

# Privates PowerShell-Ökosystem

vollwertig skripten hinter hohen Mauern

PowerShell User Group Berlin | 08.09.2021

## Get-SpeakerInfo -Brief

- **Name:** Evgenij Smirnov
- **YearOfBirth:** 1972
- **JobTitle:** { Consultant, Project Mgr. }
- **TwitterID:** @cj\_berlin
- **Employer:** SVA System Vertrieb Alexander GmbH
- **MVP:** { CDM 2020-2022 }
- **Certifications:** { MCSE, MCSA, VCP, VCAP, VCIX, CCA, QCIC }
- **UserGroups:** { WSUG-B, EXUSG, PSUGB, BerlinVMUG }



## Das übliche PowerShell-Umfeld...

- **Fremde Module: PS Gallery**
- **Eigene Module: Dateiablage**
- **Eigene Skripte: Dateiablage**
- **Versionskontrolle: GitHub (oder keine)**
- **IDE:**
  - PowerShell ISE
  - VS Code + PowerShell Extension → Installation manuell\*, Updates automatisch 😊
- **Shell:**
  - Windows PowerShell
  - PowerShell → Installation manuell\*, Updates von Microsoft Update (Beta)
  - Windows Terminal → Installation und Updates aus dem Store (nur Windows 10)

**Doch was tun,  
wenn die Infrastruktur- und Dev-Maschinen  
nicht ins Internet dürfen?**

## Das isolierte PowerShell-Umfeld

- **Fremde Module: Eigenes Repository, extern eingebracht**
- **Eigene Module: Eigenes Repository**
- **Eigene Skripte: Eigenes Repository (oder Dateiablage natürlich ;-)**
- **Versionskontrolle: Lokale Git-Repos oder eine GitLab-/Gitea-Instanz**
- **IDE:**
  - PowerShell ISE
  - VS Code + PowerShell Extension → einmalig installiert, Updates manuell ☹️
- **Shell:**
  - Windows PowerShell
  - PowerShell → Installation und Updates manuell\*
  - Windows Terminal → Installation und Updates manuell\* mit Klimmzügen

## Eigenes Repo bauen

- **Man nehme:**
  - Nuget.exe (<https://www.nuget.org/downloads> )
  - Ordner „nuget“ aus C:\Program Files\PackageManagement\ProviderAssemblies
- **Man erzeuge:**
  - Eine Fileshare, auf die AuthenticatedUsers Leserecht und Skripter Schreibrecht haben
- **Man kopiere (auf den Zielmaschinen):**
  - Nuget.exe nach C:\ProgramData\Microsoft\Windows\PowerShell\PowerShellGet
  - Nuget-Ordner nach ProviderAssemblies
- **Hilfe: OfflinePowerShellGetDeploy by KeithBankston**
  - <https://www.powershellgallery.com/packages/OfflinePowerShellGetDeploy>

## Repo registrieren

- **Eigenes Repo registrieren:**

Register-PSRepository

- Name PrivateRepo
- InstallationPolicy Trusted
- SourceLocation [\\UGSV01.ug.metabpa.org\REPO](https://UGSV01.ug.metabpa.org/REPO)
- ScriptSourceLocation [\\UGSV01.ug.metabpa.org\REPO](https://UGSV01.ug.metabpa.org/REPO)

- **Für Developer zusätzlich:**

- PublishLocation [\\UGSV01.ug.metabpa.org\REPO](https://UGSV01.ug.metabpa.org/REPO)
- ScriptPublishLocation [\\UGSV01.ug.metabpa.org\REPO](https://UGSV01.ug.metabpa.org/REPO)

- **PSGallery deregistrieren (falls aus der Vergangenheit registriert):**

Unregister-PSRepository PSGallery

## Repo befüllen und benutzen

- **Regulär veröffentlichen (es werden nur Schreibrechte auf der Share benötigt)**

```
Publish-Module -Name <Name> -Repository PrivateRepo
```

```
Publish-Module -Path <Path> -Repository PrivateRepo
```

- Bei Modulen mit Abhängigkeiten lieber zuerst installieren 😊

- **Regulär ansprechen**

```
Find-Module
```

```
Install-Module
```

```
Save-Module
```

