



МСВСфера

# Разработка решений на базе Соскріт

# Руслан Рахманбердиев

Fullstack разработчик в команде MCBSфера ОС. Более 8 лет в CloudLinux.

Разработка плагинов для хостинговых панелей управления

cPanel, Plesk, DirectAdmin

Плагины для биллинговых систем

WHMCS, Blesta

## Задача –

необходимость  
в графических утилитах



### Пример

- Интерфейс для ограничения монтирования USB-устройств
- Интерфейс настройки используемых версий ядра Linux

### Варианты

- Отдельное GUI приложение (GTK/Libadwaita)
- Доработка Центра управления Gnome
- Расширения для панели управления Cockpit

### Минусы отдельного GUI и центра управления Gnome

- Каким-то образом узнать о расширении
- В центре управления Gnome отсутствует механизм расширений
- Gnome сложности в разработке
- Можно пользоваться только локально (VNC/RDP)

# Почему Cockpit?



Готовая платформа



Понятное поведение  
и расположение



Единый вид интерфейса



Разнообразные  
инструменты для бекенда



COCKPIT



Локальный и удаленный  
доступ



Приличный функционал из  
«коробки»



Пакеты присутствуют  
в системе



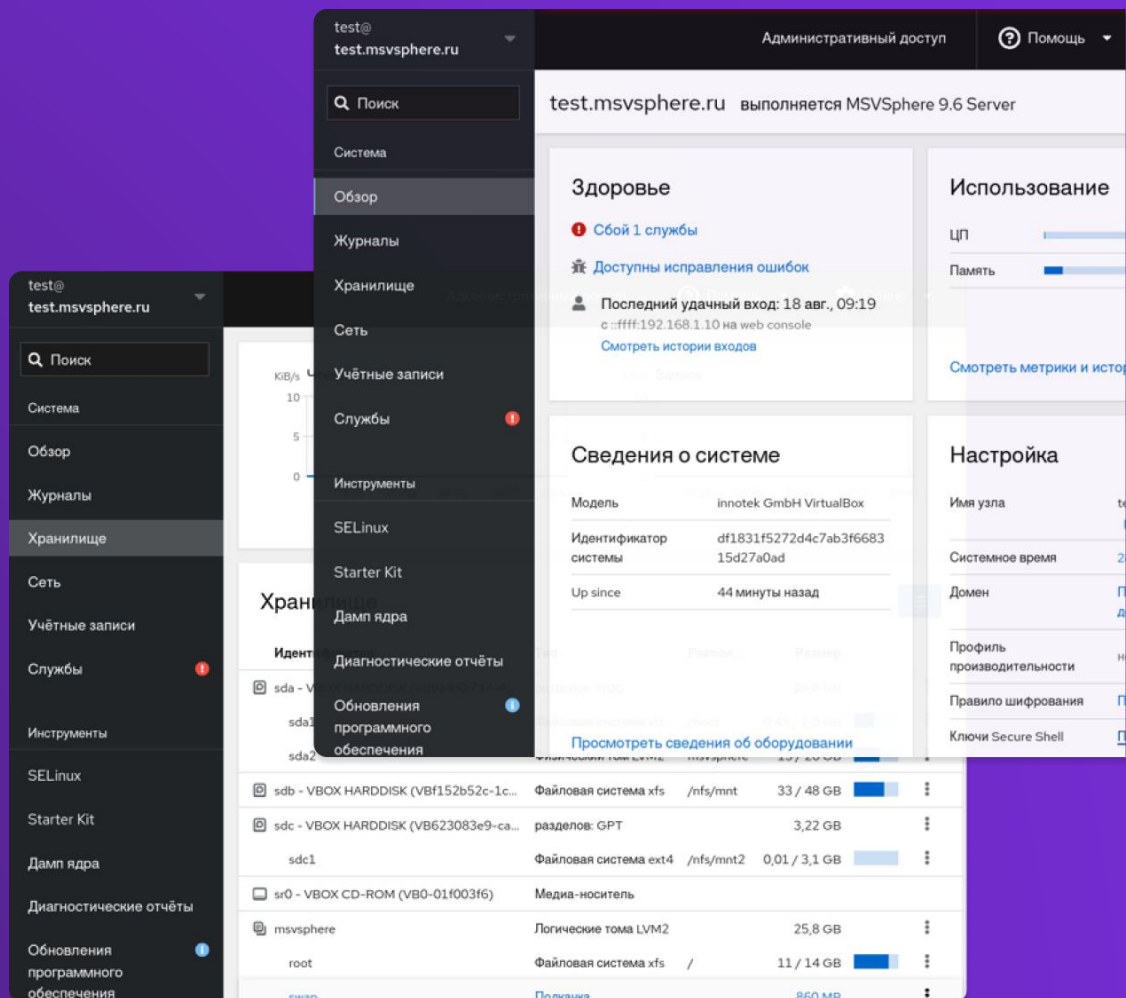
Процесс не занимает  
память постоянно

# Почему Cocksrit?



## Соскрит - открытый веб-интерфейс

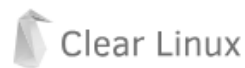
- ✓ Изменение сетевых настроек
- ✓ Настройка фаерволла
- ✓ Управление хранилищем
- ✓ Работа с контейнерами
- ✓ Просмотр журналов
- ✓ Обновление ПО
- ✓ Управление пользователями



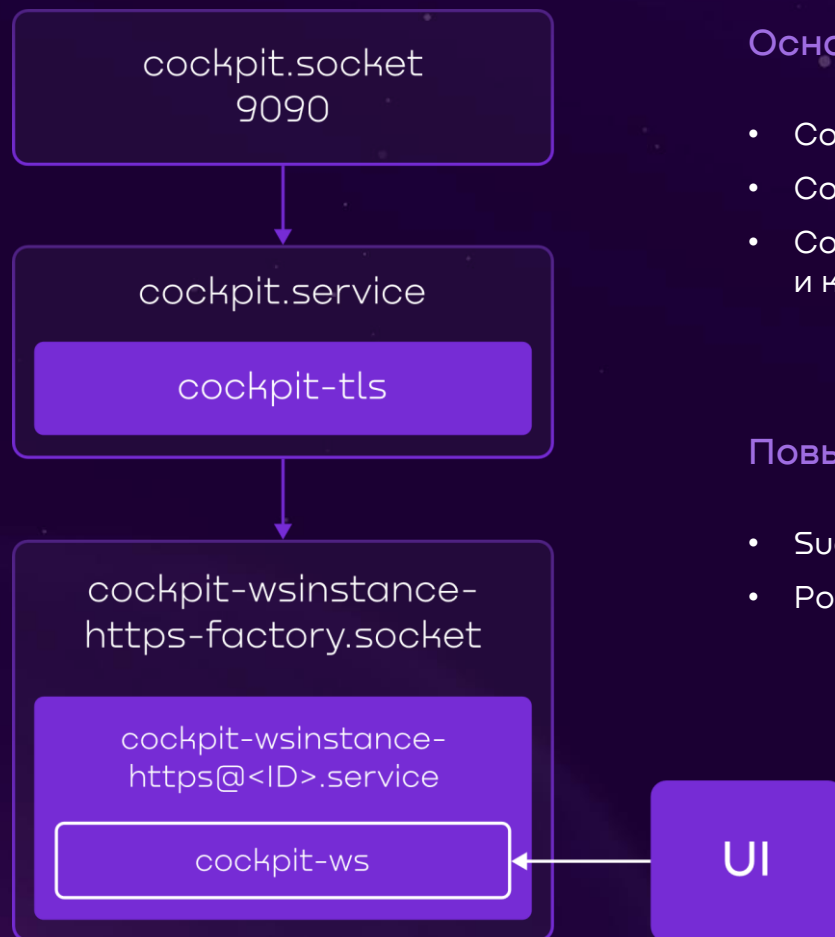
# Что такое Cockpit?

