

### 制作 USB 安装盘

如果用户制作 USB 安装盘所使用的计算机系统是 Windows, macOS 或者某个 Linux 发行版的桌面版，推荐使用 [balenaEtcher](https://www.balena.io/etcher/) 工具。具体步骤包括：

* + 下载并安装 balenaEtcher；
  + 将全新 USB 盘插入计算机；
  + 选择源 ISO 文件和目标 USB 盘，点击「复制」按钮，等待复制过程完成。

某些用户希望使用命令行创建 USB 安装盘。在 Linux 系统下，

* + 将 U 盘插入计算机，执行 lsblk 命令找出 U 盘的设备名，假设为 /dev/XYZ ；
  + 将 ISO 镜像复制到 U 盘：

dd if=uvirt.iso of=/dev/XYZ bs=1M

## 启动服务器

将 USB 安装盘插入物理服务器。

服务器开机通电后，选择进入服务器的 BIOS 或 UEFI 界面，设置 USB 安装盘作为首选启动盘，保存设置后退出。

重新启动服务器，直到出现 uVirt 安装选项界面（图4）。



欢迎使用 uVirt 虚拟化管理软件

正常安装 uVirt

调试安装 uVirt

启动急救模式

测试内存

**图4 安装选项界面**

## 选择安装选项

uVirt 初始安装界面提供 4 个安装选项：

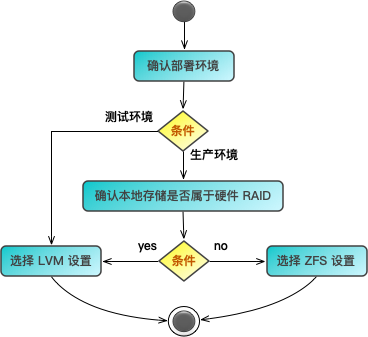
1. 正常安装：启动正常安装程序；
2. 调试安装：以调试模式启动安装程序。在执行几个关键安装步骤时，安装启动一个控制台窗口，方便用户检查调试错误。按 Ctrl+D 组合键，退出调试模式。也可以选择本选项，启动一个包含所有基本工具的实时系统，修复已有 uVirt 系统的错误。例如，修复一个降级的 ZFS 存储池或者 uVirt 系统的引导程序等；
3. 拯救启动：启动一个已有的 uVirt 系统。安装程序将搜索服务器的所有硬盘。如果找到某个硬盘上已经安装 uVirt 系统，安装程序使用 ISO 镜像中的 Linux 内核，从该硬盘启动。选择本选项，通常是因为 BIOS 无法读取 uVirt 系统盘的引导块（grub）或者引导块已经损坏；
4. 测试内存：运行 memtest86+。检测服务器内存是否功能正常。

## 正常安装

选择「正常安装」选项，开始安装过程。

### 选择目标硬盘

选择安装 uVirt 系统的目标硬盘，指定目标硬盘的文件系统类型。



**图5 选择目标硬盘设置类型**

如果是在生产环境服务器安装 uVirt，并且服务器本地存储的类型不属于带电池保护写缓存的硬件 RAID，请选择 ZFS 设置；其他情况选择默认的 LVM 设置。

**注意：**ZFS 设置不能与硬件 RAID 一起使用！

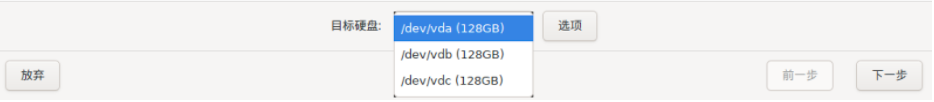
#### LVM 设置

从下拉菜单中选择一块本地硬盘作为安装 uVirt 系统的目标硬盘，即系统存储。



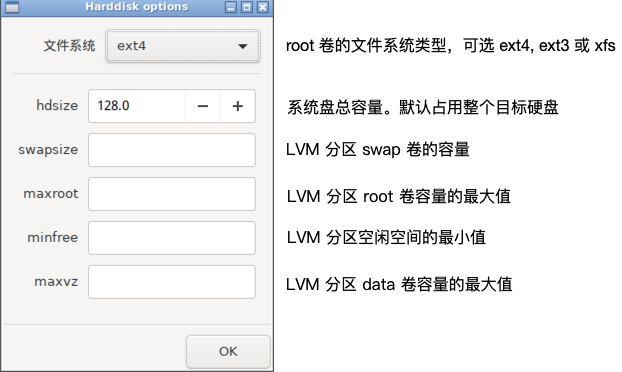
新支点虚拟化管理软件 uVirt

|  |  |
| --- | --- |
| **新支点虚拟化管理软件安装程序**将自动对评判分区，并安装所有的软件包，最后将硬盘设置为可启动。请注意：所有已经存在的分区和数据都会丢失。  点击 **下一步** 继续安装。 | * **请验证安装的目标盘**   将在下面的硬盘上安装系统。  警告：所有现有分区和数据都将丢失   * **自动分区检测**   安装程序将自动设置您的硬件。   * **图形化用户界面**   最终的配置将通过 web 浏览器访问图形用户界面完成。 |



**图5 选择目标硬盘**

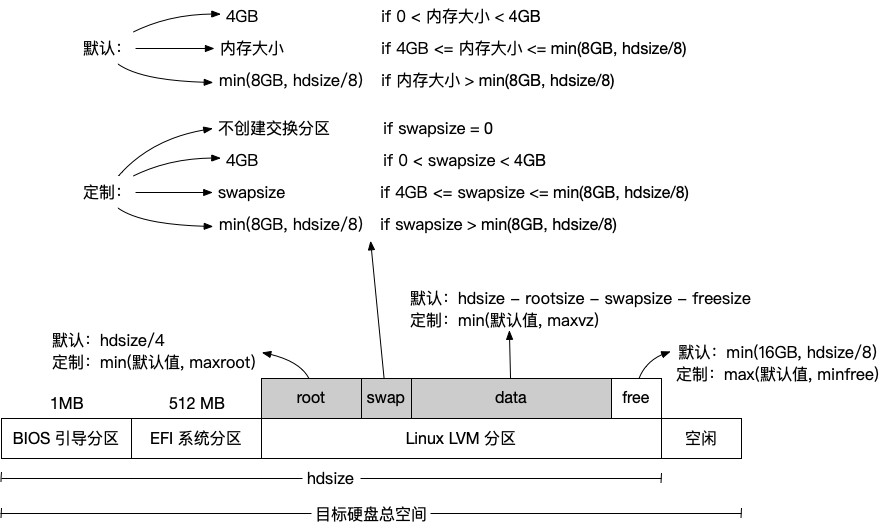
单击「选项」按钮，用户可以修改默认的 LVM 设置。



**图6 LVM 配置选项**

安装程序将系统盘划分为三个分区（见图 7）：

1. BIOS 引导分区：采用 BIOS 固件接口的服务器，从这里查找并启动 grub 引导程序。分区的大小为 1MB；
2. EFI 系统分区：采用 UEFI 固件接口的服务器，从这里查找并启动 systemd-boot 引导程序。分区的大小为 512MB；
3. Linux LVM 分区：将系统盘全部剩余空间划分为一个 Linux LVM (Logic Volume Manager, 逻辑卷管理器) 类型的分区。该分区将添加到新建的名为 uvirt 的卷组，然后创建三个逻辑卷：
   1. root: 默认格式化为 EXT4 文件系统，装载为操作系统的根目录；
   2. swap: 操作系统的交换分区；
   3. data: 一个 LVM-thin 类型的存储池。主要用途是存储虚拟机镜像，因为它能够有效地支持虚拟机镜像的快照和克隆功能。



**图7 LVM 配置选项的含义及计算公式**

LVM 配置选项决定了三个 LVM 逻辑卷的实际大小：

* + root 卷容量默认为 hdsize/4。如果设置了 maxroot，则实际容量是默认值和 maxroot 之间的最小值；
  + swap 卷容量默认值根据服务器内存大小计算。如果内存容量小于 4GB，则取值为 4GB；如果内存容量高于 8GB 和 hdsize/8 之间的最小值，则取这个最小值；其他情况，取值为内存大小。如果用户设置 swapsize 为 0，则不会创建交换分区；对于其他 swapsize 取值，采用与默认情况类似的逻辑判定 swap 卷的容量大小；
  + LVM 分区空闲空间的默认大小是 16GB 和 hdsize/8 二者之间的最小值。如果设置了 minfree，则取默认值和 minfree 之间的最大值；
  + data 卷的默认容量为 LVM 分区的全部剩余空间大小。如果用户设置了 maxvz，则取默认值和 maxvz 之间的最小值。

#### ZFS 设置

如果服务器的本地存储并不是带电池保护写缓存的硬件 RAID，建议采用 ZFS 配置实现软件 RAID。

单击「选项」按钮，从对话框的「文件系统」下拉框中选择 ZFS 及对应的 RAID 级别，包括RAID0/1/10 和 RAIDZ-1/2/3 共 6 种选项。接下来，选择构成软件 RAID 的硬盘配置，还可以调整 ZFS 高级选项。



**图8 ZFS 设置**

安装程序将创建一个名为 rpool 的存储池，在 rpool/ROOT/uvirt-1 卷被装载为根目录，而 rpool/data 卷保存虚拟机镜像。

ZFS 是由 Sun Microsystems 公司（已被 Oracle 公司收购）设计开发的逻辑卷管理器和文件系统。OpenZFS 项目是 ZFS 项目的开源后继，支持illumos, FreeBSD, Linux, macOS, NetBSD 和 Windows 操作系统。uVirt 采用 OpenZFS 项目的 ZFS on Linux (ZoL) ，提供一系列的高级存储功能：

* 不同级别的软件 RAID；
* 兼具逻辑卷管理和文件系统功能；
* 文件系统层的有效数据压缩；
* 快照和写时复制（Copy-on-Write, CoW）的克隆；
* 持续的完整性检查和自动修复；
* ……

由此可见，ZFS 实现了前述 LVM 配置中硬件 RAID + LVM + ext4/xfs/ext3 的对应功能，还提供其他高级存储能力。要在生产环境中用好 ZFS，需要用户深入了解 ZFS 的工作原理，尤其是各种配置参数的含义。对于普通用户，建议选择硬件 RAID + LVM + ext4/xfs 的配置选项，简化系统存储的管理。

使用 ZFS 的一般流程是：首先使用 zpool 命令，创建由多个虚拟设备（Virtual Device, VDEV）组成的 ZFS 存储池；然后使用 zfs 命令，从存储池中分配资源，创建指定的逻辑卷或者文件系统。在写入用户数据时，ZFS 将综合考虑同一个存储池的所有 UDEV 的空闲空间占比和设备繁忙情况，动态切分数据，将不同大小的数据片段按顺序分发到不同的 UDEV。与常规 RAID0 相比，这种方式有两点明显不同：

1. 数据片段的大小不是固定的；
2. 并不是每一个 UDEV 都会参与数据片段的保存。

ZFS 存储池支持下列类型的虚拟设备：

* disk - 一个块设备，通常位于 /dev 下。建议使用整个磁盘，而不是磁盘的一个分区；
* file - 预分配的常规文件。只适用实验目的；
* mirror - 两个或以上设备组成的镜像，数据被同等复制到所有的设备上。由 N 个设备组成的镜像，可容忍 N-1 个设备失效；
* raidz, raidz1, raidz2, raidz3 - raidz1（或者简写为 raidz）至少需要三个盘，带一种校验位，能容忍一个盘失效；raidz2 至少需要四个盘，带两种校验位，能容忍两个盘失效；raidz3 至少需要五个盘，带三种校验位，能容忍三个盘失效。上述 RAID 类型克服了传统带校验的 RAID 类型（例如 RAID5 或 RAID6 ）有可能存在数据不一致的情况。为了保证性能，通常建议 raidz 类型 UDEV 包含的物理盘总数介于 3 和 9 之间。如果要使用更多的盘，建议拆分为几个 UDEV；
* spare - 当存储池中正在使用的其他类型虚拟设备失效时，启用一个 spare 类型的虚拟设备实时替换失效的虚拟设备。spare 虚拟设备可以被多个存储池分享；
* 其他类型的虚拟设备用于 ZFS 自身数据的存储，包括 log, dedup, special 和 cache。

**注意：**UDEV 只能是由物理设备组成，不能再包含其他 UDEV；一旦一个设备添加到某个 UDEV，就不能再被移除。采用 ZFS 配置安装 uVirt 时，主要涉及 disk, mirror 和 raidz 系列类型的 UDEV。

ZFS 高级选项的含义：

* ashit - 物理盘扇区大小的指数，合法取值范围是 9-16。如果设置为 0，则将自动检测。例如，物理盘扇区的大小为 4KB，因为 4096=212，所以应该将 ashift 设置为 12；
* compress - 是否开启（on|off）数据压缩。如果开启压缩选项，默认采用 lzjb 或者（如果开启了 lz4\_compress）lz4 算法。也可以明确设定为 lzjb 或者 lz4；
* checksum - 是否开启（on|off）数据完整性的验证。如果开启验证，默认采用 fletcher4 算法。也可以明确指定为 fletcher2、fletcher4 或 sha256；
* copies - 数据副本的个数，合法取值是 1|2|3。这在磁盘级别的冗余之外，又提供一种数据冗余；
* hdsize - 引导盘占用的磁盘空间大小。所谓引导盘是指 RAID0 或者 RAID1 中的第一个盘、RAID10 中的第一个镜像、RAIDZ-1/2/3 中的所有盘。

