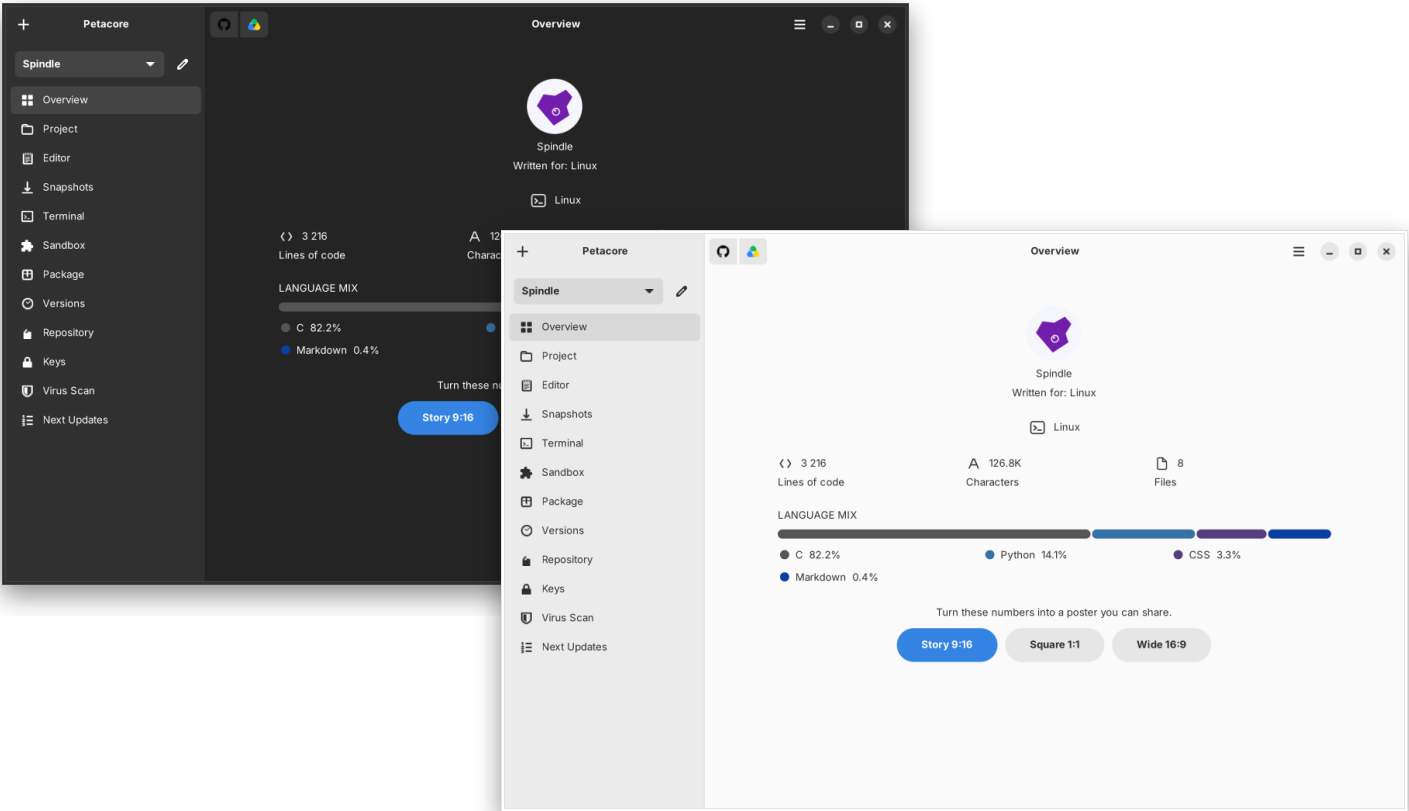


PETACORE 1.1 BETA 2 — GNOME
PETACORE 1.1 BETA 2 — KDE PLASMA

petacore

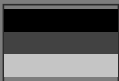
BY PETACOMM



ENG



TUR



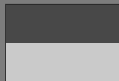
DEU



FRA



ZHO



UKR



ENG

Contents

1. Foreword.....	5
2. Technical description	5
2.1. General.....	5
2.2. System requirements.....	5
2.3. Technical specifications.....	5
2.4. Components used.....	6
3. Installation and first start	6
3.1. Installing Petacore.....	6
3.2. First start.....	6
3.3. The main window.....	7
3.4. Choosing the interface	8
4. Projects.....	8
4.1. Creating a project	8
4.2. Editing and removing a project.....	8
4.3. Overview.....	8
4.4. Project page.....	8
4.5. Editor	9
4.6. Snapshots and autosave.....	9
4.7. Terminal	9
4.8. Sandbox	9
4.9. Live Server	9
4.10. Next Updates.....	10
5. Packages and releases.....	10
5.1. Package.....	10
5.2. Signing keys.....	10
5.3. Versions.....	11
5.4. APT repository.....	11
5.5. Repository on a server	12
6. Online services.....	12
6.1. Connecting GitHub	12
6.2. Upload to GitHub	12
6.3. Cloning from GitHub	13
6.4. Google Drive	13
6.5. Backups.....	13
6.6. Changes from your other computers.....	13
6.7. Virus Scan	14
7. Settings.....	14
8. Security and privacy	14
9. Troubleshooting	15
10. Keyboard shortcuts.....	15
11. Files and uninstalling	16
11.1. Where Petacore keeps its data	16
11.2. Uninstalling	16
12. Licence and copyright	16



TUR

İçindekiler

1. Önsöz	17
2. Teknik açıklama	17
2.1. Genel.....	17
2.2. Sistem gereksinimleri.....	17
2.3. Teknik özellikler	17
2.4. Kullanılan bileşenler	18
3. Kurulum ve ilk çalıştırma	18
3.1. Petacore'u kurma.....	18
3.2. İlk çalıştırma	18
3.3. Ana pencere	19
3.4. Arayüzü seçme.....	20
4. Projeler	20
4.1. Proje oluşturma	20
4.2. Projeyi düzenleme ve kaldırma	20
4.3. Genel Bakış	20
4.4. Proje sayfası	20
4.5. Editör	21
4.6. Anlık görüntüler ve otomatik kayıt.....	21
4.7. Terminal	21
4.8. Geçici Kurulum.....	21
4.9. Canlı Sunucu.....	21
4.10. Sonraki Güncellemeler	22
5. Paketler ve yayınlar	22
5.1. Paket.....	22
5.2. İmza anahtarları.....	22
5.3. Versiyonlar	23
5.4. APT deposu	23
5.5. Sunucudaki depo.....	24
6. Çevrimiçi hizmetler	24
6.1. GitHub'ı bağlama.....	24
6.2. GitHub'a yükle.....	24
6.3. GitHub'dan proje çekme	25
6.4. Google Drive	25
6.5. Yedekler	25
6.6. Diğer bilgisayarlarınızdan gelen değişiklikler	25
6.7. Virüs Taraması	26
7. Ayarlar	26
8. Güvenlik ve gizlilik	26
9. Sorun giderme	27
10. Klavye kısayolları.....	27
11. Dosyalar ve kaldırma.....	28
11.1. Petacore verilerini nerede tutar	28
11.2. Kaldırma.....	28
12. Lisans ve telif hakkı	28



DEU

Inhaltsverzeichnis

1.	Vorwort	29
2.	Technische Beschreibung	29
2.1.	Allgemeines.....	29
2.2.	Systemvoraussetzungen	29
2.3.	Technische Daten.....	29
2.4.	Verwendete Komponenten.....	30
3.	Installation und erster Start	30
3.1.	Petacore installieren	30
3.2.	Erster Start	31
3.3.	Das Hauptfenster.....	31
3.4.	Die Oberfläche wählen	32
4.	Projekte	32
4.1.	Ein Projekt anlegen	32
4.2.	Projekt bearbeiten und entfernen	32
4.3.	Übersicht	32
4.4.	Die Seite Projekt	33
4.5.	Editor	33
4.6.	Schnappschüsse und automatisches Speichern.....	33
4.7.	Terminal	33
4.8.	Sandbox	33
4.9.	Live-Server	34
4.10.	Nächste Updates.....	34
5.	Pakete und Veröffentlichungen.....	34
5.1.	Paket.....	34
5.2.	Signatur Schlüssel	34
5.3.	Versionen.....	35
5.4.	APT-Paketquelle	35
5.5.	Paketquelle auf einem Server	36
6.	Online-Dienste	37
6.1.	GitHub verbinden.....	37
6.2.	Auf GitHub hochladen	37
6.3.	Von GitHub klonen.....	37
6.4.	Google Drive	37
6.5.	Sicherungen	38
6.6.	Änderungen von Ihren anderen Computern	38
6.7.	Virenprüfung	38
7.	Einstellungen.....	39
8.	Sicherheit und Datenschutz	39
9.	Fehlerbehebung	40
10.	Tastenkürzel	40
11.	Dateien und Deinstallation	41
11.1.	Wo Petacore seine Daten ablegt.....	41
11.2.	Deinstallation.....	41
12.	Lizenz und Urheberrecht	41



FRA

Table des matières

1.	Avant-propos.....	42
2.	Description technique	42
2.1.	Généralités	42
2.2.	Configuration requise.....	42
2.3.	Caractéristiques techniques.....	42
2.4.	Composants utilisés	43
3.	Installation et premier démarrage.....	43
3.1.	Installer Petacore.....	43
3.2.	Premier démarrage	44
3.3.	La fenêtre principale	44
3.4.	Choisir l'interface	45
4.	Projets.....	45
4.1.	Créer un projet.....	45
4.2.	Modifier et retirer un projet.....	45
4.3.	Vue d'ensemble.....	45
4.4.	La page Projet.....	46
4.5.	Éditeur.....	46
4.6.	Instantanés et sauvegarde automatique.....	46
4.7.	Terminal	46
4.8.	Bac à sable	46
4.9.	Serveur en direct.....	47
4.10.	Prochaines mises à jour	47
5.	Paquets et versions.....	47
5.1.	Paquet	47
5.2.	Clés de signature.....	47
5.3.	Versions.....	48
5.4.	Dépôt APT	48
5.5.	Dépôt sur un serveur	49
6.	Services en ligne	50
6.1.	Connecter GitHub	50
6.2.	Publier sur GitHub.....	50
6.3.	Cloner depuis GitHub.....	50
6.4.	Google Drive	50
6.5.	Sauvegardes.....	51
6.6.	Modifications de vos autres ordinateurs.....	51
6.7.	Analyse antivirus	51
7.	Paramètres	52
8.	Sécurité et confidentialité.....	52
9.	Dépannage	53
10.	Raccourcis clavier	53
11.	Fichiers et désinstallation	54
11.1.	Où Petacore conserve ses données.....	54
11.2.	Désinstallation.....	54
12.	Licence et droits d'auteur.....	54



ZHO

目录

1. 前言	55
2. 技术说明	55
2.1. 概述	55
2.2. 系统要求	55
2.3. 技术规格	55
2.4. 所用组件	56
3. 安装与首次启动	56
3.1. 安装 Petacore	56
3.2. 首次启动	57
3.3. 主窗口	57
3.4. 选择界面	58
4. 项目	58
4.1. 创建项目	58
4.2. 编辑与移除项目	58
4.3. 概览	58
4.4. “项目”页面	59
4.5. 编辑器	59
4.6. 快照与自动保存	59
4.7. 终端	59
4.8. 沙盒	59
4.9. 实时服务器	60
4.10. 后续更新	60
5. 软件包与发布	60
5.1. 软件包	60
5.2. 签名密钥	60
5.3. 版本	61
5.4. APT 软件源	61
5.5. 服务器上的软件源	62
6. 在线服务	62
6.1. 连接 GitHub	62
6.2. 上传到 GitHub	63
6.3. 从 GitHub 克隆	63
6.4. Google Drive	63
6.5. 备份	63
6.6. 来自其他电脑的改动	64
6.7. 病毒扫描	64
7. 设置	64
8. 安全与隐私	65
9. 故障排除	65
10. 键盘快捷键	66
11. 文件与卸载	66
11.1. Petacore 的数据存放位置	66
11.2. 卸载	67
12. 许可与版权	67



UKR

Зміст

1. Передмова	68
2. Технічний опис	68
2.1. Загальні відомості	68
2.2. Системні вимоги	68
2.3. Технічні характеристики	68
2.4. Використані компоненти	69
3. Встановлення та перший запуск	69
3.1. Встановлення Petacore	69
3.2. Перший запуск	70
3.3. Головне вікно	70
3.4. Вибір інтерфейсу	71
4. Проєкти	71
4.1. Створення проєкту	71
4.2. Зміна та вилучення проєкту	71
4.3. Огляд	71
4.4. Сторінка «Проєкт»	72
4.5. Редактор	72
4.6. Знімки та автозбереження	72
4.7. Термінал	72
4.8. Пісочниця	72
4.9. Живий перегляд	73
4.10. Наступні оновлення	73
5. Пакунки та випуски	73
5.1. Пакунок	73
5.2. Ключі підпису	73
5.3. Версії	74
5.4. APT-репозиторій	74
5.5. Репозиторій на сервері	75
6. Онлайн-сервіси	76
6.1. Під'єднання GitHub	76
6.2. Завантажити на GitHub	76
6.3. Клонування з GitHub	76
6.4. Google Диск	76
6.5. Резервні копії	77
6.6. Зміни з інших ваших комп'ютерів	77
6.7. Перевірка на віруси	77
7. Налаштування	78
8. Безпека та конфіденційність	78
9. Усунення несправностей	79
10. Комбінації клавіш	79
11. Файли та вилучення програми	80
11.1. Де Petacore зберігає дані	80
11.2. Вилучення програми	80
12. Ліцензія та авторські права	80

1. Foreword

Thank you for choosing Petacore, the desktop workbench for building and shipping Linux software. Petacore keeps everything a project needs in one window: its files and code, its history, its packages and signing keys, its software repository, its copy on GitHub and on Google Drive, and a virus check before anything reaches your users.

Before you start, please read this manual carefully. It describes every part of Petacore, the settings it offers, and what it does — and never does — with your files and credentials. Knowing these will let you work faster and with complete peace of mind.

Developer: Petacomm I.S.
Product: Petacomm Petacore™
Version described: 1.1 Beta 2
Project page: github.com/petacomm/petacore

2. Technical description

2.1 General

Petacore is a native desktop application for Linux. It runs entirely on your own computer and works on ordinary project folders: nothing has to be converted, and every file stays readable by any other tool. Online services (GitHub, Google Drive, VirusTotal) are used only when you connect them, with your own accounts.

Petacore comes in two interfaces that offer the same features: one for the GNOME desktop and one for KDE Plasma. The installer picks the right one automatically, and you can switch at any time.

2.2 System requirements

Item	Requirement
Operating system	Ubuntu 26.04 LTS, or another Debian- or Ubuntu-based distribution
Desktop	GNOME (GTK 4, libadwaita) or KDE Plasma (Qt 6, PySide6)
Python	3.12 or newer
Disk space	About 50 MB for Petacore, plus room for snapshots and backups
Internet	Only for GitHub, Google Drive, VirusTotal and server connections

2.3 Technical specifications

Feature	Specification
Interface languages	English, Türkçe, Deutsch, Français, 中文, Українська
Package formats	.deb (Debian, Ubuntu, Mint) and .rpm (Fedora, openSUSE, RHEL)
Package signing	GnuPG — Ed25519 (recommended) or RSA 2048 / 3072 / 4096
Version control	git; publishing to GitHub with versioned releases
Cloud copy	Google Drive through rclone, optionally end-to-end encrypted
Security scan	VirusTotal API v3, with your own key

Feature	Specification
Software repository	Signed APT archive, local or on your own server over SSH
Server connection	OpenSSH, key login only, two-factor login supported
Isolation	bubblewrap sandbox, network off by default

2.4 Components used

Petacore relies on proven, standard tools. The installer sets them up; a missing one is reported in plain words together with the command that installs it.

Component	Used for
git, gh	Version history; GitHub sign-in and publishing
clone	Google Drive connection, synchronisation and backups
gnupg	Creating signing keys and signing packages and repositories
dpkg-dev, rpm, apt-utils	Building packages and repository indexes
bubblewrap	The isolated Sandbox
openssh-client, rsync	Server connection and publishing
libsecret-tools	Keeping sign-ins and API keys in the desktop keyring
libsvg2-bin	Saving share cards as PNG images
WebKitGTK (optional)	Showing Live Server previews inside the window

3. Installation and first start

3.1 Installing Petacore

1. Unpack the Petacore archive into any folder.
2. Open a terminal in that folder and run `./install.sh` — as yourself, never with `sudo`. Petacore installs into your own home folder.
3. The installer recognises your desktop (GNOME or KDE Plasma). If it cannot tell, it asks which interface you want.
4. When system packages are needed, it asks for your password once and installs them with `apt`.
5. Petacore then appears in your applications menu, and the command `petacore` starts it from a terminal.

Petacore can also be installed from a `.deb` package. The result is the same; the installer script only adds the automatic choice of interface.

3.2 First start

On first start a short wizard asks for the language (your desktop's language is suggested), then offers to connect GitHub and Google Drive, and finally to create your first project. Every step can be skipped and done later from Settings.

3.3 The main window

On the left, the sidebar lists the pages of the active project; the project selector at its top switches between projects, and the pencil button edits the active one. At the top of the window are the GitHub button (upload to GitHub), the Google Drive button (send to Drive) and the main menu (New Project, Settings, About, Quit).

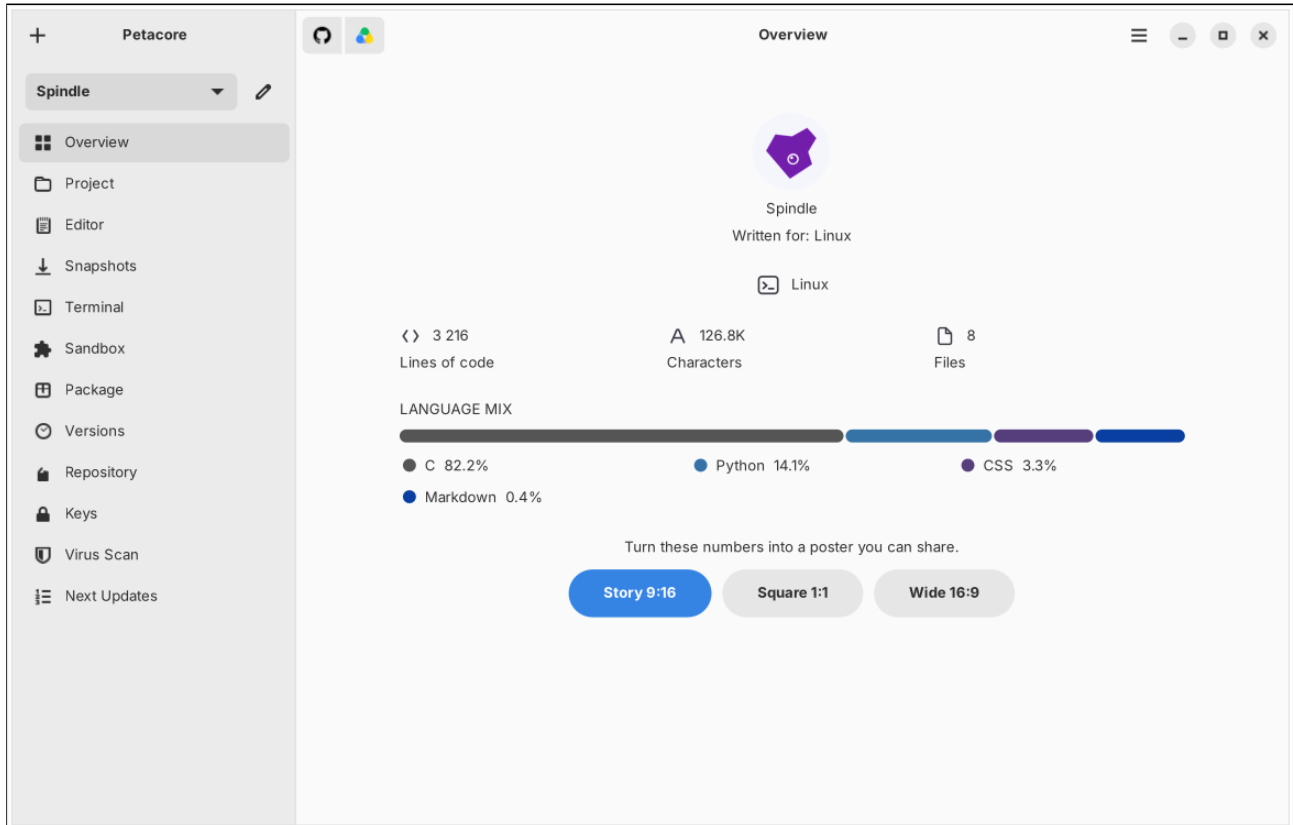


Figure 1 — The main window with the Overview page.

The pages depend on the kind of project. A project that targets an operating system gets Package, Keys and Sandbox; a web project gets Live Server instead, because it is neither packaged nor signed. Petacore decides this only on a clear match, so an unfamiliar project always keeps the full set.

Page	Purpose
Overview	Figures about the project and a shareable poster
Project	File manager for the project folder
Editor	Code editor with tabs
Snapshots	Restore points of the whole project
Terminal	A shell opened in the project folder
Sandbox / Live Server	Trying the program safely / previewing a website
Package	Building signed .deb and .rpm packages
Versions	Release history and the code of every release
Repository	Signed APT repository, locally or on a server
Keys	GPG signing keys
Virus Scan	Checking the project with VirusTotal

Page	Purpose
Next Updates	Plans for the project

3.4 Choosing the interface

Settings → Interface shows which design is in use and switches to the other one. Petacore closes and reopens in the new design; projects and settings stay as they are. If the components of the other design are missing, Petacore offers to install them.

4. Projects

4.1 Creating a project

Choose New Project from the main menu or the + button. Enter a name and the folder where the project should live.

- With a GitHub address, Petacore downloads (clones) that repository. Because that code comes from the internet, Petacore then asks whether VirusTotal should check it before you run anything in it.
- Without an address, a new local git repository is started.
- Under Encryption you may choose to keep the Google Drive copy encrypted with a 4-digit PIN, an 8-digit PIN or a passphrase of up to 4096 characters. The project on your computer stays ordinary, readable code; only the copy on Drive is encrypted — file and folder names included.

The encryption password is never stored. If it is lost, the encrypted copy on Drive cannot be opened by anyone, including Petacomm.

4.2 Editing and removing a project

The pencil button next to the project selector opens Edit Project: name, project folder, GitHub address and the installed-app folder. When an installed-app folder is set, every file you save in the Editor can be copied into the installed program at the same relative path — no terminal needed.

Remove from Petacore only takes the project off the list. No file on disk is deleted.

4.3 Overview

Overview shows the project at a glance: lines of code, characters, files, languages and size, the system it is written for (Linux, Windows, macOS, Web or cross-platform) and the language mix as a coloured bar.

Share Card turns these figures into a poster in three shapes — Story 9:16, Square 1:1 and Wide 16:9. The card shows only the name, the figures and the language mix, never file names or code; the name on the card can be changed if the project's own name is not for sharing.

4.4 Project page

The Project page is a complete file manager for the project folder. The line under the toolbar shows the git branch, how the project stands against GitHub and how many programs are running from this folder.

- Double-click opens a file in the Editor; images, audio and video open in their own viewer.
- Right-click offers Open, Rename, New Folder, Copy Path and Delete.
- Deleting is reversible: Ctrl+Z, or the undo button, brings the files back.
- Drag files in from your file manager. When names collide, you choose Replace / Merge, Keep Both or Skip.

- The search field searches the open folder; the button beside it searches the whole project. The filter limits the list to code, images, audio and video, documents or folders.
- Hold Ctrl over a file for a quick preview; Ctrl+scroll moves through long previews.
- Media Files collects every image, sound and video of the project in one place.
- The clock button opens Backups (see 6.5).

4.5 Editor

The Editor opens files in tabs, with syntax highlighting and the familiar keys of the micro editor (see chapter 10).

- Sections: select lines and add a section to give a part of a long file a name; the section list jumps straight to it.
- Focus mode dims every line outside the chosen section, so the part you are working on stands out. Esc clears the focus.
- Esc leaves a file. If it has unsaved changes, Petacore asks first: Esc again leaves without saving, Ctrl+S saves and leaves. The question can be turned off in Settings.
- When the project has an installed-app folder, saving offers to update the installed program as well.

4.6 Snapshots and autosave

A snapshot is a complete restore point of the project. Create Snapshot makes one at any time, with an optional label. Autosave makes one automatically at a set interval (10 minutes by default); the 40 most recent automatic snapshots are kept.

Restoring a snapshot first takes a safety snapshot of the current state, so a restore can itself be undone. Snapshots are stored outside the project, in your own data folder; .git, the versions folder and build leftovers are not included.

4.7 Terminal

The Terminal page opens your usual shell directly in the project folder.

4.8 Sandbox

The Sandbox runs commands in a sealed copy of your system. The real machine is read-only inside it: installing, deleting and editing only affect this throwaway session, and End Simulation removes everything without a trace.

- Try Current Project starts the Sandbox with a copy of the project.
- Install a .deb Temporarily installs a package only inside the Sandbox.
- Run App starts graphical or terminal programs from inside.
- The Sandbox has no internet by default. If something inside tries to go online, Petacore tells you and lets you allow it, keep it off, or stop asking.

Full isolation requires bubblewrap. Without it the page warns “NOT isolated” and shows the command that installs it.

4.9 Live Server

For web projects, Live Server shows the site inside Petacore without any build step and reloads it by itself whenever a file changes. F11 fills the window with the site; Esc returns.

By default the server answers only this computer. Share on the local network makes it reachable from a phone or another device on the same network and shows the address to open there. Without WebKitGTK the preview opens in your usual browser instead.

4.10 Next Updates

Next Updates keeps the plans for the project: a title, details, a priority (High, Normal, Someday) and attached documents. Plans can be edited, marked as done and copied. They are saved inside the project, so they travel with it, and can be synchronised to Google Drive.

5. Packages and releases

5.1 Package

The Package page builds an installer from the project: .deb for Debian, Ubuntu and Mint, .rpm for Fedora, openSUSE and RHEL. The package installs the program into /opt/<name> on the target system and adds it to the applications menu.

Field	Meaning
Name	Package name, in lower case
Version	For example 1.6 or 2.0.1
Maintainer	Name and e-mail address of the person responsible
Description	Shown by software centres and package managers
Licence	Chosen from the list or typed in; Compare the licences explains them
Signing	Signs the package with the active key (on by default)

Files that carry credentials — .env, private keys, .netrc, .npmrc and similar — are always left out of the package, and Petacore tells you how many it kept back.

5.2 Signing keys

A GPG key is your digital signature: it proves that a package really comes from you and has not been changed by anyone else. The Keys page lists your keys and shows which one signs.

Setting	Choices
Name, e-mail, comment	Who the key belongs to
Key type	Ed25519 (modern, fast, recommended) or RSA 4096 / 3072 / 2048 (for older systems)
Expiration	Never, 1 year, 2 years or 5 years
Passphrase	Optional. With one, GPG asks for it at every signature — more secure

- Use for signing makes a key the active signing key.
- Export public key saves the file your users verify packages with.
- The public halves of your keys can be backed up to Google Drive. Private keys never leave this computer.
- Delete cannot be undone: packages can no longer be signed with that key.