**Cockpit** поддерживается  
RedHat

**Cockpit** можно использовать во многих  
Linux-дистрибутивах



# Архитектура Cockpit



## Основные компоненты

- Cockpit-tls — HTTP TLS прокси
- Cockpit-ws — компонент веб-сервиса
- Cockpit-bridge — трансляция сообщений и команд

## Повышение привилегий

- Sudo
- Policy Kit (polkit)

## Cockpit — модульный



<https://cockpit-project.org/applications.html>

### Стандартные модули:

- `cockpit-system` - интерфейс администратора
- `cockpit-packagekit` - обновление приложений
- `cockpit-storaged` - управление локальными дисками
- `cockpit-networkmanager` - настройка сетевых интерфейсов

### Дополнительные:

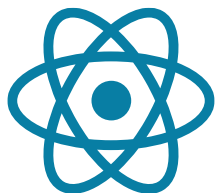
- `cockpit-machines` - управление виртуальными машинами
- `cockpit-selinux` - настройка политик SELinux



# Архитектура Cockpit

**React** (JS фреймворк) + **PatternFly** (CSS фреймворк, также поддерживаемый RedHat).

OpenSource лицензия ([LGPL v2.1+](https://www.gnu.org/licenses/lgpl-2.1.html))



<https://react.dev/>



<https://www.patternfly.org/>



Vue.js

.....

## Стартовая точка в виде репозитория

- Структура плагина
- Настройки линтера/сборки
- Базовый набор тестов
- Скрипты для помощи в сборке/деплое плагина/запуске тестов



<https://github.com/cockpit-project/starter-kit>

## Плагин в Cockpit —

каталог в определенном месте,  
с определённым названием  
и наполнением

## Наименование каталога

(`cockpit-starter-kit`)

## Метафайл

manifest.json

## Файлы локализации

(используется `gettext`)

```
[root@test /]# tree -C /usr/share/cockpit/cockpit-starter-kit
/usr/share/cockpit/cockpit-starter-kit
```

```
├── index.css
├── index.css.map
├── index.html
├── index.js
├── index.js.LEGAL.txt
├── index.js.map
├── manifest.json
├── po.de.js
└── po.manifest.de.js
```

```
[root@test /]# jq < /usr/share/cockpit/cockpit-starter-kit/manifest.json
```

```
{
  "requires": {
    "cockpit": "137"
  },
  "tools": {
    "index": {
      "label": "Starter Kit"
    }
  }
}
```

```
[root@test /]# pygmentize /usr/share/cockpit/cockpit-starter-kit/po.de.js
```

```
cockpit.locale({
  "": {
    "plural-forms": (n) => n != 1,
    "language": "de",
    "language-direction": "ltr"
  },
  "Cockpit Starter Kit": [
    null,
    "Cockpit Bausatz"
  ],
  "Running on $0": [
    null,
    "Läuft auf $0"
  ],
  "Starter Kit": [
    null,
    "Bausatz"
  ],
  "Unknown": [
    null,
    "Unbekannt"
  ]
});
```

# Разработка собственного плагина

## Интерфейс и код плагина `starter-kit`

```
import React, { useEffect, useState } from 'react';
import { Alert } from "@patternfly/react-core/dist/esm/components/Alert/index.js";
import { Card, CardBody, CardTitle } from "@patternfly/react-core/dist/esm/components/Card/index.js";

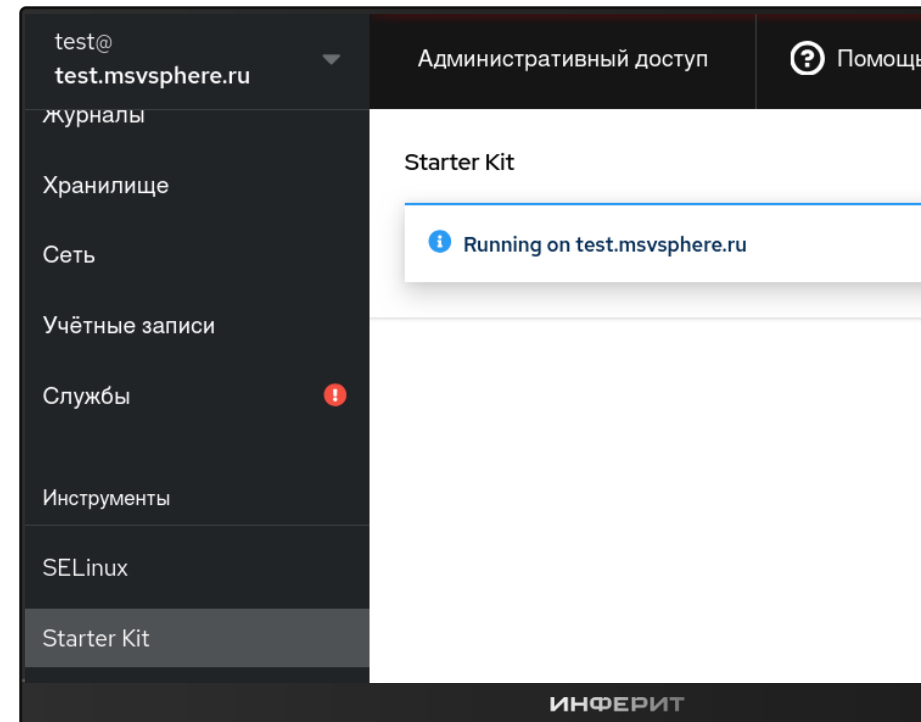
import cockpit from 'cockpit';

const _ = cockpit.getText();

export const Application = () => {
  const [hostname, setHostname] = useState(_("Unknown"));

  useEffect(() => {
    const hostname = cockpit.file('/etc/hostname');
    hostname.watch(content => setHostname(content?.trim() ?? ""));
    return hostname.close;
  }, []);

  return (
    <Card>
      <CardTitle>Starter Kit</CardTitle>
      <CardBody>
        <Alert
          variant="info"
          title={ cockpit.format(_("Running on $0"), hostname) }
        />
      </CardBody>
    </Card>
  );
};
```