### 设置操作系统信息

设置操作系统的基本信息，包括国家、时区和键盘布局。通常情况下用户接受安装程序自动检测结果即可。

### 设置管理员信息

设置系统管理员（root 用户）的密码和电子邮件地址。

要求密码至少包含 8 个字符且由大小写字母、数字和特殊字符混合组成。避免使用重复字符、键盘位置连续的字符、单词或者连续的字母或数字序列。

系统管理员的电子邮箱将收到系统自动转发的重要信息，例如，软件包升级信息、定时调度任务的出错和报警信息等。

### 配置管理网络

设定 uVirt 管理界面监听的网卡及其设置，包括网卡名称、主机名、IP 地址、网络掩码、网管和 DNS 服务器。

**注意：**可以设定管理网卡的 IPv4 地址或者 IPv6 地址，但不能选择同时使用两种网址。

### 浏览安装配置

至此，已经完成 uVirt 系统的所有安装配置。用户单击「上一步」按钮，修改已有的安装配置。如果确认无误，单击「安装」按钮，开始正式安装过程。

### 安装 uVirt 系统

安装过程包括格式化硬盘、安装操作系统和 uVirt 组件等。成功安装后，取下 USB 安装盘，重新启动服务器。

**注意：**如果安装失败，按 CTRL+ALT+F2 组合键进入终端窗口，检视安装程序日志中的出错信息。

## 自定义安装

### 在已有系统安装 uVirt

所有的 uVirt 组件已经打包为 .deb 软件包。理论上，任何 Debian 或者 Ubuntu 系统及其衍生系统都能安装 uVirt，包括国产统信 UOS 和深度操作系统等。

首先为系统添加软件包仓库，然后执行：

sudo apt update

sudo apt install uvirt

实际上，uVirt 正常工作需要完备可靠的存储设置和网络设置。已有系统的配置往往并不满足最佳使用条件，因此在生产环境中，不推荐直接在已有系统安装 uVirt。如果用户确有此需求，请联系工程中心寻求专业技术支持和服务。

# 初始化配置

在成功安装 uVirt 后，通常需要用户先完成服务器系统的初始化配置，再开始使用 uVirt。

## 系统更新

安装 uVirt 后，管理物理服务器的操作系统是 Debian GNU/Linux。Debian 系统采用 apt 作为软件包管理工具，软件仓库来源定义在 /etc/apt/sources.list 文件或者 /etc/apt/sources.list.d 目录下面以 .list 为扩展名的文件中。

建议在 /etc/apt/sources.list 文件中添加一行：

deb https://repo.udoplus.com/debian/ buster uvirt-stable

为了保证软件包的安全，我们使用 GnuPG 对软件包仓库包含的所有 Release 文件进行加密和签名。在安装 uVirt 系统的过程中，签名密钥已经被安装到服务器系统，见 /etc/apt/trusted.gpg.d/uvirt.gpg。

执行系统更新操作的命令是：

apt update && apt dist-upgrade

## 存储管理

### 系统存储

#### LVM 管理

大部分用户都直接将新支点虚拟化软件安装到服务器本地硬盘。新支点虚拟化软件uVirt 的安装光盘为服务器本地硬盘配置提供了多种选项， 目前默认的选项为LVM。默认情况下，安装程序会引导你选择一块硬盘用于安装新支点虚拟化软件，并将该硬盘格式化为物理卷（PV） 的形式以创建卷组（VG） uvirt 。下面是在一个8GB容量的小硬盘上测试安装新支点虚拟化后的LVM配置输出：

# pvs

PV VG Fmt Attr PSize PFree

/dev/sda3 uvirt lvm2 a-- 7.87g 876.00m

# vgs

VG #PV #LV #SN Attr VSize VFree

uvirt 1 3 0 wz--n- 7.87g 876.00m

安装程序会在VG上创建三个逻辑卷（LV）：

# lvs

LV VG Attr LSize Pool Origin Data% Meta%

data uvirt twi-a-tz-- 4.38g 0.00 0.63

root uvirt -wi-ao---- 1.75g

swap uvirt -wi-ao---- 896.00m

**root**

格式化为ext4文件系统，用于安装新支点虚拟化软件 uVirt。

**swap**

用于swap分区

**data**

使用 LVM-thin ，用于保存虚机镜像。 LVM-thin 在对于虚机快照和克隆的效率较高，所以非常适合用于保存虚机镜像。

安装程序会以 LVM-thin 形式创建 “data” 逻辑卷，并用于存储基于块存储的虚机镜像，而 /var/lib/vz 仅是根文件系统下的一个路径。

##### 硬件

我们强烈推荐使用基于硬件RAID控制器（带有电池保护写缓存）的服务器安装新支点虚拟化软件 uVirt。这样不仅能提高性能， 提供冗余性，还能简化故障硬盘的更换操作（可热插拔）。

LVM本身无须任何特殊的硬件支持，并且对于内存的需求也很低。

##### 启动引导程序

默认情况下，安装程序会安装两个启动引导程序。第一个分区会安装标准的GRUB启动引导程序。第二个分区会格式化为EFI系统分区 （EFI System Partition，ESP），这样就可以启动基于EFI的系统。

##### 创建卷组

假定我们现在有一块空白磁盘设备，其设备文件为 /dev/sdb ，可以用如下方法利用该磁盘创建卷组 “vmdata” 。

执行以下这些命令将会彻底销毁 /dev/sdb 上的原有数据。

首先创建一个分区。

# sgdisk -N 1 /dev/sdb

然后使用静默方式创建一个物理卷（PV），并设置元数据大小为250K。

# pvcreate --metadatasize 250k -y -ff /dev/sdb1

然后在 /dev/sdb1 上创建卷组 “vmdata” 。

# vgcreate vmdata /dev/sdb1

##### 为 /var/lib/vz 添加LV

可用如下命令添加 thin 逻辑卷（thin LV）。

# lvcreate -n <Name> -V <Size[M,G,T]> <VG>/<LVThin\_pool>

示例如下：

# lvcreate -n vz -V 10G uvirt/data

然后在该逻辑卷上创建文件系统，如下。

# mkfs.ext4 /dev/uvirt/vz

最后，将设备挂载到 /var/lib/vz 。

注意：挂载前需确保 /var/lib/vz 下为空。默认 /var/lib/vz 不为空。

为确保系统重启后自动挂载，可运行如下命令来修改 /etc/fstab 文件的配置内容。

# echo '/dev/uvirt/vz /var/lib/vz ext4 defaults 0 2' >> /etc/fstab

##### 调整thin pool的容量

可以用如下命令调整LV容量大小和元数据池大小。

# lvresize --size +<size[\M,G,T]> --poolmetadatasize +<size[\M,G]> <VG>/<LVThin\_pool>

**注意：**扩展数据池容量的同时，必须相应扩展元数据池的容量。

##### 创建LVM-thin存储池

首先要创建一个卷组(VG)，然后才能创建 thin pool 。可以参考LVM一节查看创建卷组的步骤。创建 thin pool 命令如下。

# lvcreate -L 80G -T -n vmstore vmdata

#### ZFS 管理

ZFS是由Sun微系统公司（Sun Microsystems）设计的同时集成有文件管理功能和逻辑卷管理功能的文件系统。 uVirt 引入了ZFS文件系统作为可选的文件系统和根文件系统。uVirt ISO光盘镜像已经集成 了ZFS所需的软件包，用户无须手工编译即可直接使用ZFS。

利用ZFS，用户可以在配置有限的硬件设备上尽可能多的享受企业级的文件系统服务，进一步还可以利用SSD缓存 或者全SSD技术获得极高的系统性能。利用常见的CPU和内存硬件，只需简单的配置，ZFS即可实现昂贵的基于硬件 RAID卡的管理功能。

ZFS 的技术优势包括：

* 可通过图形界面或命令行方式进行配置管理，容易使用。
* 高可靠性
* 有效防止数据损坏丢失
* 文件系统级的数据压缩
* 支持快照功能
* 支持Copy-on-write
* 多种软RAID模式：RAID0，RAID1，RAID10，RAIDZ-1，RAIDZ-2，RAIDZ-3
* 支持SSD缓存
* 自我数据修复
* 连续数据完整性检查和验证
* 支持大容量数据存储
* 有效防止数据损坏丢失
* 基于网络的数据异步复制
* 支持数据加密
* …

##### 硬件

ZFS对于内存配置的依赖较高，一般最少需要为ZFS配置8GB内存。实际生产中，最好基于你的预算配置尽可能多的内存。 为防止数据损坏，我们建议你使用高端的ECC内存条。

如果你要为ZFS配置独立的缓存盘或文件系统日志盘，最好使用SSD盘。这能够极大的提升整体性能表现。

不要使用有独立缓存管理的硬件控制器。ZFS需要能够直接访问磁盘。ZFS可以和HBA卡配合使用， 也可以和 “IT” 模式的LSI控制器及类似硬件配合使用。

如果你是在 uVirt 虚机环境里测试安装 uVirt（嵌套虚拟化），不要使用virtio类型的虚拟磁盘， 目前ZFS不支持 virtio 类型磁盘。建议为虚机配置IDE或SCSI类型的虚拟磁盘（ virtio SCSI控制器）。

##### 用于根文件系统

当你使用ISO镜像安装时，可以选择使用ZFS作为根文件系统。同时你需要在安装过程中选择配置RAID级别。

RAID0 也称为 “条带模式” 。该模式下ZFS卷的容量为所有硬盘容量之和。但RAID0不提供任何冗余性， 卷中任何一块硬盘故障都会导致整个卷不可用。

RAID1 也称为 “镜像模式” 。该模式下，数据会以复制方式同时写入所有硬盘。该模式至少需要2块容量 一样的硬盘，而整个卷的容量就等于单块盘的容量。

RAID10 该模式组合了RAID0和RAID1模式。配置使用该模式至少需要4块硬盘。

RAIDZ-1 类似于RAID5模式，提供1块硬盘故障冗余。配置使用该模式至少需要3块硬盘。

RAIDZ-2 类似于RAID5模式，提供2块硬盘故障冗余。配置使用该模式至少需要4块硬盘。

RAIDZ-3 类似于RAID5模式，提供3块硬盘故障冗余。配置使用该模式至少需要5块硬盘。

安装程序会自动完成硬盘分区，构建名为 rpool 的ZFS存储池，并在 rpool/ROOT/uvirt-1 上安装根文件系统。

安装程序还会创建一个名为 rpool/data 的子卷，用于保存虚机镜像。为便于使用新支点虚拟化工具管理该ZFS卷， 安装程序会在 /etc/uvirt/storage.cfg 文件中创建以下配置信息：

zfspool: local-zfs

pool rpool/data

sparse

content images,rootdir

安装完成后，你可以运行 zpool 命令查看ZFS存储池的状态：

# zpool status

pool: rpool

state: ONLINE

scan: none requested

config:

NAME STATE READ WRITE CKSUM

rpool ONLINE 0 0 0

mirror-0 ONLINE 0 0 0

sda2 ONLINE 0 0 0

sdb2 ONLINE 0 0 0

mirror-1 ONLINE 0 0 0

sdc ONLINE 0 0 0

sdd ONLINE 0 0 0

errors: No known data errors

可以用 zfs 命令配置管理ZFS文件系统。下面的命令用于列出安装新支点虚拟化后的ZFS文件系统：

# zfs list

NAME USED AVAIL REFER MOUNTPOINT

rpool 4.94G 7.68T 96K /rpool

rpool/ROOT 702M 7.68T 96K /rpool/ROOT

rpool/ROOT/uvirt-1 702M 7.68T 702M /

rpool/data 96K 7.68T 96K /rpool/data

rpool/swap 4.25G 7.69T 64K -

##### ZFS 管理

本节将给出ZFS日常管理操作的范例。ZFS本身非常强大，具有很多命令选项。管理ZFS最主要的两个命令是 zfs 和 zpool 。 这两个命令都有非常完善的技术手册，可以使用如下命令查看：

# man zpool

# man zfs

###### 创建新的存储池

至少需要有1块硬盘才可以创建一个新的存储池。可以用参数 ashift 指定区块大小（2的 asshift 次幂），且不能小于所用磁盘区块的大小。

zpool create -f -o ashift=12 <pool> <device>

还可以执行以下命令激活数据压缩功能。

zfs set compression=lz4 <pool>

###### 创建新的 RAID-0 存储池

至少需要1块硬盘。

zpool create -f -o ashift=12 <pool> <device1> <device2>

###### 创建新的 RAID-1 存储池

至少需要2块硬盘。

zpool create -f -o ashift=12 <pool> mirror <device1> <device2>

###### 创建新的 RAID-10 存储池

至少需要4块硬盘。

zpool create -f -o ashift=12 <pool> mirror <device1> <device2> mirror <device3> <device4>

###### 创建新的 RAIDZ-1 存储池

至少需要3块硬盘。

zpool create -f -o ashift=12 <pool> raidz1 <device1> <device2> <device3>

###### 创建新的 RAIDZ-2 存储池

至少需要4块硬盘。

zpool create -f -o ashift=12 <pool> raidz2 <device1> <device2> <device3> <device4>

###### 创建新的带有缓存（L2ARC）的存储池

可以使用独立的硬盘分区（建议使用SSD）作为缓存，以提高ZFS性能。

如下命令中， <device> 处可以列出多个硬盘设备，命令格式就像“创建新的 RAID\* 存储池”一样。

zpool create -f -o ashift=12 <pool> <device> cache <cache\_device>

###### 创建新的带日志（ZIL）的存储池

可以使用独立的硬盘分区（建议使用SSD）记录文件系统日志，以提高ZFS性能。

如下命令中， <device> 处可以列出多个硬盘设备，命令格式就像“创建新的 RAID\* 存储池”一样。

zpool create -f -o ashift=12 <pool> <device> log <log\_device>

###### 为已有的存储池添加缓存和日志盘

如果你要为一个未配置缓存和日志盘的ZFS存储池添加缓存和日志盘，首先需要使用 parted 或者 gdisk 将SSD盘划分为两个分区。

注意：确保使用GPT分区表。

日志盘的大小最大为物理内存容量的一半，通常都不大。SSD盘剩余空间可用作缓存。

zpool add -f <pool> log <device-part1> cache <device-part2>

###### 更换故障磁盘

zpool replace -f <pool> <old device> <new device>

在使用systemd-boot时更换故障的系统磁盘

sgdisk <healthy bootable device> -R <new device>

sgdisk -G <new device>

zpool replace -f <pool> <old zfs partition> <new zfs partition>

uvirt-efiboot-tool format <new disk's ESP>

uvirt-efiboot-tool init <new disk's ESP>

**注意：**ESP 是EFI系统分区，安装程序会将其设置为启动盘的第二个分区。详见 创建新分区用作同步ESP。

##### 激活邮件通知

ZFS有一个事件守护进程，专门监控ZFS内核模块产生的各类事件。当发生严重错误，例如存储池错误时，该进程还可以发送邮件通知。你可以安装 zfs-zed ：

# apt-get install zfs-zed

可以编辑配置文件 /etc/zfs/zed.d/zed.rc 以激活邮件通知功能。只需将配置参数 ZED\_EMAIL\_ADDR 前的注释符号去除即可，如下：

ZED\_EMAIL\_ADDR="root"