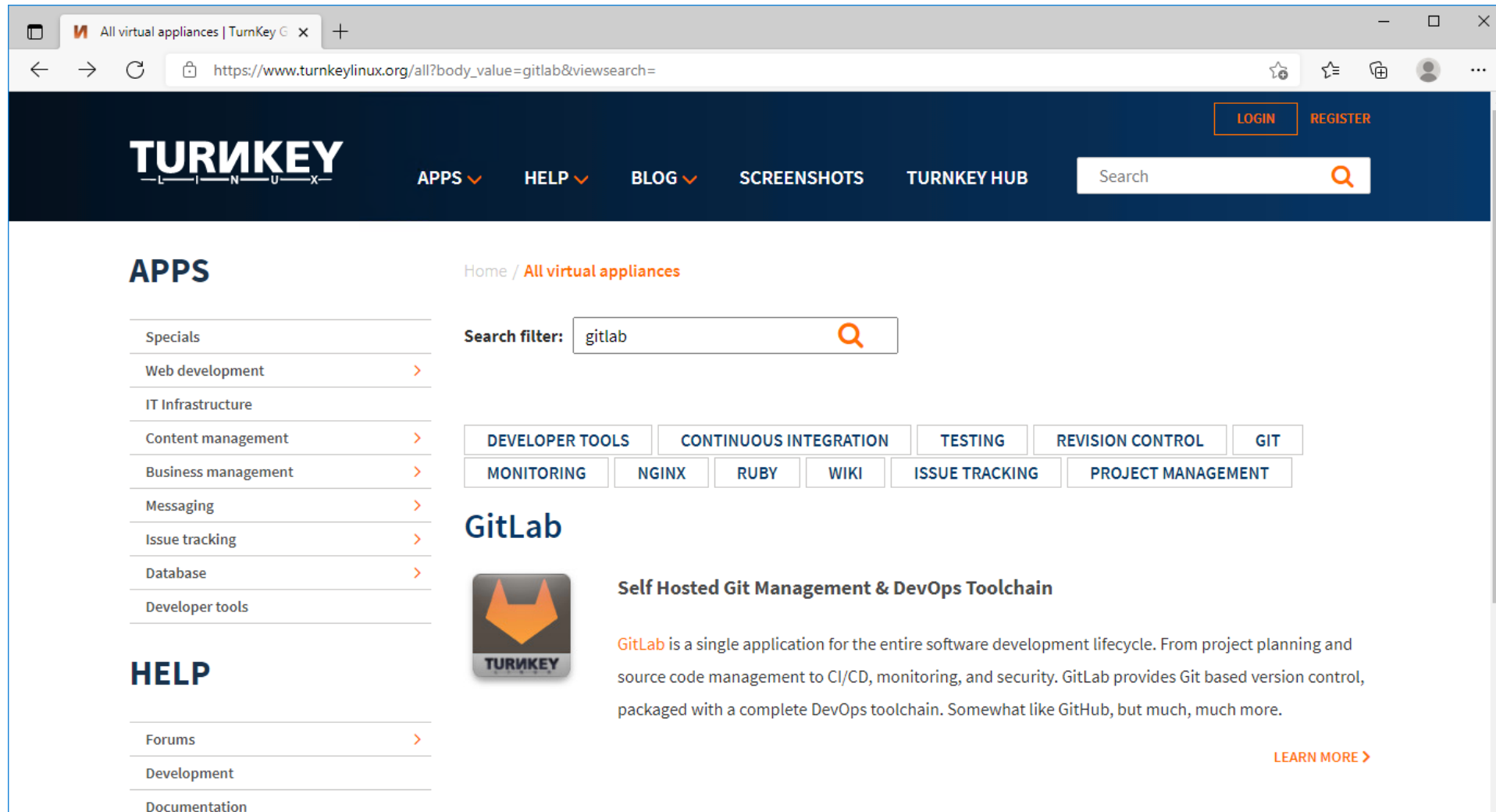
```
Update-Module
```



## Private Repo DeLuxe

- **NuGet Server:**  
<https://docs.microsoft.com/en-us/nuget/hosting-packages/nuget-server>
- **Muss kompiliert werden**
- **Kein Gallery Front-End**
- **Zugriff mit API-Keys, wie in der Gallery**
  - Diese werden manuell in die Web.Config geschrieben

## GitLab für Faule



The screenshot shows a web browser window displaying the TurnKey Linux website. The browser's address bar shows the URL `https://www.turnkeylinux.org/all?body_value=gitlab&viewsearch=`. The website's header features the TurnKey Linux logo, navigation links for APPS, HELP, BLOG, SCREENSHOTS, and TURNKEY HUB, and a search bar. The main content area is titled "APPS" and includes a search filter set to "gitlab". Below the search filter, there are several category buttons: DEVELOPER TOOLS, CONTINUOUS INTEGRATION, TESTING, REVISION CONTROL, GIT, MONITORING, NGINX, RUBY, WIKI, ISSUE TRACKING, and PROJECT MANAGEMENT. The GitLab application is highlighted, with a sub-header "Self Hosted Git Management & DevOps Toolchain" and a brief description: "GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security. GitLab provides Git based version control, packaged with a complete DevOps toolchain. Somewhat like GitHub, but much, much more." A "LEARN MORE" link is provided at the bottom right of the GitLab section. The left sidebar contains a list of application categories under "APPS" and a list of help resources under "HELP".

TurnKey Linux

APPS ▾ HELP ▾ BLOG ▾ SCREENSHOTS TURNKEY HUB

Search

Home / All virtual appliances

Search filter: gitlab

DEVELOPER TOOLS CONTINUOUS INTEGRATION TESTING REVISION CONTROL GIT

MONITORING NGINX RUBY WIKI ISSUE TRACKING PROJECT MANAGEMENT

## GitLab

**Self Hosted Git Management & DevOps Toolchain**

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security. GitLab provides Git based version control, packaged with a complete DevOps toolchain. Somewhat like GitHub, but much, much more.

[LEARN MORE >](#)

Specials

Web development >

IT Infrastructure

Content management >

Business management >

Messaging >

Issue tracking >

Database >

Developer tools

## HELP

Forums >

Development

Documentation

## GitLab für Faule

- **ISO = 1 GB**
- **Installation: 30 Minuten, davon 29,5 Minuten warten 😊**
- **Wer keine Angst vor einer Linux-Distro hat, kann auch eine eigene Appliance bauen**
  - ohne Internet-Zugang freilich schwierig
- **Wichtig: Auto DevOps global ausschalten!**

# Happy scripting!

Bei Fragen → einfach fragen 😊