

# Windows

- OpenSSH-Agent, Putty und SSH-Keys
  - OpenSSH-Agent Setup
  - Putty Setup
  - SSH-Key erstellen und einrichten
- Tweaks & Optimierung
  - Windows 11 Erinnerung deaktivieren
  - Netzwerktyp ändern
  - Bloatware loswerden

# OpenSSH-Agent, Putty und SSH-Keys

# OpenSSH-Agent Setup

Alle Befehle auf dieser Seite sind in einer Administrator Powershell auszuführen.

## 1. Setup SSH-Agent Windows Dienst

1. Mit dem Befehl kannst du überprüfen ob du den Dienst bereits installiert hast:

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

Wenn die Ausgabe so aussieht hast du bereits den notwendigen Dienst installiert und kannst bei der Installation von Putty fortfahren:

```
PS C:\Users\lukas_xc7sr14\.ssh> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'

Name : OpenSSH.Client~~~~0.0.1.0
State : Installed
```

2. Wenn bei "State : NotPresent" steht aktualisiere die optionalen Features mit diesem Befehl:

```
Get-AppXPackage -AllUsers -Name windows.immersivecontrolpanel | Foreach {Add-AppxPackage -
DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml" -Verbose}
```

3. Anschließend Installierst du den OpenSSH-Client mit dem Befehl:

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

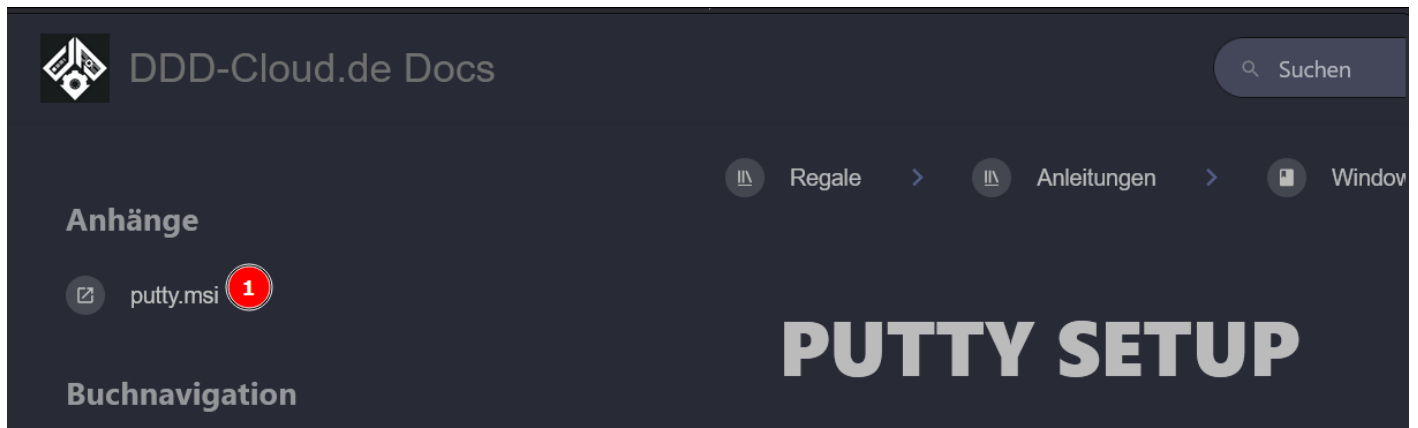
4. Als letztes noch den Dienst starten und für den Autostart beim Hochfahren eintragen:

```
Start-Service ssh-agent
Set-Service -Name ssh-agent -StartupType 'Automatic'
```

# Putty Setup

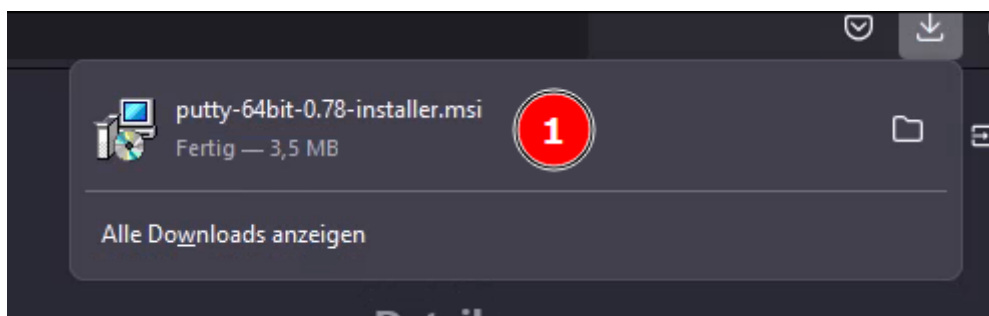
## 1. Putty Download

Klicke Links auf die Angehängte Datei und lade diese runter.