**请注意：**新支点虚拟化软件会将邮件发送给为root用户配置的电子邮件地址。

**重要：**只需激活 ZED\_MAIL\_ADDR 参数即可。其他配置参数均为可选项目，非必须项。

##### 配置ZFS内存使用上限

最好将ZFS的缓存容量ZFS ARC上限设置为物理内存容量的一半，以避免内存短缺导致系统性能变坏。具体做法是编辑配置文件 /etc/modprobe.d/zfs.conf ，插入如下内容：

options zfs zfs\_arc\_max=8589934592

上例中设置ZFS缓存容量上限为8GB。

重要：如果根文件系统也使用了ZFS，你必须在每次修改该参数后更新initramfs，如下：

update-initramfs -u

##### ZFS上的SWAP

使用 zvol 创建SWAP分区可能会导致一些问题，例如系统卡死或者很高的IO负载。特别是在向外部存储备份文件时会容易触发此类问题。

我们强烈建议为ZFS配置足够的物理内存，避免系统出现可用内存不足的情形。如果实在想要创建一个SWAP分区，最好是直接在物理磁盘上创建。 可以在安装新支点虚拟化时通过高级选项设置预留磁盘空间，以便创建SWAP。此外，你可以调低 “swappiness” 参数值。通常，设置为10比较好：

sysctl -w vm.swappiness=10

如果需要将swappiness参数设置持久化，可以编辑文件 /etc/sysctl.conf ，插入下内容：

vm.swappiness = 10

**表3 Linux内核 swappiness 参数设置表**

|  |  |
| --- | --- |
| **设置** | **对应策略** |
| vm.swappiness = 0 | 内核仅在内存耗尽时进行swap |
| vm.swappiness = 1 | 内核仅执行最低限度的swap |
| vm.swappiness = 10 | 当系统有足够多内存时，可考虑使用该值，以提高系统性能 |
| vm.swappiness = 60 | 默认设置值 |
| vm.swappiness = 100 | 内核将尽可能使用swap |

##### 加密 ZFS数据集

ZFS on Linux在0.8.0版之后引入了本地数据集加密功能。将ZFS on Linux升级后，就可以对指定存储池启用加密功能：

# zpool get feature@encryption tank

NAME PROPERTY VALUE SOURCE

tank feature@encryption disabled local

# zpool set feature@encryption=enabled

# zpool get feature@encryption tank

NAME PROPERTY VALUE SOURCE

tank feature@encryption enabled local

**警告：**目前还不支持通过Grub从加密数据集启动系统，并且在启动过程中对自动解锁加密数据集的支持也很弱。 不支持加密功能的旧版ZFS也不能解密相关数据。

**注意：**建议在启动后手工解锁存储数据集，或使用 zfs load-key 命令将启动中解锁数据集所需key信息写到定制参数中。

**警告：**在对生产数据正式启用加密功能前，建议建立并测试备份程序有效性。注意，一旦key信息丢失，将永远不可再访问加密数据。

加密功能需要在创建 datasets/zvol 时启用，并将被子数据集自动继承启用。例如，可用以下命令创建加密数据 集 tank/encrypted\_data ，并提供给 uVirt 使用。

# zfs create -o encryption=on -o keyformat=passphrase tank/encrypted\_data

Enter passphrase:

Re-enter passphrase:

# uvirtsm add zfspool encrypted\_zfs -pool tank/encrypted\_data

在该存储上创建的所有虚机的 卷/磁盘 都将被自动加密，加密秘钥将共享使用父存储的秘钥。

如需使用存储，需要使用 zfs load-key 命令加载秘钥信息：

# zfs load-key tank/encrypted\_data

Enter passphrase for 'tank/encrypted\_data':

也可以使用随机秘钥文件代替手工输入的秘钥。只需在创建数据集时设置 keylocation 和 keyformat 参数即可， 或者使用 zfs change-key 命令改变已有数据集的设置。

# dd if=/dev/urandom of=/path/to/keyfile bs=32 count=1

# zfs change-key -o keyformat=raw -o keylocation=file:///path/to/keyfile tank/encrypted\_data

注意：使用加密文件时，需要特别注意保护好加密文件，确保不被越权访问，或者意外丢失。 一旦丢失了秘钥文件，就不可能再访问相关数据。

如果虚机磁盘创建在加密数据集上，虚拟磁盘将自动继承启用 encryptionroot 参数。对每个数据集的 encryptionroot ，都需要加载秘钥信息。

具体信息可以查看 man zfs 手册的 Encryption 段落，包括 encryptionroot ， encryption ， keylocation ， keyformat 和 keystatus 属性， zfs load-key ， zfs unload-key 和 zfs change-key 命令。

##### ZFS专用设备

ZFS自从0.8.0版本后开始支持专用(special)设备。存储池中的专用设备用来存放元数据、去重表，以及小文件块（可选）。

专用设备能够有效提升存储池的速度，特别是存储池的由慢速磁介质硬盘，且由大量元数据变更时。比如新建、更新或删除大量文件的工作负载，将会 明显从专用设备中受益。ZFS数据集也可以配置专用设备来存储所有的小文件，这将进一步提升性能。专用设备请使用高速SSD硬盘。

**重要：**专用设备的冗余策略需要和存储池的匹配，因为专用设备可能是整个存储的一个单点故障。

**警告：**专用设备一旦加入，不可撤销！

1. 创建一个带专用设备的RAID-1存储池：

zpool create -f -o ashift=12 <pool> mirror <device1> <device2> special mirror <device3> <device4>

1. 为已有的RAID-1存储池添加专用设备：

zpool add <pool> special mirror <device1> <device2>

ZFS数据集暴露 special\_small\_blocks=<size> 属性来支持存储小文件。 size 为0时，将禁用专用设备存储小文件； size 可设置为 512B-128K 范围，必须是2的幂。

**重要：**如果设置 special\_small\_blocks 超过数据集的 recordsize (默认128K)，所有数据将会被写入到专用设备，请小心！

给存储池设置 special\_small\_blocks 属性，将会改变所有ZFS子数据集的该属性的默认值（比如池中所有容器将启用小文件块）

启用所有小于4K块的文件池：

zfs set special\_small\_blocks=4K <pool>

启用单个数据集的小文件块：

zfs set special\_small\_blocks=4K <pool>/<filesystem>

禁用单个数据集的小文件块：

zfs set special\_small\_blocks=0 <pool>/<filesystem>

## 网络配置

网络配置可以通过GUI进行设置，也可以手工编辑配置文件 /etc/network/interfaces 来设置，不管哪种方式， 所有的网络配置参数都保存在该文件里。Man手册文件 interfaces (5) 中有完整的设置说明。新支点虚拟化软件尽量为用户保留编辑配置文件的设置方式，但最好还是使用GUI来修改设置，这样可以有效避免配置错误。

一旦网络配置完成，可以通过命令 ifup 和 ifdown 来重启端口使配置生效。

### 应用网络修改

新支点虚拟化软件并不直接编辑配置文件 /etc/network/interfaces 。实际上，网络配置变更会首先保存在临时文件 /etc/network/interfaces.new 里，然后在重启服务器时再用该配置文件覆盖 /etc/network/interfaces 。

#### 重启节点生效

使用默认安装的 ifupdown 网络管理软件包，您需要重新启动以提交任何未决的网络更改。大多数时候，新支点虚拟化软件的基础的网络设置是稳定的，并且不会经常更改，因此不需要经常重启。

#### 通过ifupdown2重载网络

使用可选的ifupdown2网络管理软件包，您还可以实时重载(reload)网络配置，而无需重启系统。

注意：ifupdown2 不能理解 OpenVSwitch 的配置语法，因此如果采用了 OVS 网络，请不要使用 ifupdown2 。

你还可以通过WEB面板使变更的网络配置生效。具体位置在节点的「网络」面板中的 「应用配置」按钮。

**警告：**安装 ifupdown2 将会删除 ifupdown 。

要安装 ifupdown2 ，只需要简单键入：

apt install ifupdown2

### 命名规范

目前我们采用的网络设备命名规范如下：

* 网卡：en\*，即systemd类的网络接口命名。
* 网桥：vmbr[N]，其中0 ≤ N ≤ 4094（ vmbr0 - vmbr4094 ）
* 网口绑定：bond[N]，其中0 ≤ N（ bond0 ， bond1 ，…)
* VLANs：只需要将VLAN编号附加到网络设备名称后面，并用“.”分隔（eht0.50，bond1.30）

采用命名规范将网络设备名称和网络设备类型关联起来，能够大大降低网络故障排查难度。

#### Systemd网卡命名规范

Systemd采用 en 作为网卡设备名称前缀。名称后续字符由网卡驱动和命名规范匹配先后顺序决定。

* o<index>[n<phys\_port\_name>|d<dev\_port>] — 板载设备命名
* s<slot>[f<function>][n<phys\_port\_name>|d<dev\_port>] — 按设备热插拔ID命名
* [P<domain>]p<bus>s<slot>[f<function>][n<phys\_port\_name>|d<dev\_port>] — 按设备总线ID命名
* x<MAC> — 按设备MAC地址命名

最常见的命名模式如下：

* eno1 — 第一个板载NIC网卡
* enp3s0f1 — 位于3号PCI总线，0号插槽，NIC功能号为1的NIC网卡

### 网络配置规划

你需要根据当前的网络规划以及可用资源，决定具体采用的网络配置模式。可选模式包括网桥、路由以及地址转换三种类型。

#### 服务器位于内部局域网，通过外部网关与互联网连接

这种情况下最适宜采用网桥模式。这也是新安装新支点虚拟化软件默认采用的模式。 该模式下，所有虚机通过虚拟网卡与新支点虚拟化软件虚拟网桥连接。其效果类似于虚机网卡直接连接在局域网交换机上，而新支点虚拟化软件就扮演了这个交换机的角色。

#### 服务器机房托管，并分配有一段互联网公共IP地址

这种情况下，可根据IDC供应商分配的资源和权限，选择网桥模式或路由模式。

#### 服务器机房托管，但只有一个互联网公共IP地址

这种情况下，虚机访问外部互联网的唯一办法就是通过地址转换。 如果外部网络需要访问虚机，还需要配置端口转发。

为日后维护使用方便，可以配置VLANs（IEEE 802.1q）和网卡绑定，也就是 “链路聚合” 。这样就可以灵活地建立复杂虚机网络结构。

### 网桥设置

网桥相当于一个软件实现的物理交换机。所有虚机共享一个网桥，在多个域的网络环境中，也可以创建多个网桥以分别对应不同网络域。 理论上，每个新支点虚拟化软件实例最多可以支持4094个网桥。

uVirt 安装程序会创建一个名为 vmbr0 的网桥，并和检测到的服务器第一块网卡桥接。 配置文件 /etc/network/interfaces 中的对应配置信息如下：

auto lo

iface lo inet loopback

iface eno1 inet manual

auto vmbr0

iface vmbr0 inet static

address 192.168.10.2

netmask 255.255.255.0

gateway 192.168.10.1

bridge\_ports eno1

bridge\_stp off

bridge\_fd 0

在基于网桥的默认配置下，虚机看起来就和直接接入物理网络一样。尽管所有虚机共享一根网线接入网络， 但每台虚机都使用自己独立的MAC地址访问网络。

基于新支点虚拟化软件的 Ceph 服务

新支点虚拟化软件统一了计算和存储功能。集群的物理节点既可以同时用于计算（运行虚拟机和轻虚机）和多副本存储。 传统的计算资源和存储资源管理功能可以由统一的超融合应用实现，无需再部署专用存储网络设备（SANs）和网络存储 设备（NAS）。通过集成软件定义存储Ceph，新支点虚拟化软件能够直接在虚拟机服务器节点上运行和管理Ceph存储。

Ceph是一个高性能、高可靠、高可扩展的分布式对象存储和文件系统。

uVirt 集成 Ceph 的优势包括：

* 可以通过CLI和GUI轻松安装管理
* 支持thin模式存储
* 支持快照
* 自动修复
* 容量最大可扩充至exabyte级别
* 支持多种性能和冗余级别的存储池
* 多副本，高容错
* 可在低成本硬件运行
* 无需硬件raid控制器

对于中小规模部署场景，可以直接将Ceph安装到新支点虚拟化服务器，以实现RADOS块设备（RBD）功能。详见 基于Ceph RADOS块设备的后端存储。当前主流硬件拥有足够强大的CPU和内存资源，能够满足在同一节点同时运行虚机和存储服务的需要。

