

Proxmox VE Helper-Scripts вспомогательные скрипты

🌐 bafista.ru/proxmox-ve-helper-scripts-vspomogatelnye-skripty

25 мая 2024 г.

В этой небольшой короткой заметке я расскажу про вспомогательные скрипты Proxmox VE Helper-Scripts



Proxmox VE Helper-Scripts — это набор скриптов, предназначенных для автоматизации и упрощения администрирования и управления виртуализированной инфраструктурой на базе Proxmox Virtual Environment (VE). Proxmox VE является открытой платформой виртуализации, объединяющей в себе виртуализацию на основе контейнеров (LXC) и виртуализацию на основе гипервизора KVM.

Helper-скрипты предоставляют различные функции и инструменты для управления виртуальными машинами, хранилищами данных, сетевыми настройками и другими аспектами системы Proxmox VE. Они могут помочь автоматизировать рутинные задачи, упростить настройку и мониторинг системы, а также повысить эффективность работы администратора.

Эти скрипты обычно разрабатываются сообществом пользователей Proxmox VE и могут быть доступны для загрузки из различных источников. Их использование может значительно упростить работу с Proxmox VE и сделать процесс управления виртуализированной инфраструктурой более эффективным.

На сайте Proxmox VE Helper-Scripts собрано очень много различного рода скриптов от оптимизации производительности, до установки одной командой различных служб и виртуальных машин. Зайдите на сайт Proxmox VE Helper-Scripts наверняка

вы найдете парочку полезных для вас скриптов. Я выложу в этой заметке только те, которые мне нужны.

Proxmox VE Post Install

Первое, что я делаю после установки Proxmox запускаю скрипт Proxmox VE Post Install

Этот скрипт предоставляет параметры для управления репозиториями Proxmox VE, включая отключение Enterprise Repo, добавление или исправление источников PVE, включение No-Subscription Repo, добавление тестового репозитория, отключение подписки pag, обновление Proxmox VE и перезагрузку системы.

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/post-pve-install.sh)"
```

Proxmox VE Processor Microcode

В процессорах тоже бывают баги и различные обновления, почему бы не обновить их

Процессор Microcode – это уровень низкоуровневого программного обеспечения, который работает на процессоре и предоставляет исправления или обновления его прошивки. Обновления микрокода могут исправить аппаратные ошибки, повысить производительность и улучшить функции безопасности процессора.

Важно отметить, что доступность механизмов обновления прошивки, таких как Intel Management Engine (ME) или AMD Platform Security Processor (PSP), может варьироваться в зависимости от процессора и его конкретной реализации. Поэтому рекомендуется ознакомиться с документацией для вашего процессора, чтобы подтвердить, можно ли применять обновления прошивки через операционную систему.

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/microcode.sh)"
```

Proxmox VE CPU Scaling Governor

Я заметил, что после этого скрипта система работает гораздо экономичнее.

Регулятор масштабирования ЦП определяет, как частота ЦП регулируется в зависимости от рабочей нагрузки, с целью либо экономии энергии, либо повышения производительности. Масштабируя частоту вверх или вниз, операционная система может оптимизировать использование ЦП и экономить энергию, когда это возможно. Регуляторы общего масштабирования

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/scaling-governor.sh)"
```

Proxmox VE Kernel Clean

Если у вас давно установлен Proxmox и вы уже много раз его обновляли, то наверняка накопилось много ненужных Linux ядер, которые можно и даже нужно удалить.

Очистка неиспользуемых образов ядра полезна для уменьшения длины меню GRUB и освобождения дискового пространства. Удалив старые неиспользуемые ядра, система может сэкономить дисковое пространство и оптимизировать процесс загрузки.

```
bash -c "$(wget -qL0 - https://github.com/tteck/Proxmox/raw/main/misc/kernel-clean.sh)"
```

Proxmox VE Host Backup

Отличный скрипт для резервного копирования настроек Proxmox VE. Жаль только выполняется он не автоматически, а только вручную.

Этот скрипт служит универсальной утилитой резервного копирования, позволяя пользователям указывать как путь резервного копирования, так и каталог, в котором они хотят работать. Эта гибкость позволяет пользователям выбирать конкретные файлы и каталоги, которые они хотят сделать резервными, что делает их совместимыми с широким спектром хостов, не ограничиваясь Proxmox.

```
bash -c "$(wget -qL0 - https://github.com/tteck/Proxmox/raw/main/misc/host-backup.sh)"
```

Теперь вы знаете что такое Proxmox VE Helper-Scripts вспомогательные скрипты