5.3 Versions

Versions keeps every release the project has shipped, newest first, together with its packages. It has two tabs: New Version and Version Library.

- New Version builds the release from the project files as they are now: version number, release type (Stable, Release Candidate, Beta, Alpha), pre-release number, formats (.deb, .rpm), signing and release notes. The preview shows both the name people read ("1.6 Beta 2") and the version package managers compare ("1.6~beta2").
- The Version Library groups releases by series (1.6, 2.0 ...) and marks the latest one. Each release shows its packages, its signature and its notes.
- View Code opens the code of any release — read straight from that release's own package, exactly as it was shipped, read-only. Files can be filtered and their contents copied.
- Older .deb or .rpm files can be dropped onto the page to bring a release into the history; the originals stay where they are.
- Edit Notes, Add .deb to Repository and Remove Version are on each release. A removed version goes to the version trash and can be brought back.

Releases live in the project's versions folder. That folder is kept out of every package, snapshot, statistic and git history, so no package ever contains the ones before it.

5.4 APT repository

A repository turns single packages into a software source that users add once and then receive updates from, like any other. Petacore builds the tree, signs the index and keeps it in step with your packages.

Setting	Meaning
Name, Repository Folder	What it is called and where the tree is built
Public Address	The URL users install from, for example https://repo.example.com
Suite, Component, Architectures	Usual values: stable, main, amd64, arm64
Origin, Label, Description	How the repository introduces itself to apt
Signing Key	apt refuses an archive whose index is not signed
Publish By	Not from here, copy to a folder, or send over SSH (rsync)

- Add Packages, or drop .deb files onto the page. With "Add built packages to the repository" on, every new .deb goes in automatically.
- Rebuild & Sign renews the index; a warning appears whenever it no longer matches the packages.
- Publish makes the server copy match the folder; Check Server compares the two.
- Badges and filters show packages that are not indexed, not signed, not published, or superseded.
- Each package offers Verify Signature, Copy Install Command, Rename, Open Containing Folder and Remove.
- Removed packages go to the repository trash and can be put back; Remove Old Versions keeps only the newest of each.
- Copy Instructions For Users and Export Archive Key give your users everything they need to add the repository.

5.5 Repository on a server

On the Server tab Petacore manages the repository folder directly on your server, over SSH. Files are uploaded with scp; the index is built on the server and signed on your computer, so the signing key never leaves it.

1. Enter the server name or IP, port, user and the repository folder on the server, for example `/var/www/repo`.
2. Choose an SSH key, or create one for Petacore: Ed25519, protected by a passphrase of at least 8 characters that is never stored.
3. Add the line Petacore shows to `~/.ssh/authorized_keys` of that user on the server.
4. Connect. The first time, compare the server's fingerprint with the one the server reports, and confirm only if they match.

Once connected you can upload (.deb files go to the pool automatically), create folders, rename, download, move items to the server trash and restore them, and Rebuild & Sign.

Protection	What it means
Host key check	The server's identity is checked on every connection; if it changes, Petacore refuses to connect
Key login only	Passwords are never sent
Modern cryptography	Post-quantum key exchange where available, authenticated encryption
Two-factor login	Authenticator codes and FIDO2 security keys
Nothing stored	Passphrases and two-factor codes are never saved

6. Online services

6.1 Connecting GitHub

1. Press Connect with GitHub — in Settings, in the first-start wizard, or whenever GitHub is needed.
2. Your browser opens on GitHub and the one-time code is already on the clipboard.
3. Paste the code, confirm, and press Authorize. Petacore connects by itself.

If the GitHub CLI (gh) is already signed in on the computer, one click is enough. If gh is missing, Petacore offers to install it. Under Other sign-in options a personal access token can also be pasted by hand. The sign-in is kept in the desktop keyring, never in a file or a URL.

6.2 Upload to GitHub

The GitHub button at the top of the window opens Upload to GitHub. In one step the code goes to the main branch and becomes a new version on the project's GitHub page.

- Pick a version from the suggestions — for example 1.0 Beta 3, 1.0 Beta 2.1, 1.0 RC 1 or 1.0 — or type your own. A version already in the library but not yet on GitHub is suggested first.
- The tag is made from the version (v1.0-beta.3); betas, alphas and release candidates are marked as pre-releases.
- If the project has no GitHub repository yet, one is created — private by default.
- The branch is main unless you change it. Release notes are optional; left empty, GitHub writes them from the commits.

- The .deb and .rpm of that version from the Version Library are attached to the release.
- Turn off “Publish as a new version” to upload only the code.

Credential files are never uploaded. Work already on GitHub is kept: if it collides with yours, the upload stops and nothing is overwritten. A version that is already published is never replaced.

6.3 Cloning from GitHub

When a project is created from a GitHub address, Petacore asks once whether VirusTotal should check the downloaded code. Scan opens the Virus Scan page and starts the check right away.

6.4 Google Drive

Connect your own Google Drive in Settings: the browser opens, you sign in to Google and press Allow. Petacore creates a folder named “Petacomm Petacore™” and keeps each project in its own subfolder, media files included. Petacore's own records live in a hidden .petacore folder beside them.

- The Drive button sends the active project. Drive is made to match this computer — and nothing is lost (see 6.5).
- If another computer has sent a newer copy since this one last synchronised, Petacore offers to bring it in first.
- If an upload would be smaller than what Drive holds, Petacore warns first (can be turned off in Settings).
- Fetch Projects from Drive downloads projects that exist only on Drive into ~/Projects.
- Encrypted projects are sent through an encrypted layer; Drive only ever sees unreadable data.

6.5 Backups

Petacore never deletes a file when synchronising with Drive. Every file a sync replaces or removes is first kept as a backup — on Drive when you send, on this computer when a copy comes in.

- Backups are listed by day: when, from which computer, how many files and how large.
- Open a backup to see its files. Put back restores the whole backup or a single file.
- Before anything is put back, the current files are kept as a backup themselves, so this can be undone too.
- Backups are kept for 30 days by default (7, 30, 90, 365 days or forever in Settings) and then removed automatically.

Backups open from the clock button on the Project page, from the Backups button in sync notifications, and from Settings → Google Drive → Show backups.

6.6 Changes from your other computers

Computers signed in to the same Google Drive bring in each other's newest copy on their own: a few seconds after start-up and then every 10 minutes (5, 10, 30 or 60 in Settings). Projects that exist only on Drive are downloaded too, into a new folder if the name is already taken.

Situation	What Petacore does
Newer copy on Drive, no changes here	Brings it in quietly and tells you which computer sent it
Newer copy on Drive and changes here	Asks: Take Drive's copy, Send mine, or Not now
Encrypted project	Not brought in automatically — it needs the password

Whichever side you choose, the other side's files are kept in Backups. The feature is on by default and can be turned off in Settings → Google Drive.

6.7 Virus Scan

Virus Scan checks the project with the engines of VirusTotal before your users install it. The API key is yours: create a free account at virustotal.com, copy the key from your profile and enter it once. It is kept in the desktop keyring.

- Scan Project packs the project into one archive; an unchanged project is recognised by its fingerprint and never sent twice.
- Nothing is sent until you agree, once per project. Credential files, `.git`, `node_modules` and the `versions` folder are never sent.
- Results: Nothing flagged, “n of t flagged it” with the engine names, or “never seen” — the normal answer for something just built.
- One or two engines flagging a freshly built, unsigned program is usually a false positive.
- A free key allows 4 checks a minute and 500 a day.

7. Settings

Setting	Default	Description
Language	Desktop language	Six languages; Petacore restarts to apply
Appearance	Follow System	Light, dark or as the desktop
Focus mode	On	Dims lines outside the chosen section
Smaller-upload warning	On	Warns when an upload is smaller than Drive's copy
Sandbox internet	Off	Lets code in the Sandbox reach the network
Ask before leaving with Esc	On	Confirmation for unsaved files
Autosave	On, 10 minutes	Automatic snapshots
GitHub	Not connected	Connect, disconnect, optional OAuth Client ID
Google Drive	Not connected	Connect, disconnect, fetch projects
Bring in changes	On, 10 minutes	Changes from your other computers
Keep backups for	30 days	7, 30, 90, 365 days or forever
Interface	Detected	GNOME or KDE Plasma

8. Security and privacy

- Sign-ins and API keys are kept in the desktop keyring. If no keyring exists, they go into a file only you can read.
- Credential files (`.env`, private keys, `.netrc` ...) never go into a package, a GitHub upload or a VirusTotal scan.
- The GitHub sign-in is given only to `github.com`; a look-alike address gets nothing.
- Private signing keys never leave your computer; only public keys are backed up.
- Server logins use keys only; passphrases and two-factor codes are never stored.
- The Sandbox has no network unless you open it; Live Server answers only this computer unless you share it.
- Nothing is uploaded to VirusTotal without your consent.

- Synchronisation never destroys a file: what it replaces is kept in Backups.

9. Troubleshooting

Problem	Solution
Google Drive support needs rclone	<code>sudo apt install rclone</code>
The keyring can't be reached	<code>sudo apt install libsecret-tools</code> , then try again
GitHub CLI is not ready	Press Install gh, or <code>sudo apt install gh</code>
Sandbox: NOT isolated	<code>sudo apt install bubblewrap</code>
Share card saved as SVG only	<code>sudo apt install librsvg2-bin</code>
Live Server: no built-in browser	<code>sudo apt install gir1.2-webkit-6.0</code> , or use Open in Browser
Repository index cannot be built	<code>sudo apt install apt-utils dpkg-dev</code>
Server connection: OpenSSH missing	<code>sudo apt install openssh-client</code>
VirusTotal refused the key	Check the key in your VirusTotal profile; a network block can also be temporary — try again later
The signing key expired or is missing	Create a new key on the Keys page and choose Use for signing
The server's identity has changed	Do not connect. Ask the server's administrator; forget the old key only once the change is confirmed
GitHub: branch has changes that don't fit	Bring those changes in with git in the Terminal, resolve them, then upload again
Version already published on GitHub	Choose another version
Project changed in two places	Choose which copy to keep; the other side stays in Backups
Wrong password for an encrypted project	The password cannot be recovered; use the correct one
Components of the other interface missing	Press Install Now, or run the command Petacore shows

10. Keyboard shortcuts

Keys	Action
Ctrl+S	Save the file (Editor)
Ctrl+Q	Close the file (Editor)
Ctrl+F	Find; Enter goes to the next match, Esc closes
Ctrl+K / Ctrl+D	Cut line / duplicate line
Ctrl+Z / Ctrl+Y	Undo / redo
Esc	Leave the file; Esc again leaves without saving



Keys	Action
Ctrl+A / Delete	Select all / delete (Project page, undoable)
Ctrl+Z	Bring back deleted files or removed packages
Ctrl + pointer	Quick preview; Ctrl+scroll moves through it
F11 / Esc	Live Server full screen / leave full screen

11. Files and uninstalling

11.1 Where Petacore keeps its data

Location	Contents
~/.local/share/petacore-app	The program itself
~/.config/petacore	Settings (readable only by you)
~/.local/share/petacore/snapshots	Snapshots
~/.local/share/petacore/drive-backups	Backups of files replaced on this computer
<project>/versions	Release history (hidden from packages and git)
<project>/petacore	Next Updates plans
Desktop keyring	GitHub sign-in, VirusTotal key

11.2 Uninstalling

Run `./uninstall.sh` from the Petacore folder, as yourself. Removing the program and deleting your snapshots and backups are asked as two separate questions, and nothing is deleted until you confirm. Your project folders are never touched.

12. Licence and copyright

© Petacomm I.S. 2026. All rights reserved. Petacomm Petacore™ is a trademark of Petacomm I.S.

Petacore uses free and open-source components, which remain the property of their authors and are used under their own licences. GitHub, Google Drive, VirusTotal, GNOME, KDE Plasma, Ubuntu and Debian are trademarks of their respective owners.

This manual describes Petacore 1.1 Beta 2. Features may change in later versions.

1. Önsöz

Linux yazılımı geliştirmek ve yayımlamak için hazırlanmış masaüstü çalışma tezgâhı Petacore'u tercih ettiğiniz için teşekkür ederiz. Petacore bir projenin ihtiyaç duyduğu her şeyi tek pencerede toplar: dosyaları ve kodu, geçmiş, paketleri ve imza anahtarları, yazılım deposu, GitHub ve Google Drive'daki kopyası ve kullanıcılarınıza ulaşmadan önce virüs denetimi.

Kullanmaya başlamadan önce lütfen bu kılavuzu dikkatle okuyun. Kılavuz Petacore'un tüm bölümlerini, sunduğu ayarları ve dosyalarınızla, giriş bilgilerinizle neler yaptığını — ve asla yapmadığını — anlatır. Bunları bilmek daha hızlı ve tam bir gönül rahatlığıyla çalışmanızı sağlar.

Geliştirici: Petacomm I.S.

Ürün: Petacomm Petacore™

Anlatılan sürüm: 1.1 Beta 2

Proje sayfası: github.com/petacomm/petacore

2. Teknik açıklama

2.1 Genel

Petacore, Linux için yerel bir masaüstü uygulamasıdır. Tamamen kendi bilgisayarınızda çalışır ve sıradan proje klasörleriyle çalışır: hiçbir şeyin dönüştürülmesi gerekmez, her dosya başka araçlarla da okunabilir kalır. Çevrimiçi hizmetler (GitHub, Google Drive, VirusTotal) yalnızca siz bağladığınızda ve kendi hesaplarınızla kullanılır.

Petacore aynı özellikleri sunan iki arayüzle gelir: biri GNOME masaüstü, diğeri KDE Plasma için. Kurulum programı doğru olanı kendisi seçer; dilediğiniz zaman diğerine geçebilirsiniz.

2.2 Sistem gereksinimleri

Öge	Gereksinim
İşletim sistemi	Ubuntu 26.04 LTS veya Debian ya da Ubuntu tabanlı başka bir dağıtım
Masaüstü	GNOME (GTK 4, libadwaita) veya KDE Plasma (Qt 6, PySide6)
Python	3.12 veya daha yeni
Disk alanı	Petacore için yaklaşık 50 MB, ayrıca anlık görüntüler ve yedekler için yer
İnternet	Yalnızca GitHub, Google Drive, VirusTotal ve sunucu bağlantıları için

2.3 Teknik özellikler

Özellik	Ayrıntı
Arayüz dilleri	English, Türkçe, Deutsch, Français, 中文, Українська
Paket biçimleri	.deb (Debian, Ubuntu, Mint) ve .rpm (Fedora, openSUSE, RHEL)
Paket imzalama	GnuPG — Ed25519 (önerilen) veya RSA 2048 / 3072 / 4096
Sürüm denetimi	git; GitHub'a sürümlü yayın (release) gönderimi
Bulut kopyası	rclone üzerinden Google Drive, isteğe bağlı uçtan uca şifreli

Özellik	Ayrıntı
Güvenlik taraması	VirusTotal API v3, kendi anahtarınızla
Yazılım deposu	İmzalı APT arşivi; yerelde ya da SSH ile kendi sunucunuzda
Sunucu bağlantısı	OpenSSH, yalnızca anahtarla giriş, iki adımlı doğrulama desteği
Yalıtım	bubblewrap ile Geçici Kurulum; ağ varsayılan olarak kapalı

2.4 Kullanılan bileşenler

Petacore kendini kanıtlamış standart araçlara dayanır. Kurulum programı bunları hazırlar; eksik olan bir bileşen, onu kuracak komutla birlikte açık bir dille bildirilir.

Bileşen	Kullanım amacı
git, gh	Sürüm geçmişi; GitHub girişi ve yayınlama
rclone	Google Drive bağlantısı, eşitleme ve yedekler
gnupg	İmza anahtarı oluşturma; paketleri ve depoları imzalama
dpkg-dev, rpm, apt-utils	Paket ve depo dizini oluşturma
bubblewrap	Yalıtılmış Geçici Kurulum
openssh-client, rsync	Sunucu bağlantısı ve yayınlama
libsecret-tools	Giriş bilgilerini ve API anahtarlarını masaüstü anahtarlığında saklama
libsvg2-bin	Paylaşım kartlarını PNG olarak kaydetme
WebKitGTK (isteğe bağlı)	Canlı Sunucu önizlemesini pencere içinde gösterme

3. Kurulum ve ilk çalıştırma

3.1 Petacore'u kurma

1. Petacore arşivini herhangi bir klasöre açın.
2. O klasörde bir terminal açın ve `./install.sh` komutunu çalıştırın — kendi kullanıcınızla, asla `sudo` ile değil. Petacore kendi ev klasörünüze kurulur.
3. Kurulum programı masaüstünüzü (GNOME veya KDE Plasma) tanır. Anlayamazsa hangi arayüzü istediğinizi sorar.
4. Sistem paketleri gerektiğinde parolanızı bir kez sorar ve bunları `apt` ile kurar.
5. Ardından Petacore uygulamalar menüsünde görünür; terminalden `petacore` komutuyla da başlatılabilir.

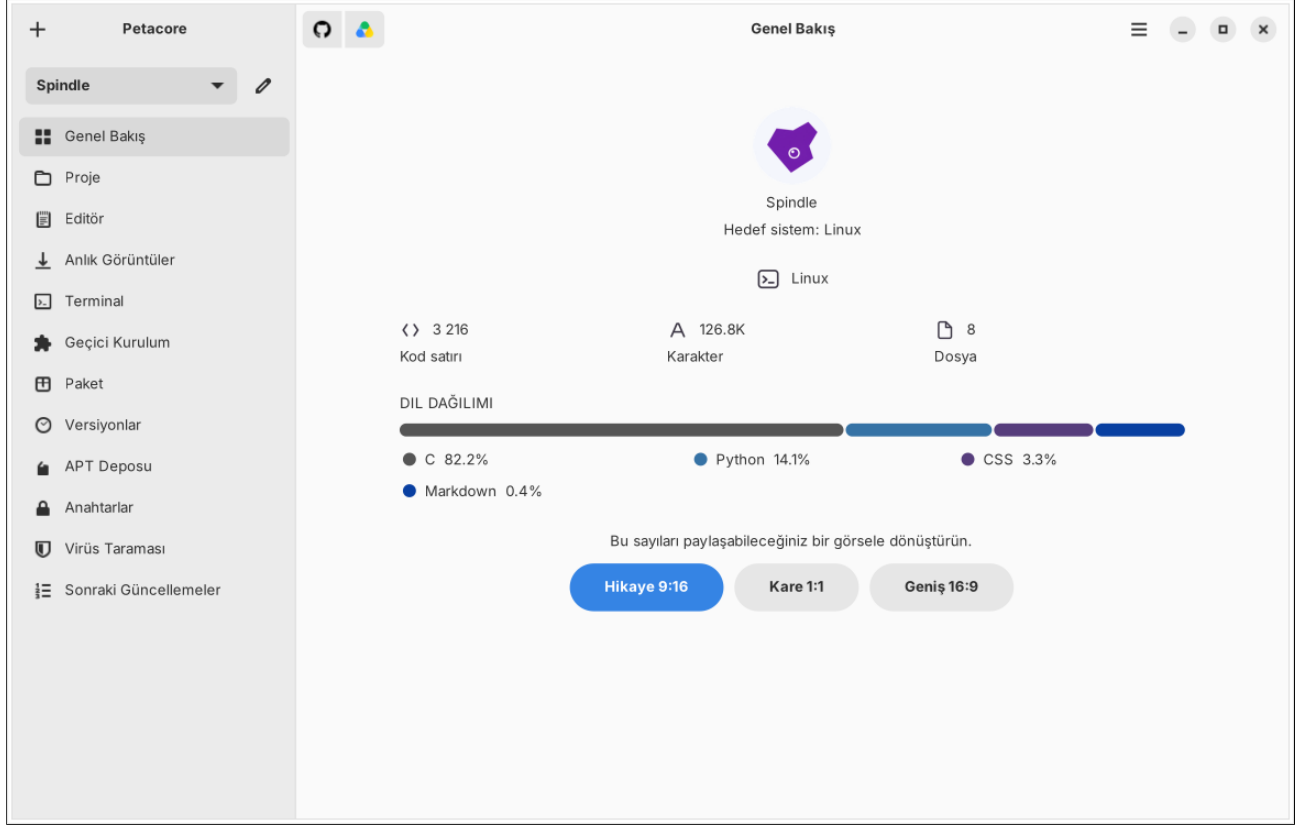
Petacore bir `.deb` paketinden de kurulabilir. Sonuç aynıdır; kurulum betiği yalnızca arayüzün kendiliğinden seçilmesini ekler.

3.2 İlk çalıştırma

İlk açılışta kısa bir sihirbaz önce dili sorar (masaüstünüzün dili önerilir), sonra GitHub ve Google Drive'ı bağlamayı, en sonunda da ilk projenizi oluşturmayı önerir. Her adım atlanabilir ve daha sonra Ayarlar'dan yapılabilir.

3.3 Ana pencere

Soldaki kenar çubuğu etkin projenin sayfalarını listeler; üstündeki proje seçici projeler arasında geçiş yapar, kalem düğmesi etkin projeyi düzenler. Pencerenin üstünde GitHub düğmesi (GitHub'a yükle), Google Drive düğmesi (Drive'a gönder) ve ana menü (Yeni Proje, Ayarlar, Petacore Hakkında, Çıkış) bulunur.



Şekil 1 — Genel Bakış sayfasıyla ana pencere.

Sayfalar projenin türüne göre değişir. Bir işletim sistemini hedefleyen proje Paket, Anahtarlar ve Geçici Kurulum sayfalarını alır; bir web projesi ise paketlenmediği ve imzalanmadığı için bunların yerine Canlı Sunucu'yu alır. Petacore bu kararı yalnızca açık bir eşleşmede verir; tanımadığı bir proje her zaman tam sayfa setini korur.

Sayfa	Amacı
Genel Bakış	Projeye ait rakamlar ve paylaşılabılır bir afiş
Proje	Proje klasörü için dosya yöneticisi
Editör	Sekmeli kod düzenleyici
Anlık Görüntüler	Tüm projenin geri dönüş noktaları
Terminal	Proje klasöründe açılan bir kabuk
Geçici Kurulum / Canlı Sunucu	Programı güvenle denemek / bir web sitesini önizlemek
Paket	İmzalı .deb ve .rpm paketleri oluşturmak
Versiyonlar	Yayın geçmişi ve her yayının kodu
APT Deposu	Yerelde veya sunucuda imzalı APT deposu
Anahtarlar	GPG imza anahtarları
Virüs Taraması	Projeyi VirusTotal ile denetlemek

Sayfa	Amacı
Sonraki Güncellemeler	Projeye ait planlar

3.4 Arayüzü seçme

Ayarlar → Arayüz, hangi tasarımın kullanıldığını gösterir ve diğerine geçer. Petacore kapanıp yeni tasarımla yeniden açılır; projeler ve ayarlar olduğu gibi kalır. Diğer tasarımın bileşenleri eksikse Petacore bunları kurmayı önerir.

4. Projeler

4.1 Proje oluşturma

Ana menüden Yeni Proje'yi ya da + düğmesini seçin. Bir ad ve projenin bulunacağı klasörü girin.

- Bir GitHub adresi verirsiniz Petacore o depoyu indirir (klonlar). Bu kod internetten geldiği için Petacore ardından, içinde bir şey çalıştırmadan önce VirusTotal'in denetleyip denetlemeyeceğini sorar.
- Adres vermezseniz yeni bir yerel git deposu başlatılır.
- Şifreleme bölümünde Google Drive kopyasının 4 haneli PIN, 8 haneli PIN veya en fazla 4096 karakterlik bir parolayla şifreli tutulmasını seçebilirsiniz. Bilgisayarınızdaki proje sıradan, okunabilir kod olarak kalır; yalnızca Drive'daki kopya — dosya ve klasör adları dahil — şifrelenir.

Şifreleme parolası hiçbir yerde saklanmaz. Kaybedilirse Drive'daki şifreli kopyayı Petacomm dahil hiç kimse açamaz.

4.2 Projeyi düzenleme ve kaldırma

Proje seçicinin yanındaki kalem düğmesi Projeyi Düzenle penceresini açar: ad, proje klasörü, GitHub adresi ve kurulu uygulama klasörü. Kurulu uygulama klasörü belirlenmişse Editör'de kaydettiğiniz her dosya, kurulu programa aynı görel yolla kopyalanabilir — terminale gerek kalmaz.

Petacore'dan Kaldır yalnızca projeyi listeden çıkarır. Diskteki hiçbir dosya silinmez.

4.3 Genel Bakış

Genel Bakış projeyi bir bakışta gösterir: kod satırı, karakter, dosya ve dil sayısı, proje boyutu, hedef sistem (Linux, Windows, macOS, Web veya çok platformlu) ve renkli bir çubukla dil dağılımı.

Paylaşım Kartı bu rakamları üç biçimde bir afişe dönüştürür — Hikaye 9:16, Kare 1:1 ve Geniş 16:9. Kartta yalnızca ad, rakamlar ve dil dağılımı yer alır; dosya adları ya da kod asla görünmez. Projenin kendi adı paylaşılmayacaksa karttaki ad değiştirilebilir.

4.4 Proje sayfası

Proje sayfası, proje klasörü için eksiksiz bir dosya yöneticisidir. Araç çubuğunun altındaki satır git dalını, projenin GitHub'a göre durumunu ve bu klasörden çalışan program sayısını gösterir.

- Çift tıklama dosyayı Editör'de açar; resim, ses ve video kendi görüntüleyicilerinde açılır.
- Sağ tık menüsünde Aç, Yeniden Adlandır, Yeni Klasör, Yolu Kopyala ve Sil bulunur.
- Silme geri alınabilir: Ctrl+Z ya da geri al düğmesi dosyaları geri getirir.
- Dosya yöneticinizden dosya sürükleyip bırakabilirsiniz. Adlar çakışırsa Değiştir / Birleştir, İkisini de Tut veya Atla seçeneklerinden birini seçersiniz.

- Arama alanı açık klasörde arar; yanındaki düğme tüm projede arar. Filtre listeyi kod, resim, ses ve video, belge veya klasörlerle sınırlar.
- Bir dosyanın üzerinde Ctrl'ye basılı tutarak hızlı önizleme açılır; uzun önizlemelerde Ctrl+kaydırma ile ilerlenir.
- Medya Dosyaları, projedeki tüm resim, ses ve videoları tek yerde toplar.
- Saat düğmesi Yedekler'i açar (bkz. 6.5).

4.5 Editör

Editör dosyaları sekmelerde, sözdizimi renklendirmesiyle ve micro düzenleyicisinin tanıdık tuşlarıyla açar (bkz. bölüm 10).

- Bölümler: uzun bir dosyanın bir kısmına ad vermek için satırları seçip bölüm ekleyin; bölüm listesi doğrudan oraya götürür.
- Odak modu seçili bölüm dışındaki satırları soluklaştırır; böylece üzerinde çalıştığınız kısım öne çıkar. Esc odağı kaldırır.
- Esc dosyadan çıkar. Kaydedilmemiş değişiklik varsa Petacore önce sorar: Esc'ye tekrar basmak kaydetmeden çıkar, Ctrl+S kaydedip çıkar. Bu soru Ayarlar'dan kapatılabilir.
- Projenin kurulu uygulama klasörü varsa, kaydederken kurulu programın da güncellenmesi önerilir.

