

Установка и настройка KVM

Виртуализация с использованием KVM (Kernel-based Virtual Machine) позволяет создавать и управлять виртуальными машинами на платформе Linux. В этой документации подробно рассмотрены требования к оборудованию и ПО, процесс установки KVM на различных дистрибутивах Linux, настройка сети и хранения для KVM, а также тонкая настройка производительности. Мы будем использовать пакетный менеджер [tdnf](#), который является стандартным для НАЙС ОС.

Требования к оборудованию и ПО

Перед установкой KVM необходимо убедиться, что ваш сервер или рабочая станция соответствует минимальным требованиям к оборудованию и ПО.

Аппаратные требования

- Процессор с поддержкой виртуализации (Intel VT-x или AMD-V).
- Минимум 2 ГБ оперативной памяти (рекомендуется 4 ГБ и более).
- Достаточно свободного дискового пространства для хранения виртуальных машин (рекомендуется 20 ГБ и более).

Программные требования

- 64-разрядная версия Linux с ядром 2.6.20 или выше.
- Права суперпользователя для установки и настройки KVM.

Установка KVM

Установка KVM может различаться в зависимости от используемого дистрибутива Linux. Ниже приведены инструкции для установки KVM на НАЙС ОС и других популярных дистрибутивах.

Установка KVM на НАЙС ОС

Для установки KVM на НАЙС ОС используйте пакетный менеджер [tdnf](#):

```
sudo tdnf install qemu-kvm libvirt virt-install bridge-utils  
sudo systemctl enable --now libvirtd
```

Настройка сети и хранения для KVM

Для полноценного использования KVM необходимо правильно настроить сеть и системы хранения. Рассмотрим, как это сделать.

Настройка сети

Для настройки сети используем сетевой мост, что позволит виртуальным машинам работать в одной сети с хостовой системой. Следуйте приведенным инструкциям для создания и

настройки сетевого моста.

Создание сетевого моста

На НАЙС ОС создайте файл конфигурации для сетевого моста:

```
sudo nano /etc/sysconfig/network-scripts/ifcfg-br0
```

Добавьте следующие строки в файл:

```
DEVICE=br0  
TYPE=Bridge  
BOOTPROTO=dhcp  
ONBOOT=yes
```

Обновление конфигурации сети

Настройте основной сетевой интерфейс (например, eth0) для использования моста:

```
sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

Обновите файл следующими строками:

```
DEVICE=eth0  
BOOTPROTO=none  
ONBOOT=yes  
BRIDGE=br0
```

Перезапустите сетевые службы для применения изменений:

```
sudo systemctl restart network
```

Настройка хранения

Для хранения виртуальных машин и их дисков используйте LVM (Logical Volume Manager) или другой тип хранилища. Рассмотрим создание логического тома для хранения виртуальных машин.

Создание логического тома

Для создания нового логического тома выполните следующие команды:

```
sudo pvcreate /dev/sdb  
sudo vgcreate kvm-vg /dev/sdb  
sudo lvcreate -L 50G -n kvm-lv kvm-vg
```

Создайте файловую систему на новом логическом томе и смонтируйте его:

```
sudo mkfs.ext4 /dev/kvm-vg/kvm-lv
sudo mkdir /mnt/kvm-storage
sudo mount /dev/kvm-vg/kvm-lv /mnt/kvm-storage
```

Тонкая настройка производительности

После установки и настройки KVM важно оптимизировать производительность виртуальных машин. Рассмотрим несколько рекомендаций и настроек для повышения производительности.

Использование Virtio драйверов

Virtio драйверы обеспечивают высокую производительность ввода-вывода в виртуальных машинах. Убедитесь, что ваши виртуальные машины используют Virtio для дисков и сетевых интерфейсов.

При создании виртуальной машины используйте следующие параметры:

```
sudo virt-install --name=myvm --ram=2048 --vcpus=2 --disk path=/mnt/kvm-storage/myvm.img,bus=virtio --network bridge=br0,model=virtio --cdrom=/path/to/ubuntu.iso
```

Настройка HugePages

Использование HugePages может значительно улучшить производительность за счет уменьшения накладных расходов на управление памятью. Для включения HugePages выполните следующие шаги:

Отредактируйте файл `/etc/sysctl.conf` и добавьте следующие строки:

```
vm.nr_hugepages=128
```

Примените изменения:

```
sudo sysctl -p
```

Убедитесь, что KVM использует HugePages, отредактировав конфигурацию виртуальной машины:

```
...
```

Настройка CPU Pinning

Закрепление виртуальных CPU (vCPU) за физическими ядрами процессора (CPU pinning) может улучшить производительность, особенно на системах с высокой нагрузкой.

Используйте `virsh` для настройки CPU pinning:

```
sudo virsh vcpupin myvm 0 2
sudo virsh vcpupin myvm 1 3
```

Настройка I/O Thread

Для улучшения производительности ввода-вывода создайте отдельные потоки I/O для дисков виртуальных машин. Отредактируйте конфигурацию виртуальной машины и добавьте:

```
...
2
...
```

Настройка и оптимизация KVM может значительно повысить производительность и эффективность виртуальных машин. Следуя приведенным рекомендациям и инструкциям, вы сможете развернуть мощную виртуальную инфраструктуру, удовлетворяющую вашим требованиям. Правильная настройка сети и систем хранения, использование передовых драйверов и технологий помогут вам достичь наилучших результатов.