为简化管理，新支点虚拟化提供了 uvirtceph 工具来安装管理Ceph服务。

Ceph包含一组后台服务，以实现RBD存储功能 ：

* Ceph Monitor（ceph-mon）
* Ceph Manager（ceph-mgr）
* Ceph OSD（ceph-osd；对象存储服务）

# 需求条件

创建超融合的 "新支点虚拟化+Ceph集群"，至少需要3台独立服务器。

更多建议参考 Ceph 网站。

## CPU

应优先选用高主频CPU以降低延时。一个简单的经验是，为每个Ceph服务分配一个CPU核心（或CPU线程），以确保Ceph拥有足够资源，能够以良好性能稳定运行。

## 内存

采用超融合部署方式时，需要密切关注内存使用情况。除了虚拟机和轻虚机必须使用的内存外，Ceph也需要有足够内存，才能提供稳定且良好的性能。一个经验是，每1TiB数据，OSD需要占用1GiB内存。OSD缓存还需要额外配置内存。

## 网络

建议为Ceph准备专用的10Gb或者更高性能的网络。如果没有10Gb交换机设备，也可以使用网状网络 。

高负载网络通信，特别是虚机恢复时的流量，将影响运行在同一网络上的服务，很有可能造成新支点虚拟化集群的崩溃。

建议认真估算网络带宽需求。单块硬盘可能不能压满1Gb链路，但多块硬盘组成的OSD就可以。主流NVME SSD完全可以压满10Gbps带宽。采用更高带宽性能的网络，可以确保网络任何时候都不会成为性能瓶颈。为此，25Gb，40Gb，100Gb的网络都值得考虑。

## 磁盘

在规划Ceph集群时，需要重点考虑恢复时间因素。对于小规模集群，恢复时间可能会非常长。推荐在小规模集群中使用固态SSD盘代替HDD硬盘，以缩短恢复时间，降低恢复期间发生二次故障的风险。

通常情况下，SSD的IOPs比传统磁盘高的多，但价格也更贵，可以参考组建不同类型的存储池，以提高恢复性能。也可以参考创建Ceph OSD，使用高速存储盘作为DB/WAL设备，加速OSDs。如果同时为多个OSDs配置了高速存储盘，需要考虑平衡OSD和WAL/DB（卷）盘的配比，以避免高速存储盘成为相关OSDs的性能瓶颈。

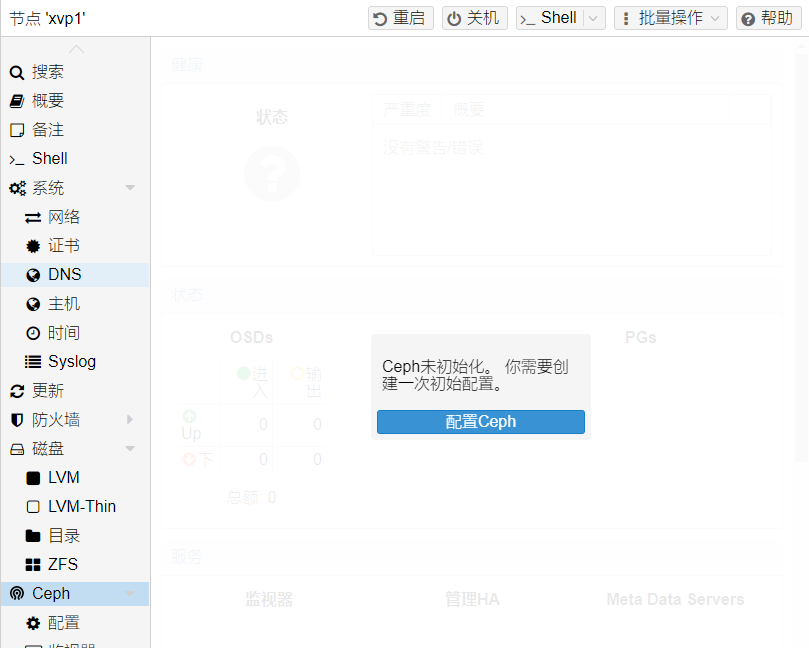
除了选择合适存储盘类型，还可以选择为单一节点配置偶数个对称存储盘，以提高Ceph性能。例如，单一节点使用4块500GB存储盘时的性能就比混合使用1块1TB盘和3块250GB盘要好。

此外，还需要妥善平衡OSD数量和单一OSD容量。大容量OSD可以增加存储密度，但也意味着在OSD故障时，Ceph需要恢复更多数据。

## 避免使用硬RAID

Ceph直接处理数据对象冗余和多重并发磁盘（OSDs）写操作，因此使用硬RAID控制器并不能提高性能和可用性。相反，Ceph需要直接控制磁盘硬件设备。硬件RAID控制器并非为Ceph所设计，其写操作管理和缓存算法可能干扰Ceph对磁盘的正常操作，从而把事情复杂化，并导致性能降低。

# 初始化Ceph安装和配置

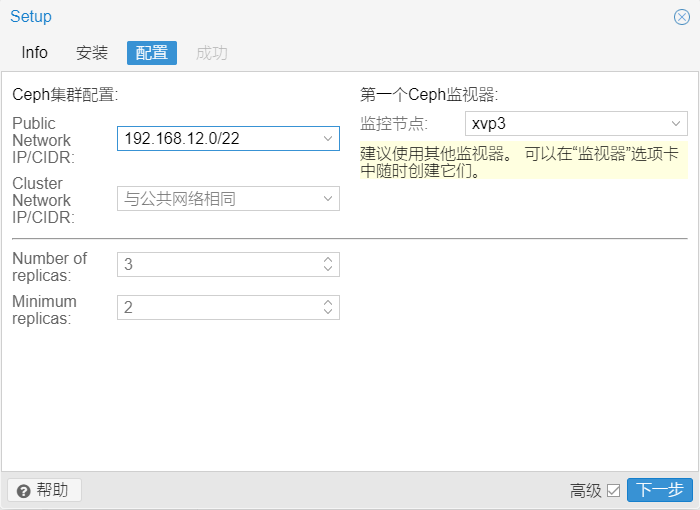


**图9 在节点上安装 Ceph**

新支点虚拟化软件提供了简单易用的Ceph安装向导。选中集群中的一个节点，然后在菜单树中打开Ceph菜单区，就可以开始安装Ceph了。

安装向导有多个步骤，每个步骤都需要执行成功才可以完成Ceph安装。开始安装操作后，向导会自动从新支点虚拟化的ceph软件包仓库下载软件包并完成安装。

完成第一个步骤后，还需要创建配置。对每个集群，生成的配置信息会通过第7章所述新支点虚拟化集群文件系统（udocfs）自动分发到其他节点，所以该操作只需要执行一次即可。



**图10 Ceph 安装配置**

创建的配置包括以下信息：

* 公共网络: 为避免影响集群通信等其他对网络延迟敏感的服务，也为了提高Ceph性能，强烈建议为Ceph准备一个专门的独立网络，将Ceph流量隔离开来。
* 集群网络: 进一步，还可以设置 集群网络 ，将 OSD 复制和心跳流量隔离出来。这将有效降低 公共网络 的负载，并有效改善大规模Ceph集群的性能。

以下两个参数项属于高级配置功能，仅供专家级用户使用。

* 副本数量: 定义一个对象被复制几份
* 最小副本数: 副本数低于该阀值时，数据将会被标记为不完全状态

此外，还需要选择第一个监视器节点。

所有配置完成后，系统会提示配置成功，并给出下一步安装指令。接下来，您创建创建额外的 监视器，创建 OSD，并至少创建一个 Pool，之后就可以使用Ceph了。

本章后续内容将介绍如何在新支点虚拟化中使用Ceph，包括前述内容，以及 CephFS。

# 安装 Ceph

在每个节点运行如下脚本命令：

uvirtceph install

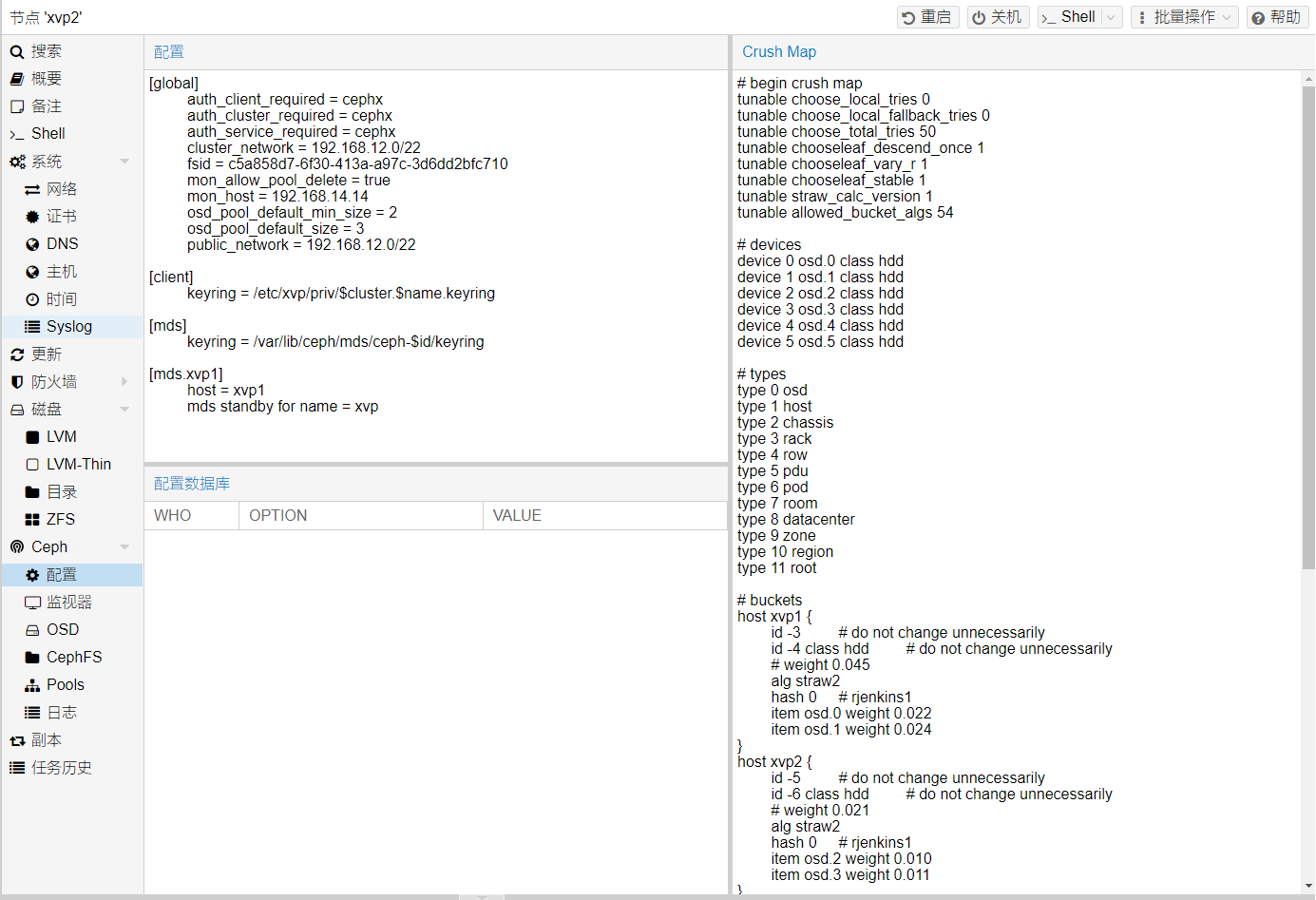
该命令将创建 apt 软件包仓库 /etc/apt/sources.list.d/ceph.list ，并安装所需软件。

# 初始化 Ceph

使用新支点虚拟化 Ceph安装向导（推荐），或在一个节点上运行以下命令：

uvirtceph init --network 10.10.10.0/24

该命令将创建为ceph创建一个专用网络，并将初始配置写入文件 /etc/uvirt/ceph.conf 。该文件将通过 udocfs 自动分发到所有新支点虚拟化服务器节点。该命令还将创建符号链接 /etc/ceph/ceph.conf 指向该配置文件，以便直接运行Ceph命令，无需另外创建配置文件。



**图11 初始化Ceph配置**

# Ceph 监视器

可以在你想要部署monitor的节点上（建议创建3个monitor），可以在GUI界面依次选择 Ceph→Monitor 选项完成monitor创建，也可以运行如下命令。