4.6 Anlık görüntüler ve otomatik kayıt

Anlık görüntü, projenin eksiksiz bir geri dönüş noktasıdır. Anlık Görüntü Oluştur istediğiniz an, isteğe bağlı bir etiketle bir tane oluşturur. Otomatik Kayıt belirli aralıklarla (varsayılan 10 dakika) kendiliğinden oluşturur; en yeni 40 otomatik görüntü saklanır.

Bir anlık görüntüye dönmekten önce mevcut durumun güvenlik görüntüsü alınır; böylece geri dönüş de geri alınabilir. Anlık görüntüler projenin dışında, kendi veri klasörünüzde saklanır; .git, versions klasörü ve derleme artıkları dahil edilmez.

4.7 Terminal

Terminal sayfası alıştığınız kabuğu doğrudan proje klasöründe açar.

4.8 Geçici Kurulum

Geçici Kurulum, komutları sisteminizin mühürlü bir kopyasında çalıştırır. Gerçek sistem burada salt okunurdur: kurma, silme ve düzenleme yalnızca bu kullan-at oturumu etkiler; Simülasyonu Bitir her şeyi iz bırakmadan kaldırır.

- Mevcut Projeyi Dene, oturumu projenin bir kopyasıyla başlatır.
- Bir .deb Dosyasını Geçici Kur, bir paketi yalnızca bu oturumun içine kurar.
- Uygulamayı Çalıştır, grafik veya terminal programlarını içeride başlatır.
- Geçici Kurulum'un varsayılan olarak interneti yoktur. İçerideki bir şey internete çıkmaya çalışırsa Petacore bunu bildirir; izin verebilir, kapalı tutabilir ya da bir daha sorulmamasını seçebilirsiniz.

Tam yalıtım için bubblewrap gerekir. O yoksa sayfa “yalıtılmamış” uyarısı verir ve kurulum komutunu gösterir.

4.9 Canlı Sunucu

Web projelerinde Canlı Sunucu, siteyi hiçbir derleme adımı olmadan Petacore içinde gösterir ve bir dosya değiştiğinde kendiliğinden yeniler. F11 siteyi tüm pencereye yayar; Esc geri döndürür.

Sunucu varsayılan olarak yalnızca bu bilgisayara yanıt verir. Yerel ağda paylaş, aynı ağdaki bir telefonda ya da başka bir cihazdan erişilmesini sağlar ve orada açılacak adresi gösterir. WebKitGTK yoksa önizleme her zamanki tarayıcınızda açılır.

4.10 Sonraki Güncellemeler

Sonraki Güncellemeler projeye ait planları tutar: bir başlık, ayrıntılar, öncelik (Yüksek, Normal, Bir ara) ve ekli belgeler. Planlar düzenlenebilir, tamamlandı olarak işaretlenebilir ve kopyalanabilir. Projenin içinde saklandıkları için projeye birlikte taşınırlar ve Google Drive ile eşitlenebilirler.

5. Paketler ve yayınlar

5.1 Paket

Paket sayfası projeden bir kurulum paketi oluşturur: Debian, Ubuntu ve Mint için .deb, Fedora, openSUSE ve RHEL için .rpm. Paket, programı hedef sistemde /opt/<ad> altına kurar ve uygulamalar menüsüne ekler.

Alan	Anlamı
Ad	Küçük harflerle paket adı
Sürüm	Örneğin 1.6 veya 2.0.1
Bakımcı	Sorumlu kişinin adı ve e-posta adresi
Açıklama	Yazılım merkezlerinde ve paket yöneticilerinde görünür
Lisans	Listeden seçilir ya da yazılır; Lisansları karşılaştırmak farklarını anlatır
İmzalama	Paketi etkin anahtarla imzalar (varsayılan olarak açık)

Giriş bilgisi taşıyan dosyalar — .env, özel anahtarlar, .netrc, .npmrc ve benzerleri — pakete asla girmez; Petacore kaç tanesini dışarıda bıraktığını bildirir.

5.2 İmza anahtarları

Bir GPG anahtarı dijital imzanızdır: bir paketin gerçekten sizden geldiğini ve kimse tarafından değiştirilmediğini kanıtlar. Anahtarlar sayfası anahtarlarınızı listeler ve hangisinin imzaladığını gösterir.

Ayar	Seçenekler
Ad, e-posta, yorum	Anahtarın kime ait olduğu
Anahtar türü	Ed25519 (modern, hızlı, önerilen) veya eski sistemler için RSA 4096 / 3072 / 2048
Geçerlilik süresi	Süresiz, 1 yıl, 2 yıl veya 5 yıl
Parola	İsteğe bağlı. Belirlenirse GPG her imzada sorar — daha güvenli

- İmzalama için kullan, bir anahtarı etkin imza anahtarı yapar.
- Genel anahtarı dışa aktar, kullanıcılarınızın paketleri doğrulayacağı dosyayı kaydeder.
- Anahtarlarınızın genel yarıları Google Drive'a yedeklenebilir. Özel anahtarlar bu bilgisayardan asla çıkmaz.
- Silme geri alınamaz: o anahtarla artık paket imzalanamaz.

5.3 Versiyonlar

Versiyonlar, projenin yayımladığı her sürümü paketleriyle birlikte, en yenisi başta olmak üzere saklar. İki sekmesi vardır: Yeni Versiyon ve Versiyon Kütüphanesi.

- Yeni Versiyon, sürümü proje dosyalarının şu anki halinden oluşturur: sürüm numarası, yayın türü (Kararlı, Sürüm Adayı, Beta, Alfa), ön sürüm numarası, biçimler (.deb, .rpm), imzalama ve sürüm notları. Önizleme hem insanların okuduğu adı ("1.6 Beta 2") hem de paket yöneticilerinin karşılaştığı sürümü ("1.6~beta2") gösterir.
- Versiyon Kütüphanesi yayınları serilere (1.6, 2.0 ...) göre gruplar ve en güncelini işaretler. Her yayın paketlerini, imzasını ve notlarını gösterir.
- Kodu Görüntüle, herhangi bir yayının kodunu açar — doğrudan o yayının kendi paketinden, yayımlandığı haliyle ve salt okunur olarak. Dosyalar süzülebilir, içerikleri kopyalanabilir.
- Eski .deb veya .rpm dosyalarını sayfaya bırakarak bir yayını geçmişe ekleyebilirsiniz; asılları yerinde kalır.
- Notları Düzenle, .deb'i Depoya Ekle ve Versiyonu Kaldır her yayında bulunur. Kaldırılan sürüm sürüm çöpüne gider ve geri getirilebilir.

Yayınlar projenin versions klasöründe durur. Bu klasör her paketin, anlık görüntünün, istatistiğin ve git geçmişinin dışında tutulur; böylece hiçbir paket kendinden öncekileri içermez.

5.4 APT deposu

Bir depo tekil paketleri, kullanıcıların bir kez ekleyip sonra güncellemelerini aldığı bir yazılım kaynağına dönüştürür. Petacore ağacı kurar, dizini imzalar ve paketlerinizle uyumlu tutar.

Ayar	Anlamı
Ad, Depo Klasörü	Deponun adı ve ağacın kurulduğu yer
Genel Adres	Kullanıcıların kurulum yapacağı URL, örneğin https://repo.example.com
Suite, Bileşen, Mimariler	Olağan değerler: stable, main, amd64, arm64
Origin, Label, Açıklama	Deponun kendini apt'ye nasıl tanıttığı
İmza Anahtarı	apt, dizini imzasız bir arşivi reddeder
Yayın Yöntemi	Buradan yayımlanmaz, bir klasöre kopyalanır veya SSH (rsync) ile gönderilir

- Paket Ekle'yi kullanın ya da .deb dosyalarını sayfaya bırakın. "Üretilen paketleri depoya ekle" açıkken her yeni .deb kendiliğinden eklenir.
- Yeniden Kur ve İmzala dizini yeniler; dizin paketlerle uyuşmadığında bir uyarı görünür.
- Yayınla sunucudaki kopyayı klasörle eşler; Sunucuyu Denetle ikisini karşılaştırır.
- Rozetler ve filtreler dizinde olmayan, imzasız, yayımlanmamış ya da yerine yenisi gelmiş paketleri gösterir.
- Her pakette İmzayı Doğrula, Kurulum Komutunu Kopyala, Yeniden Adlandır, Bulunduğu Klasörü Aç ve Depodan Kaldır bulunur.
- Kaldırılan paketler depo çöpüne gider ve geri konabilir; Eski Sürümleri Kaldır her paketin yalnızca en yenisini bırakır.
- Kullanıcı Yönergelerini Kopyala ve Arşiv Anahtarını Dışa Aktar, kullanıcılarınızın depoyu eklemesi için gereken her şeyi verir.

5.5 Sunucudaki depo

Sunucu sekmesinde Petacore depo klasörünü SSH üzerinden doğrudan sunucunuzda yönetir. Dosyalar scp ile yüklenir; izin sunucuda oluşturulur ve bilgisayarınızda imzalanır, böylece imza anahtarı bilgisayarınızdan asla çıkmaz.

1. Sunucu adını veya IP'sini, portu, kullanıcıyı ve sunucudaki depo klasörünü girin, örneğin `/var/www/repo`.
2. Bir SSH anahtarı seçin ya da Petacore için yeni bir tane oluşturun: Ed25519, en az 8 karakterlik ve asla saklanmayan bir parolayla korunur.
3. Petacore'un gösterdiği satırı sunucuda o kullanıcının `~/.ssh/authorized_keys` dosyasına ekleyin.
4. Bağlan'a basın. İlk seferde sunucunun parmak izini sunucunun kendi bildirdiğiyle karşılaştırın ve yalnızca eşleşiyorsa onaylayın.

Bağlandıktan sonra dosya yükleyebilir (.deb dosyaları kendiliğinden havuza konur), klasör oluşturabilir, yeniden adlandırabilir, indirebilir, öğeleri sunucu çöpüne taşıyıp geri getirebilir ve Yeniden Kur ve İmzala'yı kullanabilirsiniz.

Koruma	Anlamı
Sunucu kimliği denetimi	Sunucunun kimliği her bağlantıda denetlenir; değişirse Petacore bağlanmayı reddeder
Yalnızca anahtarla giriş	Parolalar asla gönderilmez
Modern şifreleme	Mümkün olduğunda kuantum sonrası anahtar değişimi, kimliği doğrulanmış şifreleme
İki adımlı giriş	Doğrulayıcı uygulama kodları ve FIDO2 güvenlik anahtarları
Hiçbir şey saklanmaz	Parolalar ve iki adımlı doğrulama kodları asla kaydedilmez

6. Çevrimiçi hizmetler

6.1 GitHub'ı bağlama

1. GitHub ile bağlan düğmesine basın — Ayarlar'da, ilk açılış sihirbazında ya da GitHub gerektiği her an.
2. Tarayıcınızda GitHub açılır ve tek kullanımlık kod zaten panoya kopyalanmıştır.
3. Kodu yapıştırın, onaylayın ve Authorize'a basın. Petacore kendiliğinden bağlanır.

Bilgisayarda GitHub CLI (gh) zaten oturum açmışsa tek tık yeter. gh yoksa Petacore onu kurmayı önerir. Diğer giriş seçenekleri altında kişisel erişim belirteci (token) elle de yapıştırılabilir. Giriş bilgisi masaüstü anahtarlığında saklanır; asla bir dosyada ya da adreste tutulmaz.

6.2 GitHub'a yükle

Pencerenin üstündeki GitHub düğmesi GitHub'a yükle penceresini açar. Tek adımda kod main dalına gönderilir ve projenin GitHub sayfasında yeni bir sürüm olarak yayınlanır.

- Önerilerden bir sürüm seçin — örneğin 1.0 Beta 3, 1.0 Beta 2.1, 1.0 RC 1 veya 1.0 — ya da kendiniz yazın. Kütüphanede olup henüz GitHub'da olmayan sürüm ilk sırada önerilir.
- Etiket sürümden oluşturulur (v1.0-beta.3); beta, alfa ve sürüm adayları ön sürüm olarak işaretlenir.
- Projenin henüz bir GitHub deposu yoksa oluşturulur — varsayılan olarak gizli.
- Siz değiştirmedikçe dal main'dir. Sürüm notları isteğe bağlıdır; boş bırakılırsa GitHub bunları commit'lerden kendisi yazar.

- O sürümün Versiyon Kütüphanesi'ndeki .deb ve .rpm paketleri yayına eklenir.
- Yalnızca kodu göndermek için “Yeni sürüm olarak yayınla” seçeneğini kapatın.

Giriş bilgisi içeren dosyalar asla gönderilmez. GitHub'daki mevcut çalışma korunur: sizinkiyle çakışırsa gönderim durur ve hiçbir şeyin üzerine yazılmaz. Zaten yayınlanmış bir sürümün yerine asla yenisi konmaz.

6.3 GitHub'dan proje çekme

Bir GitHub adresinden proje oluşturulduğunda Petacore, indirilen kodun VirusTotal ile denetlenip denetlenmeyeceğini bir kez sorar. Tara, Virüs Taraması sayfasını açar ve denetimi hemen başlatır.

6.4 Google Drive

Kendi Google Drive hesabınızı Ayarlar'dan bağlayın: tarayıcı açılır, Google'a giriş yapıp İzin ver'e basarsınız. Petacore “Petacomm Petacore™” adlı bir klasör oluşturur ve her projeyi, medya dosyaları dahil, kendi alt klasöründe tutar. Petacore'un kendi kayıtları bunların yanında gizli bir .petacore klasöründe durur.

- Drive düğmesi etkin projeyi gönderir. Drive bu bilgisayardaki hale getirilir — ve hiçbir şey kaybolmaz (bkz. 6.5).
- Bu bilgisayarın son eşitlemesinden sonra başka bir bilgisayar daha yeni bir kopya göndermişse Petacore önce onu almayı önerir.
- Gönderim Drive'dakinden küçük olacaksa Petacore önce uyarır (Ayarlar'dan kapatılabilir).
- Drive'dan Projeleri Getir, yalnızca Drive'da bulunan projeleri ~/Projects altına indirir.
- Şifreli projeler şifreli bir katmandan gönderilir; Drive yalnızca okunamaz veri görür.

6.5 Yedekler

Petacore, Drive ile eşitlerken hiçbir dosyayı silmez. Bir eşitlemenin değiştirdiği ya da kaldırdığı her dosya önce yedeklenir — gönderirken Drive'da, bir kopya gelirken bu bilgisayarda.

- Yedekler günlere göre listelenir: ne zaman, hangi bilgisayardan, kaç dosya ve ne büyüklükte.
- Bir yedeği açınca dosyaları görünür. Geri koy, yedeğin tamamını ya da tek bir dosyayı geri getirir.
- Bir şey geri konmadan önce mevcut dosyalar da yedeklenir; böylece bu işlem de geri alınabilir.
- Yedekler varsayılan olarak 30 gün saklanır (Ayarlar'da 7, 30, 90, 365 gün ya da süresiz) ve sonra kendiliğinden silinir.

Yedekler Proje sayfasındaki saat düğmesinden, eşitleme bildirimlerindeki Yedekler düğmesinden ve Ayarlar → Google Drive → Yedekleri göster'den açılır.

6.6 Diğer bilgisayarlarınızdan gelen değişiklikler

Aynı Google Drive hesabına bağlı bilgisayarlar birbirinin en yeni kopyasını kendiliğinden alır: açılıştan birkaç saniye sonra, ardından her 10 dakikada bir (Ayarlar'da 5, 10, 30 veya 60). Yalnızca Drive'da bulunan projeler de indirilir; ad zaten kullanılıyorsa yeni bir klasöre.

Durum	Petacore ne yapar
Drive'da daha yeni kopya var, burada değişiklik yok	Sessizce alır ve hangi bilgisayarın gönderdiğini bildirir
Drive'da daha yeni kopya var ve burada da değişiklik var	Sorar: Drive'dakini al, Benimkini gönder veya Şimdi değil
Şifreli proje	Kendiliğinden alınmaz — parola gerekir

Hangi tarafı seçerseniz seçin, diğer taraftaki dosyalar Yedekler'de saklanır. Özellik varsayılan olarak açıktır ve Ayarlar → Google Drive'dan kapatılabilir.

6.7 Virüs Taraması

Virüs Taraması, projeyi kullanıcılarınız kurmadan önce VirusTotal motorlarıyla denetler. API anahtarı size aittir: virustotal.com'da ücretsiz bir hesap açın, anahtarı profilinizden kopyalayıp bir kez girin. Anahtar masaüstü anahtarlığında saklanır.

- Projeyi Tara, projeyi tek bir arşiv halinde paketler; değişmemiş bir proje parmak izinden tanınır ve iki kez gönderilmez.
- Siz onaylamadan hiçbir şey gönderilmez; onay her proje için bir kez istenir. Giriş bilgisi dosyaları, .git, node_modules ve versions klasörü asla gönderilmez.
- Sonuçlar: Hiçbir şey işaretlenmedi, motor adlarıyla birlikte "t motordan n tanesi işaretledi" ya da "hiç görülmedi" — yeni oluşturulmuş bir dosya için olağan yanıt.
- Yeni oluşturulmuş, imzasız bir programı bir iki motorun işaretlemesi çoğunlukla yanlış alarmdır.
- Ücretsiz anahtar dakikada 4, günde 500 denetime izin verir.

7. Ayarlar

Ayar	Varsayılan	Açıklama
Dil	Masaüstü dili	Altı dil; uygulamak için Petacore yeniden başlar
Görünüm	Sistemi İzle	Açık, koyu veya masaüstüne göre
Odak modu	Açık	Seçili bölüm dışındaki satırları soluklaştırır
Küçük gönderim uyarısı	Açık	Gönderim Drive'daki kopyadan küçükse uyarır
Geçici Kurulum'da internet	Kapalı	Geçici Kurulum'daki kodun ağa çıkmasına izin verir
Esc ile çıkarken sor	Açık	Kaydedilmemiş dosyalar için onay
Otomatik Kayıt	Açık, 10 dakika	Otomatik anlık görüntüler
GitHub	Bağlı değil	Bağlan, bağlantıyı kes, isteğe bağlı OAuth Client ID
Google Drive	Bağlı değil	Bağlan, bağlantıyı kes, projeleri getir
Değişiklikleri otomatik çek	Açık, 10 dakika	Diğer bilgisayarlarınızdan gelen değişiklikler
Yedekleri sakla	30 gün	7, 30, 90, 365 gün veya süresiz
Arayüz	Algılanan	GNOME veya KDE Plasma

8. Güvenlik ve gizlilik

- Giriş bilgileri ve API anahtarları masaüstü anahtarlığında saklanır. Anahtarlık yoksa yalnızca sizin okuyabildiğiniz bir dosyaya yazılır.
- Giriş bilgisi taşıyan dosyalar (.env, özel anahtarlar, .netrc ...) asla bir pakete, GitHub gönderimine ya da VirusTotal taramasına girmez.
- GitHub girişi yalnızca github.com'a verilir; ona benzeyen bir adres hiçbir şey alamaz.

- Özel imza anahtarları bilgisayarınızdan asla çıkmaz; yalnızca genel anahtarlar yedeklenir.
- Sunucu girişleri yalnızca anahtarla yapılır; parolalar ve iki adımlı doğrulama kodları asla saklanmaz.
- Geçici Kurulum siz açmadıkça ağa çıkamaz; Canlı Sunucu siz paylaşmadıkça yalnızca bu bilgisayara yanıt verir.
- Onayınız olmadan VirusTotal'a hiçbir şey gönderilmez.
- Eşitleme hiçbir dosyayı yok etmez: değiştirdiği her şey Yedekler'de saklanır.

9. Sorun giderme

Sorun	Çözüm
Google Drive desteği için rclone gerekiyor	<code>sudo apt install rclone</code>
Anahtarlığa ulaşılamıyor	<code>sudo apt install libsecret-tools</code> , ardından yeniden deneyin
GitHub CLI hazır değil	gh'yi kur düğmesine basın ya da <code>sudo apt install gh</code>
Geçici Kurulum: yalıtılmamış	<code>sudo apt install bubblewrap</code>
Paylaşım kartı yalnızca SVG kaydedildi	<code>sudo apt install librsvg2-bin</code>
Canlı Sunucu: yerleşik tarayıcı yok	<code>sudo apt install gir1.2-webkit-6.0</code> ya da Tarayıcıda Aç
Depo dizini oluşturulamıyor	<code>sudo apt install apt-utils dpkg-dev</code>
Sunucu bağlantısı: OpenSSH eksik	<code>sudo apt install openssh-client</code>
VirusTotal anahtarı reddetti	Anahtarı VirusTotal profilinizden kontrol edin; ağ engeli geçici de olabilir — daha sonra yeniden deneyin
İmza anahtarının süresi doldu ya da anahtar yok	Anahtarlar sayfasında yeni bir anahtar oluşturup imzalama için kullan'ı seçin
Sunucunun kimliği değişti	Bağlanmayın. Sunucu yöneticisine danışın; eski anahtarı yalnızca değişiklik doğrulandıktan sonra unutturun
GitHub: dalda uyuşmayan değişiklikler var	Bu değişiklikleri Terminal'de git ile alın, çakışmaları çözün, sonra yeniden gönderin
Sürüm GitHub'da zaten yayında	Başka bir sürüm seçin
Proje iki yerde değişti	Hangi kopyanın kalacağını seçin; diğer taraf Yedekler'de durur
Şifreli projede yanlış parola	Parola kurtarılamaz; doğru parolayı kullanın
Diğer arayüzün bileşenleri eksik	Şimdi Kur'a basın ya da Petacore'un gösterdiği komutu çalıştırın

10. Klavye kısayolları

Tuşlar	İşlev
Ctrl+S	Dosyayı kaydet (Editör)
Ctrl+Q	Dosyayı kapat (Editör)
Ctrl+F	Bul; Enter sonraki eşleşmeye gider, Esc kapatır

Tuřlar	İřlev
Ctrl+K / Ctrl+D	Satırı kes / satırı çoęalt
Ctrl+Z / Ctrl+Y	Geri al / yinele
Esc	Dosyadan çık; tekrar Esc kaydetmeden çıkar
Ctrl+A / Delete	Tümünü seç / sil (Proje sayfası, geri alınabilir)
Ctrl+Z	Silinen dosyaları ya da kaldırılan paketleri geri getir
Ctrl + işareti	Hızlı önizleme; Ctrl+kaydırma ile ilerlenir
F11 / Esc	Canlı Sunucu tam ekran / tam ekrandan çık

11. Dosyalar ve kaldırma

11.1 Petacore verilerini nerede tutar

Konum	İçerik
~/local/share/petacore-app	Programın kendisi
~/config/petacore	Ayarlar (yalnızca siz okuyabilirsiniz)
~/local/share/petacore/snapshots	Anlık görüntüler
~/local/share/petacore/drive-backups	Bu bilgisayarda deęiřtirilen dosyaların yedekleri
<proje>/versions	Yayın geęmiři (paketlerden ve git'ten gizli)
<proje>/petacore	Sonraki Güncellemeler planları
Masaüstü anahtarlığı	GitHub giriři, VirusTotal anahtarı

11.2 Kaldırma

Petacore klasöründe `./uninstall.sh` komutunu kendi kullanıcınızla çalıştırın. Programın kaldırılması ile anlık görüntülerinizin ve yedeklerinizin silinmesi iki ayrı soru olarak sorulur ve siz onaylamadan hiçbir řey silinmez. Proje klasörlerinize asla dokunulmaz.

12. Lisans ve telif hakkı

© Petacomm I.S. 2026. Tüm hakları saklıdır. Petacomm Petacore™, Petacomm I.S.'nin ticari markasıdır.

Petacore özgür ve açık kaynaklı bileřenler kullanır; bunlar yazarlarının mülkiyetinde kalır ve kendi lisansları altında kullanılır. GitHub, Google Drive, VirusTotal, GNOME, KDE Plasma, Ubuntu ve Debian ilgili sahiplerinin ticari markalarıdır.

Bu kılavuz Petacore 1.1 Beta 2'yi anlatır. Özellikler sonraki sürümlerde deęiřebilir.

1. Vorwort

Vielen Dank, dass Sie sich für Petacore entschieden haben, die Desktop-Werkbank zum Entwickeln und Ausliefern von Linux-Software. Petacore vereint alles, was ein Projekt braucht, in einem Fenster: seine Dateien und seinen Code, seine Geschichte, seine Pakete und Signaturschlüssel, seine Paketquelle, seine Kopie auf GitHub und in Google Drive sowie eine Virenprüfung, bevor etwas Ihre Nutzer erreicht.

Bitte lesen Sie diese Anleitung vor der ersten Verwendung sorgfältig durch. Sie beschreibt jeden Bereich von Petacore, alle Einstellungen und was Petacore mit Ihren Dateien und Zugangsdaten tut — und niemals tut. Mit diesem Wissen arbeiten Sie schneller und mit voller Gelassenheit.

Entwickler: Petacomm I.S.
Produkt: Petacomm Petacore™
Beschriebene Version: 1.1 Beta 2
Projektseite: github.com/petacomm/petacore

2. Technische Beschreibung

2.1 Allgemeines

Petacore ist eine native Desktop-Anwendung für Linux. Sie läuft vollständig auf Ihrem eigenen Computer und arbeitet mit gewöhnlichen Projektordnern: Nichts muss umgewandelt werden, und jede Datei bleibt mit jedem anderen Werkzeug lesbar. Online-Dienste (GitHub, Google Drive, VirusTotal) werden nur genutzt, wenn Sie sie verbinden — mit Ihren eigenen Konten.

Petacore gibt es mit zwei Oberflächen, die dieselben Funktionen bieten: eine für den GNOME-Desktop und eine für KDE Plasma. Das Installationsprogramm wählt automatisch die passende; Sie können jederzeit wechseln.

2.2 Systemvoraussetzungen

Merkmal	Voraussetzung
Betriebssystem	Ubuntu 26.04 LTS oder eine andere Debian- bzw. Ubuntu-basierte Distribution
Desktop	GNOME (GTK 4, libadwaita) oder KDE Plasma (Qt 6, PySide6)
Python	3.12 oder neuer
Speicherplatz	Etwa 50 MB für Petacore, zusätzlich Platz für Schnappschüsse und Sicherungen
Internet	Nur für GitHub, Google Drive, VirusTotal und Serververbindungen

2.3 Technische Daten

Funktion	Angabe
Sprachen der Oberfläche	English, Türkçe, Deutsch, Français, 中文, Українська
Paketformate	.deb (Debian, Ubuntu, Mint) und .rpm (Fedora, openSUSE, RHEL)
Paketsignatur	GnuPG — Ed25519 (empfohlen) oder RSA 2048 / 3072 / 4096
Versionsverwaltung	git; Veröffentlichung auf GitHub mit versionierten Releases

Funktion	Angabe
Cloud-Kopie	Google Drive über rclone, auf Wunsch Ende-zu-Ende-verschlüsselt
Sicherheitsprüfung	VirusTotal API v3, mit Ihrem eigenen Schlüssel
Paketquelle	Signiertes APT-Archiv, lokal oder per SSH auf Ihrem eigenen Server
Serververbindung	OpenSSH, nur Anmeldung per Schlüssel, Zwei-Faktor-Anmeldung möglich
Isolation	bubblewrap-Sandbox, Netzwerk standardmäßig aus

2.4 Verwendete Komponenten

Petacore setzt auf bewährte Standardwerkzeuge. Das Installationsprogramm richtet sie ein; fehlt eine Komponente, wird das in klaren Worten gemeldet — zusammen mit dem Befehl, der sie installiert.