# Разработка собственного плагина



МСВСфера

API предоставляет всё необходимое



Работа с файлами



Вызов исполняемых  
файлов



Взаимодействие с DBUS



HTTP клиент



Локализация



Чтение данных  
о пользователе и текущей  
сессии

## Дополнительные примеры

- Интеграция терминала
- Запуск длительного процесса
- Пингер (чтение потока данных)



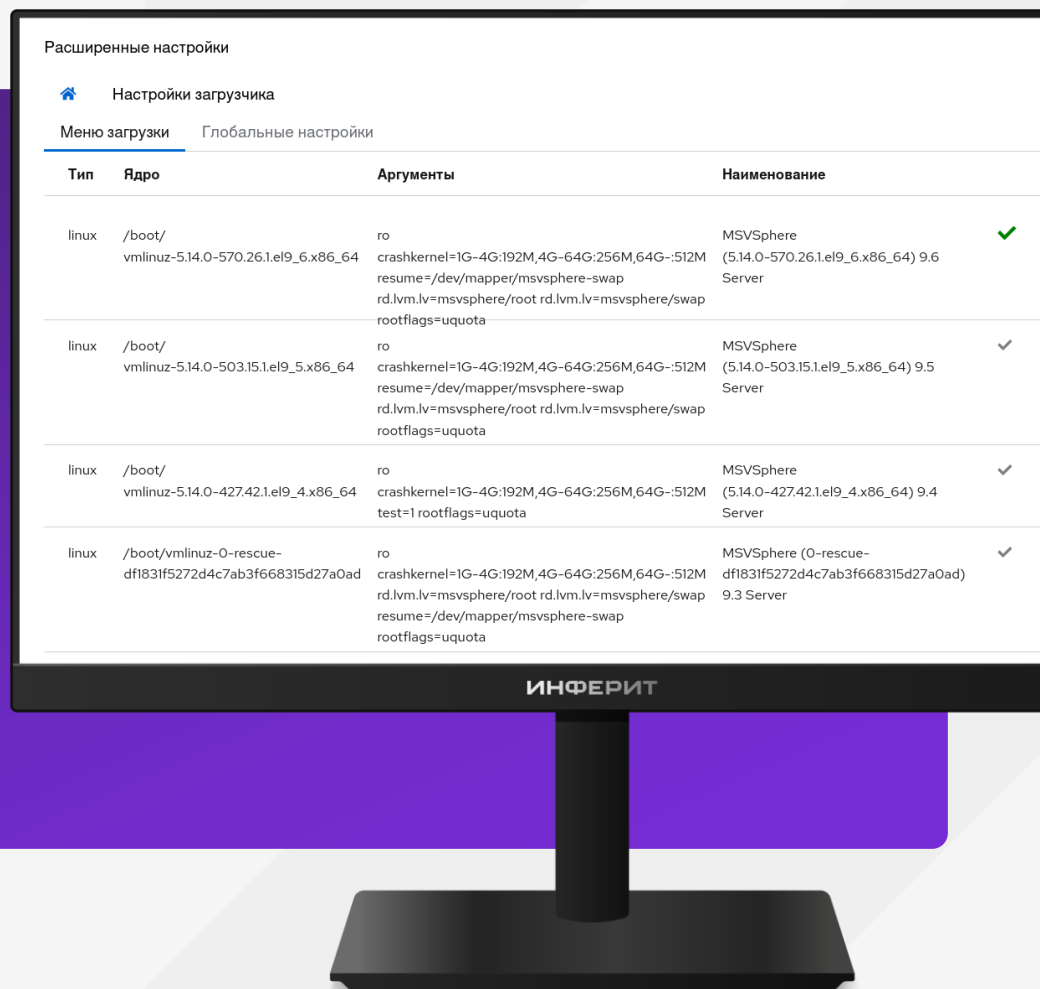
<https://github.com/cockpit-project/cockpit/tree/main/examples>