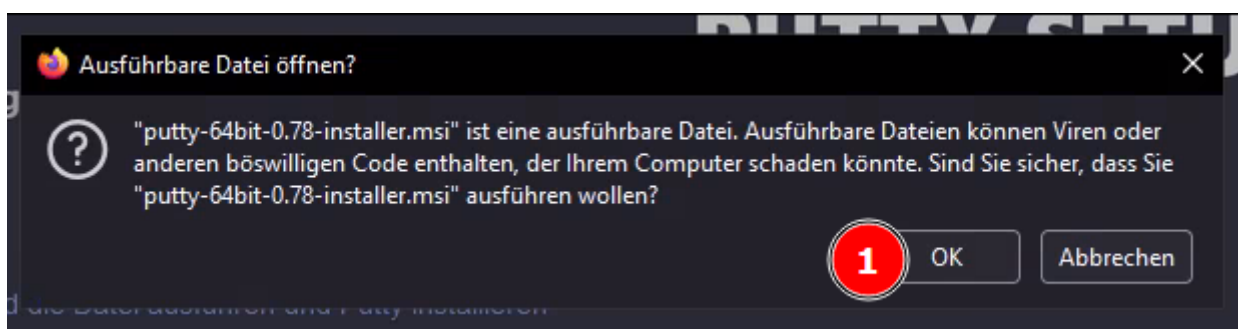


## 2. Installation

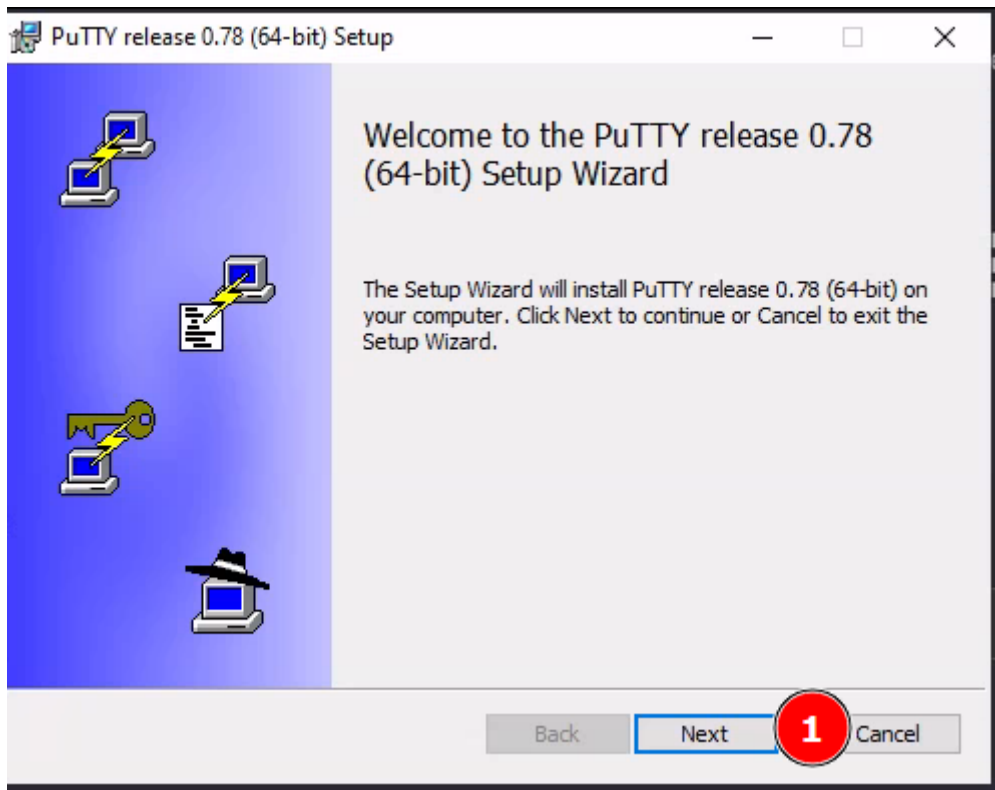
1. Doppelklicken zum Ausführen



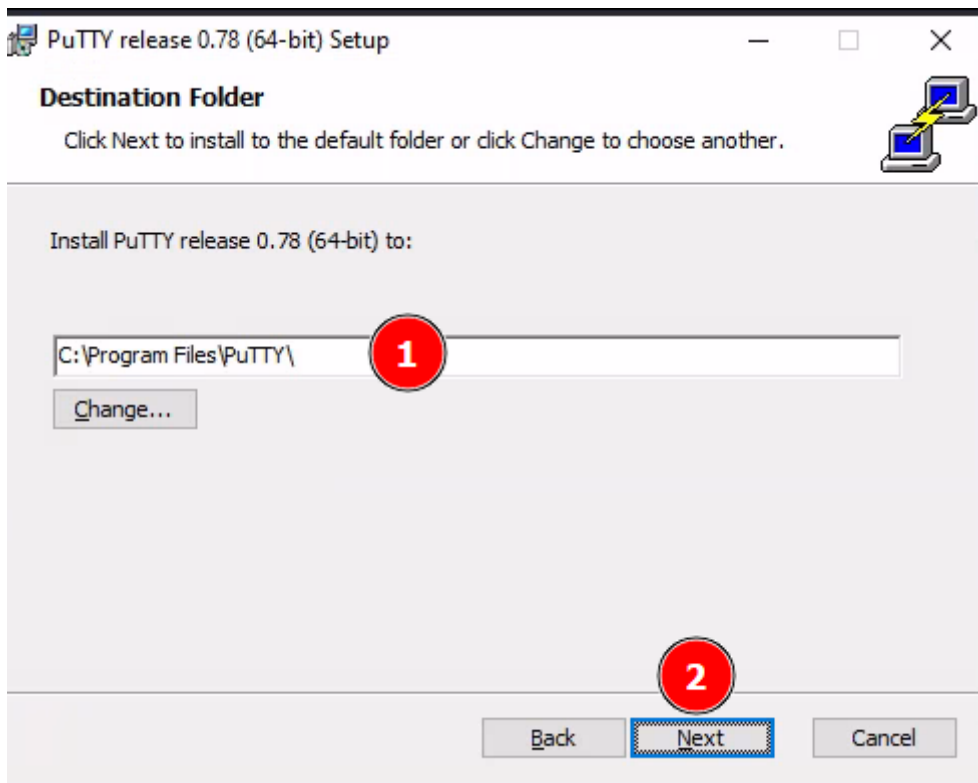
1. Dialog mit OK bestätigen



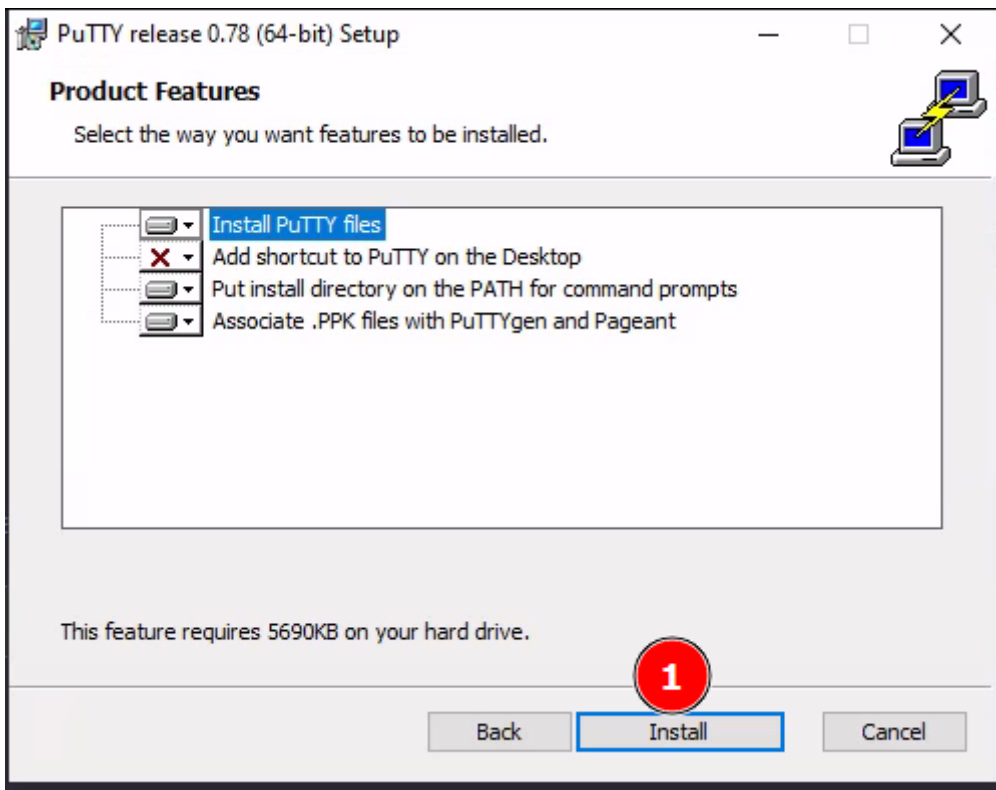
1. Bestätigen mit "Next"



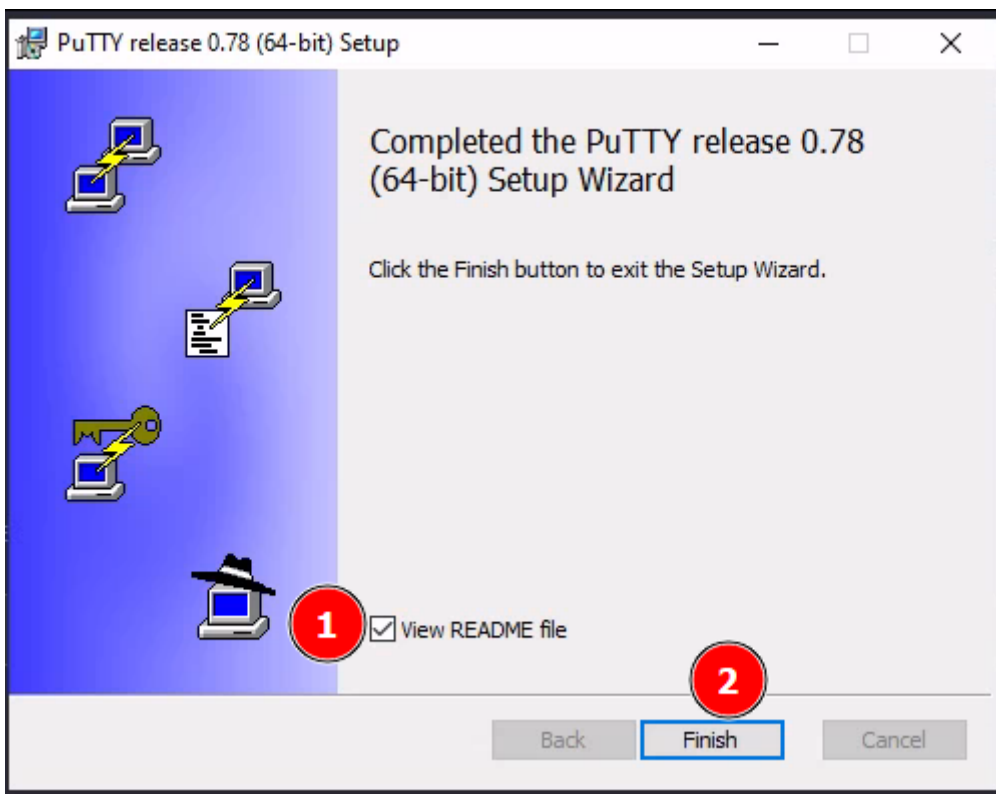
1. Pfad anpassen wenn gewünscht
2. Bestätigen mit "Next"