Komponente	Verwendet für
git, gh	Versionsgeschichte; GitHub-Anmeldung und Veröffentlichung
rclone	Verbindung zu Google Drive, Synchronisierung und Sicherungen
gnupg	Signaturschlüssel erstellen; Pakete und Paketquellen signieren
dpkg-dev, rpm, apt-utils	Pakete und Paketquellen-Indizes erzeugen
bubblewrap	Die isolierte Sandbox
openssh-client, rsync	Serververbindung und Veröffentlichung
libsecret-tools	Anmeldungen und API-Schlüssel im Schlüsselbund des Desktops ablegen
libsvg2-bin	Teilen-Karten als PNG-Bilder speichern
WebKitGTK (optional)	Live-Server-Vorschau direkt im Fenster anzeigen

3. Installation und erster Start

3.1 Petacore installieren

1. Entpacken Sie das Petacore-Archiv in einen beliebigen Ordner.
2. Öffnen Sie in diesem Ordner ein Terminal und führen Sie `./install.sh` aus — als Sie selbst, niemals mit `sudo`. Petacore wird in Ihren eigenen Persönlichen Ordner installiert.
3. Das Installationsprogramm erkennt Ihren Desktop (GNOME oder KDE Plasma). Gelingt das nicht, fragt es, welche Oberfläche Sie möchten.
4. Werden Systempakete benötigt, fragt es einmal nach Ihrem Passwort und installiert sie mit `apt`.
5. Danach erscheint Petacore im Anwendungsmenü; im Terminal startet es mit dem Befehl `petacore`.

Petacore lässt sich auch aus einem `.deb`-Paket installieren. Das Ergebnis ist dasselbe; das Installationsskript ergänzt nur die automatische Wahl der Oberfläche.

3.2 Erster Start

Beim ersten Start fragt ein kurzer Assistent nach der Sprache (die Sprache Ihres Desktops wird vorgeschlagen), bietet dann an, GitHub und Google Drive zu verbinden, und schließlich, Ihr erstes Projekt anzulegen. Jeder Schritt kann übersprungen und später in den Einstellungen nachgeholt werden.

3.3 Das Hauptfenster

Links listet die Seitenleiste die Seiten des aktiven Projekts auf; die Projektauswahl darüber wechselt zwischen Projekten, der Stift-Knopf bearbeitet das aktive Projekt. Oben im Fenster befinden sich der GitHub-Knopf (Auf GitHub hochladen), der Google-Drive-Knopf (an Drive senden) und das Hauptmenü (Neues Projekt, Einstellungen, Über Petacore, Beenden).

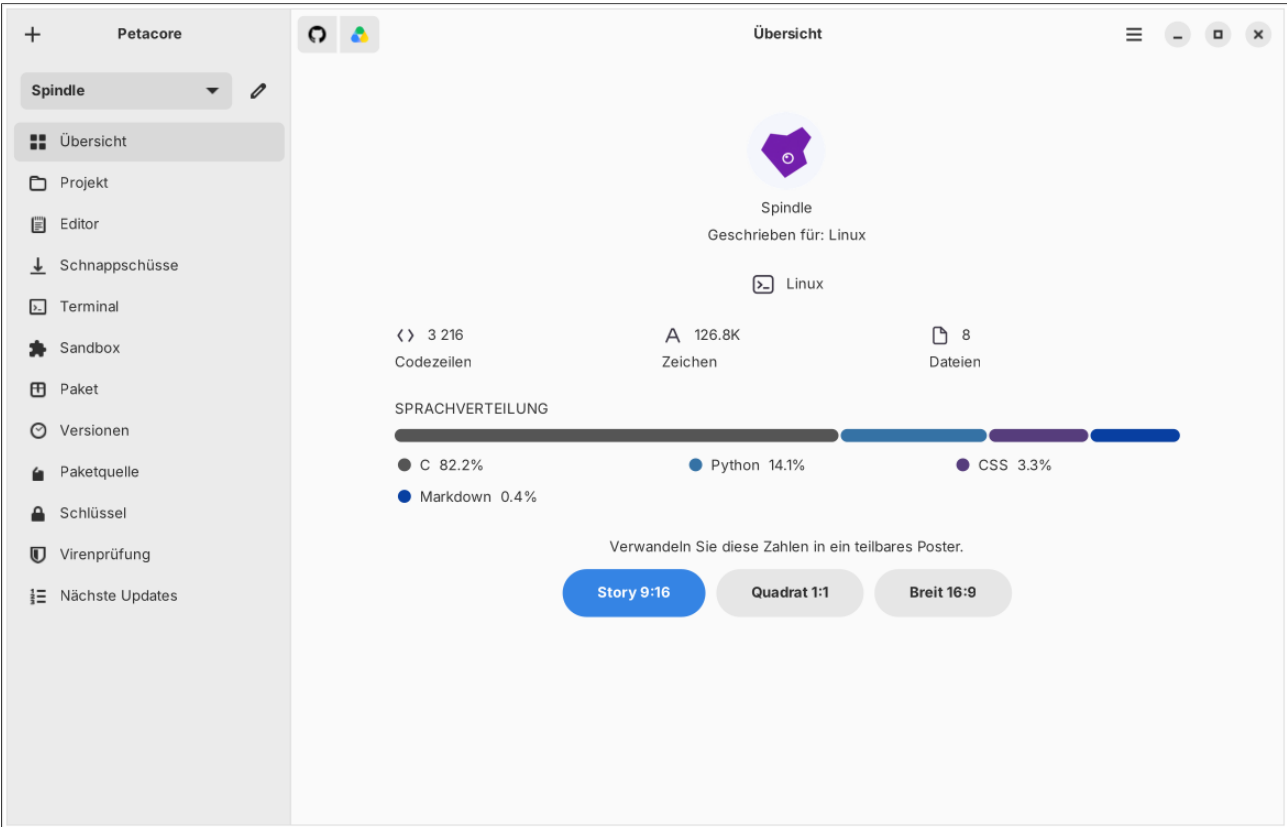


Abbildung 1 — Das Hauptfenster mit der Seite Übersicht.

Welche Seiten erscheinen, hängt von der Art des Projekts ab. Ein Projekt für ein Betriebssystem erhält Paket, Schlüssel und Sandbox; ein Webprojekt erhält stattdessen Live-Server, da es weder paketiert noch signiert wird. Petacore entscheidet das nur bei eindeutiger Erkennung — ein unbekanntes Projekt behält immer alle Seiten.

Seite	Zweck
Übersicht	Kennzahlen des Projekts und ein Poster zum Teilen
Projekt	Dateiverwaltung für den Projektordner
Editor	Code-Editor mit Reitern
Schnappschüsse	Wiederherstellungspunkte des ganzen Projekts
Terminal	Eine Shell im Projektordner
Sandbox / Live-Server	Das Programm gefahrlos testen / eine Website in der Vorschau sehen

Seite	Zweck
Paket	Signierte .deb- und .rpm-Pakete erzeugen
Versionen	Veröffentlichungsverlauf und der Code jeder Version
Paketquelle	Signierte APT-Paketquelle, lokal oder auf einem Server
Schlüssel	GPG-Signaturschlüssel
Virenprüfung	Das Projekt mit VirusTotal prüfen
Nächste Updates	Pläne für das Projekt

3.4 Die Oberfläche wählen

Einstellungen → Oberfläche zeigt, welches Design aktiv ist, und wechselt zum anderen. Petacore schließt sich und öffnet sich im neuen Design wieder; Projekte und Einstellungen bleiben erhalten. Fehlen die Komponenten des anderen Designs, bietet Petacore an, sie zu installieren.

4. Projekte

4.1 Ein Projekt anlegen

Wählen Sie Neues Projekt im Hauptmenü oder den Knopf +. Geben Sie einen Namen und den Ordner ein, in dem das Projekt liegen soll.

- Mit einer GitHub-Adresse lädt Petacore dieses Repository herunter (Klonen). Da der Code aus dem Internet stammt, fragt Petacore anschließend, ob VirusTotal ihn prüfen soll, bevor Sie darin etwas ausführen.
- Ohne Adresse wird ein neues lokales git-Repository angelegt.
- Unter Verschlüsselung können Sie die Kopie in Google Drive mit einer 4-stelligen PIN, einer 8-stelligen PIN oder einer Passphrase mit bis zu 4096 Zeichen verschlüsseln lassen. Das Projekt auf Ihrem Computer bleibt gewöhnlicher, lesbarer Code; verschlüsselt wird nur die Kopie in Drive — einschließlich der Datei- und Ordnernamen.

Das Verschlüsselungspasswort wird nirgends gespeichert. Geht es verloren, kann niemand — auch Petacore nicht — die verschlüsselte Kopie in Drive öffnen.

4.2 Projekt bearbeiten und entfernen

Der Stift-Knopf neben der Projektauswahl öffnet Projekt bearbeiten: Name, Projektordner, GitHub-Adresse und Ordner der installierten Anwendung. Ist ein solcher Ordner gesetzt, kann jede im Editor gespeicherte Datei unter demselben relativen Pfad in das installierte Programm kopiert werden — ganz ohne Terminal.

Aus Petacore entfernen nimmt das Projekt nur aus der Liste. Keine Datei auf dem Datenträger wird gelöscht.

4.3 Übersicht

Die Übersicht zeigt das Projekt auf einen Blick: Codezeilen, Zeichen, Dateien, Sprachen und Projektgröße, das Zielsystem (Linux, Windows, macOS, Web oder plattformübergreifend) und die Sprachverteilung als farbigen Balken.

Die Teilen-Karte macht aus diesen Zahlen ein Poster in drei Formaten — Story 9:16, Quadrat 1:1 und Breit 16:9. Die Karte zeigt nur den Namen, die Zahlen und die Sprachverteilung, niemals Dateinamen oder Code; der Name auf der Karte lässt sich ändern, falls der eigentliche Projektname nicht geteilt werden soll.

4.4 Die Seite Projekt

Die Seite Projekt ist eine vollständige Dateiverwaltung für den Projektordner. Die Zeile unter der Werkzeugleiste zeigt den git-Zweig, den Stand gegenüber GitHub und wie viele Programme aus diesem Ordner laufen.

- Ein Doppelklick öffnet eine Datei im Editor; Bilder, Audio und Video öffnen sich in ihrer eigenen Ansicht.
- Ein Rechtsklick bietet Öffnen, Umbenennen, Neuer Ordner, Pfad kopieren und Löschen.
- Löschen ist umkehrbar: Strg+Z oder der Rückgängig-Knopf holt die Dateien zurück.
- Ziehen Sie Dateien aus Ihrer Dateiverwaltung hinein. Bei gleichen Namen wählen Sie Ersetzen / Zusammenführen, Beide behalten oder Überspringen.
- Das Suchfeld durchsucht den geöffneten Ordner, der Knopf daneben das ganze Projekt. Der Filter beschränkt die Liste auf Code, Bilder, Audio und Video, Dokumente oder Ordner.
- Halten Sie Strg über einer Datei für eine Schnellvorschau gedrückt; mit Strg+Mausrad blättern Sie durch lange Vorschauen.
- Mediendateien sammelt alle Bilder, Töne und Videos des Projekts an einem Ort.
- Der Uhr-Knopf öffnet die Sicherungen (siehe 6.5).

4.5 Editor

Der Editor öffnet Dateien in Reitern, mit Syntaxhervorhebung und den vertrauten Tasten des Editors micro (siehe Kapitel 10).

- Abschnitte: Markieren Sie Zeilen und fügen Sie einen Abschnitt hinzu, um einem Teil einer langen Datei einen Namen zu geben; die Abschnittsliste springt direkt dorthin.
- Der Fokusmodus dunkelt alle Zeilen außerhalb des gewählten Abschnitts ab, sodass der Teil, an dem Sie arbeiten, hervortritt. Esc hebt den Fokus auf.
- Esc verlässt eine Datei. Gibt es ungespeicherte Änderungen, fragt Petacore zuerst: Ein weiteres Esc verlässt die Datei ohne Speichern, Strg+S speichert und verlässt sie. Die Rückfrage lässt sich in den Einstellungen abschalten.
- Hat das Projekt einen Ordner der installierten Anwendung, bietet Petacore beim Speichern an, auch das installierte Programm zu aktualisieren.

4.6 Schnappschüsse und automatisches Speichern

Ein Schnappschuss ist ein vollständiger Wiederherstellungspunkt des Projekts. Schnappschuss erstellen legt jederzeit einen an, auf Wunsch mit Bezeichnung. Das automatische Speichern legt in festen Abständen (standardmäßig 10 Minuten) selbst einen an; die 40 neuesten automatischen Schnappschüsse bleiben erhalten.

Vor dem Wiederherstellen wird ein Sicherheits-Schnappschuss des aktuellen Stands angelegt — auch eine Wiederherstellung lässt sich also rückgängig machen. Schnappschüsse liegen außerhalb des Projekts in Ihrem eigenen Datenordner; .git, der Ordner versions und Build-Reste gehören nicht dazu.

4.7 Terminal

Die Seite Terminal öffnet Ihre gewohnte Shell direkt im Projektordner.

4.8 Sandbox

Die Sandbox führt Befehle in einer abgeschotteten Kopie Ihres Systems aus. Das echte System ist darin schreibgeschützt: Installieren, Löschen und Bearbeiten betreffen nur diese Wegwerf-Sitzung, und Simulation beenden entfernt alles spurlos.

- Aktuelles Projekt testen startet die Sandbox mit einer Kopie des Projekts.

- .deb temporär installieren installiert ein Paket nur innerhalb der Sandbox.
- App starten startet grafische oder Terminal-Programme darin.
- Die Sandbox hat standardmäßig kein Internet. Versucht etwas darin, online zu gehen, meldet Petacore das; Sie können es erlauben, gesperrt lassen oder nicht mehr gefragt werden.

Vollständige Isolation erfordert bubblewrap. Fehlt es, warnt die Seite „NICHT isoliert“ und zeigt den Installationsbefehl.

4.9 Live-Server

Bei Webprojekten zeigt der Live-Server die Website ohne Build-Schritt direkt in Petacore und lädt sie von selbst neu, sobald sich eine Datei ändert. F11 füllt das Fenster mit der Website; Esc kehrt zurück.

Standardmäßig antwortet der Server nur diesem Computer. Im lokalen Netzwerk freigeben macht ihn von einem Telefon oder einem anderen Gerät im selben Netz erreichbar und zeigt die dort zu öffnende Adresse. Ohne WebKitGTK öffnet sich die Vorschau stattdessen in Ihrem gewohnten Browser.

4.10 Nächste Updates

Nächste Updates verwaltet die Pläne für das Projekt: Titel, Details, Priorität (Hoch, Normal, Irgendwann) und angehängte Dokumente. Pläne lassen sich bearbeiten, als erledigt markieren und kopieren. Sie werden im Projekt gespeichert, wandern also mit ihm, und können mit Google Drive synchronisiert werden.

5. Pakete und Veröffentlichungen

5.1 Paket

Die Seite Paket erzeugt aus dem Projekt ein Installationspaket: .deb für Debian, Ubuntu und Mint, .rpm für Fedora, openSUSE und RHEL. Das Paket installiert das Programm auf dem Zielsystem nach /opt/<Name> und trägt es ins Anwendungsmenü ein.

Feld	Bedeutung
Name	Paketname in Kleinbuchstaben
Version	Zum Beispiel 1.6 oder 2.0.1
Betreuer	Name und E-Mail-Adresse der verantwortlichen Person
Beschreibung	Wird von Softwarezentren und Paketverwaltungen angezeigt
Lizenz	Aus der Liste gewählt oder eingetippt; Lizenzen vergleichen erklärt sie
Signieren	Signiert das Paket mit dem aktiven Schlüssel (standardmäßig an)

Dateien mit Zugangsdaten — .env, private Schlüssel, .netrc, .npmrc und Ähnliches — kommen niemals ins Paket; Petacore teilt mit, wie viele es zurückgehalten hat.

5.2 Signaturschlüssel

Ein GPG-Schlüssel ist Ihre digitale Unterschrift: Er belegt, dass ein Paket wirklich von Ihnen stammt und von niemandem verändert wurde. Die Seite Schlüssel listet Ihre Schlüssel auf und zeigt, welcher signiert.

Einstellung	Möglichkeiten
Name, E-Mail, Kommentar	Wem der Schlüssel gehört
Schlüsseltyp	Ed25519 (modern, schnell, empfohlen) oder RSA 4096 / 3072 / 2048 (für ältere Systeme)
Ablauf	Nie, 1 Jahr, 2 Jahre oder 5 Jahre
Passphrase	Optional. Ist eine gesetzt, fragt GPG bei jeder Signatur danach — sicherer

- Zum Signieren verwenden macht einen Schlüssel zum aktiven Signaturschlüssel.
- Öffentlichen Schlüssel exportieren speichert die Datei, mit der Ihre Nutzer Pakete prüfen.
- Die öffentlichen Hälften Ihrer Schlüssel lassen sich in Google Drive sichern. Private Schlüssel verlassen diesen Computer nie.
- Löschen ist endgültig: Mit diesem Schlüssel kann dann nicht mehr signiert werden.

5.3 Versionen

Versionen bewahrt jede veröffentlichte Version des Projekts samt ihren Paketen auf, die neueste zuerst. Die Seite hat zwei Reiter: Neue Version und Versionsbibliothek.

- Neue Version erzeugt die Version aus dem aktuellen Stand der Projektdateien: Versionsnummer, Veröffentlichungsart (Stabil, Release Candidate, Beta, Alpha), Vorabversionsnummer, Formate (.deb, .rpm), Signatur und Versionshinweise. Die Vorschau zeigt den Namen, den Menschen lesen („1.6 Beta 2“), und die Version, die Paketverwaltungen vergleichen („1.6~beta2“).
- Die Versionsbibliothek gruppiert Versionen nach Reihen (1.6, 2.0 ...) und hebt die neueste hervor. Jede Version zeigt ihre Pakete, ihre Signatur und ihre Hinweise.
- Code ansehen öffnet den Code jeder Version — direkt aus dem Paket dieser Version gelesen, genau so wie ausgeliefert, nur lesend. Dateien lassen sich filtern und ihr Inhalt kopieren.
- Ältere .deb- oder .rpm-Dateien können auf die Seite gezogen werden, um eine Version in den Verlauf aufzunehmen; die Originale bleiben, wo sie sind.
- Hinweise bearbeiten, .deb zur Paketquelle hinzufügen und Version entfernen finden sich bei jeder Version. Eine entfernte Version landet im Versions-Papierkorb und lässt sich zurückholen.

Die Versionen liegen im Ordner versions des Projekts. Dieser Ordner wird aus jedem Paket, jedem Schnappschuss, jeder Statistik und der git-Geschichte herausgehalten — kein Paket enthält je seine Vorgänger.

5.4 APT-Paketquelle

Eine Paketquelle macht aus einzelnen Paketen eine Softwarequelle, die Nutzer einmal hinzufügen und aus der sie dann wie gewohnt Aktualisierungen erhalten. Petacore erzeugt den Verzeichnisbaum, signiert den Index und hält ihn mit Ihren Paketen im Einklang.

Einstellung	Bedeutung
Name, Ordner der Paketquelle	Bezeichnung und Ort, an dem der Baum erzeugt wird
Öffentliche Adresse	Die URL, von der Nutzer installieren, z. B. https://repo.example.com
Suite, Komponente, Architekturen	Übliche Werte: stable, main, amd64, arm64
Origin, Label, Beschreibung	Wie sich die Paketquelle gegenüber apt vorstellt

Einstellung	Bedeutung
Signaturschlüssel	apt verweigert ein Archiv mit unsigniertem Index
Veröffentlichen über	Nicht von hier, in einen Ordner kopieren oder per SSH (rsync) senden

- Pakete hinzufügen oder .deb-Dateien auf die Seite ziehen. Mit „Erzeugte Pakete zur Paketquelle hinzufügen“ kommt jedes neue .deb automatisch hinein.
- Neu erzeugen & signieren erneuert den Index; ein Hinweis erscheint, sobald er nicht mehr zu den Paketen passt.
- Veröffentlichen gleicht die Kopie auf dem Server an den Ordner an; Server prüfen vergleicht beide.
- Abzeichen und Filter zeigen Pakete, die nicht im Index, nicht signiert, nicht veröffentlicht oder überholt sind.
- Jedes Paket bietet Signatur prüfen, Installationsbefehl kopieren, Umbenennen, Enthaltenden Ordner öffnen und Entfernen.
- Entfernte Pakete landen im Papierkorb der Paketquelle und lassen sich zurücklegen; Alte Versionen entfernen behält nur die neueste jedes Pakets.
- Anleitung für Nutzer kopieren und Archivschlüssel exportieren geben Ihren Nutzern alles, was sie zum Hinzufügen der Paketquelle brauchen.

5.5 Paketquelle auf einem Server

Im Reiter Server verwaltet Petacore den Ordner der Paketquelle direkt auf Ihrem Server über SSH. Dateien werden mit scp hochgeladen; der Index wird auf dem Server erzeugt und auf Ihrem Computer signiert — der Signaturschlüssel verlässt ihn also nie.

1. Geben Sie Servername oder IP, Port, Benutzer und den Ordner der Paketquelle auf dem Server ein, z. B. `/var/www/repo`.
2. Wählen Sie einen SSH-Schlüssel oder erstellen Sie einen für Petacore: Ed25519, geschützt durch eine Passphrase mit mindestens 8 Zeichen, die nie gespeichert wird.
3. Fügen Sie die von Petacore angezeigte Zeile in `~/.ssh/authorized_keys` dieses Benutzers auf dem Server ein.
4. Verbinden. Vergleichen Sie beim ersten Mal den Fingerabdruck des Servers mit dem, den der Server selbst meldet, und bestätigen Sie nur bei Übereinstimmung.

Nach dem Verbinden können Sie hochladen (.deb-Dateien kommen automatisch in den Pool), Ordner anlegen, umbenennen, herunterladen, Einträge in den Server-Papierkorb verschieben und zurückholen sowie Neu erzeugen & signieren.

Schutz	Bedeutung
Prüfung des Host-Schlüssels	Die Identität des Servers wird bei jeder Verbindung geprüft; ändert sie sich, verweigert Petacore die Verbindung
Nur Schlüssel-Anmeldung	Passwörter werden nie gesendet
Moderne Kryptografie	Post-Quanten-Schlüsselaustausch, wo verfügbar; authentifizierte Verschlüsselung
Zwei-Faktor-Anmeldung	Codes aus Authenticator-Apps und FIDO2-Sicherheitsschlüssel
Nichts gespeichert	Passphrasen und Zwei-Faktor-Codes werden nie gesichert

6. Online-Dienste

6.1 GitHub verbinden

1. Klicken Sie auf Mit GitHub verbinden — in den Einstellungen, im Assistenten beim ersten Start oder wann immer GitHub benötigt wird.
2. Ihr Browser öffnet GitHub; der Einmal-Code liegt bereits in der Zwischenablage.
3. Fügen Sie den Code ein, bestätigen Sie und klicken Sie auf „Authorize“. Petacore verbindet sich von selbst.

Ist die GitHub CLI (gh) auf dem Computer bereits angemeldet, genügt ein Klick. Fehlt gh, bietet Petacore an, es zu installieren. Unter Weitere Anmeldeoptionen lässt sich auch ein persönliches Zugriffstoken von Hand einfügen. Die Anmeldung liegt im Schlüsselbund des Desktops, niemals in einer Datei oder einer Adresse.

6.2 Auf GitHub hochladen

Der GitHub-Knopf oben im Fenster öffnet Auf GitHub hochladen. In einem Schritt gelangt der Code in den Zweig main und wird zu einer neuen Version auf der GitHub-Seite des Projekts.

- Wählen Sie eine Version aus den Vorschlägen — etwa 1.0 Beta 3, 1.0 Beta 2.1, 1.0 RC 1 oder 1.0 — oder geben Sie eine eigene ein. Eine Version, die schon in der Bibliothek, aber noch nicht auf GitHub ist, wird zuerst vorgeschlagen.
- Das Tag wird aus der Version gebildet (v1.0-beta.3); Beta-, Alpha- und Release-Candidate-Versionen werden als Vorabversion markiert.
- Hat das Projekt noch kein GitHub-Repository, wird eines angelegt — standardmäßig privat.
- Der Zweig ist main, sofern Sie ihn nicht ändern. Versionshinweise sind optional; bleiben sie leer, schreibt GitHub sie aus den Commits.
- Die .deb- und .rpm-Pakete dieser Version aus der Versionsbibliothek werden an die Version angehängt.
- Schalten Sie „Als neue Version veröffentlichen“ aus, um nur den Code hochzuladen.

Dateien mit Zugangsdaten werden nie hochgeladen. Was bereits auf GitHub liegt, bleibt erhalten: Kollidiert es mit Ihrem Stand, bricht das Hochladen ab, und nichts wird überschrieben. Eine bereits veröffentlichte Version wird nie ersetzt.

6.3 Von GitHub klonen

Wird ein Projekt aus einer GitHub-Adresse angelegt, fragt Petacore einmal, ob VirusTotal den heruntergeladenen Code prüfen soll. Prüfen öffnet die Seite Virenprüfung und startet die Prüfung sofort.

6.4 Google Drive

Verbinden Sie Ihr eigenes Google Drive in den Einstellungen: Der Browser öffnet sich, Sie melden sich bei Google an und klicken auf „Zulassen“. Petacore legt einen Ordner „Petacomm Petacore™“ an und führt jedes Projekt in einem eigenen Unterordner, Mediendateien eingeschlossen. Petacores eigene Aufzeichnungen liegen daneben in einem verborgenen Ordner .petacore.

- Der Drive-Knopf sendet das aktive Projekt. Drive wird an diesen Computer angeglichen — und nichts geht verloren (siehe 6.5).
- Hat ein anderer Computer seit der letzten Synchronisierung dieses Computers eine neuere Kopie gesendet, bietet Petacore an, sie zuerst zu übernehmen.
- Wäre ein Upload kleiner als der Inhalt in Drive, warnt Petacore vorher (in den Einstellungen abschaltbar).

- Projekte von Drive holen lädt Projekte, die nur in Drive liegen, nach ~/Projects herunter.
- Verschlüsselte Projekte werden über eine verschlüsselte Schicht gesendet; Drive sieht nur unlesbare Daten.

6.5 Sicherungen

Petacore löscht beim Synchronisieren mit Drive keine einzige Datei. Jede Datei, die eine Synchronisierung ersetzt oder entfernt, wird zuerst gesichert — in Drive beim Senden, auf diesem Computer beim Übernehmen einer Kopie.

- Sicherungen werden nach Tagen aufgelistet: wann, von welchem Computer, wie viele Dateien und wie groß.
- Öffnen Sie eine Sicherung, um ihre Dateien zu sehen. Zurückholen stellt die ganze Sicherung oder eine einzelne Datei wieder her.
- Bevor etwas zurückgeholt wird, werden die aktuellen Dateien selbst gesichert — auch das lässt sich also rückgängig machen.
- Sicherungen bleiben standardmäßig 30 Tage erhalten (in den Einstellungen 7, 30, 90, 365 Tage oder unbegrenzt) und werden dann automatisch entfernt.