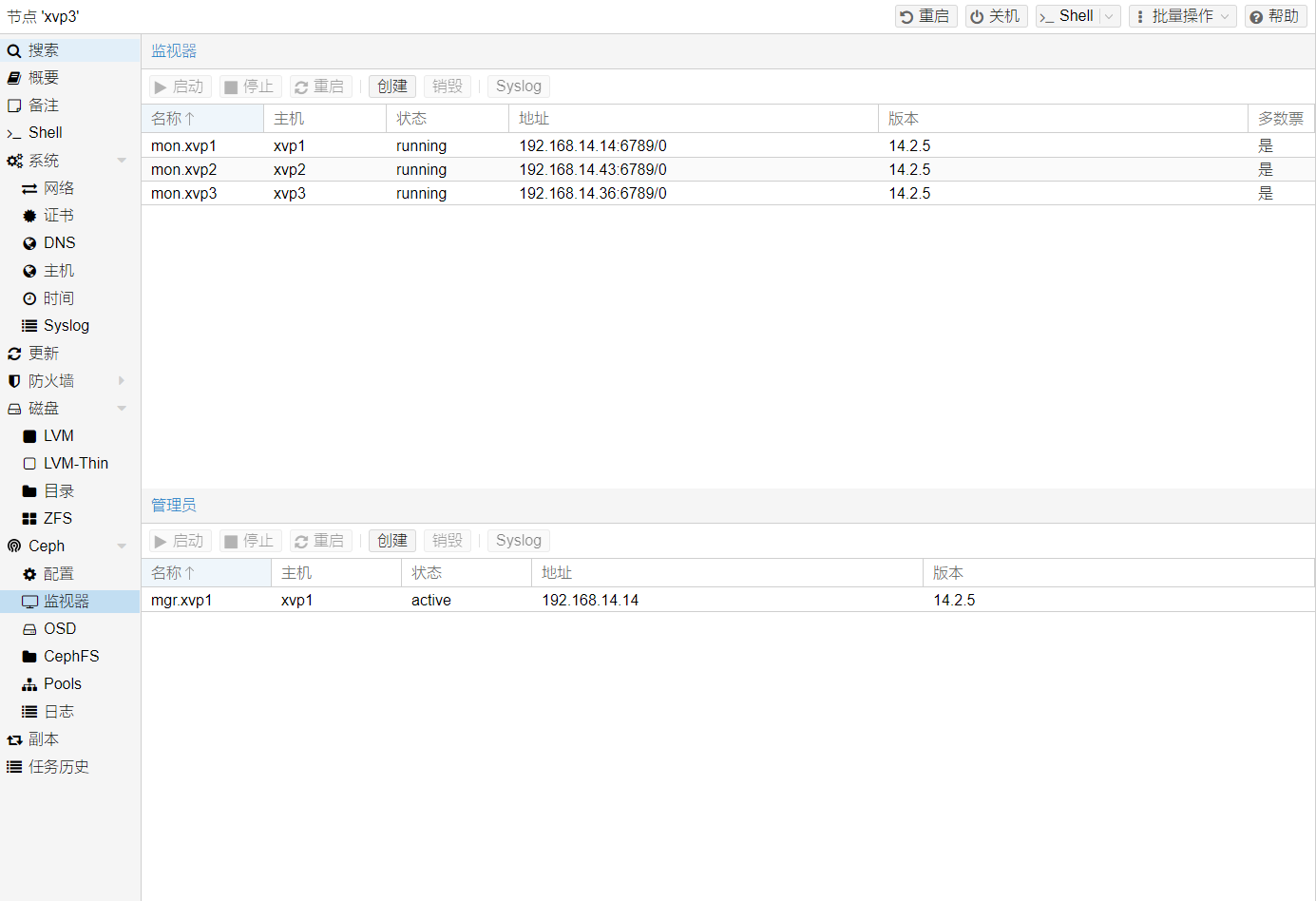
uvirtceph createmon

运行该命令将默认同时安装Ceph Manager（ceph-mgr）。如果不需要安装，可以在执行命令行增加选项 -exclude-manager 。

## 创建监视器

Ceph Monitor(MON)负责管理集群全局数据。如需要实现HA，则至少需要创建3个Monitor。安装向导会自动创建一个monitor。对中小规模集群而言，最多3个monitor就够了，只有大型集群才需要更多的monitor。

uvirtceph mon create



**图12 Ceph 监控器**

## 销毁监视器

要通过GUI删除Ceph Monitor，请首先在树形视图中选择一个节点，然后转到 Ceph → Monitor 面板。选择 MON ，然后单击 销毁 按钮。

要通过CLI删除Ceph Monitor，请首先连接到运行 MON 的节点。然后执行以下命令：

uvirtceph mon destroy

# Ceph管理器

Ceph管理器是独立于monitor的另一个服务。主要提供监控Ceph集群运行状态的接口。从 luminous版本开始，ceph-mgr 服务成为luminous的必选组件。Ceph manager一般和monitor同时完成安装。

## 创建管理器

可以安装多个管理器，但同一时间只能一个生效。

uvirtceph mgr create

**注意：**建议将Ceph Manager部署在monitor节点上。并考虑安装多个Cepnmanager，以满足高可用要求。

## 销毁管理器

在树形视图中选择一个节点，转到 Ceph → 监视器 面板，选择 管理器 并点击 销毁 按钮。

通过命令行删除管理器，需要先连接到管理器所在节点，然后执行如下命令：

uvirtceph mgr destroy

**注意：**Ceph集群在没有管理器时也可以工作，但是部分功能会失效，比如集群状态或高可用。

# Ceph OSDs

Ceph Object Storage Daemons 专门存储对象数据。推荐一个物硬盘一个OSD。

**注意：**默认一个对象4MiB大小

## 创建 OSD

可以通过GUI或命令行创建OSD。创建命令如下：

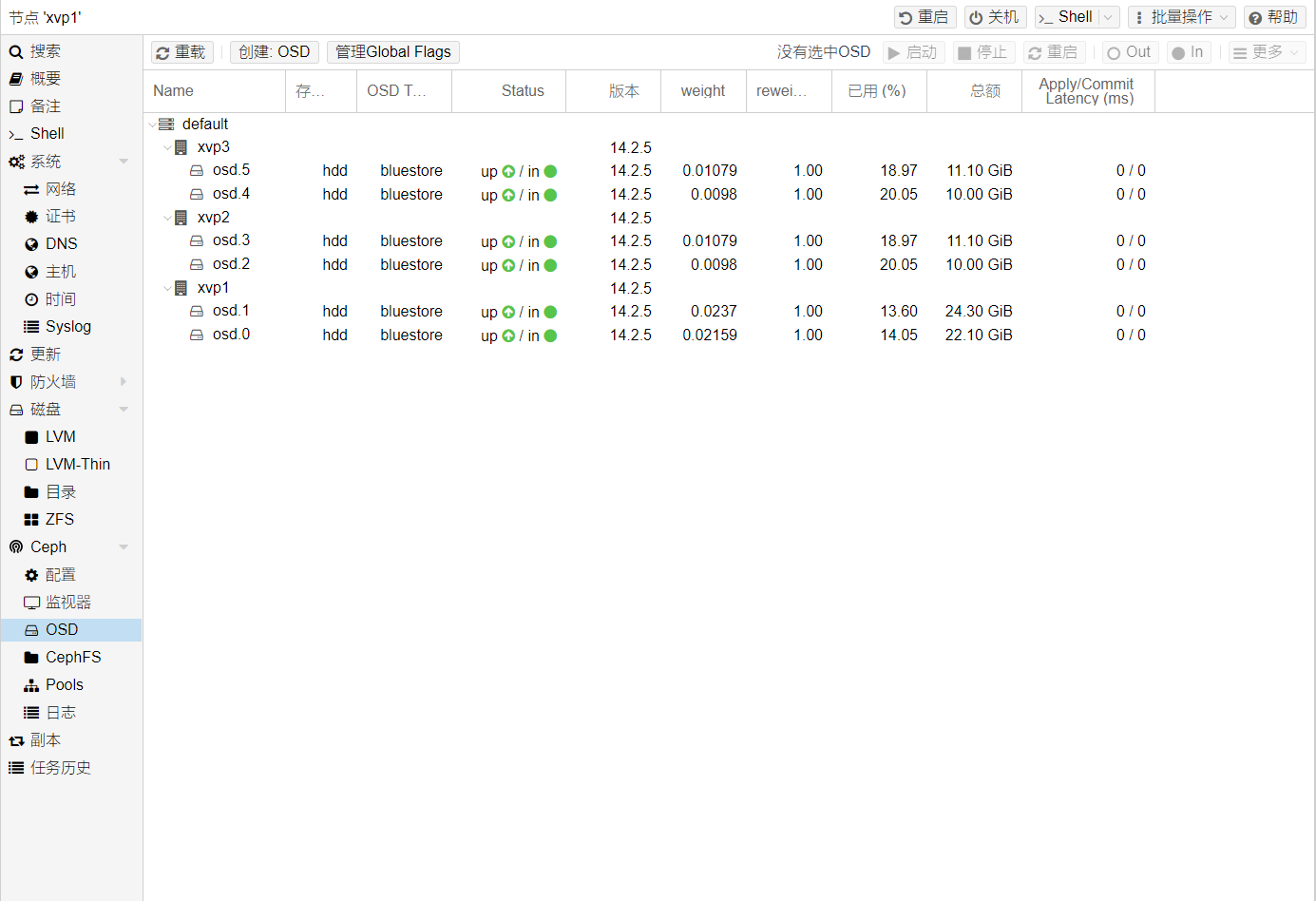
uvirtceph osd create /dev/sd[X]

建议至少为Ceph集群创建12个OSD，并平均分配到集群的各节点。在最小的3节点集群下，每个节点部署4个OSD。

如磁盘之前已经被格式化过（如ZFS/RAID/OSD），可用以下命令删除分区表、引导扇区和其他OSD遗留数据。

ceph-volume lvm zap /dev/sd[X] –destroy

**注意：**以上命令将删除磁盘上的所有数据。



**图13 OSD 状态**

### Ceph Bluestore

从Kraken版本开始，Ceph引入一种新的Ceph OSD存储类型，即Bluestore。而在luminous版Ceph中，该类型已成为默认OSD类型。

uvirtceph osd create /dev/sd[X]

### Block.db and block.wal

如果要为OSD配置专门独立的DB/WAL设备，可以通过 -db\_dev 和 -wal\_dev 选项指定设备。如果不指定专门设备，WAL数据将保存在DB上。

uvirtceph osd create /dev/sd[X] -db\_dev /dev/sd[Y] -wal\_dev /dev/sd[Z]

你可以用-db\_size和-wal\_size参数直接设置相关设备大小。如果未明确设置，将依次尝试使用以下值：

* 使用ceph配置中的bluestore\_block\_{db,wal}\_size
  + …数据库，osd区块
  + …数据库，global区块
  + …文件，osd区块
  + …文件，global区块
* OSD大小的10%(DB)/1%(WAL)

DB保存了BlueStore的内部元数据。WAL是BlueStore的内部卷，用于保存预写日志。建议采用高性能SSD或NVRAM作为DB/WAL，以提高性能。

### Ceph Filestore

在Ceph luminous之前，Ceph OSD都采用Filestore存储模式。从Ceph Nautilus开始，新支点虚拟化不再支持使用 uvirtceph 创建该类OSD。如果确实需要创建Filestore类的OSD，可以使用 ceph-volume 命令。

ceph-volume lvm create --filestore --data /dev/sd[X] --journal /dev/sd[Y]

## 销毁OSD

可以在GUI树形视图中上选择一个 新支点虚拟化 节点，转到 Ceph → OSD 面板。选择一个OSD。下一步 点击 OUT 按钮。一旦OSD状态从 in 变为 out 后，点击 停止 按钮。只要其状态从 up 变为 down ，从 更多下拉菜单中点击 销毁 。

从命令行销毁一个OSD：

ceph osd out <ID>

systemctl stop ceph-osd@<ID>.service

第一个命令通知Ceph不要将分发给该OSD。第二个命令停止OSD服务。到这个时候，数据还没有丢失。

下面的命令会销毁OSD。指定 -cleanup 选项将会额外销毁分区表。

uvirtceph osd destroy <ID>

警告：上面的命令销毁了磁盘上数据！

# Ceph Pools

## 创建 Ceph Pool

存储池pool是一组存储对象的逻辑集合。具体是由一组Placement Groups（PG，pg\_num）的对象集合组成。 当不使用任何选项时，默认将创建128个PG，并使用3副本模式，降级模式最低使用2副本模式。

默认PG数量适用于2-5块磁盘的场景。当集群内PG数量太少或太多时，Ceph将发出 HEALTH\_WARNING 告警。

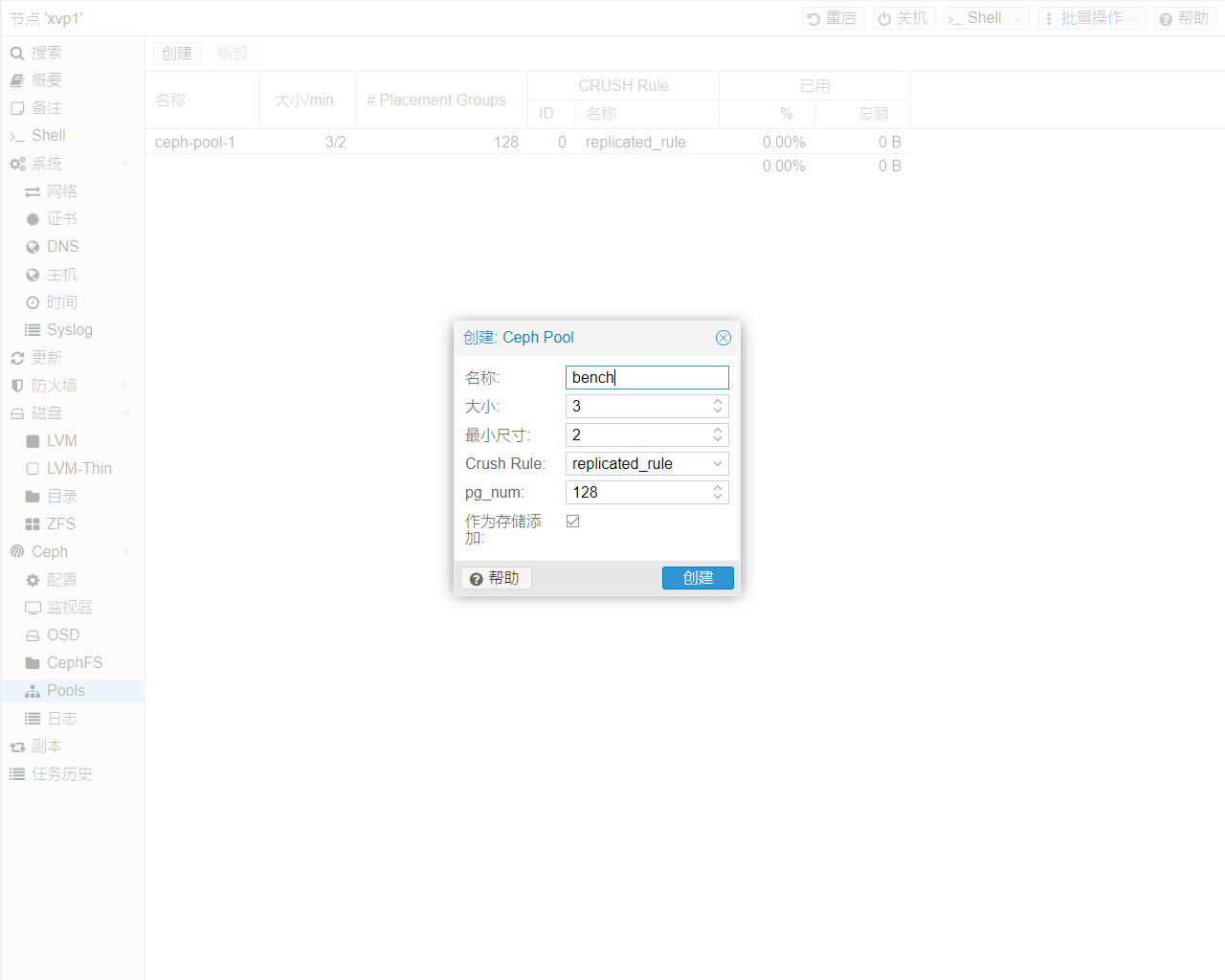
建议根据你的具体配置计算确定PG数量。互联网上可以找到PG数量计算公式或 PG数量计算器。 Ceph一旦投入运行，PG数量只可以调整增加，但不能再减少。