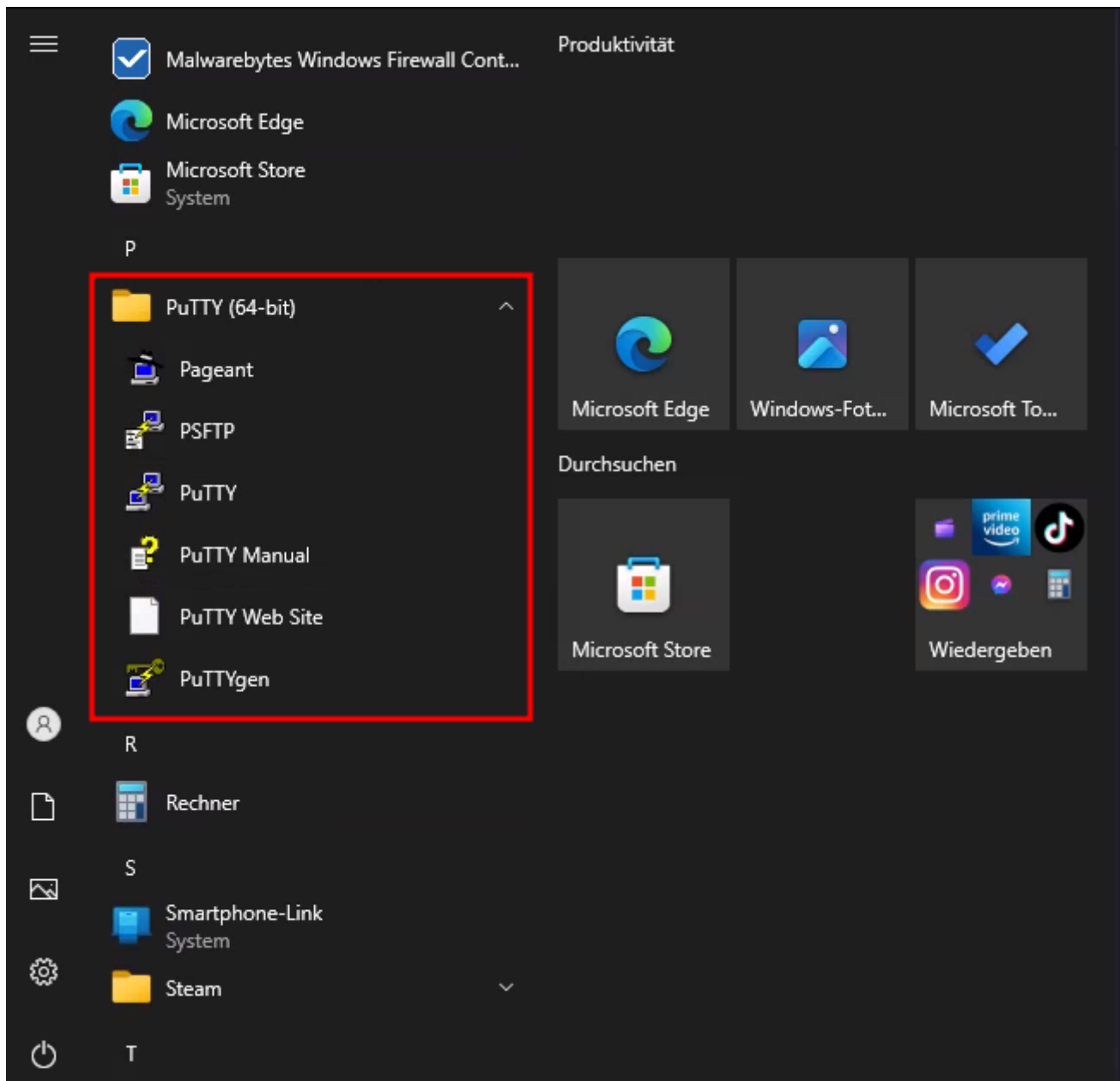
1. Installation mit "Install" starten



1. View Readme abwählen
2. Installation mit "Finish" abschließen



Wenn die Liste an Programmen gleich ist war die Installation erfolgreich

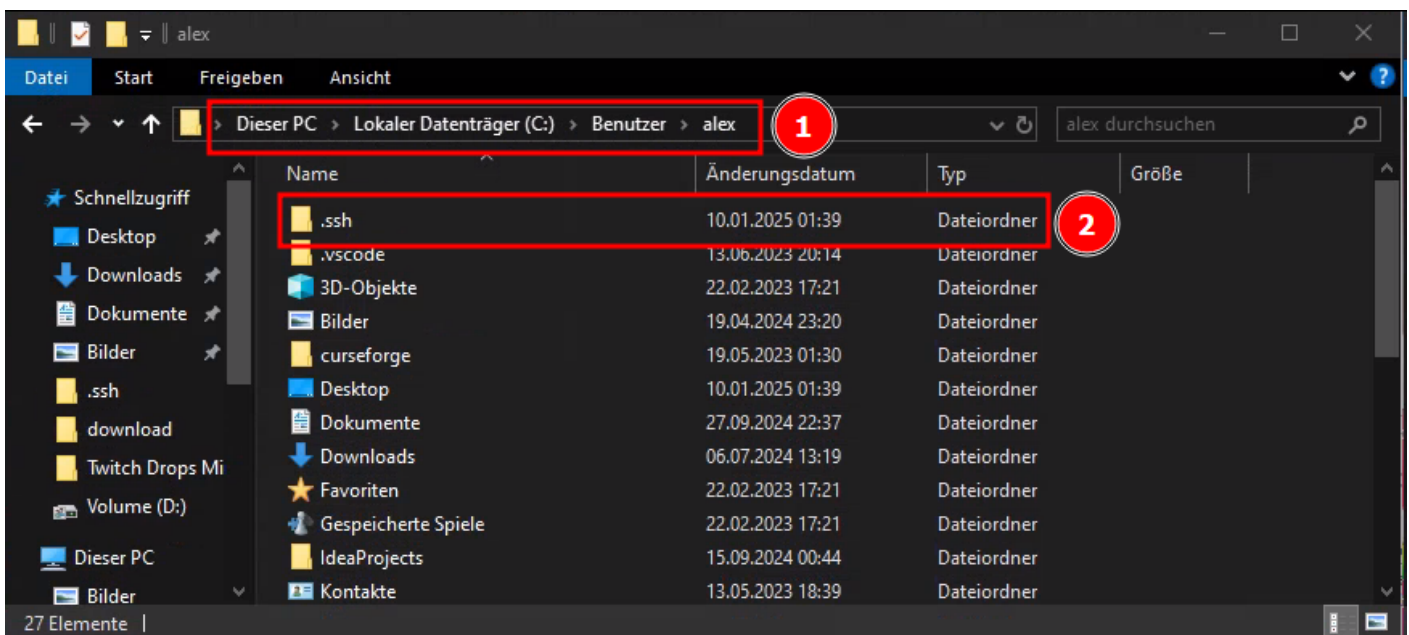


# SSH-Key erstellen und einrichten

Teil 1 und Teil 2 müssen vorher gemacht werden!

