Инсталятор НАЙС.ОС

Краткий обзор

5. Что такое UI config

UI config — это файл в формате `JSON`, в котором можно указать параметры для отображения дополнительных экранов в процессе установки НАЙС.ОС.

Все ключи верхнего уровня являются **необязательными**, то есть вы можете указывать только те, которые вам нужны.

5.1 download_screen — экран загрузки файла по URL

Этот параметр позволяет добавить в интерфейс установки экран, на котором пользователь может вручную ввести URL для загрузки файла. После установки указанный файл будет сохранён в заданном месте на установленной системе.

Важно: этот экран отображается только если также включён `network_screen`.

Поддерживает следующие поля:

- `"title"` (обязательный) — короткий заголовок, отображаемый на экране;
- `"intro"` (обязательный) — краткое описание (1–2 строки), объясняющее пользователю, какой URL нужно ввести;
- `"destination"` (обязательный) — путь в установленной системе, куда будет сохранён файл после загрузки.

Пример использования: вы хотите, чтобы пользователь ввёл ссылку на кастомный скрипт или конфигурацию, который автоматически загрузится и сохранится в `/etc/custom/init.sh` после установки.

1. Что такое Kickstart

Kickstart — это специальный файл, в котором заранее указано, как нужно

устанавливать систему НАЙС.ОС: какой язык, разметка диска, какие пакеты ставить и т.д. Он помогает полностью автоматизировать установку — от начала до конца без участия пользователя.

Это особенно полезно, если вы ставите систему много раз — например, на сервера, в виртуальные машины или в корпоративной сети.

2. Как использовать Kickstart

Чтобы установка прошла автоматически, установщику НАЙС.ОС нужно «подсунуть» файл конфигурации Kickstart. Есть два основных способа:

- **Для виртуальных машин в ESXi** — самый простой способ;
- **Для обычных компьютеров или других виртуалок** — через параметр загрузки `ks=` .

2.1 Если вы устанавливаете в ESXi (виртуализация от VMware)

Здесь всё просто — достаточно добавить файл Kickstart в настройки виртуальной машины. Есть два варианта:

- `guestinfo.kickstart.data=<base64-кодированный текст файла>`
  — содержимое Kickstart-файла вставляется прямо в настройки (но предварительно кодируется в base64);
- `guestinfo.kickstart.url=<ссылка на файл>`
  — указать ссылку, откуда файл можно скачать (например, с локального сервера или интернета).

Поддерживаются ссылки начинающиеся с: `http://` , `https://` , или `https+insecure://` (последний вариант не проверяет SSL-сертификат сайта).

2.2 Если вы устанавливаете на обычный ПК, сервер или в другую виртуалку

В этом случае нужно указать путь к Kickstart-файлу прямо в момент загрузки системы — то есть, когда появляется меню установки (GRUB или аналог).

Просто добавьте в параметры загрузки строку:

```
ks=<путь до файла>
```

Ниже — примеры, как правильно указать путь.

3. Примеры, как указать путь к файлу Kickstart

3.1 Файл лежит прямо в ISO-образе

```
ks=cdrom:/isolinux/sample_ks.cfg
```

Это работает, если вы добавили файл `sample_ks.cfg` внутрь ISO. Установщик сам найдёт его на подключённом диске.

3.2 Файл находится на флешке или внешнем диске

```
ks=/dev/sdb1:/kickstart/naisos.cfg
```

Здесь `/dev/sdb1` — это имя устройства (например, флешки), а `/kickstart/naisos.cfg` — путь до файла на этом устройстве.

3.3 Файл находится на удалённом сервере (по сети)

```
ks=https://example.com/kickstart/naisos.cfg
```

Установщик скачает файл с сервера. Можно использовать обычный `http`, защищённый `https`, или `https+insecure` (если не хотите проверять SSL-сертификат).

1. Настройка имени хоста

Скрипт `hostname.py` устанавливает имя хоста системы, заданное в конфигурации установки. Это имя сохраняется в системных файлах и применяется при запуске.

- `/etc/hostname` — устанавливается заданное имя хоста;

- `/etc/hosts` — добавляется строка для корректного разрешения имени хоста.

Результат: система получает уникальное сетевое имя, которое используется в терминале, журналах и сетевых службах.

2. Установка пароля root

Скрипт `password.py` задаёт пароль для суперпользователя `root` и настраивает срок его действия, если это указано.

- `/etc/passwd` — обновляется информация о пароле root;
- `/etc/shadow` — устанавливается хеш пароля;
- `/etc/login.defs` — при необходимости обновляется срок действия пароля.

Также запускаются системные команды `pwconv`, `grpconv`, `chage` для применения изменений.

3. Добавление публичного ключа

Скрипт `public_key.py` добавляет SSH-публичный ключ в аккаунт `root` и разрешает вход по SSH.

- `/root/.ssh/authorized_keys` — добавляется публичный ключ;
- `/etc/ssh/sshd_config.d/200-allow-root-login.conf` — включается доступ root по SSH.

Это позволяет администратору подключаться к системе сразу после установки, без ввода пароля.

4. Настройка локали

Скрипт `locale.py` задаёт системную локаль (например, `ru_RU.UTF-8`).

- `/etc/locale.conf` — задаётся системная локаль.

Это влияет на язык интерфейса, формат времени и кодировку по умолчанию.

5. Настройка идентификатора системы

Скрипт `machine_id.py` генерирует уникальный идентификатор системы или создаёт пустой файл для автоматической генерации при загрузке.

- `/etc/machine-id` — создаётся или очищается в зависимости от режима установки (live или образ).

Это предотвращает дублирование идентификаторов при массовом развёртывании систем.

6. Предустановочные скрипты

Скрипт `preinstall.py` выполняет пользовательские скрипты, заданные в конфигурации, перед началом установки.

- `/tmp/pre-install/` — временное место хранения пользовательских скриптов.

Позволяет выполнить любые действия до начала установки пакетов и настройки системы.

7. Скрипты перед установкой пакетов

Скрипт `prepkgsinstall.py` запускает пользовательские скрипты до установки программных пакетов.

- `/tmp/prepkgs-install/` — временное хранилище скриптов;
- Устанавливается переменная окружения `POI_ROOT` для скриптов.

Это может быть полезно для настройки репозиториев, прокси и других параметров окружения.

8. Пост-установочные скрипты

Скрипт `postinstall.py` выполняет пользовательские скрипты после завершения установки.

- `/tmp/post-install/` — временное размещение скриптов, которые запускаются внутри установленной системы.

Обычно используется для финальной настройки, установки дополнительных пакетов, активации сервисов.