可以在GUI主机管理界面选择 Ceph → Pools 或直接用命令行创建pool。

uvirtceph pool create <name>

如果要自动获取pool的存储定义，可以在GUI上勾选 Add storages 框，或使用命令行创建pool时使用选项 -–add\_storages 。

关于Ceph pool管理的进一步信息可以查看 Ceph pool管理手册。



**图14 创建 Ceph Pool**

## 销毁Pools

可以在GUI树形视图中上选择一个 新支点虚拟化 节点，转到 Ceph → Pools 面板，选择一个pool，点击 销毁 按钮。 要确认真的销毁pool，你需要输入pool名称。

也可以运行如下命令销毁pool，指定 -remove\_storages 选项也会删除关联的存储：

uvirtceph pool destroy <name>

删除pool中的数据是一个后台任务，需要一定时间。你可以观察到集群中的磁盘利用率在下降。

# Ceph CRUSH和设备类别

Ceph是以算法 Controlled Replication Under Scalable Hashing(CRUSH)为基础创建的。

CRUSH算法用于计算数据存取的位置，且无需中心索引服务的支持。CRUSH基于构成存储池pool的OSD、buckets（设备位置）和rulesets（数据复制规则）来完成计算。

关于CRUSH图的进一步信息，可以查看Ceph官方文档中CRUSH图 一节。

调整该图可以反映不同层次的复制关系。对象副本可以分布在不同地方（例如，各故障区域），并同时保持期望的分布。

常见用法是为不同的Ceph pool配置不同类别的磁盘。为此，Ceph luminous引入了设备类的概念，以简化ruleset的创建。

设备类信息可以用命令 ceph osd tree 查看。各个类代表了各自的根位置。命令如下。

ceph osd crush tree --show-shadow

以上命令的输出示例如下：

ID CLASS WEIGHT TYPE NAME

-16 nvme 2.18307 root default~nvme

-13 nvme 0.72769 host sumi1~nvme

12 nvme 0.72769 osd.12

-14 nvme 0.72769 host sumi2~nvme

13 nvme 0.72769 osd.13

-15 nvme 0.72769 host sumi3~nvme

14 nvme 0.72769 osd.14

-1 7.70544 root default

-3 2.56848 host sumi1

12 nvme 0.72769 osd.12

-5 2.56848 host sumi2

13 nvme 0.72769 osd.13

-7 2.56848 host sumi3

14 nvme 0.72769 osd.14

如果要让pool将对象保存在指定设备类上，需要用指定设备类创建ruleset。

ceph osd crush rule create-replicated <rule-name> <root> <failure-domain> <class>

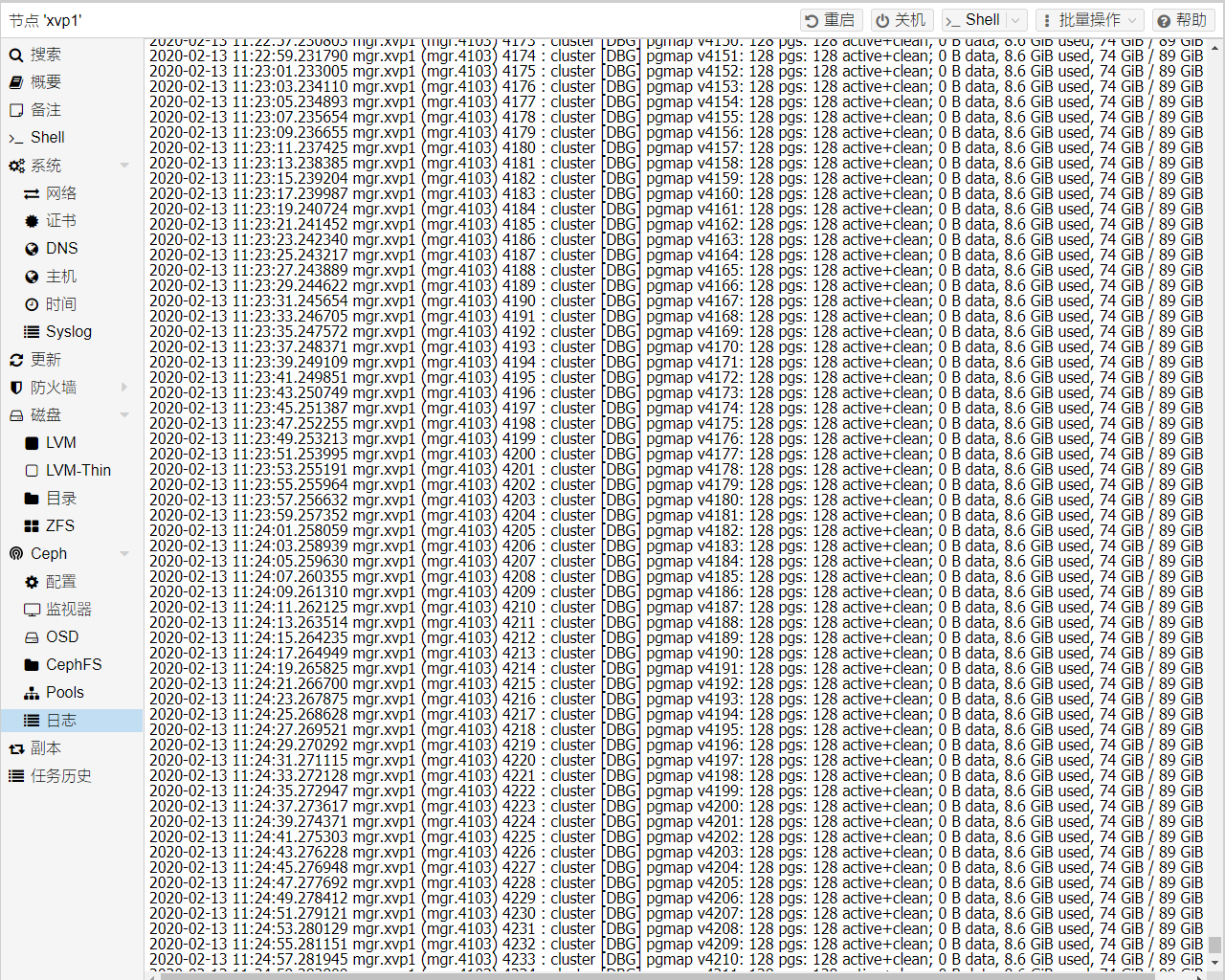
|  |  |
| --- | --- |
| <rule-name> | 规则名称，用于和pool关联（见GUI和CLI） |
| <root> | 规则所属的CRUSH根名称（默认ceph root为“default”） |
| <failure-domain> | 对象所属的故障域（通常为host） |
| <class> | 要使用的OSD存储类名称（例如nvme，ssd，hdd） |

在CRUSH图中创建了规则后，可以指示pool启用该ruleset。

ceph osd pool set <pool-name> crush\_rule <rule-name>

**提示：**如果pool中已有数据，则现有数据将根据规则移动位置。这有可能对集群性能产生重大影响。你也可以新建一个存储池，然后将磁盘逐个迁移过去。

# Ceph 客户端



**图15 Ceph 日志**

接下来可以配置新支点虚拟化使用pool存储虚拟机或轻虚机镜像。通过GUI增加RBD存储即可（参见 基于Ceph RADOS块设备的后端存储 (RBD)）。

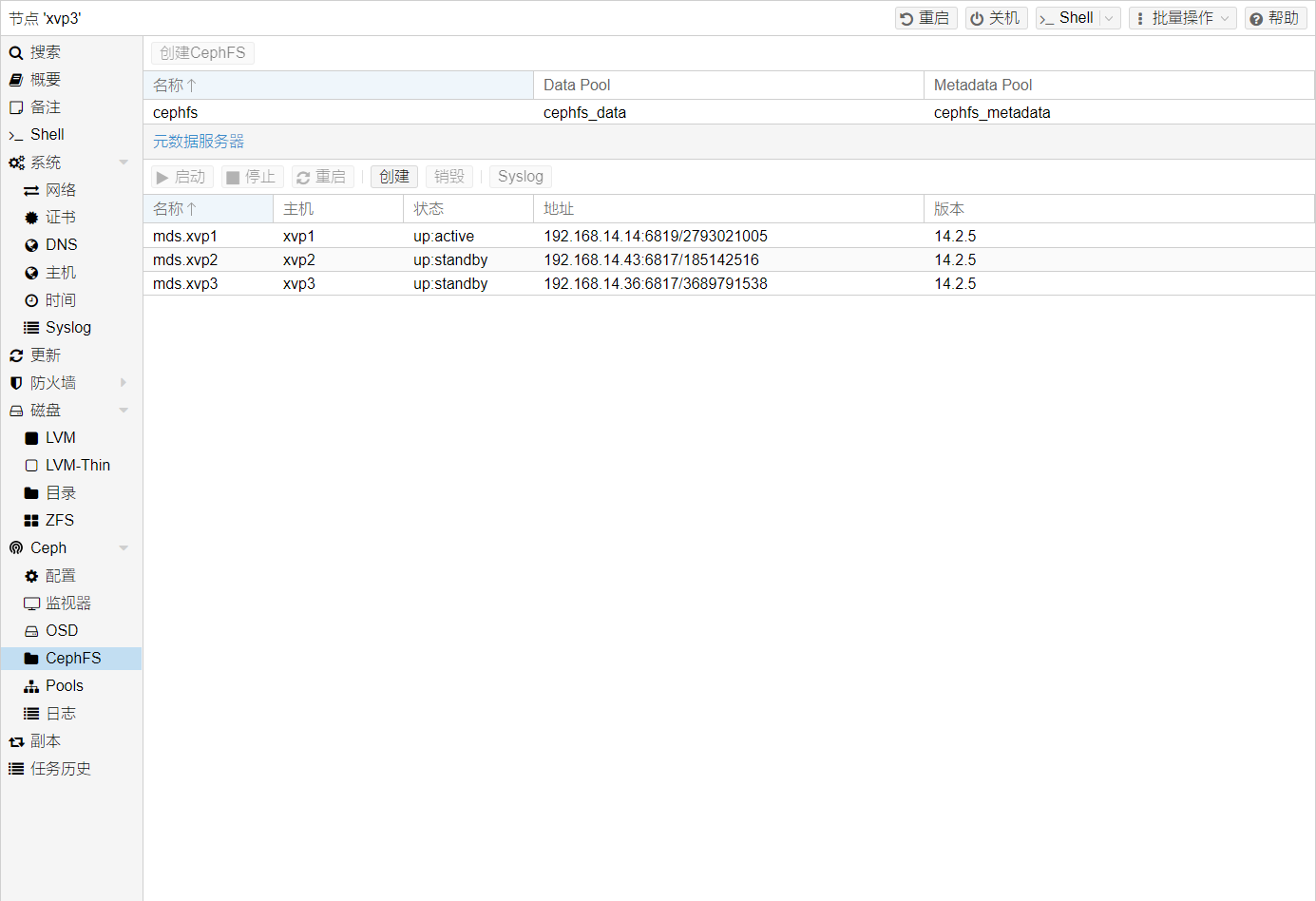
也可以将keyring复制到外部Ceph集群指定位置。如果Ceph就安装在新支点虚拟化节点，该操作将自动完成。

文件名称需要采用 <storage\_id> + .keyring 的格式。其中 <storage\_id> 配置文件 /etc/uvirt/storage.cfg 中rbd:后面的存储名称。下面例子中采用 my-ceph-storage 的名称。

mkdir /etc/uvirt/priv/ceph

cp /etc/ceph/ceph.client.admin.keyring /etc/uvirt/priv/ceph/my-ceph-storage.keyring

# CephFS



**图16 CephFS 控制界面**

Ceph也支持基于RADOS块设备的文件系统。元数据服务器（ MDS ）可以将RADOS块映射为文件和目录，并提供兼容POSIX标准的多副本文件系统。用户可以很容易在Ceph上建立高可用的集群共享文件系统。元数据服务器可以确保文件平均分布在整个Ceph集群上，在高负载下也能有效避免单一节点过载，而这往往NFS等传统共享文件系统的一大痛点。

新支点虚拟化支持创建超融合的CephFS，也支持挂载外部 CephFS as storage，并可用于保存备份，ISO文件，轻虚机模板等。

## 元数据服务器（MDS）

为了使用CephFS，至少需要配置一个元数据服务器。通过新支点虚拟化的GUI界面，可以很容易创建元数据服务器，只需依次在打开 Node→CephFS 控制面板即可找到操作界面，也可以通过执行以下命令：

uvirtceph mds create

一个集群内也可以创建多个元数据服务器。但默认设置同一时间只能有一个元数据服务器处于活动状态。如果 MDS ，或者其所在节点失去响应（或者崩溃），某个 standby 的 MDS 将自动转为 active 。可以通过设置 hotstandby 参数加速主备切换，或者在对应MDS的 ceph.conf 文件中进行如下设置：

mds standby replay = true

启用该设置后，该备用MDS将持续轮训活动MDS的状态，相当于处于一种温备状态，能够在主MDS宕机后更快接管。当然，持续轮训会消耗一定资源，并对活动 MDS 的性能产生一定影响。