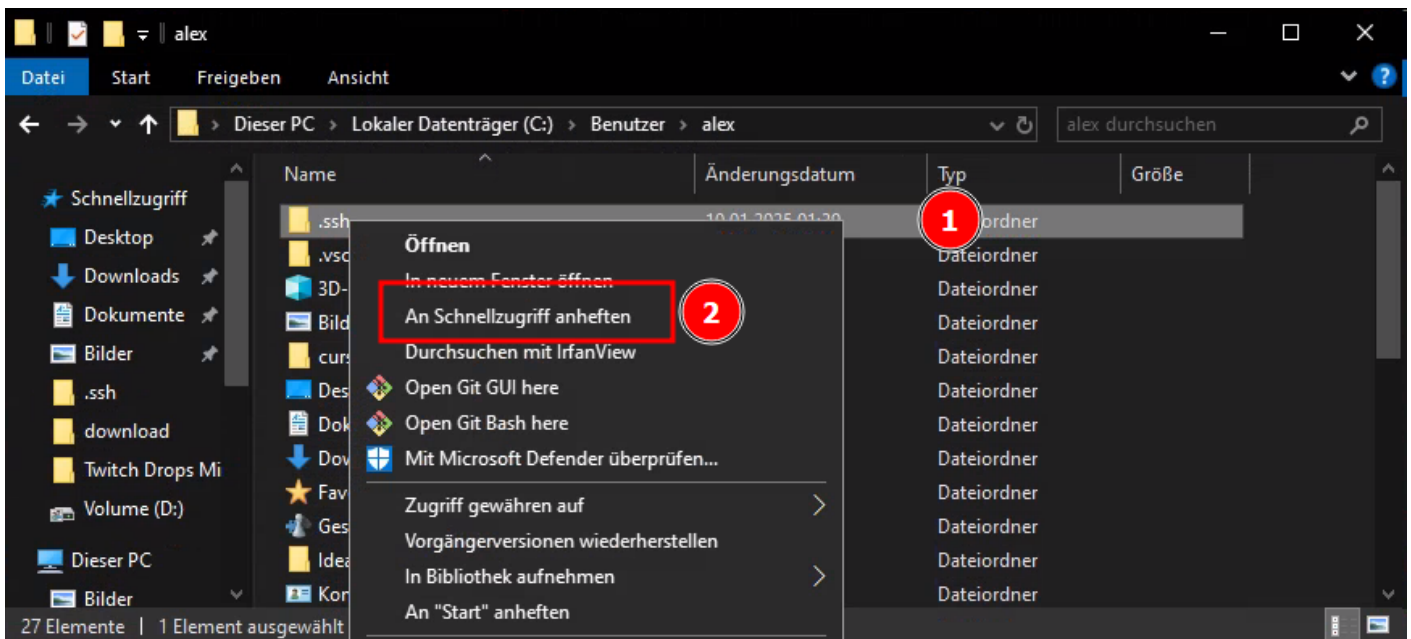
## 1. Vorbereitung

1. Geh in deinen Benutzer Ordner unter `C:\Users\DEIN-BENUTZER`
2. Lege den Ordner `.ssh` an



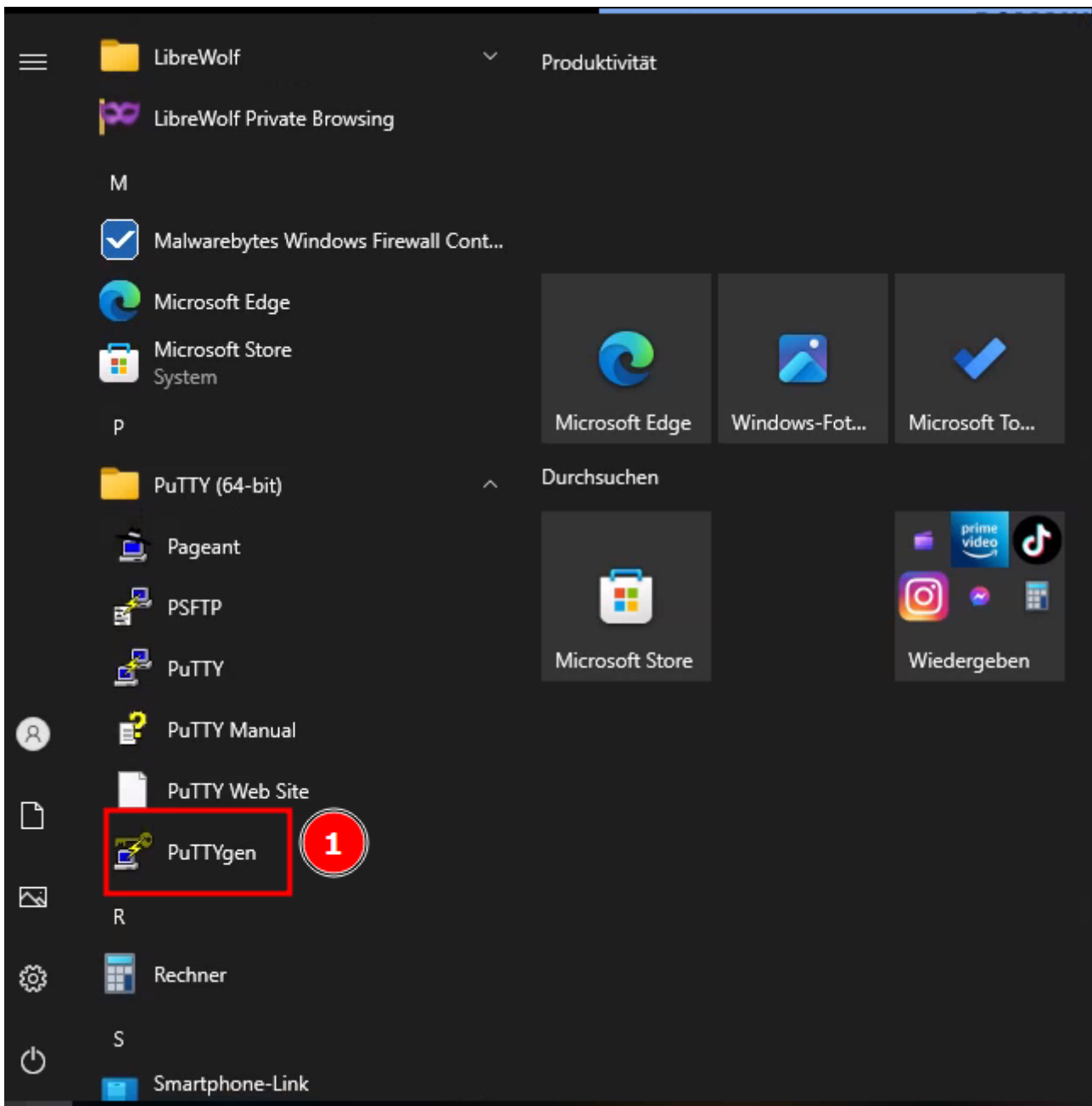
1. Anschließend den Ordner per Rechtsklick auswählen
2. Mit [An Schnellzugriff anheften](#) Links hinzufügen