# Пример реализации

## Плагин для управления настройками загрузчика

### Что может:

- Управлять настройками по умолчанию
- Изменять аргументы для выбранного ядра
- Изменять глобальные настройки загрузчика



# Пример реализации


## Как работает:

- Для бекенда используется CLI скрипт
- UI во время действий выполняет скрипт
- Результат возвращается в JSON формате
- Ответ обрабатывается и отображается пользователю
- Поддерживает grub.cfg, entries



3-5 дней  
на разработку

Расширенные настройки

 Настройки загрузчика

Меню загрузки    Глобальные настройки

Таймаут ожидания действия пользователя	<input type="text" value="5"/> <small>Значение ожидания в секундах. -1: ожидание без таймаута, 0: не отображать меню выбора загрузчика</small>
Выбора источника ввода данных при загрузке	<input type="text"/>
Выбора источника вывода данных при загрузке	<input type="text" value="console"/>
Аргументы ядра	<input type="text" value='"crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/msvsphere-swap rd.lvm.lv=msvsphere/root rootflags=uquota"'/> <small>Аргументы будут применены для новых версий ядер.</small>
Ядро для загрузки по умолчанию	<input type="text" value="saved"/> <small>В случае значения "saved", для загрузки по умолчанию, будет использовано последнее выбранное в меню ядро. Выбрать ядро по умолчанию можно во вкладке "Меню загрузки".</small>

☒ Сохранить

ИНФЕРИТ



## Часть функционала вынесли в отдельный пакет («базовый»)

- Добавляет основное меню
- Читает конфиги установленных плагинов
- Добавляет на страницу секцию с названием, описанием, иконкой и ссылкой плагина



## Зависимый плагин

- JSON конфиг
- Собственный функционал

# Пример реализации



МСВСфера



- Aide
- Bootloader
- Multi-seat
- Quota
- USBGuard
- LibreOffice
- Виртуальные терминалы



<https://git.inferitos.ru/explore/repos?q=msvsphere%2Fcockpit-msvsphere>

The screenshot shows the MSVCSфера Cockpit interface. The top bar includes the user 'test@test.msvsphere.ru', 'Административный доступ', 'Помощь', and 'Сеанс'. The left sidebar lists system components: журналы, Хранилище, Сеть, Учётные записи, Службы, Инструменты, SELinux, Starter Kit, Дамп ядра, Диагностические отчёты, Обновления программного обеспечения, Приложения, and 'Расширенные настройки' (highlighted). The main content area, titled 'Расширенные настройки', displays settings for several components:

- Aide**: Средство настройки AIDE и проверки ФС.
- Bootloader**: Средство для управления настройками загрузчика.
- LibreOffice**: Средство для настройки безопасности LibreOffice.
- Multi-seat**: Средство настройки мультитерминального режима, позволяющего обеспечить одновременную работу нескольких пользователей на одном компьютере.
- Quota**: Средство для управления дисковыми квотами.
- USBGuard**: Средство для контроля подключения USB устройств.
- Виртуальные терминалы**: Средство для отключения виртуальных текстовых терминалов.

The bottom of the interface features the 'ИНФЕРИТ' logo.

# Спасибо за внимание!

Готов ответить на вопросы



МСВСфера