### Multiple Active MDS

从Luminous（12.2.x）版本开始，可以有多个活动的元数据服务器同时运行，但这通常只在多个并发客户端的场景中有意义， MDS 很少成为性能瓶颈。如果想使用该特性，请参考 Ceph文档。

## 创建CephFS

在新支点虚拟化下，可以通过Web GUI、CLI、外部API接口等多种方式轻松创建CephFS。前置条件如下：

* 安装Ceph软件包，如果之前已经安装了Ceph，也可以再次安装升级到最新版，同时确保CephFS相关的包也被安装上。
* 设置Monitors
* 设置OSDs
* 创建至少一个MDS

完成以上操作后，就可以通过Web GUI的 Node -> CephFS 面板或者命令行工具 uvirtceph 创建CephFS了。示例如下：

uvirtceph fs create --pg\_num 128 --add-storage

上面的命令将创建一个名为 cephfs 的CephFS存储池，数据存储名称为 cephfs\_data ，配置128个数据集，元数据存储名称为 cephfs\_metadata, 配置32个数据集，也就是数据存储的四分之一。可查看 新支点虚拟化托管的Ceph Pool 或Ceph文档以确定适当的存储集数量（pg\_num） Ceph PG组。此外， --add-storage 参数将自动把创建成功的CephFS添加到新支点虚拟化的存储配置文件中。

## 删除 CephFS

**警告：**删除操作后，CephFS上所有数据都将不可继续使用。且该操作无法撤销。

如果确认需要删除 CephFS，首先需要停止或删除所有的元数据服务器（ MDS ）。该操作可通过Web GUi或如下命令进行：

uvirtceph mds destroy NAME

该命令需要在所有运行了MDS的新支点虚拟化节点上执行。

然后可用如下命令删除CephFS：

ceph fs rm NAME --yes-i-really-mean-it

该命令只需在某个运行了Ceph的节点执行一次即可。然后可以删除创建的数据存储和元数据存储。可在Web GUI操作或运行以下命令行：

uvirtceph pool destroy NAME

# Ceph维护

## 替换OSD

一个通用的Ceph维护任务是替换OSD硬盘。如果一个硬盘损坏了，你可以参考 销毁OSD 执行相关操作。Ceph将会在其它OSD上生成对应的副本。一旦OSD故障被检测到或OSD被主动停止， Ceph将自动开启这种再平衡动作(rebalance)。

注意：使用pool默认的 size/min\_size(3/2) ，仅当有 size+1 个节点处于可用时，数据恢复才会开始执行。 原因是Ceph对象平衡器 CRUSH 默认将一个完整节点作为 故障域 。

要更换仍在工作中的硬盘，请在GUI界面上按照 销毁OSD指示操作。 在停止OSD并将其销毁之前，需要等待集群显示为 HEALTH\_OK 。

命令行操作命令如下：

ceph osd out osd.<id>

如果OSD已经被安全移除，可以用下面的命令检查：

ceph osd safe-to-destroy osd.<id>

一旦以上检查告诉您已安全移除OSD，则可以继续以下命令：

systemctl stop ceph-osd@<id>.service

uvirtceph osd destroy <id>

用新盘替换旧盘，并参照前面的 创建OSD 中的步骤执行。

## 修剪丢弃

最好在虚拟机或轻虚机上定期运行 fstrim (丢弃)命令。这将释放文件系统不再使用的数据块。 这将会降低数据占用率和资源负载。大多数现代操作系统都会定期向硬盘发布此类丢弃命令。您 只需要确保虚机启用了 磁盘丢弃选项。

## 清洗和深度清洗

Ceph通过 清理(scrub) PG组来确保数据完整性。Ceph检查PG组中的每个对象的健康状况。 清理有两种形式：每日低成本的元数据检查和每周的深度数据检查。每周一次的深度清洗(deep scrub) 会读取对象并使用指纹(checksum)来确保数据完整性。如果正在运行的清洗影响到了业务的性能， 可以调整执行清洗的时间。

# Ceph监控和故障排查

最好从安装Ceph后就开始持续监控Ceph的健康状态。可以通过ceph自带工具，也可以通过新支点虚拟化 API监控。

以下命令可以查看集群是否健康(HEALTH\_OK)，或是否存在警告(HEALTH\_WARN)或错误(HEALTH\_ERR)。 如果集群状态不健康，以下命令还可以查看当前事件和活动情况概览。

# single time output

uvirt# ceph -s

# continuously output status changes (press CTRL+C to stop)

uvirt# ceph -w

如果要查看进一步详细信息，可以查看 /var/log/ceph/ 下的日志文件，每个ceph服务都会在该目录下有一个日志文件。如果日志信息不够详细， 还可以进一步调整 日志记录级别。 可以在官网查看 Ceph集群故障排查。

# 证书管理

## 集群通信认证

新支点虚拟化集群内置了自签名的CA证书，并以此为每个节点生成一个签名证书。 这些证书用于集群内 uvirtproxy 服务通信加密，以及SPICE控制台的Shell/Console通信加密。

CA证书和密钥保存在集群文件系统(udocfs)上。

## 认证API和Web GUI界面

新支点虚拟化每个服务器的 uvirtproxy 服务都提供了REST API接口和Web GUI界面。

在 uvirtproxy 中可用的CA认证方式如下：

1. 默认情况下，使用指定节点认证文件/etc/uvirt/nodes/NODENAME/uvirt-ssl.pem。 该证书由集群CA签发，因此浏览器和操作系统默认不信任该证书。需要手工将新支点虚拟化的CA根证书安装到浏览器中成为受信任的根证书。
2. 使用外部证书（例如，由商业CA签发的证书）。
3. 使用ACME（例如，Let’s Encrypt）获得可信证书。

第2、3种认证方式使用 /etc/uvirt/local/uvirtproxy-ssl.pem （和 /etc/uvirt/local/uvirtproxy-ssl.key ，这个key需要免密）。

可以通过新支点虚拟化节点管理命令管理证书（详情见man手册 uvirtnode(1) ）。

**注意：**不要替换或手工修改自动生成的节点证书文件 /etc/uvirt/local/uvirt-ssl.pem 和 /etc/uvirt/local/uvirt-ssl.key ，以及集群CA文件 /etc/uvirt/uvirt-root-ca.pem 和 /etc/uvirt/priv/uvirt-root-ca.key 。

### 安装集群CA根证书到浏览器

推荐使用此方法解决证书安全提醒问题。新支点虚拟化的集群CA根证书内容如下：

-----BEGIN CERTIFICATE-----

MIIGVjCCBD6gAwIBAgIJAOVZe2OoL1ZAMA0GCSqGSIb3DQEBBQUAMHkxIjAgBgNV

BAMTGVhpaUNsb3VkIFZpcnR1YWwgUGxhdGZvcm0xLTArBgNVBAsTJGE5ZjA5ZWI1

LWYxNmQtNDdmMS04NzNiLTU4OGIzZjIyMTJkMjEkMCIGA1UEChMbWGlpQ2xvdWQg

Q2x1c3RlciBNYW5hZ2VyIENBMCAXDTIwMDYyMzAyMTY0OVoYDzIwNTAwNjE2MDIx

NjQ5WjB5MSIwIAYDVQQDExlYaWlDbG91ZCBWaXJ0dWFsIFBsYXRmb3JtMS0wKwYD

VQQLEyRhOWYwOWViNS1mMTZkLTQ3ZjEtODczYi01ODhiM2YyMjEyZDIxJDAiBgNV

BAoTG1hpaUNsb3VkIENsdXN0ZXIgTWFuYWdlciBDQTCCAiIwDQYJKoZIhvcNAQEB

BQADggIPADCCAgoCggIBAMnKOCB8wcJhGaQMOSsKkifUO/TV2RcJwO42S5vfNrn5

atWO1+1MilHUZvt6ofAxutRelAuiqDnqDcfTlrF2wLAFjdhpSk0VhngTwRJRX595

PiEeNxFIK/g9dbYpxkZfy6/fgs8FyqJC4JmUUA22r8BsZssEbAeWzvocZAczwUBa

+eFfyAsAoBoLsLwguJo7npLavf+jmAzwTDDjDKHWW2/sCvgTcWGOFYeZqyaegyTV

/AdDjjzv2myNB36pQvGecVgEb3bowyEQDk9OeMJeM7J6aiHE/1JMseStuGw+nRWM

qJ0RpC6fi8wbJl7i1xAOKMR8k/OD1jdA3Fs3YlKtN2GADaPHj0Ctdk9CT0n8o0xv

tjR1kqJncjopRZ3gLyY74c8ceLZ7QmSGmFfwKlyiImrzqjFivvOj3LfuHFRAbube

vtVfRCqvw37YBsX5ExtDmRMJMU0C/vVzQbWL77V/ciLULYoay4nouhuzp4T2Kmuj

Qa4mB6YhVNDdtoJAl0PH9ER0dlaft/jAOdAIH3BqZ4FwfW4Sx3KQISmsjzoOhak3

7VnWAuC577SUoJTJv48daMZTD4ouIi2q/V8vT+5mWhw731yHCALMaA58mi6urPtC

2yFbwgFPrp8erH1sXr8uF2TMAxFIKsF/Uw6HCKgAG/xMy8sDWEdh131cdn8e60Wt

AgMBAAGjgd4wgdswHQYDVR0OBBYEFDL27KLnC4oGZI48a5kxfxJFCkKtMIGrBgNV

HSMEgaMwgaCAFDL27KLnC4oGZI48a5kxfxJFCkKtoX2kezB5MSIwIAYDVQQDExlY

aWlDbG91ZCBWaXJ0dWFsIFBsYXRmb3JtMS0wKwYDVQQLEyRhOWYwOWViNS1mMTZk

LTQ3ZjEtODczYi01ODhiM2YyMjEyZDIxJDAiBgNVBAoTG1hpaUNsb3VkIENsdXN0

ZXIgTWFuYWdlciBDQYIJAOVZe2OoL1ZAMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcN

AQEFBQADggIBAEGiJ7aK7EYxXAn3sbXEbBXBFa5cok/klan1oqr1AP13IC75Qxoo

z90wcMs0VBH7iKCF4y0gO5ebkBBT2o+ImXz03X44KSLUAJVY4VMc5dAvXG1iFvab

4Wx2MkOjLsj/UZ1pwUrOcnjcSuteT/q9CpxT+72RUZaxQnGR2BoO/29kUEwmY3pm

QT8tJ1i7syigltErCgNcVrNy5Yxkb5YcJt8yAhzbrZBGbwjb4qY/JS3LmJ45f0b7

9CX8FgWjYP299u2pZEbwShadVy3trODj8dI5QFljN88KVs9509KHsqsm/dyHUl9T

NcIVV9I1khZevW/T7wfI4YnxhRVWMlULfXdS4WhS4+EPjUtgdfaCRw/tp5vtyL/u

tRepP1DBYXHOXRN7l/MPzo6rv+kOkUyvYBOeWtUO1UGr6ozbe/dMHhJuSmeJI0cD

ZdP72MlTgNKIyIT806ToJ/o81PhzgTgcoxL0J7VTsqrl4jGj8B15s4gXaTUnwtIj

3lSMOi9ru9GChdKNe+2sHJIIo/97K80mp7cWSfaFSpLiJkMoh7ooniQ/awY53TFk

WHVOrHb7Ds1Tf6bKAgrYR4ZAo7pCl2lO5HryIMg2OrQCIXajj80MxABB6C5Ib1GZ

WM2C60+L+iEU83iMh9lErYwNzhP3Bg88Dgy0yZ6pf+oefR/xl4oWrqM+

-----END CERTIFICATE-----

在桌面上新建一个文件 uvirt-ca.cer ，并将上面的CA根证书内容拷贝到文件中。 针对不同浏览器的根证书导入方法，将CA根证书安装到浏览器中使之成为"受信任的根证书"。 这样我们用浏览器访问新支点虚拟化时，就不会再提醒证书不安全了。

Chrome系列浏览器的CA根证书安装方法："设置" → "隐私和安全" → "管理证书" → "受信任的根证书颁发机构" → "导入"

Firefox系列浏览器的CA根证书安装方法： "选项" → "高级" → "证书" → "查看证书" → "导入" → "信任使用此CA标识的网站"

IE系列浏览器的CA根证书安装方法： "鼠标右键点击证书文件安装" → "将所有的证书放入下列存储" → "受信任的根证书颁发机构"

MacOSX Safari根证书安装方法： "钥匙串" → "证书" → "导入项目" → "右键此证书" → "显示简介" → "信任" → "使用此证书时" → "始终信任"

### 通过ACME获取可信证书

新支点虚拟化内置了自动化证书管理 Automatic Certificate Management Enviroment，即ACME协议， 可以通过Let’s Encrypt轻松获取可信TLS证书，从而被大多数操作系统和浏览器直接接受。

目前实现的ACME端点是 Let‘s Encrypt(LE) 及其staging环境（详见https://letsencrypt.org），均使用标准的 HTTP challenge 模式。

使用 Let’s Encrypt 前需准备好以下条件：

* 80端口对互联网开放访问权限。
* 没有其他服务使用80端口。
* 请求的（子）域名能够被解析为节点的互联网公共IP地址。
* 需要接受 Let’s Encrypt 的 ToS 。

当前GUI仅使用默认ACME账号。

**示例 使用 uvirtnode 命令获取 Let’s Encrypt 证书**

root@udo:~# uvirtnode acme account register default mail@example.invalid

Directory endpoints:

0) Let's Encrypt V2 (https://acme-v02.api.letsencrypt.org/directory)

1) Let's Encrypt V2 Staging (https://acme-staging-v02.api.letsencrypt.org/directory)

2) Custom

Enter selection:

1

Attempting to fetch Terms of Service from 'https://acme-staging-v02.api.letsencrypt.org/directory'..