Die Sicherungen öffnen Sie über den Uhr-Knopf auf der Seite Projekt, über den Knopf Sicherungen in Synchronisierungsmeldungen und über Einstellungen → Google Drive → Sicherungen anzeigen.

6.6 Änderungen von Ihren anderen Computern

Computer mit demselben Google Drive übernehmen gegenseitig die neueste Kopie von selbst: wenige Sekunden nach dem Start und danach alle 10 Minuten (5, 10, 30 oder 60 in den Einstellungen). Projekte, die nur in Drive liegen, werden ebenfalls heruntergeladen — in einen neuen Ordner, falls der Name schon vergeben ist.

Situation	Was Petacore tut
Neuere Kopie in Drive, hier keine Änderungen	Übernimmt sie still und meldet, welcher Computer sie gesendet hat
Neuere Kopie in Drive und Änderungen hier	Fragt: Drive-Kopie übernehmen, Meine senden oder Nicht jetzt
Verschlüsseltes Projekt	Wird nicht automatisch übernommen — es braucht das Passwort

Egal, wofür Sie sich entscheiden: Die Dateien der anderen Seite bleiben in den Sicherungen erhalten. Die Funktion ist standardmäßig eingeschaltet und lässt sich unter Einstellungen → Google Drive abschalten.

6.7 Virenprüfung

Die Virenprüfung lässt das Projekt von den Engines von VirusTotal prüfen, bevor Ihre Nutzer es installieren. Der API-Schlüssel gehört Ihnen: Legen Sie ein kostenloses Konto auf virustotal.com an, kopieren Sie den Schlüssel aus Ihrem Profil und geben Sie ihn einmal ein. Er wird im Schlüsselbund des Desktops abgelegt.

- Projekt prüfen packt das Projekt in ein einziges Archiv; ein unverändertes Projekt wird an seinem Fingerabdruck erkannt und nie zweimal gesendet.
- Nichts wird gesendet, bevor Sie zustimmen — einmal pro Projekt. Dateien mit Zugangsdaten, .git, node_modules und der Ordner versions werden nie gesendet.
- Ergebnisse: Nichts gemeldet, „n von t haben es gemeldet“ mit den Namen der Engines, oder „nie gesehen“ — die übliche Antwort für etwas gerade Erzeugtes.
- Melden ein, zwei Engines ein frisch erzeugtes, unsigned Programm, ist das meist ein Fehlalarm.
- Ein kostenloser Schlüssel erlaubt 4 Prüfungen pro Minute und 500 pro Tag.

7. Einstellungen

Einstellung	Standard	Beschreibung
Sprache	Desktop-Sprache	Sechs Sprachen; Petacore startet zur Übernahme neu
Darstellung	System folgen	Hell, dunkel oder wie der Desktop
Fokusmodus	An	Dunkelt Zeilen außerhalb des gewählten Abschnitts ab
Warnung bei kleinerem Upload	An	Warnt, wenn ein Upload kleiner als die Kopie in Drive ist
Internet in der Sandbox	Aus	Erlaubt Code in der Sandbox den Netzzugang
Vor Verlassen mit Esc fragen	An	Rückfrage bei ungespeicherten Dateien
Automatisches Speichern	An, 10 Minuten	Automatische Schnappschüsse
GitHub	Nicht verbunden	Verbinden, trennen, optionale OAuth-Client-ID
Google Drive	Nicht verbunden	Verbinden, trennen, Projekte holen
Änderungen übernehmen	An, 10 Minuten	Änderungen von Ihren anderen Computern
Sicherungen aufbewahren	30 Tage	7, 30, 90, 365 Tage oder unbegrenzt
Oberfläche	Erkannt	GNOME oder KDE Plasma

8. Sicherheit und Datenschutz

- Anmeldungen und API-Schlüssel liegen im Schlüsselbund des Desktops. Gibt es keinen, kommen sie in eine Datei, die nur Sie lesen können.
- Dateien mit Zugangsdaten (.env, private Schlüssel, .netrc ...) gelangen nie in ein Paket, einen GitHub-Upload oder eine VirusTotal-Prüfung.
- Die GitHub-Anmeldung wird nur an github.com übergeben; eine täuschend ähnliche Adresse erhält nichts.
- Private Signaturschlüssel verlassen Ihren Computer nie; gesichert werden nur öffentliche Schlüssel.
- Server-Anmeldungen erfolgen nur per Schlüssel; Passphrasen und Zwei-Faktor-Codes werden nie gespeichert.
- Die Sandbox hat kein Netzwerk, solange Sie es nicht öffnen; der Live-Server antwortet nur diesem Computer, solange Sie ihn nicht freigeben.
- Ohne Ihre Zustimmung wird nichts an VirusTotal gesendet.
- Synchronisieren zerstört nie eine Datei: Was es ersetzt, bleibt in den Sicherungen erhalten.

9. Fehlerbehebung

Problem	Lösung
Google-Drive-Unterstützung braucht rclone	<code>sudo apt install rclone</code>
Der Schlüsselbund ist nicht erreichbar	<code>sudo apt install libsecret-tools</code> , dann erneut versuchen
GitHub CLI ist nicht bereit	gh installieren anklicken oder <code>sudo apt install gh</code>
Sandbox: NICHT isoliert	<code>sudo apt install bubblewrap</code>
Teilen-Karte nur als SVG gespeichert	<code>sudo apt install librsvg2-bin</code>
Live-Server: kein eingebauter Browser	<code>sudo apt install gir1.2-webkit-6.0</code> oder Im Browser öffnen
Index der Paketquelle lässt sich nicht erzeugen	<code>sudo apt install apt-utils dpkg-dev</code>
Serververbindung: OpenSSH fehlt	<code>sudo apt install openssh-client</code>
VirusTotal hat den Schlüssel abgelehnt	Schlüssel im VirusTotal-Profil prüfen; eine Netzsperrung kann auch vorübergehend sein — später erneut versuchen
Signaturschlüssel abgelaufen oder nicht vorhanden	Auf der Seite Schlüssel einen neuen Schlüssel erstellen und Zum Signieren verwenden wählen
Die Identität des Servers hat sich geändert	Nicht verbinden. Den Server-Administrator fragen; den alten Schlüssel erst nach bestätigter Änderung vergessen
GitHub: Zweig enthält unpassende Änderungen	Diese Änderungen im Terminal mit git holen, Konflikte lösen, dann erneut hochladen
Version bereits auf GitHub veröffentlicht	Eine andere Version wählen
Projekt an zwei Orten geändert	Wählen, welche Kopie bleiben soll; die andere Seite bleibt in den Sicherungen
Falsches Passwort für ein verschlüsseltes Projekt	Das Passwort ist nicht wiederherstellbar; das richtige verwenden
Komponenten der anderen Oberfläche fehlen	Jetzt installieren anklicken oder den angezeigten Befehl ausführen

10. Tastenkürzel

Tasten	Aktion
Strg+S	Datei speichern (Editor)
Strg+Q	Datei schließen (Editor)
Strg+F	Suchen; Eingabe springt zum nächsten Treffer, Esc schließt
Strg+K / Strg+D	Zeile ausschneiden / Zeile verdoppeln

Tasten	Aktion
Strg+Z / Strg+Y	Rückgängig / Wiederholen
Esc	Datei verlassen; ein weiteres Esc verlässt sie ohne Speichern
Strg+A / Entf	Alles auswählen / löschen (Seite Projekt, umkehrbar)
Strg+Z	Gelöschte Dateien oder entfernte Pakete zurückholen
Strg + Mauszeiger	Schnellvorschau; Strg+Mausrad blättert darin
F11 / Esc	Live-Server im Vollbild / Vollbild verlassen

11. Dateien und Deinstallation

11.1 Wo Petacore seine Daten ablegt

Ort	Inhalt
~/local/share/petacore-app	Das Programm selbst
~/config/petacore	Einstellungen (nur für Sie lesbar)
~/local/share/petacore/snapshots	Schnappschüsse
~/local/share/petacore/drive-backups	Sicherungen der auf diesem Computer ersetzten Dateien
<Projekt>/versions	Veröffentlichungsverlauf (aus Paketen und git herausgehalten)
<Projekt>/petacore	Pläne aus Nächste Updates
Schlüsselbund des Desktops	GitHub-Anmeldung, VirusTotal-Schlüssel

11.2 Deinstallation

Führen Sie im Petacore-Ordner `./uninstall.sh` als Sie selbst aus. Das Entfernen des Programms und das Löschen Ihrer Schnappschüsse und Sicherungen werden als zwei getrennte Fragen gestellt, und nichts wird gelöscht, bevor Sie bestätigen. Ihre Projektordner werden nie angetastet.

12. Lizenz und Urheberrecht

© Petacomm I.S. 2026. Alle Rechte vorbehalten. Petacomm Petacore™ ist eine Marke von Petacomm I.S.

Petacore verwendet freie und quelloffene Komponenten; sie bleiben Eigentum ihrer Urheber und werden unter deren eigenen Lizenzen genutzt. GitHub, Google Drive, VirusTotal, GNOME, KDE Plasma, Ubuntu und Debian sind Marken ihrer jeweiligen Inhaber.

Diese Anleitung beschreibt Petacore 1.1 Beta 2. Funktionen können sich in späteren Versionen ändern.

1. Avant-propos

Nous vous remercions d'avoir choisi Petacore, l'atelier de bureau pour développer et distribuer des logiciels Linux. Petacore rassemble dans une seule fenêtre tout ce dont un projet a besoin : ses fichiers et son code, son historique, ses paquets et ses clés de signature, son dépôt de logiciels, sa copie sur GitHub et sur Google Drive, ainsi qu'une analyse antivirus avant que quoi que ce soit n'atteigne vos utilisateurs.

Avant de commencer, veuillez lire attentivement ce manuel. Il décrit chaque partie de Petacore, les réglages proposés et ce que Petacore fait — et ne fait jamais — de vos fichiers et de vos identifiants. Les connaître vous permettra de travailler plus vite, en toute sérénité.

Développeur : Petacomm I.S.

Produit : Petacomm Petacore™

Version décrite : 1.1 Beta 2

Page du projet : github.com/petacomm/petacore

2. Description technique

2.1 Généralités

Petacore est une application de bureau native pour Linux. Elle fonctionne entièrement sur votre propre ordinateur et travaille sur des dossiers de projet ordinaires : rien n'a besoin d'être converti, et chaque fichier reste lisible par n'importe quel autre outil. Les services en ligne (GitHub, Google Drive, VirusTotal) ne sont utilisés que si vous les connectez, avec vos propres comptes.

Petacore existe en deux interfaces offrant les mêmes fonctions : l'une pour le bureau GNOME, l'autre pour KDE Plasma. Le programme d'installation choisit automatiquement la bonne, et vous pouvez en changer à tout moment.

2.2 Configuration requise

Élément	Exigence
Système d'exploitation	Ubuntu 26.04 LTS ou une autre distribution basée sur Debian ou Ubuntu
Bureau	GNOME (GTK 4, libadwaita) ou KDE Plasma (Qt 6, PySide6)
Python	3.12 ou plus récent
Espace disque	Environ 50 Mo pour Petacore, plus la place des instantanés et des sauvegardes
Internet	Uniquement pour GitHub, Google Drive, VirusTotal et les connexions aux serveurs

2.3 Caractéristiques techniques

Fonction	Caractéristique
Langues de l'interface	English, Türkçe, Deutsch, Français, 中文, Українська
Formats de paquet	.deb (Debian, Ubuntu, Mint) et .rpm (Fedora, openSUSE, RHEL)
Signature des paquets	GnuPG — Ed25519 (recommandé) ou RSA 2048 / 3072 / 4096

Fonction	Caractéristique
Gestion de versions	git; publication sur GitHub avec des versions (releases)
Copie dans le nuage	Google Drive via rclone, chiffrée de bout en bout si vous le souhaitez
Analyse de sécurité	API VirusTotal v3, avec votre propre clé
Dépôt de logiciels	Archive APT signée, en local ou sur votre propre serveur via SSH
Connexion au serveur	OpenSSH, connexion par clé uniquement, double authentification prise en charge
Isolation	Bac à sable bubblewrap, réseau coupé par défaut

2.4 Composants utilisés

Petacore s'appuie sur des outils standard et éprouvés. Le programme d'installation les met en place; s'il en manque un, Petacore le signale clairement, avec la commande qui l'installe.

Composant	Utilisé pour
git, gh	Historique des versions; connexion à GitHub et publication
rclone	Connexion à Google Drive, synchronisation et sauvegardes
gnupg	Création des clés; signature des paquets et des dépôts
dpkg-dev, rpm, apt-utils	Construction des paquets et des index de dépôt
bubblewrap	Le bac à sable isolé
openssh-client, rsync	Connexion au serveur et publication
libsecret-tools	Conserver connexions et clés d'API dans le trousseau du bureau
libsvg2-bin	Enregistrer les cartes de partage en PNG
WebKitGTK (facultatif)	Afficher l'aperçu du serveur en direct dans la fenêtre

3. Installation et premier démarrage

3.1 Installer Petacore

1. Décompressez l'archive de Petacore dans le dossier de votre choix.
2. Ouvrez un terminal dans ce dossier et lancez `./install.sh` — en tant que vous-même, jamais avec `sudo`. Petacore s'installe dans votre dossier personnel.
3. Le programme d'installation reconnaît votre bureau (GNOME ou KDE Plasma). S'il n'y parvient pas, il vous demande quelle interface vous souhaitez.
4. Lorsque des paquets système sont nécessaires, il demande une seule fois votre mot de passe et les installe avec `apt`.
5. Petacore apparaît ensuite dans le menu des applications; la commande `petacore` le lance depuis un terminal.

Petacore peut aussi être installé à partir d'un paquet `.deb`. Le résultat est le même; le script d'installation ajoute seulement le choix automatique de l'interface.

3.2 Premier démarrage

Au premier démarrage, un court assistant demande la langue (celle de votre bureau est proposée), propose ensuite de connecter GitHub et Google Drive, puis de créer votre premier projet. Chaque étape peut être passée et faite plus tard depuis les Paramètres.

3.3 La fenêtre principale

À gauche, la barre latérale liste les pages du projet actif; le sélecteur de projet en haut permet de passer d'un projet à l'autre, et le bouton crayon modifie le projet actif. En haut de la fenêtre se trouvent le bouton GitHub (Publier sur GitHub), le bouton Google Drive (envoyer vers Drive) et le menu principal (Nouveau projet, Paramètres, À propos de Petacore, Quitter).

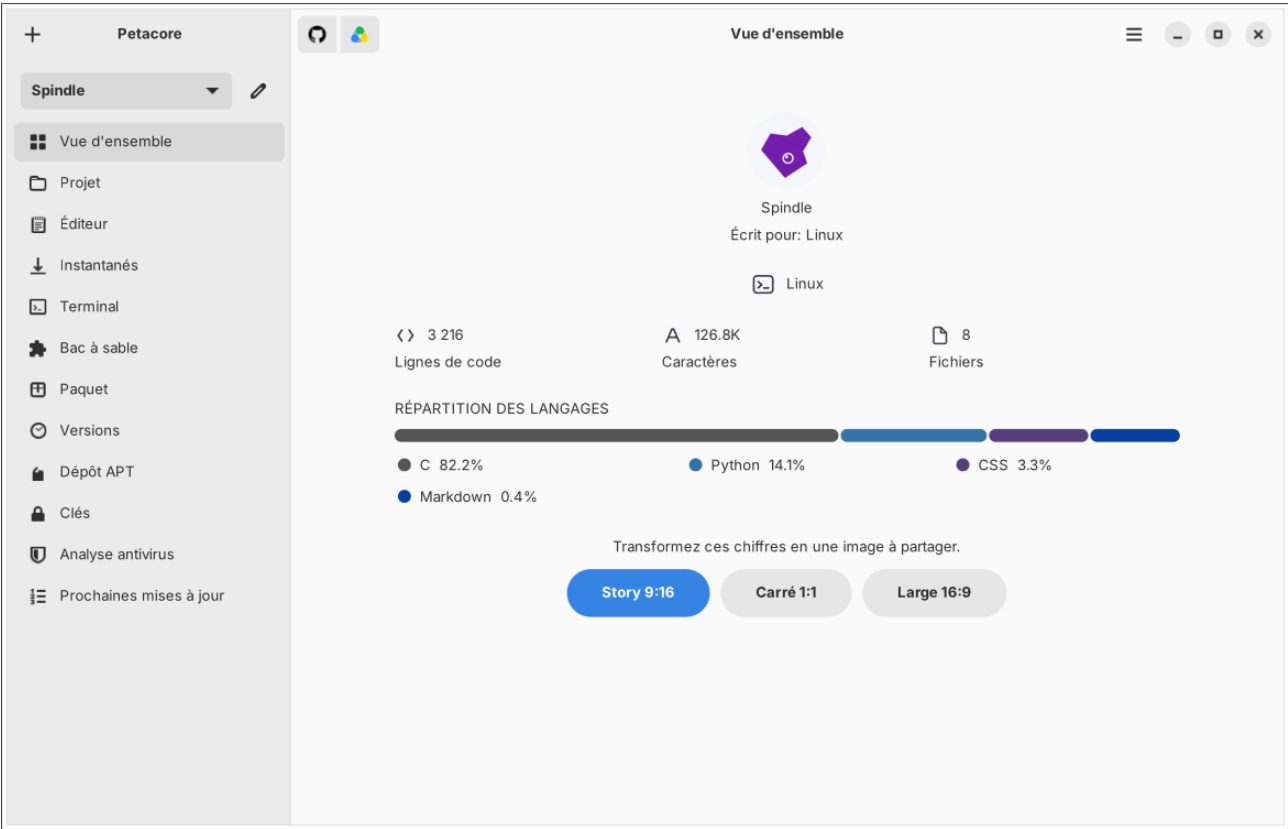


Figure 1 — La fenêtre principale avec la page Vue d'ensemble.

Les pages dépendent du type de projet. Un projet destiné à un système d'exploitation reçoit Paquet, Clés et Bac à sable; un projet web reçoit à la place le Serveur en direct, car il n'est ni emballé ni signé. Petacore n'en décide qu'en cas de correspondance nette: un projet inconnu garde toujours l'ensemble complet.

Page	Rôle
Vue d'ensemble	Chiffres du projet et affiche à partager
Projet	Gestionnaire de fichiers du dossier du projet
Éditeur	Éditeur de code à onglets
Instantanés	Points de restauration de tout le projet
Terminal	Un shell ouvert dans le dossier du projet
Bac à sable / Serveur en direct	Essayer le programme sans risque / prévisualiser un site web

Page	Rôle
Paquet	Construire des paquets .deb et .rpm signés
Versions	Historique des versions et code de chacune
Dépôt APT	Dépôt APT signé, en local ou sur un serveur
Clés	Clés de signature GPG
Analyse antivirus	Analyser le projet avec VirusTotal
Prochaines mises à jour	Projets d'évolution

3.4 Choisir l'interface

Paramètres → Interface indique le style utilisé et bascule vers l'autre. Petacore se ferme puis se rouvre dans le nouveau style; projets et réglages restent intacts. S'il manque des composants de l'autre style, Petacore propose de les installer.

4. Projets

4.1 Créer un projet

Choisissez Nouveau projet dans le menu principal ou le bouton +. Saisissez un nom et le dossier où le projet doit se trouver.

- Avec une adresse GitHub, Petacore télécharge (clone) ce dépôt. Ce code venant d'Internet, Petacore vous demande ensuite si VirusTotal doit l'analyser avant que vous n'y exécutiez quoi que ce soit.
- Sans adresse, un nouveau dépôt git local est créé.
- Sous Chiffrement, vous pouvez faire chiffrer la copie sur Google Drive avec un code PIN à 4 chiffres, un code PIN à 8 chiffres ou une phrase secrète de 4096 caractères au plus. Le projet sur votre ordinateur reste du code ordinaire et lisible; seule la copie sur Drive est chiffrée — noms de fichiers et de dossiers compris.

Le mot de passe de chiffrement n'est jamais enregistré. S'il est perdu, personne — Petacomm compris — ne peut ouvrir la copie chiffrée sur Drive.

4.2 Modifier et retirer un projet

Le bouton crayon à côté du sélecteur de projet ouvre Modifier le projet: nom, dossier du projet, adresse GitHub et dossier de l'application installée. Si ce dernier est défini, chaque fichier enregistré dans l'Éditeur peut être copié dans le programme installé, au même chemin relatif — sans passer par le terminal.

Retirer de Petacore enlève seulement le projet de la liste. Aucun fichier n'est supprimé du disque.

4.3 Vue d'ensemble

La Vue d'ensemble montre le projet d'un coup d'œil: lignes de code, caractères, fichiers, langages et taille, le système visé (Linux, Windows, macOS, Web ou multiplateforme) et la répartition des langages sous forme de barre colorée.

La Carte de partage transforme ces chiffres en affiche sous trois formats — Story 9:16, Carré 1:1 et Large 16:9. La carte n'affiche que le nom, les chiffres et la répartition des langages, jamais de noms de fichiers ni de code; le nom affiché peut être changé si celui du projet ne doit pas être partagé.

4.4 La page Projet

La page Projet est un gestionnaire de fichiers complet pour le dossier du projet. La ligne sous la barre d'outils indique la branche git, l'état du projet par rapport à GitHub et le nombre de programmes lancés depuis ce dossier.

- Un double-clic ouvre un fichier dans l'Éditeur; images, sons et vidéos s'ouvrent dans leur propre visionneuse.
- Le clic droit propose Ouvrir, Renommer, Nouveau dossier, Copier le chemin et Supprimer.
- La suppression est réversible: Ctrl+Z, ou le bouton Annuler, ramène les fichiers.
- Glissez des fichiers depuis votre gestionnaire de fichiers. En cas de noms identiques, choisissez Remplacer / Fusionner, Garder les deux ou Ignorer.
- Le champ de recherche cherche dans le dossier ouvert; le bouton voisin cherche dans tout le projet. Le filtre limite la liste au code, aux images, aux sons et vidéos, aux documents ou aux dossiers.
- Maintenez Ctrl au-dessus d'un fichier pour un aperçu rapide; Ctrl+molette fait défiler les longs aperçus.
- Fichiers multimédias réunit en un seul endroit toutes les images, tous les sons et toutes les vidéos du projet.
- Le bouton horloge ouvre les Sauvegardes (voir 6.5).

4.5 Éditeur

L'Éditeur ouvre les fichiers dans des onglets, avec coloration syntaxique et les raccourcis familiers de l'éditeur micro (voir chapitre 10).

- Sections: sélectionnez des lignes et ajoutez une section pour nommer une partie d'un long fichier; la liste des sections y mène directement.
- Le mode focus atténue toutes les lignes hors de la section choisie, pour faire ressortir la partie sur laquelle vous travaillez. Échap annule le focus.
- Échap quitte un fichier. S'il contient des modifications non enregistrées, Petacore demande d'abord: un second Échap quitte sans enregistrer, Ctrl+S enregistre et quitte. Cette question peut être désactivée dans les Paramètres.
- Si le projet a un dossier d'application installée, l'enregistrement propose de mettre aussi à jour le programme installé.

4.6 Instantanés et sauvegarde automatique

Un instantané est un point de restauration complet du projet. Créer un instantané en fait un à tout moment, avec une étiquette facultative. La sauvegarde automatique en crée à intervalle régulier (10 minutes par défaut); les 40 instantanés automatiques les plus récents sont conservés.

Avant toute restauration, un instantané de sécurité de l'état actuel est pris: une restauration peut donc elle-même être annulée. Les instantanés sont stockés hors du projet, dans votre propre dossier de données; .git, le dossier versions et les résidus de compilation n'y figurent pas.

4.7 Terminal

La page Terminal ouvre votre shell habituel directement dans le dossier du projet.

4.8 Bac à sable

Le Bac à sable exécute des commandes dans une copie scellée de votre système. La vraie machine y est en lecture seule: installer, supprimer et modifier n'affectent que cette session jetable, et Terminer la simulation efface tout, sans laisser de trace.

- Essayer le projet actuel démarre le Bac à sable avec une copie du projet.
- Installer un .deb temporairement installe un paquet uniquement dans le Bac à sable.

- Lancer l'app démarre des programmes graphiques ou en terminal à l'intérieur.
- Le Bac à sable n'a pas accès à Internet par défaut. Si quelque chose tente d'aller en ligne, Petacore vous prévient et vous laisse autoriser, refuser ou ne plus être sollicité.

L'isolation complète nécessite bubblewrap. Sans lui, la page affiche «NON isolé» et la commande pour l'installer.

4.9 Serveur en direct

Pour les projets web, le Serveur en direct affiche le site dans Petacore sans étape de compilation et le recharge de lui-même dès qu'un fichier change. F11 étend le site à toute la fenêtre; Échap revient en arrière.

Par défaut, le serveur ne répond qu'à cet ordinateur. Partager sur le réseau local le rend accessible depuis un téléphone ou un autre appareil du même réseau et affiche l'adresse à y ouvrir. Sans WebKitGTK, l'aperçu s'ouvre dans votre navigateur habituel.

4.10 Prochaines mises à jour

Prochaines mises à jour conserve les projets d'évolution: un titre, des détails, une priorité (Haute, Normale, Un jour) et des documents joints. Ils peuvent être modifiés, marqués comme faits et copiés. Enregistrés dans le projet, ils voyagent avec lui et peuvent être synchronisés avec Google Drive.

5. Paquets et versions

5.1 Paquet

La page Paquet construit un programme d'installation à partir du projet: .deb pour Debian, Ubuntu et Mint, .rpm pour Fedora, openSUSE et RHEL. Le paquet installe le programme dans /opt/<nom> sur le système cible et l'ajoute au menu des applications.

Champ	Signification
Nom	Nom du paquet, en minuscules
Version	Par exemple 1.6 ou 2.0.1
Mainteneur	Nom et adresse e-mail de la personne responsable
Description	Affichée par les logithèques et les gestionnaires de paquets
Licence	Choisie dans la liste ou saisie; Comparer les licences les explique
Signature	Signe le paquet avec la clé active (activée par défaut)

Les fichiers contenant des identifiants — .env, clés privées, .netrc, .npmrc et semblables — ne sont jamais mis dans le paquet, et Petacore indique combien il en a écartés.

5.2 Clés de signature

Une clé GPG est votre signature numérique: elle prouve qu'un paquet vient bien de vous et que personne ne l'a modifié. La page Clés liste vos clés et indique celle qui signe.

Réglage	Choix
Nom, e-mail, commentaire	À qui appartient la clé

Réglage	Choix
Type de clé	Ed25519 (moderne, rapide, recommandé) ou RSA 4096 / 3072 / 2048 (systèmes plus anciens)
Expiration	Jamais, 1 an, 2 ans ou 5 ans
Phrase secrète	Facultative. Si elle est définie, GPG la demande à chaque signature — plus sûr

- Utiliser pour signer fait d'une clé la clé de signature active.
- Exporter la clé publique enregistre le fichier avec lequel vos utilisateurs vérifient les paquets.
- Les moitiés publiques de vos clés peuvent être sauvegardées sur Google Drive. Les clés privées ne quittent jamais cet ordinateur.
- La suppression est définitive : cette clé ne pourra plus signer de paquets.