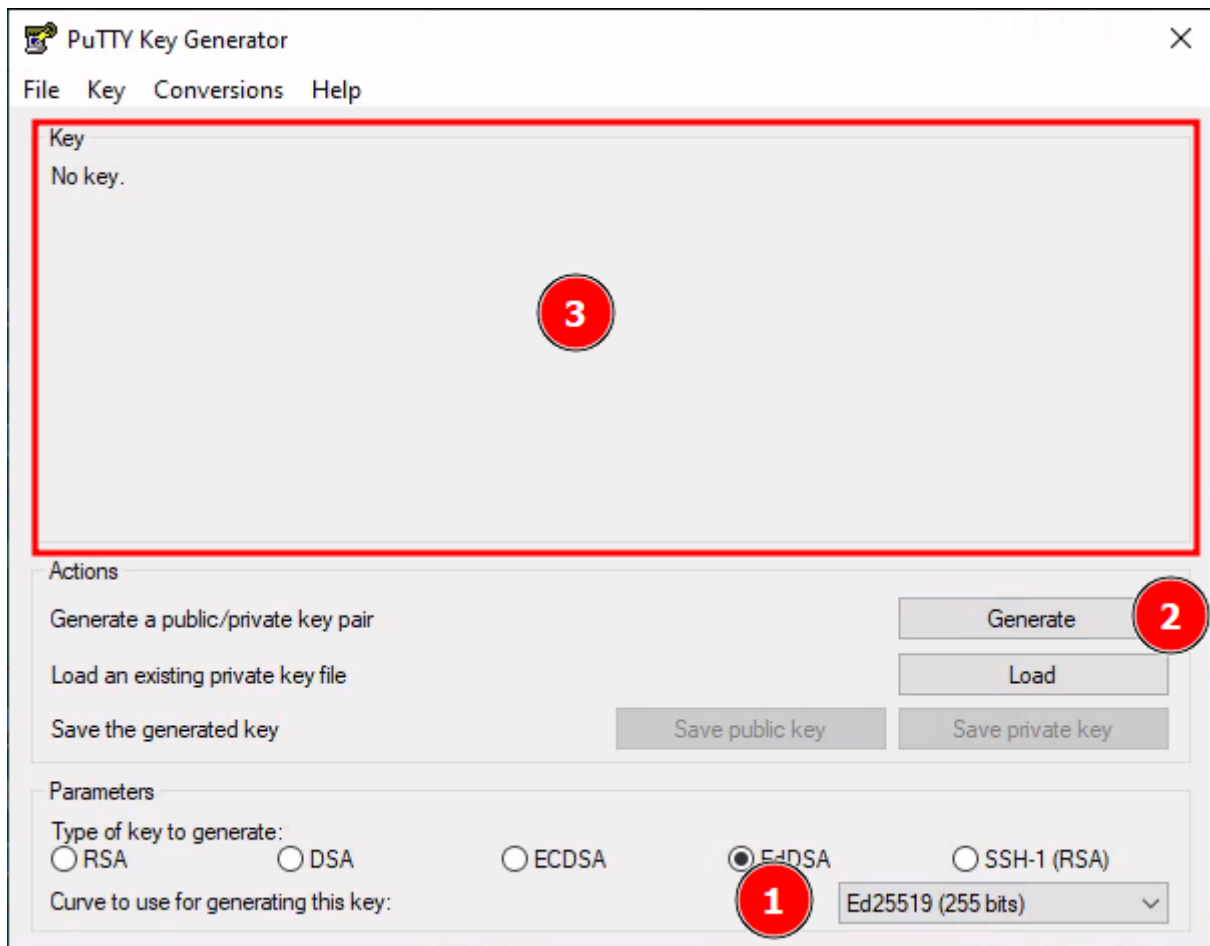


## 2. Schlüssel Erstellung

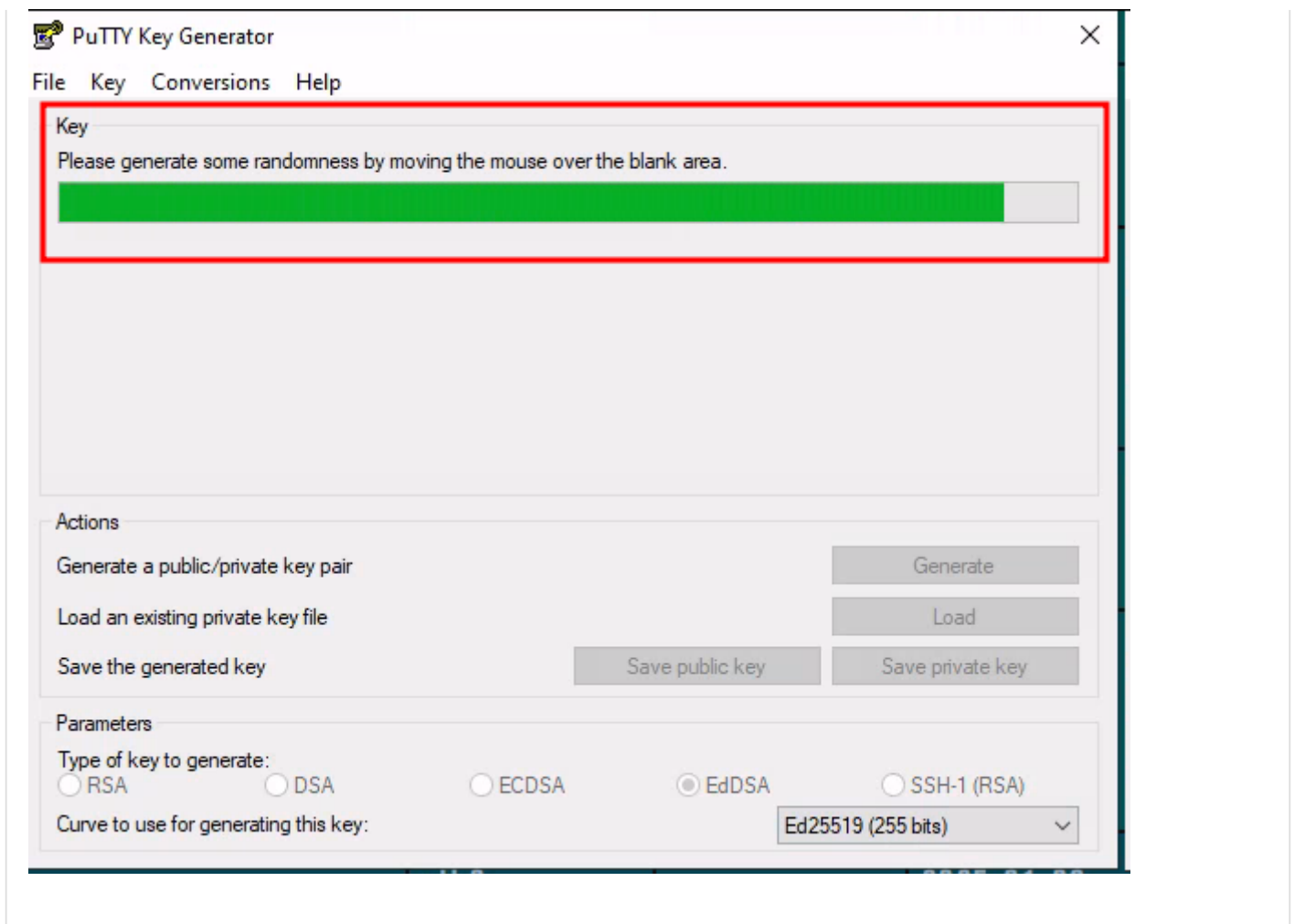
1. Dann wird PuTTYgen gestartet



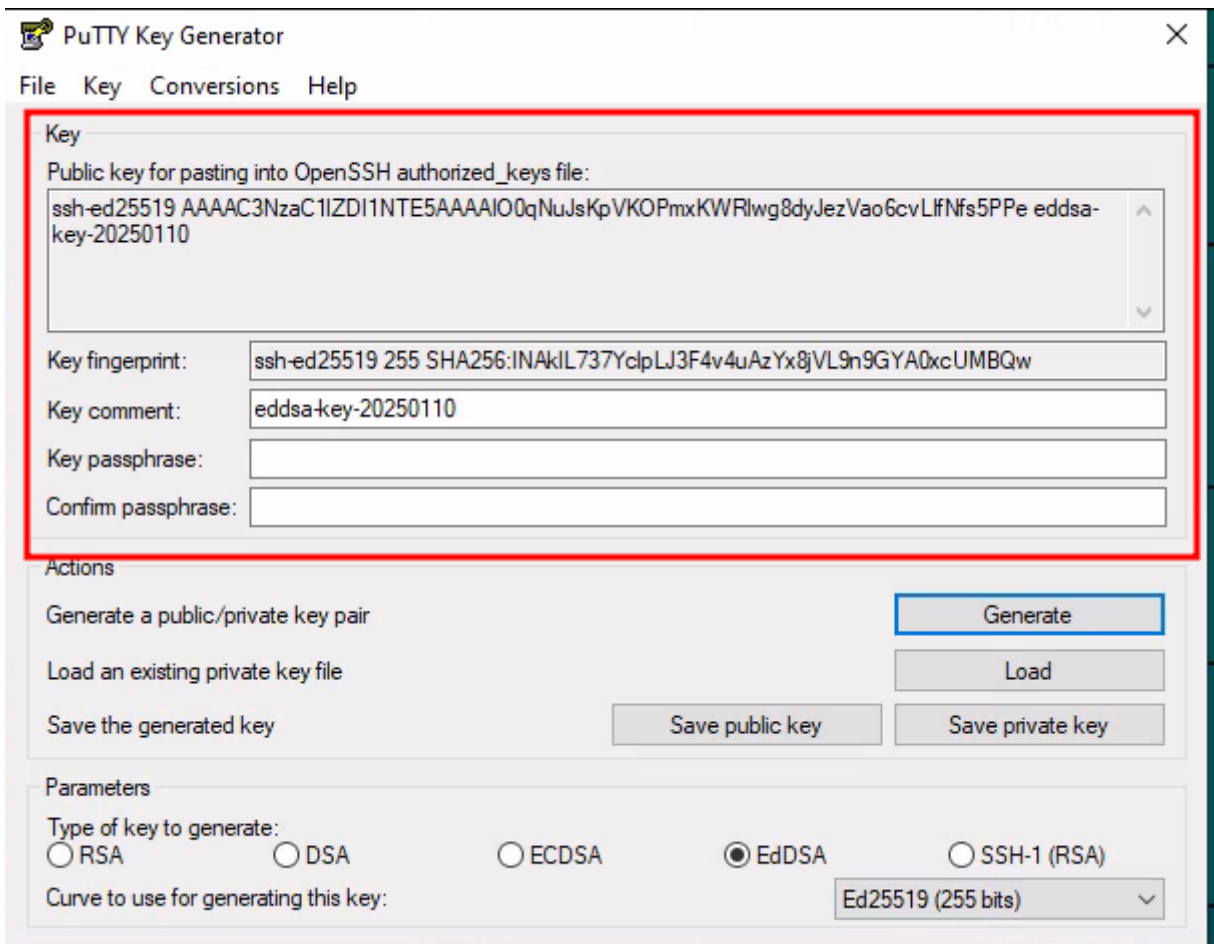
1. Dann wird auf den Schlüsseltyp **EdDSA** gestellt
2. Mit Generate starten wir die Generation
3. Maus in dem Feld solange bewegen bis Text erscheint



## Generationsfortschritt




Der Schlüssel ist fertig wenn das Fenster so aussieht:



Jetzt den erzeugten Schlüssel 3 mal speichern

## Speichern des Putty Private Keys

Auf "Save private key" klicken

 PuTTY Key Generator ×

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized\_keys file:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO0qNuJsKpVKOPmxKWRlwg8dyJezVao6cvLifNfs5PPe eddsa-key-20250110
```

Key fingerprint: ssh-ed25519 255 SHA256:INakIL737YclpLJ3F4v4uAzYx8jVL9n9GYA0xcUMBQw

Key comment: eddsa-key-20250110

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key **1**

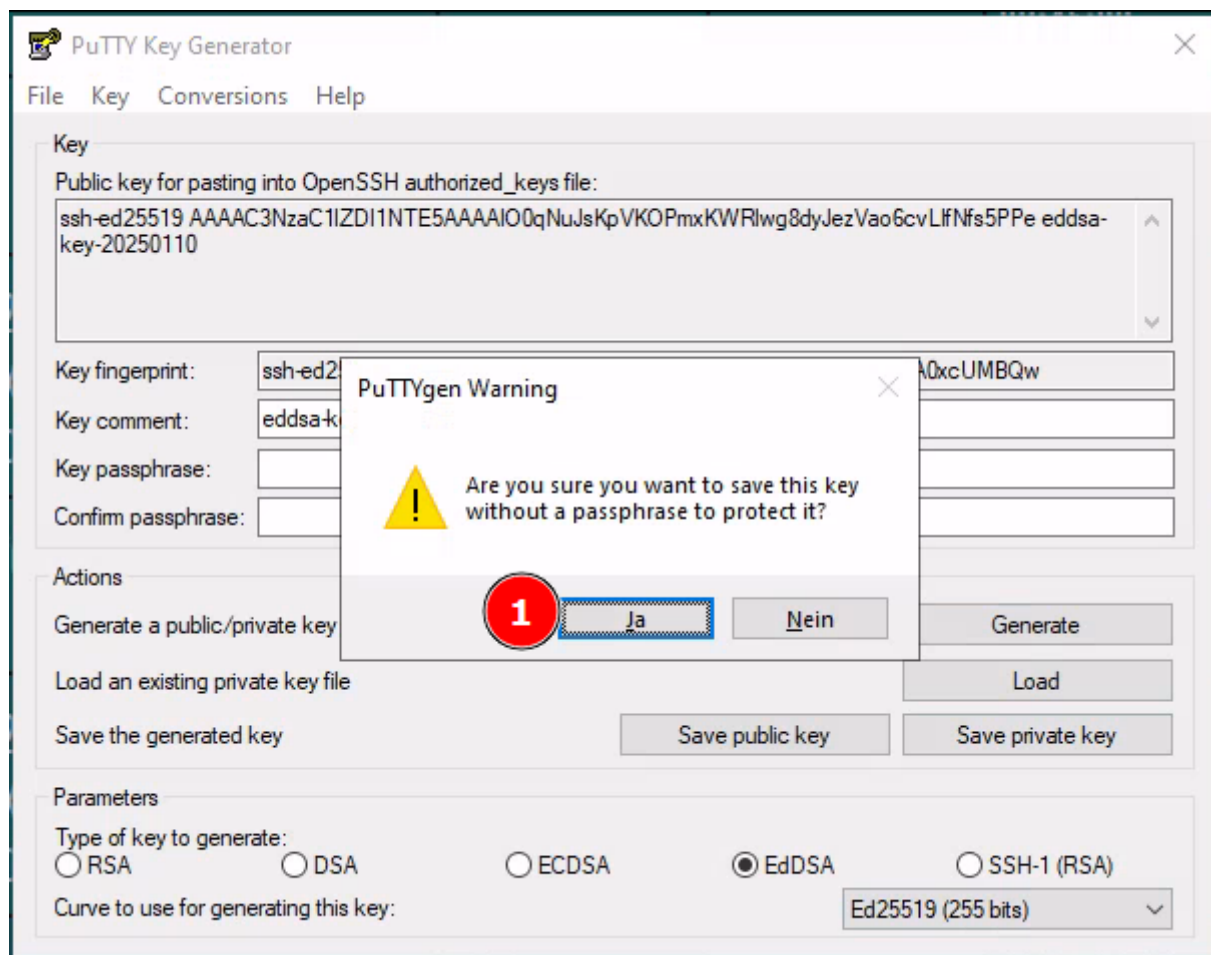
Parameters

Type of key to generate:

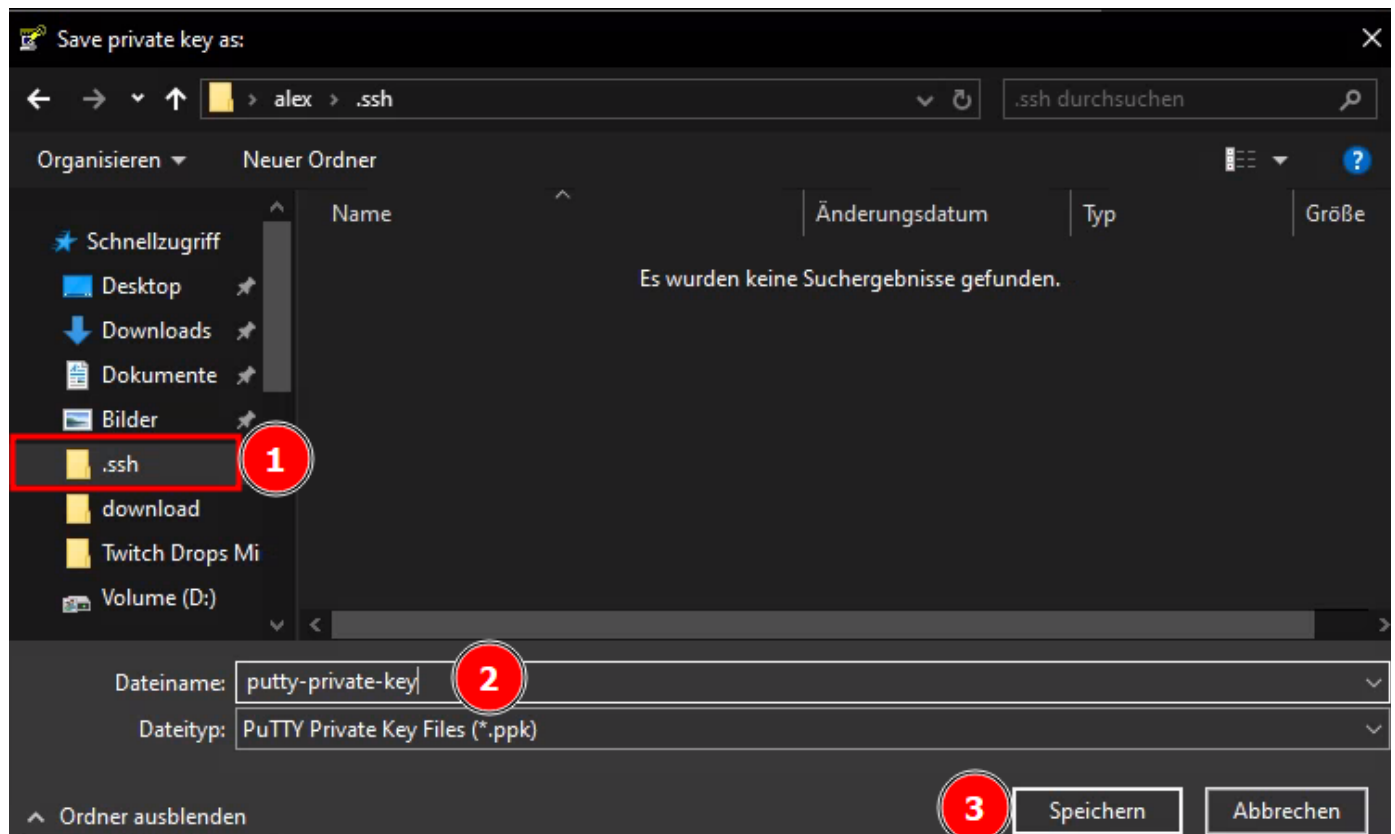
☐ RSA ☐ DSA ☐ ECDSA ☒ EdDSA ☐ SSH-1 (RSA)

Curve to use for generating this key: Ed25519 (255 bits) ▼

Dialog mit **Ja** bestätigen



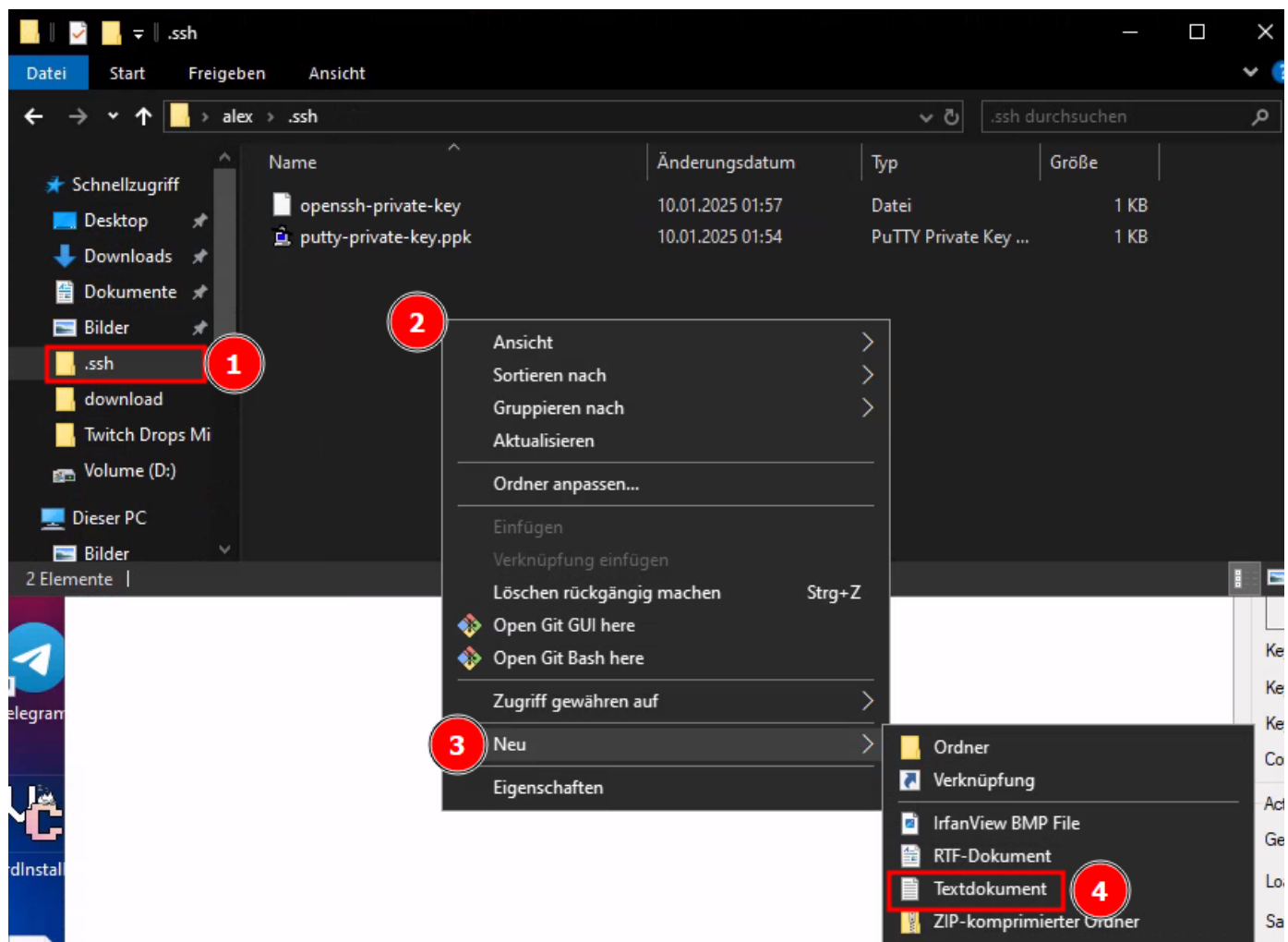
1. Auf `.ssh` klicken
2. Bei Dateinamen `putty-private-key` eintragen
3. Und auf `Speichern` klicken