Terms of Service: https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the above terms? [y|N]y

Attempting to register account with 'https://acme-staging-v02.api.letsencrypt.org/directory'..

Generating ACME account key..

Registering ACME account..

Registration successful, account URL: 'https://acme-staging-v02.api.letsencrypt.org/acme/acct/xxxxxxx'

Task OK

root@udo:~# uvirtnode acme account list

default

root@udo:~# uvirtnode config set --acme domains=example.invalid

root@udo:~# uvirtnode acme cert order

Loading ACME account details

Placing ACME order

Order URL: https://acme-staging-v02.api.letsencrypt.org/acme/order/xxxxxxxxxxxxxx

Getting authorization details from

'https://acme-staging-v02.api.letsencrypt.org/acme/authz/xxxxxxxxxxxxxxxxxxxxx-xxxxxxxxxxxxx-xxxxxxx'

... pending!

Setting up webserver

Triggering validation

Sleeping for 5 seconds

Status is 'valid'!

All domains validated!

Creating CSR

Finalizing order

Checking order status

valid!

Downloading certificate

Setting uvirtproxy certificate and key

Restarting uvirtproxy

Task OK

### 从 staging 切换至常规ACME目录

并不支持直接修改账户的ACME目录。如果想将一个账户的ACME目录从 staging 修改为常规方式，需要先删停用账户，然后新建一个账户。

该步骤也适用于修改GUI中的默认ACME账户。

**示例 将默认的ACME账户从staging切换至常规目录**

root@udo:~# uvirtnode acme account info default

Directory URL: https://acme-staging-v02.api.letsencrypt.org/directory

Account URL: https://acme-staging-v02.api.letsencrypt.org/acme/acct/6332194

Terms Of Service: https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Account information:

ID: xxxxxxx

Contact:

- mailto:example@udoplus.com

Creation date: 2018-07-31T08:41:44.54196435Z

Initial IP: 192.0.2.1

Status: valid

root@udo:~# uvirtnode acme account deactivate default

Renaming account file from '/etc/uvirt/priv/acme/default' to '/etc/uvirt/priv/acme/\_deactivated\_default\_4'

Task OK

root@udo:~# uvirtnode acme account register default example@udoplus.com

Directory endpoints:

0) Let's Encrypt V2 (https://acme-v02.api.letsencrypt.org/directory)

1) Let's Encrypt V2 Staging (https://acme-staging-v02.api.letsencrypt.org/directory)

2) Custom

Enter selection:

0

Attempting to fetch Terms of Service from 'https://acme-v02.api.letsencrypt.org/directory'..

Terms of Service: https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the above terms? [y|N]y

Attempting to register account with 'https://acme-v02.api.letsencrypt.org/directory'..

Generating ACME account key..

Registering ACME account..

Registration successful, account URL: 'https://acme-v02.api.letsencrypt.org/acme/acct/39335247'

Task OK

### ACME证书自动更新

新支点虚拟化服务器节点成功配置ACME证书后（使用 uvirtnode 命令或GUI均可），证书将通过 uvirt-daily-update.service 自动更新。 目前，证书将在过期后或到期前30天内尝试自动更新。

# 主机引导程序

新支点虚拟化目前有两种引导程序可以选用，具体根据安装时采用的磁盘配置决定。

对于使用ZFS为根文件系统的EFI系统，将使用 systemd-boot 。其他部署方式将使用标准 grub 引导程序 （在Debian上安装新支点虚拟化时将使用这种方式）

## 安装程序分区方案

新支点虚拟化安装程序会在系统引导盘创建3个分区。引导磁盘类型如下：

* 可使用 ext4 或 xfs 文件系统磁盘。
* 可使用ZFS的第一个 vdev ：
  + 第一个磁盘，配置为RAID0
  + 所有磁盘，配置为RAID1，RAIDZ1，RAIDZ2，RAIDZ3
  + 前两个磁盘，配置为RAID10

创建的分区有：

* BIOS Boot分区，容量1MB（gdisk类型EF02）
* EFI系统分区，容量512MB（ESP，gdisk类型EF00）
* 第三个分区，容量按照 hdsize 参数决定，或使用所选磁盘的剩余空间

在BIOS模式（ --target i386-pc ）下， grub 将安装在所有可启动磁盘的BIOS Boot分区，以兼容旧的系统。

### Grub

多年来，grub 是Linux系统事实上的标准引导程序，文档十分完善。

Grub将从 /boot 目录获取Linux内核和initrd镜像，配置文件 /boot/grub/grub.cfg 包含了内核安装时的相关信息。

#### 配置信息

修改 grub 配置信息，可以通过 /etc/default/grub ，或配置 /etc/default/grub.d 下的相关文件。 并可以运行如下命令生成新的 /boot/grub/grub.cfg ：

update-grub

### Systemd-boot

Systemd-boot 是一个轻量级的EFI引导程序。它从EFI服务分区（ESP）读取内核和initrd镜像。直接从ESP加载内核的好处是， 不需要为访问存储设备而反复实现驱动。在使用ZFS作为根文件系统的场景中，这就意味着可以直接使用主存储池的所有功能特性， 不再局限于 grub 实现的功能子集，或专门创建独立的boot存储池。

在使用磁盘阵列时（RAID1，RAID10，RAIDZ），所有可引导启动的磁盘（属于第一个 vdev 的磁盘）都会分配一个ESP， 以确保第一个启动设备故障时系统仍然能够启动。各个ESP的数据由内核钩子脚本 /etc/kernel/postinst.d/zz-uvirt-efiboot 自动同步。 该脚本将指定版本的内核和initrd镜像复制到每一个ESP根目录的 EFI/udo/ 目录，并创建相应的配置文件， 路径是 loader/entries/udo-\*.conf 。辅助脚本 uvirt-efiboot-tool 用于管理同步的ESP和相关内容。

默认配置的内核版本如下：

* 当前运行的内核
* 软件包升级新安装的内核
* 已安装的版本最新的两个内核
* 最近两个内核系列（如4.15、5.0）的最新版本内核
* 手工指定的内核（详见后续内容）

日常运行中，ESP将不会被挂载到文件系统，而 grub 会将ESP挂载到 /boot/efi 下。这能够有效避免ESP的vfat文件系统损坏， 如系统崩溃可能导致文件系统故障，并避免了主引导设备故障时手工调整 /etc/fstab 配置的麻烦。

#### 配置信息

sysemd-boot 的配置信息保存在 loader/loader.conf 中。详细配置信息参见man手册的 loader.conf(5) 页面中。

每条启动引导记录都保存在 loader/entries/ 的独立目录下。

引导记录配置文件示例 entry.conf 如下（ / 相对于ESP的根目录）：

title UDO

version 5.0.15-1-uvirt

options root=ZFS=rpool/ROOT/uvirt-1 boot=zfs

linux /EFI/udo/5.0.15-1-uvirt/vmlinuz-5.0.15-1-uvirt

initrd /EFI/udo/5.0.15-1-uvirt/initrd.img-5.0.15-1-uvirt

#### 手工指定可启动内核

可以使用 uvirt-efiboot-tool kernel add 命令将指定内核和 initrd image 添加到可启动内核列表中。

以下命令将 5.0.15-1-uvirt 版本的内核添加到内核列表，并同步更新所有ESP：

uvirt-efiboot-tool kernel add 5.0.15-1-uvirt

uvirt-efiboot-tool kernel list 将列出当前可用的所有版本内核：

# uvirt-efiboot-tool kernel list

Manually selected kernels:

5.0.15-1-uvirt

Automatically selected kernels:

5.0.12-1-uvirt

4.15.18-18-uvirt

uvirt-efiboot-tool kernel remove 命令能够从内核列表删除指定版本内核，示例如下：

uvirt-efiboot-tool kernel remove 5.0.15-1-uvirt

手工添加或删除内核后，需要执行 uvirt-efiboot-tool refresh 更新所有的ESP。

#### 创建新分区并用于同步ESP

如果需要将一个分区格式化并初始化为同步ESP，比如在替换 rpool 中的故障 vdev 后， 或将一个不支持同步机制的旧文件系统转换为ESP，可以使用 uvirt-kernel-helpers 软件包中的 uvirt-efiboot-tool 命令。

警告：命令 format 将格式化指定分区，请务必确保使用正确的设备分区参数！

以下示例将一个空分区 /dev/sda2 格式化为ESP：

uvirt-efiboot-tool format /dev/sda2

以下命令将一个已存在的但未挂载的ESP分区 /dev/sda2 加入新支点虚拟化的内核升级同步机制：

uvirt-efiboot-tool init /dev/sda2

命令执行后， /etc/kernel/uvirt-efiboot-uuids 将新增一行对应于新增分区的UUID信息， init 命令将自动触发所有ESP更新。

#### 更新所有ESP配置

如需复制并配置所有可用内核，同时确保 /etc/kernel/uvirt-efiboot-uuids 所列出的所有ESP分区保持同步，可执行以下命令：

uvirt-efiboot-tool refresh

（效果等同于基于 grub 引导的系统上执行 update-grub ）

该命令一般在修改过内核命令行，或需要同步所有内核和initrd镜像时执行。

注意：update-initramfs 和 apt （当需要时）都会自动触发更新。

### 编辑内核命令行

根据使用的引导程序不同，可以按以下方式修改内核命令行：

#### Grub

使用Grub引导时，内核命令行参数保存在 /etc/default/grub 的 GRUB\_CMDLINE\_LINUX\_DEFAULT 参数中。运行 update-grub 命令， 可以将其内容附加在 /boot/grub/grub.cfg 中所有的linux项目后。

#### Systemd-boot

使用 systemd-boot 引导时，内核命令行参数保存在 /etc/kernel/cmdline 中。运行 /etc/kernel/postinst.d/zz-uvirt-efiboot 会将其添加到配置文件 loader/entries/udo-\*.conf 中的 option 行中。

新支点虚拟化防火墙

新支点虚拟化防火墙为你的IT基础设施提供了一种简单易用的防护手段。你既可以为集群内的所有主机设置防火墙策略，也可以为单个虚拟机和轻虚机定义策略。防火墙宏，安全组，IP集和别名等特性将大大简化策略配置管理。 尽管所有的防火墙策略都保存在集群文件系统，但基于 iptables 的防火墙服务在每个节点都是独立运行的，从而为虚拟机提供了完全隔离的防护。这套分布式部署的防火墙较传统防火墙提供了更高的带宽。 新支点虚拟化防火墙完全支持IPv4和IPv6。IPv6的支持是完全透明的，我们默认自动对两种协议通信同时进行过滤和检测。所以没有必要为IPv6专门建立并维护防火墙策略。

# 区域

新支点虚拟化防火墙将网络划分为不同区域，对每个区域，你都可以对流入/流出流量定义防火墙策略。

## Host

流出/流入集群节点的网络通信

## VM

流出/流入虚拟机的网络通信

# 配置文件

防火墙相关的配置文件全部保存在新支点虚拟化集群文件系统中，所以能够自动在所有节点间同步复制， 而防火墙管理服务 uvirt-firewall 将在防火墙策略改变后自动更新底层 iptables 策略。 你可以在WebGUI界面完成所有的防火墙配置（例如通过， 数据中心 → 防火墙 ，或者通过， 节点 → 防火墙 ）， 或者也可以直接用你喜欢的编辑器编辑配置文件。防火墙配置文件按段落(section)把 键-值 策略对组织起来。 以 # 字符开头的行和空行被当作注释处理。每个段落开头第一行格式都是 [段落名] 。

## 集群级别的防火墙配置

作用域为整个集群的防火墙配置保存在

/etc/uvirt/firewall/cluster.fw

该配置文件由以下段落构成：

[OPTIONS] 该段落用于设置整个集群的防火墙配置项。

ebtables: <boolean> (default = 1) 集群范围内启用ebtables。

enable: <integer> (0 -N) 启用/禁用集群范围的防火墙。

log\_ratelimit: [enable=]<1|0> [,burst=<integer>] [,rate=<rate>] 设置日志记录速度阀值。

burst=<integer> (0 - N) (default = 5) 将被记录的初始突发包。

enable=<boolean> (default = 1) 启用或禁用阀值

rate=<rate> (default = 1/second) 突发缓冲区重新填充频度。

policy\_in: <ACCEPT | DROP | REJECT> 流入方向的防火墙策略。

policy\_out: <ACCEPT | DROP | REJECT> 流出方向的防火墙策略。

[RULES] 该段落用于设置所有节点公共的防火墙策略。

[IPSET <name>] 整个集群范围内有效的IP集合定义。

[GROUP <name>] 整个集群范围内有效的组定义。

[ALIASES] 整个集群范围内有效的别名定义。

### 启用防火墙

防火墙默认是被完全禁用的。你可以按如下方式设置启用参数项：

[OPTIONS]

# enable firewall (cluster wide setting, default is disabled)

enable: 1

启用防火墙后，默认所有主机的通信都将被阻断。唯一例外是集群网络内的WebGUI（端口6666）和ssh（端口22）访问可以继续使用。

如果你希望远程管理新支点虚拟化服务器，你需要首先配置防火墙策略，允许远程IP访问WebGUI（端口6666）。根据需要，你还可以开通ssh（端口22）或SPICE（端口3128）的访问权限。

请在启用防火墙前先打开到新支点虚拟化服务器的一个SSH连接，这样即使策略配置有误，也还可以通过该连接访问服务器。

为简化配置，你可以创建一个名为“管理地址”的IPSet，并把所有的远程管理终端IP地址添加进去。这样就可以创建策略允许所有的远程地址访问WebGUI。