5.3 Versions

Versions conserve chaque version publiée du projet avec ses paquets, la plus récente en premier. La page a deux onglets : Nouvelle version et Bibliothèque des versions.

- Nouvelle version construit la version à partir des fichiers du projet tels qu'ils sont : numéro de version, type de version (Stable, Version candidate, Bêta, Alpha), numéro de préversion, formats (.deb, .rpm), signature et notes de version. L'aperçu montre à la fois le nom lu par les gens (« 1.6 Beta 2 ») et la version comparée par les gestionnaires de paquets (« 1.6~beta2 »).
- La Bibliothèque des versions regroupe les versions par série (1.6, 2.0...) et signale la plus récente. Chacune montre ses paquets, sa signature et ses notes.
- Voir le code ouvre le code de n'importe quelle version — lu directement dans son propre paquet, exactement tel qu'il a été livré, en lecture seule. Les fichiers peuvent être filtrés et leur contenu copié.
- D'anciens fichiers .deb ou .rpm peuvent être déposés sur la page pour ajouter une version à l'historique ; les originaux restent à leur place.
- Modifier les notes, Ajouter le .deb au dépôt et Retirer la version figurent sur chaque version. Une version retirée va dans la corbeille des versions et peut être restaurée.

Les versions se trouvent dans le dossier versions du projet. Ce dossier est tenu à l'écart de chaque paquet, instantané, statistique et de l'historique git : aucun paquet ne contient jamais ceux qui l'ont précédé.

5.4 Dépôt APT

Un dépôt transforme des paquets isolés en une source de logiciels que les utilisateurs ajoutent une fois, puis dont ils reçoivent les mises à jour comme pour toute autre. Petacore construit l'arborescence, signe l'index et le maintient en accord avec vos paquets.

Réglage	Signification
Nom, Dossier du dépôt	Son nom et l'endroit où l'arborescence est construite
Adresse publique	L'URL d'installation pour les utilisateurs, par exemple https://repo.example.com
Suite, Composant, Architectures	Valeurs usuelles : stable, main, amd64, arm64
Origin, Label, Description	La façon dont le dépôt se présente à apt
Clé de signature	apt refuse une archive dont l'index n'est pas signé

Réglage	Signification
Publier par	Pas d'ici, copie vers un dossier, ou envoi par SSH (rsync)

- Ajouter des paquets, ou déposer des fichiers .deb sur la page. Avec «Ajouter les paquets créés au dépôt», chaque nouveau .deb y entre automatiquement.
- Régénérer et signer renouvelle l'index; un avertissement apparaît dès qu'il ne correspond plus aux paquets.
- Publier aligne la copie du serveur sur le dossier; Vérifier le serveur compare les deux.
- Badges et filtres montrent les paquets non indexés, non signés, non publiés ou remplacés.
- Chaque paquet propose Vérifier la signature, Copier la commande d'installation, Renommer, Ouvrir le dossier et Retirer du dépôt.
- Les paquets retirés vont dans la corbeille du dépôt et peuvent être remis; Retirer les anciennes versions ne garde que la plus récente de chacun.
- Copier les instructions pour les utilisateurs et Exporter la clé de l'archive donnent à vos utilisateurs tout le nécessaire pour ajouter le dépôt.

5.5 Dépôt sur un serveur

Dans l'onglet Serveur, Petacore gère le dossier du dépôt directement sur votre serveur, via SSH. Les fichiers sont envoyés avec scp; l'index est construit sur le serveur et signé sur votre ordinateur, si bien que la clé de signature ne le quitte jamais.

1. Saisissez le nom ou l'IP du serveur, le port, l'utilisateur et le dossier du dépôt sur le serveur, par exemple /var/www/repo.
2. Choisissez une clé SSH, ou créez-en une pour Petacore: Ed25519, protégée par une phrase secrète d'au moins 8 caractères qui n'est jamais enregistrée.
3. Ajoutez la ligne affichée par Petacore au fichier ~/.ssh/authorized_keys de cet utilisateur sur le serveur.
4. Connectez-vous. La première fois, comparez l'empreinte du serveur avec celle que le serveur indique lui-même, et ne confirmez que si elles concordent.

Une fois connecté, vous pouvez envoyer des fichiers (les .deb vont automatiquement dans le pool), créer des dossiers, renommer, télécharger, placer des éléments dans la corbeille du serveur et les restaurer, et Régénérer et signer.

Protection	Ce que cela signifie
Vérification de la clé d'hôte	L'identité du serveur est contrôlée à chaque connexion; si elle change, Petacore refuse de se connecter
Connexion par clé uniquement	Aucun mot de passe n'est jamais envoyé
Cryptographie moderne	Échange de clés post-quantique si disponible, chiffrement authentifié
Double authentification	Codes d'applications d'authentification et clés de sécurité FIDO2
Rien n'est conservé	Phrases secrètes et codes à usage unique ne sont jamais enregistrés

6. Services en ligne

6.1 Connecter GitHub

1. Cliquez sur Se connecter avec GitHub — dans les Paramètres, dans l'assistant du premier démarrage, ou chaque fois que GitHub est nécessaire.
2. Votre navigateur s'ouvre sur GitHub et le code à usage unique est déjà dans le presse-papiers.
3. Collez le code, confirmez, puis cliquez sur «Authorize». Petacore se connecte de lui-même.

Si GitHub CLI (gh) est déjà connecté sur l'ordinateur, un clic suffit. Si gh manque, Petacore propose de l'installer. Sous Autres options de connexion, un jeton d'accès personnel peut aussi être collé à la main. La connexion est conservée dans le trousseau du bureau, jamais dans un fichier ni dans une adresse.

6.2 Publier sur GitHub

Le bouton GitHub en haut de la fenêtre ouvre Publier sur GitHub. En une seule étape, le code part sur la branche main et devient une nouvelle version sur la page GitHub du projet.

- Choisissez une version parmi les suggestions — par exemple 1.0 Beta 3, 1.0 Beta 2.1, 1.0 RC 1 ou 1.0 — ou saisissez la vôtre. Une version déjà dans la bibliothèque mais pas encore sur GitHub est proposée en premier.
- L'étiquette est formée à partir de la version (v1.0-beta.3); les bêtas, alphas et versions candidates sont marquées comme préversions.
- Si le projet n'a pas encore de dépôt GitHub, il est créé — privé par défaut.
- La branche est main, sauf si vous la changez. Les notes de version sont facultatives; laissées vides, GitHub les rédige à partir des commits.
- Les paquets .deb et .rpm de cette version, issus de la Bibliothèque des versions, sont joints à la version.
- Désactivez «Publier comme nouvelle version» pour n'envoyer que le code.

Les fichiers d'identifiants ne sont jamais envoyés. Le travail déjà présent sur GitHub est conservé: s'il entre en conflit avec le vôtre, l'envoi s'arrête et rien n'est écrasé. Une version déjà publiée n'est jamais remplacée.

6.3 Cloner depuis GitHub

Lorsqu'un projet est créé à partir d'une adresse GitHub, Petacore demande une fois si VirusTotal doit analyser le code téléchargé. Analyser ouvre la page Analyse antivirus et lance aussitôt l'analyse.

6.4 Google Drive

Connectez votre propre Google Drive dans les Paramètres: le navigateur s'ouvre, vous vous connectez à Google et cliquez sur «Autoriser». Petacore crée un dossier «Petacomm Petacore™» et range chaque projet dans son propre sous-dossier, fichiers multimédias compris. Les données propres à Petacore se trouvent à côté, dans un dossier caché .petacore.

- Le bouton Drive envoie le projet actif. Drive est aligné sur cet ordinateur — et rien n'est perdu (voir 6.5).
- Si un autre ordinateur a envoyé une copie plus récente depuis la dernière synchronisation de celui-ci, Petacore propose de la récupérer d'abord.
- Si un envoi devait être plus petit que le contenu de Drive, Petacore prévient d'abord (désactivable dans les Paramètres).
- Récupérer les projets depuis Drive télécharge dans ~/Projects les projets qui n'existent que sur Drive.
- Les projets chiffrés passent par une couche de chiffrement; Drive ne voit que des données illisibles.

6.5 Sauvegardes

Petacore ne supprime jamais de fichier en synchronisant avec Drive. Chaque fichier qu'une synchronisation remplace ou retire est d'abord conservé en sauvegarde — sur Drive lors d'un envoi, sur cet ordinateur lors de l'arrivée d'une copie.

- Les sauvegardes sont classées par jour : quand, depuis quel ordinateur, combien de fichiers et quelle taille.
- Ouvrez une sauvegarde pour voir ses fichiers. Remettre restaure toute la sauvegarde ou un seul fichier.
- Avant toute remise en place, les fichiers actuels sont eux-mêmes sauvegardés : l'opération peut donc aussi être annulée.
- Les sauvegardes sont conservées 30 jours par défaut (7, 30, 90, 365 jours ou toujours, dans les Paramètres), puis supprimées automatiquement.

Les sauvegardes s'ouvrent depuis le bouton horloge de la page Projet, depuis le bouton Sauvegardes des notifications de synchronisation, et depuis Paramètres → Google Drive → Afficher les sauvegardes.

6.6 Modifications de vos autres ordinateurs

Les ordinateurs connectés au même Google Drive récupèrent d'eux-mêmes la copie la plus récente des autres : quelques secondes après le démarrage, puis toutes les 10 minutes (5, 10, 30 ou 60 dans les Paramètres). Les projets présents uniquement sur Drive sont aussi téléchargés, dans un nouveau dossier si le nom est déjà pris.

Situation	Ce que fait Petacore
Copie plus récente sur Drive, aucune modification ici	La récupère discrètement et indique quel ordinateur l'a envoyée
Copie plus récente sur Drive et modifications ici	Demande : Prendre la copie de Drive, Envoyer la mienne ou Pas maintenant
Projet chiffré	Pas de récupération automatique — le mot de passe est nécessaire

Quel que soit votre choix, les fichiers de l'autre côté sont conservés dans les Sauvegardes. La fonction est activée par défaut et peut être désactivée dans Paramètres → Google Drive.

6.7 Analyse antivirus

L'Analyse antivirus fait vérifier le projet par les moteurs de VirusTotal avant que vos utilisateurs ne l'installent. La clé d'API vous appartient : créez un compte gratuit sur virustotal.com, copiez la clé depuis votre profil et saisissez-la une fois. Elle est conservée dans le trousseau du bureau.

- Analyser le projet regroupe le projet en une seule archive ; un projet inchangé est reconnu à son empreinte et n'est jamais envoyé deux fois.
- Rien n'est envoyé sans votre accord, donné une fois par projet. Fichiers d'identifiants, .git, node_modules et dossier versions ne sont jamais envoyés.
- Résultats : Rien de signalé, « n sur t l'ont signalé » avec le nom des moteurs, ou « jamais vu » — la réponse habituelle pour un fichier qui vient d'être construit.
- Un ou deux moteurs qui signalent un programme fraîchement construit et non signé, c'est le plus souvent un faux positif.
- Une clé gratuite permet 4 analyses par minute et 500 par jour.

7. Paramètres

Réglage	Par défaut	Description
Langue	Langue du bureau	Six langues; Petacore redémarre pour l'appliquer
Apparence	Suivre le système	Clair, sombre ou comme le bureau
Mode focus	Activé	Atténue les lignes hors de la section choisie
Alerte d'envoi plus petit	Activée	Prévient si un envoi est plus petit que la copie sur Drive
Internet dans le bac à sable	Désactivé	Laisse le code du bac à sable accéder au réseau
Demander avant de quitter avec Échap	Activé	Confirmation pour les fichiers non enregistrés
Sauvegarde automatique	Activée, 10 minutes	Instantanés automatiques
GitHub	Non connecté	Connexion, déconnexion, Client ID OAuth facultatif
Google Drive	Non connecté	Connexion, déconnexion, récupération des projets
Récupérer les modifications	Activé, 10 minutes	Modifications de vos autres ordinateurs
Conserver les sauvegardes	30 jours	7, 30, 90, 365 jours ou toujours
Interface	Détectée	GNOME ou KDE Plasma

8. Sécurité et confidentialité

- Connexions et clés d'API sont conservées dans le trousseau du bureau. En son absence, elles vont dans un fichier que vous seul pouvez lire.
- Les fichiers d'identifiants (.env, clés privées, .netrc...) n'entrent jamais dans un paquet, un envoi sur GitHub ou une analyse VirusTotal.
- La connexion GitHub n'est transmise qu'à github.com; une adresse qui lui ressemble ne reçoit rien.
- Les clés de signature privées ne quittent jamais votre ordinateur; seules les clés publiques sont sauvegardées.
- Les connexions aux serveurs se font uniquement par clé; phrases secrètes et codes à usage unique ne sont jamais conservés.
- Le Bac à sable n'a pas de réseau tant que vous ne l'ouvrez pas; le Serveur en direct ne répond qu'à cet ordinateur tant que vous ne le partagez pas.
- Rien n'est envoyé à VirusTotal sans votre accord.
- La synchronisation ne détruit jamais un fichier: ce qu'elle remplace est conservé dans les Sauvegardes.

9. Dépannage

Problème	Solution
Google Drive nécessite rclone	<code>sudo apt install rclone</code>
Le trousseau est inaccessible	<code>sudo apt install libsecret-tools</code> , puis réessayer
GitHub CLI n'est pas prêt	Cliquer sur Installer gh, ou <code>sudo apt install gh</code>
Bac à sable: NON isolé	<code>sudo apt install bubblewrap</code>
Carte de partage enregistrée en SVG seulement	<code>sudo apt install librsvg2-bin</code>
Serveur en direct: pas de navigateur intégré	<code>sudo apt install gir1.2-webkit-6.0</code> , ou Ouvrir dans le navigateur
Impossible de construire l'index du dépôt	<code>sudo apt install apt-utils dpkg-dev</code>
Connexion au serveur: OpenSSH absent	<code>sudo apt install openssh-client</code>
VirusTotal a refusé la clé	Vérifier la clé dans votre profil VirusTotal; un blocage réseau peut aussi être passager — réessayer plus tard
Clé de signature expirée ou absente	Créer une nouvelle clé sur la page Clés et choisir Utiliser pour signer
L'identité du serveur a changé	Ne pas se connecter. Interroger l'administrateur du serveur; n'oublier l'ancienne clé qu'une fois le changement confirmé
GitHub: la branche contient des modifications incompatibles	Récupérer ces modifications avec git dans le Terminal, résoudre les conflits, puis renvoyer
Version déjà publiée sur GitHub	Choisir une autre version
Projet modifié à deux endroits	Choisir la copie à garder; l'autre côté reste dans les Sauvegardes
Mauvais mot de passe pour un projet chiffré	Le mot de passe ne peut pas être récupéré; utiliser le bon
Composants de l'autre interface absents	Cliquer sur Installer maintenant, ou lancer la commande indiquée

10. Raccourcis clavier

Touches	Action
Ctrl+S	Enregistrer le fichier (Éditeur)
Ctrl+Q	Fermer le fichier (Éditeur)
Ctrl+F	Rechercher; Entrée passe au résultat suivant, Échap ferme
Ctrl+K / Ctrl+D	Couper la ligne / dupliquer la ligne
Ctrl+Z / Ctrl+Y	Annuler / rétablir

Touches	Action
Échap	Quitter le fichier ; un second Échap quitte sans enregistrer
Ctrl+A / Suppr	Tout sélectionner / supprimer (page Projet, réversible)
Ctrl+Z	Ramener des fichiers supprimés ou des paquets retirés
Ctrl + pointeur	Aperçu rapide ; Ctrl+molette pour le parcourir
F11 / Échap	Serveur en direct en plein écran / quitter le plein écran

11. Fichiers et désinstallation

11.1 Où Petacore conserve ses données

Emplacement	Contenu
~/local/share/petacore-app	Le programme lui-même
~/config/petacore	Paramètres (lisibles par vous seul)
~/local/share/petacore/snapshots	Instantanés
~/local/share/petacore/drive-backups	Sauvegardes des fichiers remplacés sur cet ordinateur
<projet>/versions	Historique des versions (tenu hors des paquets et de git)
<projet>/petacore	Projets de Prochaines mises à jour
Trousseau du bureau	Connexion GitHub, clé VirusTotal

11.2 Désinstallation

Lancez `./uninstall.sh` depuis le dossier de Petacore, en tant que vous-même. La suppression du programme et celle de vos instantanés et sauvegardes font l'objet de deux questions distinctes, et rien n'est supprimé avant votre confirmation. Vos dossiers de projet ne sont jamais touchés.

12. Licence et droits d'auteur

© Petacomm I.S. 2026. Tous droits réservés. Petacomm Petacore™ est une marque de Petacomm I.S.

Petacore utilise des composants libres et open source, qui restent la propriété de leurs auteurs et sont utilisés selon leurs propres licences. GitHub, Google Drive, VirusTotal, GNOME, KDE Plasma, Ubuntu et Debian sont des marques de leurs propriétaires respectifs.

Ce manuel décrit Petacore 1.1 Beta 2. Les fonctions peuvent évoluer dans les versions ultérieures.

1. 前言

感谢您选择 Petacore——一款用于开发和发布 Linux 软件的桌面工作台。Petacore 把一个项目所需的一切集中在同一个窗口中：文件与代码、历史记录、软件包与签名密钥、软件源、GitHub 和 Google Drive 上的副本，以及在交付给用户之前的病毒扫描。

开始使用之前，请仔细阅读本手册。手册介绍了 Petacore 的每个部分、提供的各项设置，以及它会如何处理——以及绝不会如何处理——您的文件和登录凭据。了解这些内容，能让您工作得更快，也更安心。

开发者： Petacomm I.S.

产品： Petacomm Petacore™

对应版本： 1.1 Beta 2

项目主页： github.com/petacomm/petacore

2. 技术说明

2.1 概述

Petacore 是一款原生的 Linux 桌面应用。它完全在您自己的电脑上运行，直接处理普通的项目文件夹：无需任何转换，每个文件都可以继续用其他工具打开。在线服务（GitHub、Google Drive、VirusTotal）只有在您主动连接时才会使用，并且使用的是您自己的账户。

Petacore 提供两套功能完全相同的界面：一套适用于 GNOME 桌面，一套适用于 KDE Plasma。安装程序会自动选择合适的界面，您也可以随时切换。

2.2 系统要求

项目	要求
操作系统	Ubuntu 26.04 LTS，或其他基于 Debian / Ubuntu 的发行版
桌面环境	GNOME (GTK 4、libadwaita) 或 KDE Plasma (Qt 6、PySide6)
Python	3.12 或更高版本
磁盘空间	Petacore 本身约 50 MB，另需为快照和备份预留空间
网络	仅在使用 GitHub、Google Drive、VirusTotal 及连接服务器时需要

2.3 技术规格

功能	规格
界面语言	English、Türkçe、Deutsch、Français、中文、Українська
软件包格式	.deb (Debian、Ubuntu、Mint) 和 .rpm (Fedora、openSUSE、RHEL)
软件包签名	GnuPG——Ed25519 (推荐) 或 RSA 2048 / 3072 / 4096

功能	规格
版本控制	git；可向 GitHub 发布带版本号的 Release
云端副本	通过 rclone 使用 Google Drive，可选端到端加密
安全扫描	VirusTotal API v3，使用您自己的密钥
软件源	签名的 APT 仓库，可放在本地，也可通过 SSH 放在您自己的服务器上
服务器连接	OpenSSH，仅限密钥登录，支持双重验证
隔离	基于 bubblewrap 的沙盒，默认断开网络

2.4 所用组件

Petacore 建立在成熟的标准工具之上。安装程序会把它们准备好；如果缺少某个组件，Petacore 会用通俗的语言说明，并给出安装它的命令。

组件	用途
git、gh	版本历史；GitHub 登录与发布
rclone	连接 Google Drive、同步与备份
gnupg	创建签名密钥，为软件包和软件源签名
dpkg-dev、rpm、apt-utils	构建软件包和软件源索引
bubblewrap	隔离的沙盒
openssh-client、rsync	连接服务器与发布
libsecret-tools	把登录信息和 API 密钥保存在桌面密钥环中
libsvg2-bin	把分享卡片保存为 PNG 图片
WebKitGTK（可选）	在窗口内显示实时服务器的预览

3. 安装与首次启动

3.1 安装 Petacore

1. 把 Petacore 压缩包解压到任意文件夹。
2. 在该文件夹中打开终端，运行 `./install.sh`——请以您自己的身份运行，切勿使用 `sudo`。Petacore 会安装到您的个人主目录。
3. 安装程序会识别您的桌面（GNOME 或 KDE Plasma）；如果无法判断，会询问您想要哪种界面。
4. 需要安装系统软件包时，它只会询问一次密码，然后用 `apt` 安装。
5. 完成后，Petacore 会出现在应用菜单中；也可以在终端里用 `petacore` 命令启动。

Petacore 也可以通过 .deb 软件包安装，结果完全相同；安装脚本只是额外提供了自动选择界面的功能。

3.2 首次启动

首次启动时，一个简短的向导会先询问语言（默认建议桌面所用的语言），然后提议连接 GitHub 和 Google Drive，最后帮您创建第一个项目。每一步都可以跳过，之后在“设置”中完成。

3.3 主窗口

左侧边栏列出当前项目的各个页面；顶部的项目选择器用于在项目之间切换，铅笔按钮用于编辑当前项目。窗口顶部是 GitHub 按钮（上传到 GitHub）、Google Drive 按钮（发送到 Drive）以及主菜单（新建项目、设置、关于 Petacore、退出）。

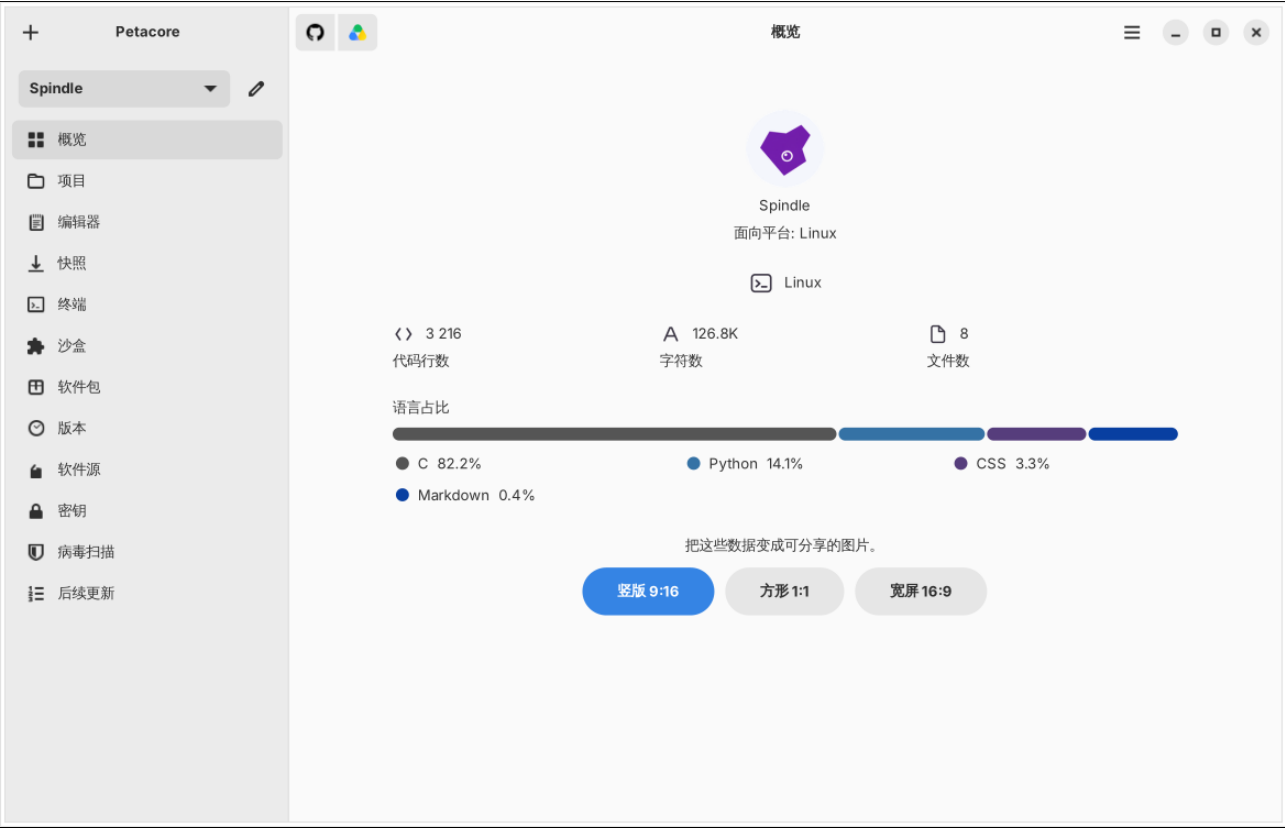


图 1 — 显示“概览”页面的主窗口。

显示哪些页面取决于项目类型。面向操作系统的项目会有“软件包”“密钥”和“沙盒”；网页项目既不打包也不签名，因此改为显示“实时服务器”。只有在识别结果明确时 Petacore 才会这样调整，无法识别的项目始终保留全部页面。

页面	用途
概览	项目的各项数据，以及可分享的海报
项目	项目文件夹的文件管理器
编辑器	多标签页代码编辑器
快照	整个项目的还原点

页面	用途
终端	在项目文件夹中打开的 Shell
沙盒 / 实时服务器	安全地试运行程序 / 预览网站
软件包	构建已签名的 .deb 和 .rpm 软件包
版本	发布历史，以及每个版本的代码
软件源	本地或服务器上的签名 APT 软件源
密钥	GPG 签名密钥
病毒扫描	用 VirusTotal 检查项目
后续更新	项目的计划

3.4 选择界面

“设置 → 界面”会显示当前使用的设计，并可切换到另一种。Petacore 会关闭并以新的设计重新打开，项目和设置保持不变。如果缺少另一种设计所需的组件，Petacore 会提议为您安装。

4. 项目

4.1 创建项目

在主菜单中选择“新建项目”，或点击 + 按钮。输入名称以及项目所在的文件夹。

- 如果填写了 GitHub 地址，Petacore 会下载（克隆）该仓库。由于这些代码来自互联网，Petacore 随后会询问：在运行其中任何内容之前，是否先让 VirusTotal 检查一遍。
- 如果不填写地址，则会新建一个本地 git 仓库。
- 在“加密”中，您可以选择用 4 位 PIN、8 位 PIN 或最长 4096 个字符的密码短语来加密 Google Drive 上的副本。电脑上的项目仍然是普通、可读的代码；只有 Drive 上的副本会被加密——包括文件名和文件夹名。

加密密码不会保存在任何地方。一旦遗失，任何人（包括 Petacomm）都无法打开 Drive 上的加密副本。

4.2 编辑与移除项目

项目选择器旁的铅笔按钮会打开“编辑项目”：名称、项目文件夹、GitHub 地址以及已安装应用的文件夹。设置了已安装应用的文件夹后，在编辑器中保存的每个文件都可以按相同的相对路径复制到已安装的程序中——无需使用终端。

“从 Petacore 移除”只会把项目从列表中去掉，不会删除磁盘上的任何文件。

4.3 概览

“概览”让您一眼看清项目：代码行数、字符数、文件数、语言数和项目大小，面向的平台（Linux、Windows、macOS、Web 或跨平台），以及用彩色条显示的语言占比。

“分享卡片”可把这些数据做成三种尺寸的海报——竖版 9:16、方形 1:1 和宽屏 16:9。卡片上只显示名称、数据和语言占比，绝不会显示文件名或代码；如果不便公开项目本名，可以修改卡片上显示的名称。

4.4 “项目”页面

“项目”页面是项目文件夹的完整文件管理器。工具栏下方的一行显示 git 分支、项目相对于 GitHub 的状态，以及从该文件夹运行的程序数量。

- 双击可在编辑器中打开文件；图片、音频和视频会在各自的查看器中打开。
- 右键菜单提供“打开”“重命名”“新建文件夹”“复制路径”和“删除”。
- 删除可以撤销：按 Ctrl+Z 或点击撤销按钮即可恢复文件。
- 可以从文件管理器中拖入文件。遇到同名文件时，可选择“替换 / 合并”“全部保留”或“跳过”。
- 搜索框在当前文件夹中搜索；旁边的按钮则搜索整个项目。筛选器可只显示代码、图片、音视频、文档或文件夹。
- 在文件上按住 Ctrl 可快速预览；较长的预览可用 Ctrl+滚轮浏览。
- “媒体文件”把项目中的所有图片、音频和视频集中到一处。
- 时钟按钮用于打开“备份”（见 6.5）。

4.5 编辑器

编辑器以标签页方式打开文件，提供语法高亮，并沿用 micro 编辑器的常用快捷键（见第 10 章）。

- 代码段：选中若干行并添加代码段，即可为长文件的某一部分命名；在代码段列表中点一下即可跳转过去。
- 专注模式会淡化所选代码段以外的所有行，让您正在处理的部分更突出。按 Esc 取消专注。
- 按 Esc 离开文件。如果有未保存的修改，Petacore 会先询问：再按一次 Esc 不保存直接离开，按 Ctrl+S 保存后离开。这一询问可以在“设置”中关闭。
- 如果项目设置了已安装应用的文件夹，保存时会提议同时更新已安装的程序。

4.6 快照与自动保存

快照是整个项目的完整还原点。“创建快照”可随时创建一个，并可附上标签。自动保存会按设定的间隔（默认 10 分钟）自动创建快照，并保留最近的 40 个自动快照。

还原快照前，会先为当前状态创建一个安全快照，因此还原操作本身也可以撤销。快照保存在项目之外、您自己的数据文件夹中；.git、versions 文件夹和构建产生的临时文件不包含在内。

4.7 终端

“终端”页面会直接在项目文件夹中打开您惯用的 Shell。

4.8 沙盒

沙盒在您系统的一个密封副本中运行命令。在沙盒里，真实系统是只读的：安装、删除和修改只影响这个用完即弃的会话，“结束模拟”会把一切清除得无影无踪。

- “试运行当前项目”会带着项目副本启动沙盒。
- “临时安装 .deb”只把软件包安装在沙盒内部。

- “运行应用”可在沙盒中启动图形程序或终端程序。
- 沙盒默认不能访问互联网。如果其中有程序试图联网，Petacore 会告诉您，您可以选择允许、保持关闭，或不再询问。

完整隔离需要 bubblewrap。如果没有安装，页面会提示“未隔离”，并给出安装命令。

4.9 实时服务器

对于网页项目，实时服务器无需任何构建步骤即可在 Petacore 中显示网站，并在文件改动时自动刷新。按 F11 让网站铺满窗口，按 Esc 返回。

默认情况下，服务器只响应这台电脑。开启“在局域网内共享”后，同一网络中的手机或其他设备也能访问，并会显示在那里打开的地址。如果没有 WebKitGTK，预览会改在您常用的浏览器中打开。

4.10 后续更新

“后续更新”用于记录项目的计划：标题、详情、优先级（高、普通、以后）以及附加的文档。计划可以编辑、标记为完成，也可以复制。它们保存在项目内部，会随项目一起迁移，并可同步到 Google Drive。

5. 软件包与发布

5.1 软件包

“软件包”页面会把项目构建为安装包：面向 Debian、Ubuntu 和 Mint 的 .deb，以及面向 Fedora、openSUSE 和 RHEL 的 .rpm。安装包会把程序安装到目标系统的 /opt/<名称> 下，并加入应用菜单。

字段	含义
名称	软件包名称，使用小写字母
版本	例如 1.6 或 2.0.1
维护者	负责人的姓名和电子邮件地址
描述	在软件中心和包管理器中显示
许可证	从列表中选择或手动填写；“比较许可证”会解释各自的区别
签名	用当前密钥为软件包签名（默认开启）

含有凭据的文件——.env、私钥、.netrc、.npmrc 等——绝不会被打进软件包，Petacore 会告诉您排除了多少个。

5.2 签名密钥

GPG 密钥就是您的数字签名：它证明软件包确实出自您手，且未被任何人篡改。“密钥”页面列出您的所有密钥，并标明当前用于签名的是哪一个。

设置	选项
姓名、电子邮件、备注	密钥的所有者
密钥类型	Ed25519（现代、快速，推荐），或面向旧系统的 RSA 4096 / 3072 / 2048
有效期	永不过期、1 年、2 年或 5 年
密码短语	可选。设置后，GPG 每次签名都会询问——更安全

- “用于签名”会把某个密钥设为当前签名密钥。
- “导出公钥”会保存一个文件，用户可用它验证您的软件包。
- 密钥的公钥部分可以备份到 Google Drive；私钥绝不会离开这台电脑。
- 删除无法撤销：此后无法再用该密钥为软件包签名。

5.3 版本

“版本”页面按从新到旧的顺序保存项目发布过的每个版本及其软件包。它有两个标签页：“新版本”和“版本库”。

- “新版本”根据项目文件的当前状态构建一个版本：版本号、发布类型（稳定版、候选版、测试版、内测版）、预发布编号、格式（.deb、.rpm）、签名以及版本说明。预览会同时显示给人看的名称（“1.6 Beta 2”）和包管理器比较用的版本号（“1.6~beta2”）。
- “版本库”按系列（1.6、2.0……）对版本分组，并标出最新版本。每个版本都会显示其软件包、签名和说明。
- “查看代码”可打开任意版本的代码——直接从该版本自己的软件包中读取，与当初发布的内容完全一致，且为只读。文件可以筛选，内容可以复制。
- 把旧的 .deb 或 .rpm 文件拖到页面上，即可把该版本加入历史记录；原文件保持不动。
- 每个版本都提供“编辑说明”“将 .deb 加入软件源”和“移除版本”。移除的版本会进入版本回收站，可以恢复。

各个版本保存在项目的 versions 文件夹中。该文件夹不会进入任何软件包、快照、统计数据或 git 历史，因此任何软件包都不会包含它之前的版本。

5.4 APT 软件源

软件源能把一个个独立的软件包变成一个软件来源：用户只需添加一次，之后就能像其他软件一样收到更新。Petacore 会生成目录结构、为索引签名，并让索引始终与您的软件包保持一致。

设置	含义
名称、软件源文件夹	软件源的名称，以及生成目录结构的位置
公开地址	用户安装时使用的 URL，例如 https://repo.example.com
Suite、组件、架构	常用值：stable、main、amd64、arm64
Origin、Label、描述	软件源向 apt 介绍自己的方式
签名密钥	apt 会拒绝索引未签名的仓库

设置	含义
发布方式	不从此处发布、复制到某个文件夹，或通过 SSH (rsync) 发送

- 使用“添加软件包”，或把 .deb 文件拖到页面上。开启“把生成的软件包加入软件源”后，每个新的 .deb 都会自动加入。
- “重建并签名”会更新索引；一旦索引与软件包不一致，就会出现提示。
- “发布”让服务器上的副本与文件夹保持一致；“检查服务器”会比较两者。
- 标记和筛选器会显示未编入索引、未签名、未发布或已被新版本取代的软件包。
- 每个软件包都提供“验证签名”“复制安装命令”“重命名”“打开所在文件夹”和“从软件源移除”。
- 移除的软件包会进入软件源回收站，可以放回；“移除旧版本”只保留每个软件包的最新版本。
- “复制给用户的说明”和“导出软件源密钥”为用户提供添加软件源所需的一切。

5.5 服务器上的软件源

在“服务器”标签页中，Petacore 通过 SSH 直接管理您服务器上的软件源文件夹。文件用 scp 上传；索引在服务器上生成、在您的电脑上签名，因此签名密钥绝不会离开您的电脑。

1. 填写服务器名称或 IP、端口、用户名，以及服务器上的软件源文件夹，例如 /var/www/repo。
2. 选择一个 SSH 密钥，或为 Petacore 新建一个：Ed25519，由至少 8 个字符的密码短语保护，该密码短语绝不保存。
3. 把 Petacore 显示的那一行添加到服务器上该用户的 ~/.ssh/authorized_keys 文件中。
4. 点击连接。首次连接时，请把服务器指纹与服务器自己报告的指纹进行比对，只有一致时才确认。

连接后，您可以上传文件（.deb 文件会自动放入 pool）、新建文件夹、重命名、下载、把条目移到服务器回收站并恢复，以及执行“重建并签名”。

保护措施	含义
主机密钥检查	每次连接都会核对服务器身份；一旦发生变化，Petacore 会拒绝连接
仅限密钥登录	绝不发送密码
现代加密	在可用时使用后量子密钥交换，并采用认证加密
双重验证登录	身份验证器应用的验证码和 FIDO2 安全密钥
不保存任何内容	密码短语和双重验证码绝不保存

6. 在线服务

6.1 连接 GitHub

1. 点击“使用 GitHub 连接”——可在“设置”中、首次启动向导中，或任何需要 GitHub 的时候点击。
2. 浏览器会打开 GitHub，一次性验证码已经复制到剪贴板。

3. 粘贴验证码并确认，然后点击 Authorize。Petacore 会自动完成连接。

如果这台电脑上的 GitHub CLI (gh) 已经登录，只需点一下即可。如果没有 gh，Petacore 会提议为您安装。在“其他登录方式”中也可以手动粘贴个人访问令牌。登录信息保存在桌面密钥环中，绝不会写入文件或网址。

6.2 上传到 GitHub

窗口顶部的 GitHub 按钮会打开“上传到 GitHub”。只需一步，代码就会推送到 main 分支，并在项目的 GitHub 页面上成为一个新版本。

- 从建议中选择一个版本——例如 1.0 Beta 3、1.0 Beta 2.1、1.0 RC 1 或 1.0——也可以自行输入。已在版本库中但尚未发布到 GitHub 的版本会排在最前面。
- 标签根据版本生成 (v1.0-beta.3)；测试版、内测版和候选版会被标记为预发布版本。
- 如果项目还没有 GitHub 仓库，会自动创建一个——默认为私有。
- 除非您修改，分支始终为 main。版本说明可选；留空时，GitHub 会根据提交记录自动生成。
- 版本库中该版本的 .deb 和 .rpm 软件包会附加到发布中。
- 关闭“作为新版本发布”即可只上传代码。

含有凭据的文件绝不会上传。GitHub 上已有的内容会被保留：如果与您的内容冲突，上传会停止，不会覆盖任何内容。已经发布的版本绝不会被替换。

6.3 从 GitHub 克隆

用 GitHub 地址创建项目时，Petacore 会询问一次是否让 VirusTotal 检查下载的代码。点击“扫描”会打开“病毒扫描”页面并立即开始检查。

6.4 Google Drive

在“设置”中连接您自己的 Google Drive：浏览器打开后，登录 Google 并点击“允许”。Petacore 会创建一个名为“Petacomm Petacore™”的文件夹，把每个项目（包括媒体文件）放在各自的子文件夹中。Petacore 自己的记录则放在旁边一个隐藏的 .petacore 文件夹中。

- Drive 按钮会发送当前项目。Drive 会与这台电脑保持一致——并且不会丢失任何内容（见 6.5）。
- 如果在这台电脑上次同步之后，另一台电脑又发送了更新的副本，Petacore 会提议先把它取回。
- 如果上传的内容会比 Drive 中的少，Petacore 会先发出警告（可在“设置”中关闭）。
- “从 Drive 获取项目”会把只存在于 Drive 上的项目下载到 ~/Projects。
- 加密项目会经过一个加密层发送；Drive 上只能看到无法读取的数据。

6.5 备份

与 Drive 同步时，Petacore 从不删除任何文件。同步时被替换或移除的每个文件都会先保存为备份——发送时备份在 Drive 上，取回副本时备份在这台电脑上。

- 备份按日期列出：时间、来自哪台电脑、文件数量和大小。
- 打开一个备份即可看到其中的文件。“放回”可以恢复整个备份，也可以只恢复单个文件。
- 在放回任何内容之前，当前的文件也会先被备份，因此这一步同样可以撤销。
- 备份默认保留 30 天（可在“设置”中选择 7、30、90、365 天或永久），到期后自动删除。

您可以通过“项目”页面上的时钟按钮、同步通知中的“备份”按钮，或“设置 → Google Drive → 查看备份”打开备份。

6.6 来自其他电脑的改动

登录同一个 Google Drive 的电脑会自动取回彼此的最新副本：启动几秒后检查一次，此后每 10 分钟检查一次（可在“设置”中选择 5、10、30 或 60 分钟）。只存在于 Drive 上的项目也会被下载；如果同名文件夹已存在，则下载到一个新文件夹。

情况	Petacore 的处理方式
Drive 上有更新的副本，本机没有改动	静默取回，并告诉您是哪台电脑发送的
Drive 上有更新的副本，本机也有改动	询问您：采用 Drive 上的副本、上传我的，或暂不处理
加密项目	不会自动取回——需要输入密码

无论选择哪一方，另一方的文件都会保留在备份中。此功能默认开启，可在“设置 → Google Drive”中关闭。

6.7 病毒扫描

“病毒扫描”会在用户安装之前，让 VirusTotal 的各个引擎检查您的项目。API 密钥属于您自己：在 [virustotal.com](https://www.virustotal.com) 免费注册账户，从个人资料中复制密钥并输入一次即可。密钥保存在桌面密钥环中。

- “扫描项目”会把项目打包成一个压缩包；未改动的项目会通过指纹识别出来，绝不会重复发送。
- 未经您同意不会发送任何内容，每个项目只需同意一次。凭据文件、.git、node_modules 和 versions 文件夹绝不会被发送。
- 结果分为：未发现问题；“t 个引擎中有 n 个报告”，并列引擎名称；或“从未见过”——对刚构建的文件来说，这是正常的结果。
- 刚构建、未签名的程序被一两个引擎报告，通常属于误报。
- 免费密钥每分钟可检查 4 次，每天 500 次。

7. 设置

设置	默认值	说明
语言	桌面语言	共六种语言；切换后 Petacore 会重新启动
外观	跟随系统	浅色、深色或与桌面一致
专注模式	开启	淡化所选代码段以外的行
上传变小警告	开启	当上传内容小于 Drive 中的副本时发出警告
沙盒联网	关闭	允许沙盒中的代码访问网络
用 Esc 离开前询问	开启	离开未保存的文件时进行确认
自动保存	开启，10 分钟	自动创建快照
GitHub	未连接	连接、断开，可选的 OAuth 客户端 ID

设置	默认值	说明
Google Drive	未连接	连接、断开、获取项目
自动取回改动	开启, 10 分钟	取回您其他电脑上的改动
备份保留时长	30 天	7、30、90、365 天或永久
界面	自动检测	GNOME 或 KDE Plasma

8. 安全与隐私

- 登录信息和 API 密钥保存在桌面密钥环中；如果没有密钥环，则保存在一个只有您能读取的文件中。
- 凭据文件（.env、私钥、.netrc.....）绝不会进入软件包、GitHub 上传或 VirusTotal 扫描。
- GitHub 登录信息只会提供给 github.com；仿冒的地址什么也拿不到。
- 签名私钥绝不会离开您的电脑，只有公钥会被备份。
- 登录服务器只使用密钥；密码短语和双重验证码绝不保存。
- 除非您主动开启，沙盒无法联网；除非您主动共享，实时服务器只响应这台电脑。
- 未经您同意，不会向 VirusTotal 发送任何内容。
- 同步绝不会销毁文件：被替换的内容都会保留在备份中。

9. 故障排除

问题	解决方法
使用 Google Drive 需要 rclone	<code>sudo apt install rclone</code>
无法访问密钥环	<code>sudo apt install libsecret-tools</code> ，然后重试
GitHub CLI 尚未就绪	点击“安装 gh”，或运行 <code>sudo apt install gh</code>
沙盒：未隔离	<code>sudo apt install bubblewrap</code>
分享卡片只保存为 SVG	<code>sudo apt install librsvg2-bin</code>
实时服务器：没有内置浏览器	<code>sudo apt install gir1.2-webkit-6.0</code> ，或使用“在浏览器中打开”
无法生成软件源索引	<code>sudo apt install apt-utils dpkg-dev</code>
连接服务器：缺少 OpenSSH	<code>sudo apt install openssh-client</code>
VirusTotal 拒绝了密钥	在 VirusTotal 个人资料中核对密钥；网络拦截也可能是暂时的——请稍后再试
签名密钥已过期或不存在	在“密钥”页面新建密钥，并选择“用于签名”
服务器身份已改变	不要连接。请联系服务器管理员；确认变更属实后，才可忘记旧密钥



问题	解决方法
GitHub：分支中有不相容的改动	在“终端”中用 git 取回这些改动并解决冲突，然后重新上传
该版本已在 GitHub 发布	换一个版本
项目在两处都有改动	选择要保留的副本；另一方会保留在备份中
加密项目的密码错误	密码无法找回；请使用正确的密码
缺少另一种界面的组件	点击“立即安装”，或运行 Petacore 显示的命令

10. 键盘快捷键

按键	操作
Ctrl+S	保存文件（编辑器）
Ctrl+Q	关闭文件（编辑器）
Ctrl+F	查找；Enter 跳到下一个结果，Esc 关闭
Ctrl+K / Ctrl+D	剪切当前行 / 复制当前行
Ctrl+Z / Ctrl+Y	撤销 / 重做
Esc	离开文件；再按一次 Esc 不保存直接离开
Ctrl+A / Delete	全选 / 删除（“项目”页面，可撤销）
Ctrl+Z	恢复已删除的文件或已移除的软件包
Ctrl + 指针	快速预览；Ctrl+滚轮浏览
F11 / Esc	实时服务器全屏 / 退出全屏

11. 文件与卸载

11.1 Petacore 的数据存放位置

位置	内容
~/.local/share/petacore-app	程序本身
~/.config/petacore	设置（仅您本人可读）
~/.local/share/petacore/snapshots	快照
~/.local/share/petacore/drive-backups	这台电脑上被替换文件的备份
<项目>/versions	发布历史（不进入软件包和 git）
<项目>/petacore	“后续更新”中的计划

位置	内容
桌面密钥环	GitHub 登录信息、VirusTotal 密钥

11.2 卸载

在 Petacore 文件夹中以您自己的身份运行 `./uninstall.sh`。卸载程序与删除快照和备份是两个独立的问题，在您确认之前不会删除任何内容。您的项目文件夹绝不会被触及。

12. 许可与版权

© Petacomm I.S. 2026。保留所有权利。Petacomm Petacore™ 是 Petacomm I.S. 的商标。

Petacore 使用了自由及开源组件，这些组件仍归其作者所有，并按各自的许可证使用。GitHub、Google Drive、VirusTotal、GNOME、KDE Plasma、Ubuntu 和 Debian 是其各自所有者的商标。

本手册对应 Petacore 1.1 Beta 2。后续版本的功能可能有所变化。

1. Передмова

Дякуємо, що обрали Petacore — настільну робочу станцію для створення та випуску програм для Linux. Petacore зводить в одне вікно все, що потрібно проєкту: його файли й код, історію, пакунки та ключі підпису, репозиторій програм, копію на GitHub і на Google Диску, а також перевірку на віруси, перш ніж щось потрапить до ваших користувачів.

Перш ніж почати роботу, уважно прочитайте цей посібник. Він описує кожен частину Petacore, доступні налаштування і те, що Petacore робить — і чого ніколи не робить — із вашими файлами та обліковими даними. Знаючи це, ви працюватимете швидше й цілком спокійно.

Розробник: Petacomm I.S.
Продукт: Petacomm Petacore™
Описана версія: 1.1 Beta 2
Сторінка проєкту: github.com/petacomm/petacore

2. Технічний опис

2.1 Загальні відомості

Petacore — це нативна настільна програма для Linux. Вона повністю працює на вашому комп'ютері й має справу зі звичайними теками проєктів: нічого не треба перетворювати, і кожен файл лишається доступним для будь-якого іншого інструмента. Онлайн-сервіси (GitHub, Google Диск, VirusTotal) використовуються лише тоді, коли ви їх під'єднаєте, — і лише з вашими власними обліковими записами.

Petacore має два інтерфейси з однаковими можливостями: один для стільниці GNOME, другий для KDE Plasma. Інстальатор сам вибирає відповідний, а перемкнутися можна будь-коли.

2.2 Системні вимоги

Параметр	Вимога
Операційна система	Ubuntu 26.04 LTS або інший дистрибутив на основі Debian чи Ubuntu
Стільниця	GNOME (GTK 4, libadwaita) або KDE Plasma (Qt 6, PySide6)
Python	3.12 або новіший
Місце на диску	Близько 50 МБ для Petacore, а також місце для знімків і резервних копій
Інтернет	Лише для GitHub, Google Диска, VirusTotal і з'єднань із сервером

2.3 Технічні характеристики

Можливість	Характеристика
Мови інтерфейсу	English, Türkçe, Deutsch, Français, 中文, Українська
Формати пакунків	.deb (Debian, Ubuntu, Mint) і .rpm (Fedora, openSUSE, RHEL)
Підпис пакунків	GnuPG — Ed25519 (рекомендовано) або RSA 2048 / 3072 / 4096
Керування версіями	git; публікація на GitHub із випусками за версіями

Можливість	Характеристика
Хмарна копія	Google Диск через rclone, за бажанням із наскрізним шифруванням
Перевірка безпеки	VirusTotal API v3, із вашим власним ключем
Репозиторій програм	Підписаний APT-архів — локально або на вашому сервері через SSH
З'єднання з сервером	OpenSSH, вхід лише за ключем, підтримка двофакторного входу
Ізоляція	Пісочниця на основі bubblewrap, мережу типово вимкнено

2.4 Використані компоненти

Petacore спирається на перевірені стандартні інструменти. Інстальатор їх налаштовує; якщо якогось бракує, Petacore зрозуміло про це повідомляє й показує команду для встановлення.

Компонент	Призначення
git, gh	Історія версій; вхід у GitHub і публікація
rclone	Під'єднання до Google Диска, синхронізація та резервні копії
gnupg	Створення ключів підпису; підпис пакунків і репозиторіїв
dpkg-dev, rpm, apt-utils	Збирання пакунків та індексів репозиторію
bubblewrap	Ізольована Пісочниця
openssh-client, rsync	З'єднання з сервером і публікація
libsecret-tools	Зберігання даних входу та API-ключів у в'язці ключів стільниці
librsvg2-bin	Збереження карток для поширення у форматі PNG
WebKitGTK (необов'язково)	Показ Живого перегляду просто у вікні

3. Встановлення та перший запуск

3.1 Встановлення Petacore

1. Розпакуйте архів Petacore в будь-яку теку.
2. Відкрийте в цій теці термінал і запустіть `./install.sh` — від власного імені, ніколи через `sudo`. Petacore встановлюється у вашу домашню теку.
3. Інстальатор розпізнає вашу стільницю (GNOME чи KDE Plasma). Якщо не вдасться, він запитає, який інтерфейс вам потрібен.
4. Коли потрібні системні пакунки, він один раз запитує пароль і встановлює їх через `apt`.
5. Після цього Petacore з'являється в меню програм; із термінала його запускає команда `petacore`.

Petacore можна встановити й із пакунка `.deb`. Результат той самий; скрипт інстальатора лише додає автоматичний вибір інтерфейсу.

3.2 Перший запуск

Під час першого запуску короткий помічник спершу запитує мову (пропонується мова вашої стільниці), далі пропонує під'єднати GitHub і Google Диск, а наприкінці — створити перший проєкт. Кожен крок можна пропустити й виконати пізніше в Налаштуваннях.

3.3 Головне вікно

Ліва бічна панель містить сторінки активного проєкту; перемикач проєктів угорі дає змогу переходити між проєктами, а кнопка з олівцем — змінити активний. У верхній частині вікна розташовані кнопка GitHub (Завантажити на GitHub), кнопка Google Диска (надіслати на Диск) і головне меню (Новий проєкт, Налаштування, Про Petacore, Вийти).

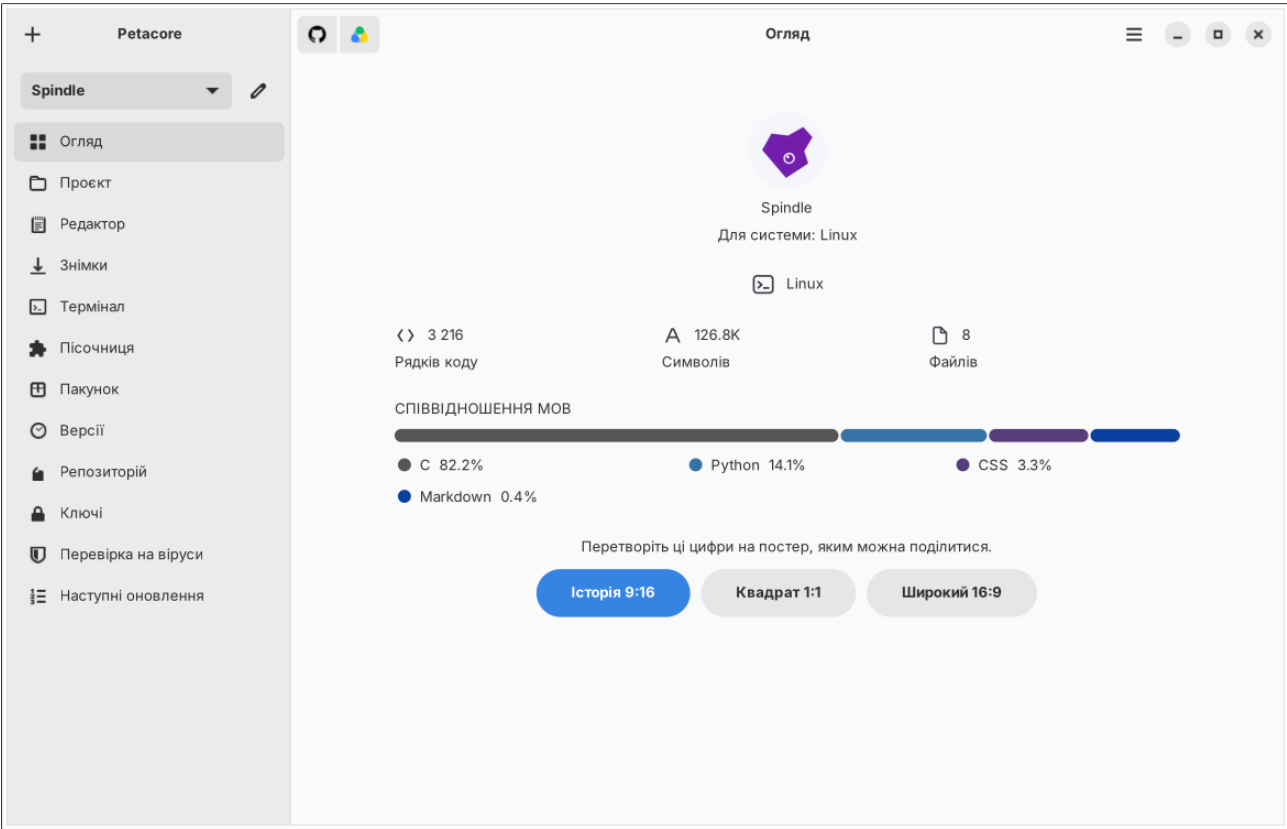


Рисунок 1 — Головне вікно зі сторінкою «Огляд».

Набір сторінок залежить від типу проєкту. Проєкт для операційної системи отримує Пакунок, Ключі й Пісочницю; вебпроєкт натомість отримує Живий перегляд, адже його не пакують і не підписують. Petacore ухвалює таке рішення лише за чіткого збігу, тож незнайомий проєкт завжди зберігає повний набір сторінок.

Сторінка	Призначення
Огляд	Показники проєкту та постер для поширення
Проект	Файловий менеджер для теки проєкту
Редактор	Редактор коду з вкладками
Знімки	Точки відновлення всього проєкту
Термінал	Командна оболонка, відкрита в теці проєкту
Пісочниця / Живий перегляд	Безпечна спроба програми / перегляд вебсайту

Сторінка	Призначення
Пакунок	Збирання підписаних пакунків .deb і .rpm
Версії	Історія випусків і код кожного з них
Репозиторій	Підписаний APT-репозиторій — локально або на сервері
Ключі	Ключі підпису GPG
Перевірка на віруси	Перевірка проєкту через VirusTotal
Наступні оновлення	Плани щодо проєкту

3.4 Вибір інтерфейсу

Налаштування → Інтерфейс показує, який дизайн використовується, і перемикає на інший. Petascore закривається й знову відкривається в новому дизайні; проєкти й налаштування лишаються незмінними. Якщо компонентів іншого дизайну бракує, Petascore запропонує їх установити.

4. Проєкти

4.1 Створення проєкту

Виберіть «Новий проєкт» у головному меню або натисніть кнопку +. Введіть назву й теку, де розташовуватиметься проєкт.

- Якщо вказати адресу на GitHub, Petascore завантажить (клонує) цей репозиторій. Оскільки код надходить з інтернету, Petascore потім запитає, чи перевірити його через VirusTotal, перш ніж ви щось у ньому запускатимете.
- Без адреси створюється новий локальний git-репозиторій.
- У розділі «Шифрування» можна вибрати, щоб копія на Google Диску була зашифрована 4-значним PIN-кодом, 8-значним PIN-кодом або пароллю фразою завдовжки до 4096 символів. Проєкт на вашому комп'ютері лишається звичайним, читабельним кодом; шифрується лише копія на Диску — разом із назвами файлів і тек.

Пароль шифрування ніде не зберігається. Якщо його втрачено, зашифровану копію на Диску не зможе відкрити ніхто — зокрема й Petascore.

4.2 Зміна та вилучення проєкту

Кнопка з олівцем біля перемикача проєктів відкриває «Змінити проєкт»: назва, тека проєкту, адреса на GitHub і тека встановленої програми. Якщо цю теку задано, кожен файл, збережений у Редакторі, можна скопіювати у встановлену програму за тим самим відносним шляхом — без терміналу.

«Прибрати з Petascore» лише прибирає проєкт зі списку. Жоден файл на диску не видаляється.

4.3 Огляд

«Огляд» показує проєкт з одного погляду: кількість рядків коду, символів, файлів і мов, розмір проєкту, систему, для якої його написано (Linux, Windows, macOS, Web чи кросплатформний), а також співвідношення мов у вигляді кольорової смуги.

«Картка для поширення» перетворює ці показники на постер у трьох форматах — Історія 9:16, Квадрат 1:1 і Широкий 16:9. На картці лише назва, показники та співвідношення мов, жодних назв файлів чи коду; назву на картці можна змінити, якщо справжню назву проєкту не варто оприлюднювати.

4.4 Сторінка «Проект»

Сторінка «Проект» — це повноцінний файловий менеджер для теки проекту. Рядок під панеллю інструментів показує git-гілку, стан проекту щодо GitHub і кількість програм, запущених із цієї теки.

- Подвійне клацання відкриває файл у Редакторі; зображення, аудіо й відео відкриваються у власних переглядачах.
- Контекстне меню містить «Відкрити», «Переименувати», «Нова тека», «Копіювати шлях» і «Видалити».
- Видалення можна скасувати: Ctrl+Z або кнопка скасування повертає файли.
- Перетягуйте файли з файлового менеджера. Якщо назви збігаються, виберіть «Замінити / об'єднати», «Залишити обидва» або «Пропустити».
- Поле пошуку шукає у відкритій теці; кнопка поруч — у всьому проекті. Фільтр обмежує список кодом, зображеннями, аудіо й відео, документами або теками.
- Утримуйте Ctrl над файлом, щоб швидко його переглянути; Ctrl+коліщатко прокручує довгий перегляд.
- «Медіафайли» збирає в одному місці всі зображення, звуки й відео проекту.
- Кнопка з годинником відкриває «Резервні копії» (див. 6.5).

4.5 Редактор

Редактор відкриває файли у вкладках, із підсвічуванням синтаксису та звичними клавішами редактора місго (див. розділ 10).

- Розділи: виділіть рядки й додайте розділ, щоб дати назву частині довгого файлу; список розділів веде просто до неї.
- Режим фокусування притлумлює всі рядки поза вибраним розділом, тож частина, над якою ви працюєте, вирізняється. Esc знімає фокус.
- Esc закриває файл. Якщо в ньому є незбережені зміни, Petacore спершу запитас: повторне натискання Esc закриває без збереження, Ctrl+S зберігає й закриває. Це запитання можна вимкнути в Налаштуваннях.
- Якщо проект має теку встановленої програми, під час збереження Petacore запропонує оновити й встановлену програму.

4.6 Знімки та автозбереження

Знімок — це повна точка відновлення проекту. «Створити знімок» створює його будь-коли, за бажанням із міткою. Автозбереження створює знімки саме через заданий інтервал (типово 10 хвилин); зберігаються 40 найновіших автоматичних знімків.

Перед відновленням знімка спершу створюється запобіжний знімок поточного стану, тож і відновлення можна скасувати. Знімки зберігаються поза проектом, у вашій власній теці даних; .git, тека versions і залишки збирання до них не входять.

4.7 Термінал

Сторінка «Термінал» відкриває вашу звичну командну оболонку просто в теці проекту.

4.8 Пісочниця

Пісочниця виконує команди в запечатаній копії вашої системи. Справжня система всередині доступна лише для читання: встановлення, видалення й зміни стосуються тільки цього одноразового сеансу, а «Завершити симуляцію» прибирає все без сліду.

- «Спробувати поточний проект» запускає Пісочницю з копією проекту.

- «Тимчасово встановити .deb» встановлює пакунок лише всередині Пісочниці.
- «Запустити програму» запускає всередині графічні або термінальні програми.
- Типово Пісочниця не має доступу до інтернету. Якщо щось усередині намагається вийти в мережу, Petascore повідомить про це й дасть змогу дозволити, залишити вимкненим або більше не питати.

Повна ізоляція потребує bubblewrap. Без нього сторінка попереджає «НЕ ізольовано» й показує команду для встановлення.

4.9 Живий перегляд

Для вебпроектів Живий перегляд показує сайт просто в Petascore без жодного збирання й сам оновлює його, щойно змінюється файл. F11 розгортає сайт на все вікно; Esc повертає назад.

Типово сервер відповідає лише цьому комп'ютеру. «Відкрити в локальній мережі» робить його доступним із телефона чи іншого пристрою в тій самій мережі й показує адресу, яку там слід відкрити. Без WebKitGTK перегляд відкривається у вашому звичному браузері.

4.10 Наступні оновлення

«Наступні оновлення» зберігає плани щодо проєкту: заголовки, подробиці, пріоритет (Високий, Звичайний, Колись) і вкладені документи. Плани можна змінювати, позначати виконаними й копіювати. Вони зберігаються всередині проєкту, тож переїжджають разом із ним, і можуть синхронізуватися з Google Диском.

5. Пакунки та випуски

5.1 Пакунок

Сторінка «Пакунок» збирає з проєкту інсталяційний пакунок: .deb для Debian, Ubuntu й Mint, .rpm для Fedora, openSUSE й RHEL. Пакунок встановлює програму в /opt/<назва> у цільовій системі й додає її до меню програм.

Поле	Значення
Назва	Назва пакунка малими літерами
Версія	Наприклад, 1.6 або 2.0.1
Супровідник	Ім'я та адреса електронної пошти відповідальної особи
Опис	Показується в центрах програм і менеджерах пакунків
Ліцензія	Вибирається зі списку або вводиться; «Порівняти ліцензії» пояснює відмінності
Підпис	Підписує пакунок активним ключем (типово увімкнено)

Файли з обліковими даними — .env, закриті ключі, .netrc, .prmtc тощо — ніколи не потрапляють до пакунка, а Petascore повідомляє, скільки їх залишено поза ним.

5.2 Ключі підпису

Ключ GPG — це ваш цифровий підпис: він доводить, що пакунок справді від вас і ніхто його не змінював. Сторінка «Ключі» показує ваші ключі та те, який із них підписує.

Параметр	Варіанти
Ім'я, електронна пошта, коментар	Кому належить ключ
Тип ключа	Ed25519 (сучасний, швидкий, рекомендований) або RSA 4096 / 3072 / 2048 (для старіших систем)
Термін дії	Безстроково, 1 рік, 2 роки або 5 років
Парольна фраза	Необов'язково. Якщо її задано, GPG питатиме її під час кожного підпису — так безпечніше

- «Використовувати для підпису» робить ключ активним ключем підпису.
- «Експортувати відкритий ключ» зберігає файл, за яким ваші користувачі перевіряють пакунки.
- Відкриті частини ваших ключів можна зберегти на Google Диску. Закриті ключі ніколи не залишають цей комп'ютер.
- Видалення не можна скасувати: цим ключем більше не вдасться підписувати пакунки.

5.3 Версії

«Версії» зберігає кожен випуск проєкту разом із його пакунками, найновіший — першим. Сторінка має дві вкладки: «Нова версія» і «Бібліотека версій».

- «Нова версія» збирає випуск із поточного стану файлів проєкту: номер версії, тип випуску (Стабільна, Кандидат у випуск, Бета, Альфа), номер попереднього випуску, формати (.deb, .rpm), підпис і примітки до випуску. Попередній перегляд показує і назву, яку читають люди («1.6 Beta 2»), і версію, яку порівнюють менеджери пакунків («1.6~beta2»).
- «Бібліотека версій» групує випуски за серіями (1.6, 2.0...) і позначає найновіший. Кожен випуск показує свої пакунки, підпис і примітки.
- «Переглянути код» відкриває код будь-якого випуску — прочитаний просто з його власного пакунка, точно таким, яким його випущено, лише для читання. Файли можна фільтрувати, а їхній вміст — копіювати.
- Старі файли .deb або .rpm можна перетягнути на сторінку, щоб додати випуск до історії; оригінали лишаються на місці.
- «Змінити примітки», «Додати .deb до репозиторію» і «Вилучити версію» є в кожного випуску. Вилучена версія потрапляє до кошика версій, і її можна повернути.

Випуски зберігаються в теці проєкту `versions`. Ця тека не потрапляє до жодного пакунка, знімка, статистики чи git-історії, тож жоден пакунок не містить попередніх.

5.4 APT-репозиторій

Репозиторій перетворює окремі пакунки на джерело програм, яке користувачі додають один раз і далі отримують із нього оновлення, як із будь-якого іншого. Petascore будує дерево, підписує індекс і підтримує його відповідність вашим пакункам.

Параметр	Значення
Назва, тека репозиторію	Як він називається й де будується дерево
Публічна адреса	URL, з якої встановлюють користувачі, наприклад <code>https://repo.example.com</code>
Suite, компонент, архітектури	Звичні значення: <code>stable</code> , <code>main</code> , <code>amd64</code> , <code>arm64</code>

Параметр	Значення
Origin, Label, опис	Як репозиторій представляється apt
Ключ підпису	apt відмовляється від архіву з непідписаним індексом
Спосіб публікації	Не звідси, копіювання в теку або надсилання через SSH (rsync)

- Натисніть «Додати пакунки» або перетягніть файли .deb на сторінку. Якщо ввімкнено «Додавати зібрані пакунки до репозиторію», кожен новий .deb потрапляє туди автоматично.
- «Перезібрати й підписати» оновлює індекс; попередження з'являється щоразу, коли він перестав відповідати пакунам.
- «Опублікувати» узгоджує копію на сервері з текою; «Перевірити сервер» порівнює їх.
- Позначки й фільтри показують пакунки, що не внесені до індексу, не підписані, не опубліковані або застарілі.
- Для кожного пакунка є «Перевірити підпис», «Копіювати команду встановлення», «Перейменувати», «Відкрити теку з файлом» і «Вилучити з репозиторію».
- Вилучені пакунки потрапляють до кошика репозиторію, і їх можна повернути; «Вилучити старі версії» залишає лише найновішу версію кожного пакунка.
- «Копіювати інструкції для користувачів» і «Експортувати ключ архіву» дають користувачам усе потрібне, щоб додати репозиторій.

5.5 Репозиторій на сервері

На вкладці «Сервер» Petacore керує текою репозиторію просто на вашому сервері через SSH. Файли вивантажуються через scp; індекс будується на сервері, а підписується на вашому комп'ютері, тож ключ підпису ніколи його не залишає.

1. Вкажіть назву або IP сервера, порт, користувача й теку репозиторію на сервері, наприклад /var/www/hero.
2. Виберіть SSH-ключ або створіть новий для Petacore: Ed25519, захищений паролем фразою щонайменше з 8 символів, яка ніколи не зберігається.
3. Додайте рядок, який показує Petacore, до ~/.ssh/authorized_keys цього користувача на сервері.
4. Під'єднайтеся. Першого разу порівняйте відбиток сервера з тим, який повідомляє сам сервер, і підтверджуйте лише за збігу.

Після під'єднання можна вивантажувати файли (пакунки .deb автоматично потрапляють до пулу), створювати теки, перейменовувати, завантажувати, переміщувати елементи до кошика сервера й повертати їх, а також виконувати «Перезібрати й підписати».

Захист	Що це означає
Перевірка ключа сервера	Ідентичність сервера перевіряється під час кожного з'єднання; якщо вона змінилася, Petacore відмовляється під'єднуватися
Вхід лише за ключем	Паролі ніколи не надсилаються
Сучасна криптографія	Постквантовий обмін ключами, де він доступний, і автентифіковане шифрування
Двофакторний вхід	Коди з застосунків-автентифікаторів і ключі безпеки FIDO2
Нічого не зберігається	Парольні фрази й двофакторні коди ніколи не записуються

6. Онлайн-сервіси

6.1 Під'єднання GitHub

1. Натисніть «Під'єднати через GitHub» — у Налаштуваннях, у помічнику першого запуску або будь-коли, коли потрібен GitHub.
2. У браузері відкривається GitHub, а одноразовий код уже скопійовано в буфер обміну.
3. Вставте код, підтвердьте й натисніть Authorize. Petacore під'єднається сам.

Якщо GitHub CLI (gh) на комп'ютері вже має виконаний вхід, достатньо одного натискання. Якщо gh немає, Petacore запропонує його встановити. В «Інших способах входу» можна також вручну вставити персональний токен доступу. Дані входу зберігаються у в'язці ключів стільниці — ніколи у файлі чи в адресі.

6.2 Завантажити на GitHub

Кнопка GitHub угорі вікна відкриває «Завантажити на GitHub». За один крок код потрапляє до гілки main і стає новою версією на сторінці проєкту на GitHub.

- Виберіть версію із запропонованих — наприклад, 1.0 Beta 3, 1.0 Beta 2.1, 1.0 RC 1 чи 1.0 — або введіть власну. Версія, яка вже є в бібліотеці, але ще не на GitHub, пропонується першою.
- Тег утворюється з версії (v1.0-beta.3); бета-, альфа-версії та кандидати у випуск позначаються як попередні випуски.
- Якщо в проєкті ще немає репозиторію на GitHub, його буде створено — типово приватним.
- Гілка — main, якщо ви її не змініте. Примітки до випуску необов'язкові; якщо лишити їх порожніми, GitHub складе їх із комітів.
- Пакунки .deb і .rpm цієї версії з Бібліотеки версій додаються до випуску.
- Вимкніть «Опублікувати як нову версію», щоб надіслати лише код.

Файли з обліковими даними ніколи не надсилаються. Те, що вже є на GitHub, зберігається: якщо воно конфліктує з вашим, надсилання зупиняється й нічого не перезаписується. Уже опубліковану версію ніколи не буде замінено.

6.3 Клонування з GitHub

Коли проєкт створюється за адресою на GitHub, Petacore один раз запитує, чи перевірити завантажений код через VirusTotal. «Перевірити» відкриває сторінку «Перевірка на віруси» й одразу запускає перевірку.

6.4 Google Диск

Під'єднайте власний Google Диск у Налаштуваннях: відкриється браузер, ви ввійдете в Google і натиснете «Дозволити». Petacore створить теку «Petacore Petacore™» і зберігатиме кожен проєкт в окремій підтеці, разом із медіафайлами. Власні записи Petacore лежать поруч, у прихованій теці .petacore.

- Кнопка Диска надсилає активний проєкт. Диск приводиться у відповідність до цього комп'ютера — і нічого не губиться (див. 6.5).
- Якщо після останньої синхронізації цього комп'ютера інший надіслав новішу копію, Petacore запропонує спершу отримати її.
- Якщо вивантаження виявиться меншим за вміст Диска, Petacore спершу попередить (це можна вимкнути в Налаштуваннях).
- «Отримати проєкти з Диска» завантажує в ~/Projects проєкти, які є лише на Диску.

- Зашифровані проєкти надсилаються через шар шифрування; Диск бачить лише нечитабельні дані.

6.5 Резервні копії

Під час синхронізації з Диском Petacore ніколи не видаляє файлів. Кожен файл, який синхронізація замінює або прибирає, спершу зберігається як резервна копія — на Диску під час надсилання, на цьому комп'ютері під час отримання копії.

- Резервні копії впорядковано за днями: коли, з якого комп'ютера, скільки файлів і якого розміру.
- Відкрийте резервну копію, щоб побачити її файли. «Повернути» відновлює всю копію або окремий файл.
- Перш ніж щось повернути, поточні файли самі зберігаються як резервна копія, тож і це можна скасувати.
- Типово резервні копії зберігаються 30 днів (у Налаштуваннях: 7, 30, 90, 365 днів або безстроково), після чого видаляються автоматично.

Резервні копії відкриваються кнопкою з годинником на сторінці «Проект», кнопкою «Резервні копії» в сповіщеннях про синхронізацію та через Налаштування → Google Диск → Показати резервні копії.

6.6 Зміни з інших ваших комп'ютерів

Комп'ютери з тим самим Google Диском самі отримують найновішу копію одне одного: за кілька секунд після запуску, а далі кожні 10 хвилин (у Налаштуваннях: 5, 10, 30 або 60). Проєкти, які є лише на Диску, теж завантажуються — у нову теку, якщо назву вже зайнято.

Ситуація	Що робить Petacore
На Диску новіша копія, тут змін немає	Тихо отримує її й повідомляє, який комп'ютер її надіслав
На Диску новіша копія, і тут є зміни	Запитує: Узяти копію з Диска, Надіслати мою чи Не зараз
Зашифрований проєкт	Не отримується автоматично — потрібен пароль

Хоч би що ви обрали, файли іншої сторони збережуться в резервних копіях. Функцію типово ввімкнено; вимкнути її можна в Налаштування → Google Диск.

6.7 Перевірка на віруси

«Перевірка на віруси» дає рушіям VirusTotal перевірити проєкт, перш ніж його встановлять ваші користувачі. API-ключ належить вам: створіть безкоштовний обліковий запис на [virustotal.com](https://www.virustotal.com), скопіюйте ключ зі свого профілю й один раз введіть його. Він зберігається у в'язці ключів стільниці.

- «Перевірити проєкт» пакує проєкт в один архів; незмінений проєкт розпізнається за відбитком і ніколи не надсилається вдруге.
- Нічого не надсилається без вашої згоди — її запитують один раз для кожного проєкту. Файли з обліковими даними, .git, node_modules і тека versions не надсилаються ніколи.
- Результати: нічого не виявлено; «п із т позначили» з назвами рушіїв; або «ще не бачили» — звична відповідь для щойно зібраного файлу.
- Якщо щойно зібрану непідписану програму позначили один-два рушії, найчастіше це хибне спрацювання.
- Безкоштовний ключ дає змогу виконувати 4 перевірки на хвилину й 500 на день.

7. Налаштування

Параметр	Типово	Опис
Мова	Мова стільниці	Шість мов; щоб застосувати, Petacore перезапускається
Вигляд	Як у системі	Світлий, темний або як у стільниці
Режим фокусування	Увімкнено	Притлумлює рядки поза вибраним розділом
Попередження про менше вивантаження	Увімкнено	Попереджає, якщо вивантаження менше за копію на Диску
Інтернет у Пісочниці	Вимкнено	Дає коду в Пісочниці доступ до мережі
Питати перед виходом через Esc	Увімкнено	Підтвердження для незбережених файлів
Автозбереження	Увімкнено, 10 хвилин	Автоматичні знімки
GitHub	Не під'єднано	Під'єднати, від'єднати, необов'язковий OAuth Client ID
Google Диск	Не під'єднано	Під'єднати, від'єднати, отримати проєкти
Отримувати зміни	Увімкнено, 10 хвилин	Зміни з інших ваших комп'ютерів
Зберігати резервні копії	30 днів	7, 30, 90, 365 днів або безстроково
Інтерфейс	Визначено автоматично	GNOME або KDE Plasma

8. Безпека та конфіденційність

- Дані входу й API-ключі зберігаються у в'язці ключів стільниці. Якщо її немає, вони записуються у файл, який можете прочитати лише ви.
- Файли з обліковими даними (.env, закриті ключі, .netrc...) ніколи не потрапляють до пакунка, на GitHub чи на перевірку VirusTotal.
- Дані входу GitHub передаються лише github.com; схожа на нього адреса нічого не отримає.
- Закриті ключі підпису ніколи не залишають ваш комп'ютер; резервуються лише відкриті ключі.
- Вхід на сервери — лише за ключем; паролні фрази й двофакторні коди ніколи не зберігаються.
- Пісочниця не має мережі, доки ви її не відкриєте; Живий перегляд відповідає лише цьому комп'ютеру, доки ви не відкриєте його в мережі.
- Без вашої згоди нічого не надсилається до VirusTotal.
- Синхронізація ніколи не знищує файлів: усе, що вона замінює, зберігається в резервних копіях.

9. Усунення несправностей

Проблема	Розв'язання
Для Google Диска потрібен rclone	<code>sudo apt install rclone</code>
В'язка ключів недоступна	<code>sudo apt install libsecret-tools</code> , потім спробуйте ще раз
GitHub CLI не готовий	Натисніть «Встановити gh» або <code>sudo apt install gh</code>
Пісочниця: HE ізольовано	<code>sudo apt install bubblewrap</code>
Картку збережено лише як SVG	<code>sudo apt install librsvg2-bin</code>
Живий перегляд: немає вбудованого браузера	<code>sudo apt install gir1.2-webkit-6.0</code> або «Відкрити в браузері»
Не вдається зібрати індекс репозиторію	<code>sudo apt install apt-utils dpkg-dev</code>
З'єднання з сервером: бракує OpenSSH	<code>sudo apt install openssh-client</code>
VirusTotal відхилив ключ	Перевірте ключ у профілі VirusTotal; блокування мережі може бути й тимчасовим — спробуйте пізніше
Ключ підпису прострочений або відсутній	Створіть новий ключ на сторінці «Ключі» й виберіть «Використовувати для підпису»
Ідентичність сервера змінилася	Не під'єднуйтеся. Зверніться до адміністратора сервера; забувайте старий ключ лише після підтвердження зміни
GitHub: у гілці є несумісні зміни	Отримайте ці зміни через git у Терміналі, розв'яжіть конфлікти й надішліть знову
Версію вже опубліковано на GitHub	Виберіть іншу версію
Проект змінено у двох місцях	Виберіть, яку копію залишити; інша сторона збережеться в резервних копіях
Неправильний пароль зашифрованого проєкту	Пароль неможливо відновити; введіть правильний
Бракує компонентів іншого інтерфейсу	Натисніть «Установити зараз» або виконайте команду, яку показує Petacore

10. Комбінації клавіш

Клавіші	Дія
Ctrl+S	Зберегти файл (Редактор)
Ctrl+Q	Закрити файл (Редактор)
Ctrl+F	Пошук; Enter — до наступного збігу, Esc — закрити
Ctrl+K / Ctrl+D	Вирізати рядок / продублювати рядок

Клавiші	Дiя
Ctrl+Z / Ctrl+Y	Скасувати / повторити
Esc	Закрити файл; повторний Esc — закрити без збереження
Ctrl+A / Delete	Виділити все / видалити (сторінка «Проект», можна скасувати)
Ctrl+Z	Повернути видалені файли чи вилучені пакунки
Ctrl + вказівник	Швидкий перегляд; Ctrl+коліщатко — прокручування
F11 / Esc	Живий перегляд на весь екран / вийти з повноекранного режиму

11. Файли та вилучення програми

11.1 Де Petacore зберігає дані

Розташування	Вміст
~/local/share/petacore-app	Сама програма
~/config/petacore	Налаштування (доступні для читання лише вам)
~/local/share/petacore/snapshots	Знімки
~/local/share/petacore/drive-backups	Резервні копії файлів, замінені на цьому комп'ютері
<проект>/versions	Історія випусків (поза пакунками й git)
<проект>/petacore	Плани з «Наступних оновлень»
В'язка ключів стільниці	Дані входу GitHub, ключ VirusTotal

11.2 Вилучення програми

Запустіть `./uninstall.sh` у теці Petacore від власного імені. Вилучення програми та видалення ваших знімків і резервних копій — це два окремі запитання, і нічого не видається, доки ви не підтвердите. Ваших тек із проектами скрипт ніколи не торкається.

12. Ліцензія та авторські права

© Petacomm I.S. 2026. Усі права захищено. Petacomm Petacore™ — торговельна марка Petacomm I.S.

Petacore використовує вільні компоненти з відкритим кодом, які лишаються власністю своїх авторів і використовуються на умовах їхніх ліцензій. GitHub, Google Drive, VirusTotal, GNOME, KDE Plasma, Ubuntu і Debian — торговельні марки відповідних власників.

Цей посібник описує Petacore 1.1 Beta 2. У наступних версіях можливості можуть змінитися.

petacore

PETACOMM PETACORE™ · 1.1 BETA 2