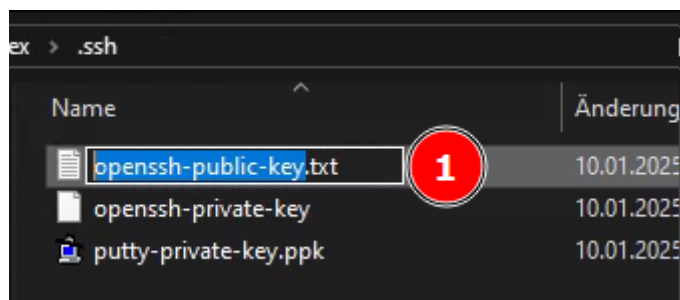
## Speichern des OpenSSH Public Keys

0. Den **Windows Explorer** öffnen
1. Auf **.ssh** klicken
2. Rechtsklick
3. Auf **Neu** gehen
4. **Textdokument** auswählen

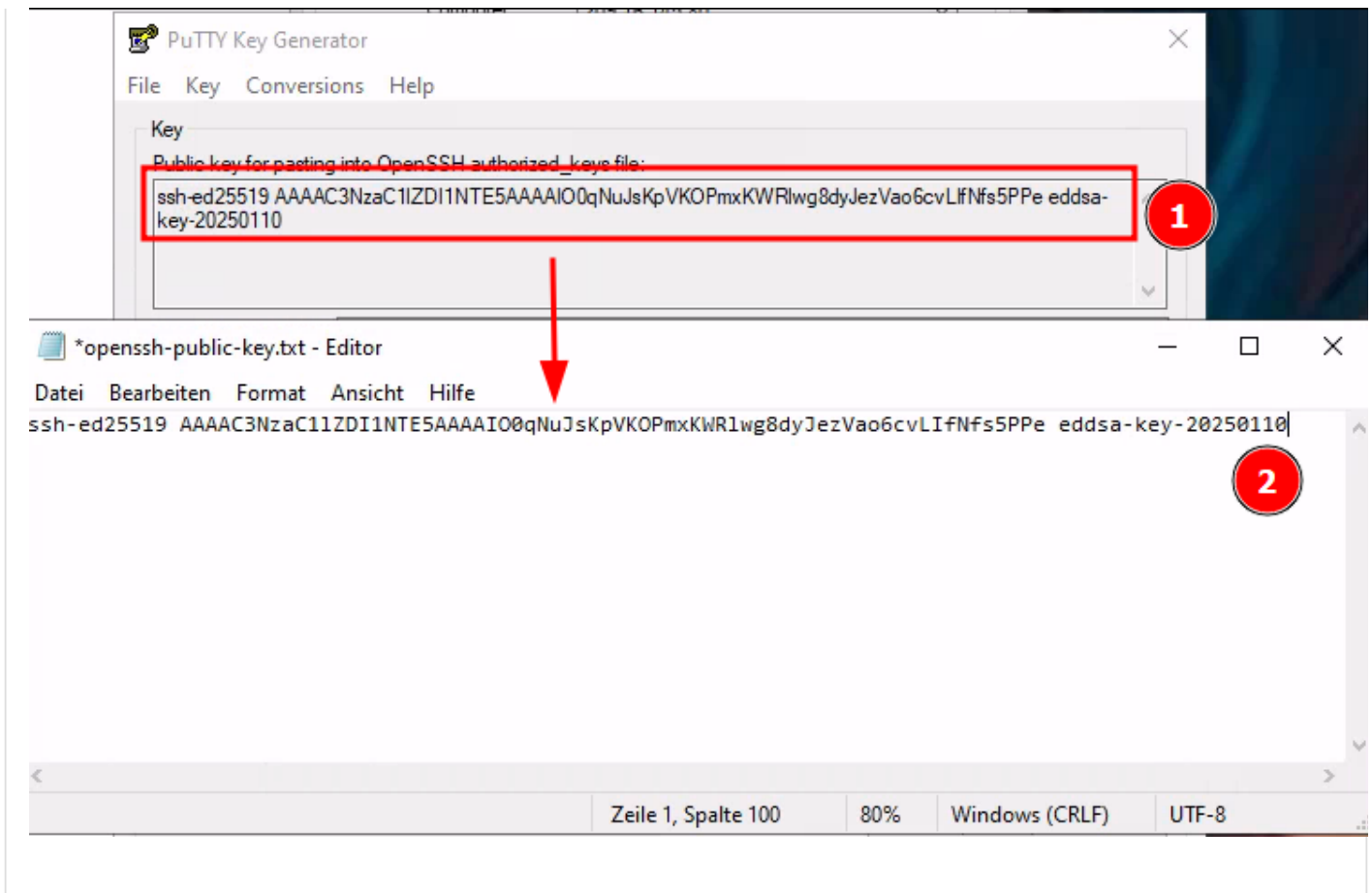




1. Die Datei **openssh-public-key.txt** nennen und mit Doppelklick öffnen

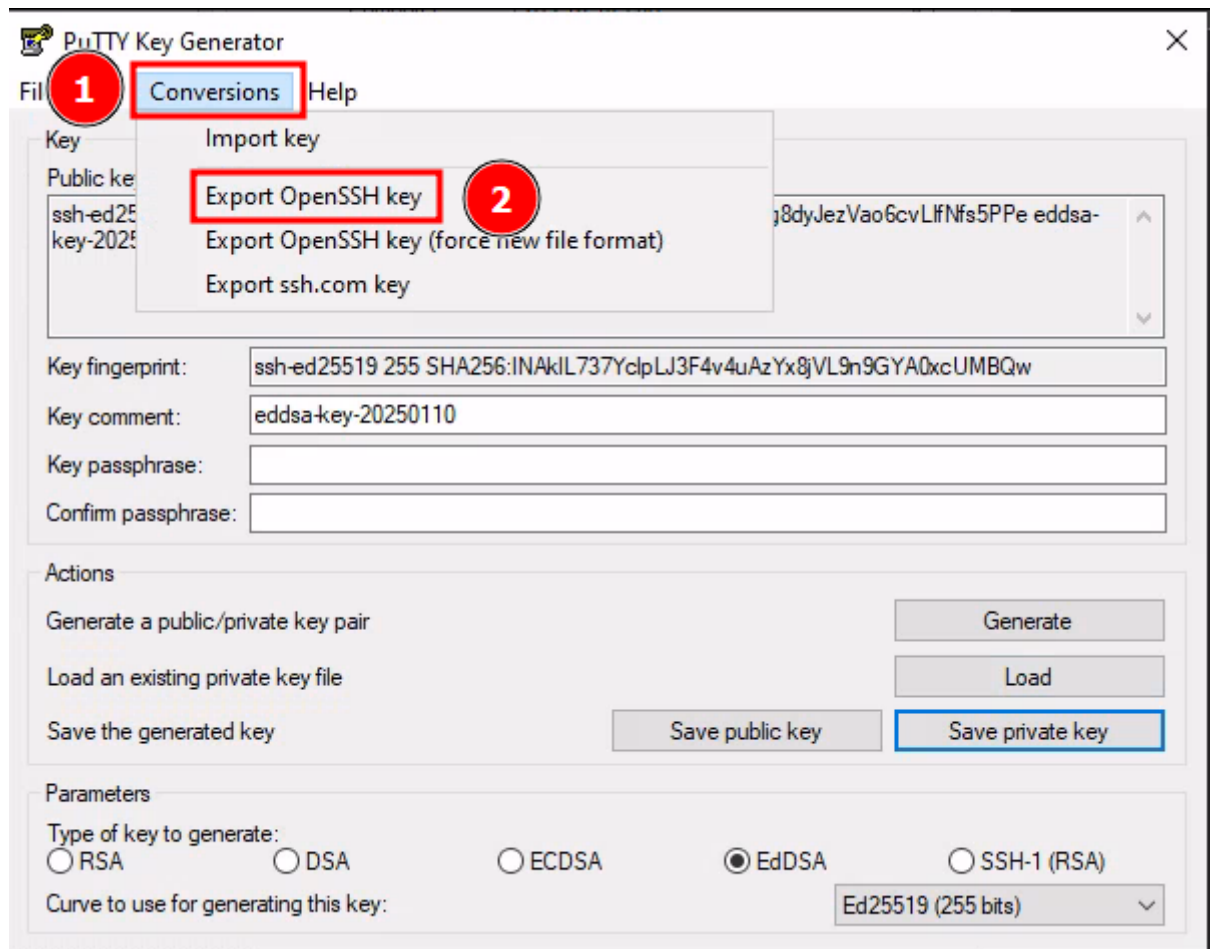


1. Den Inhalt aus dem Textfeld im **PuTTYgen** kopieren
2. In den Editor einfügen und speichern

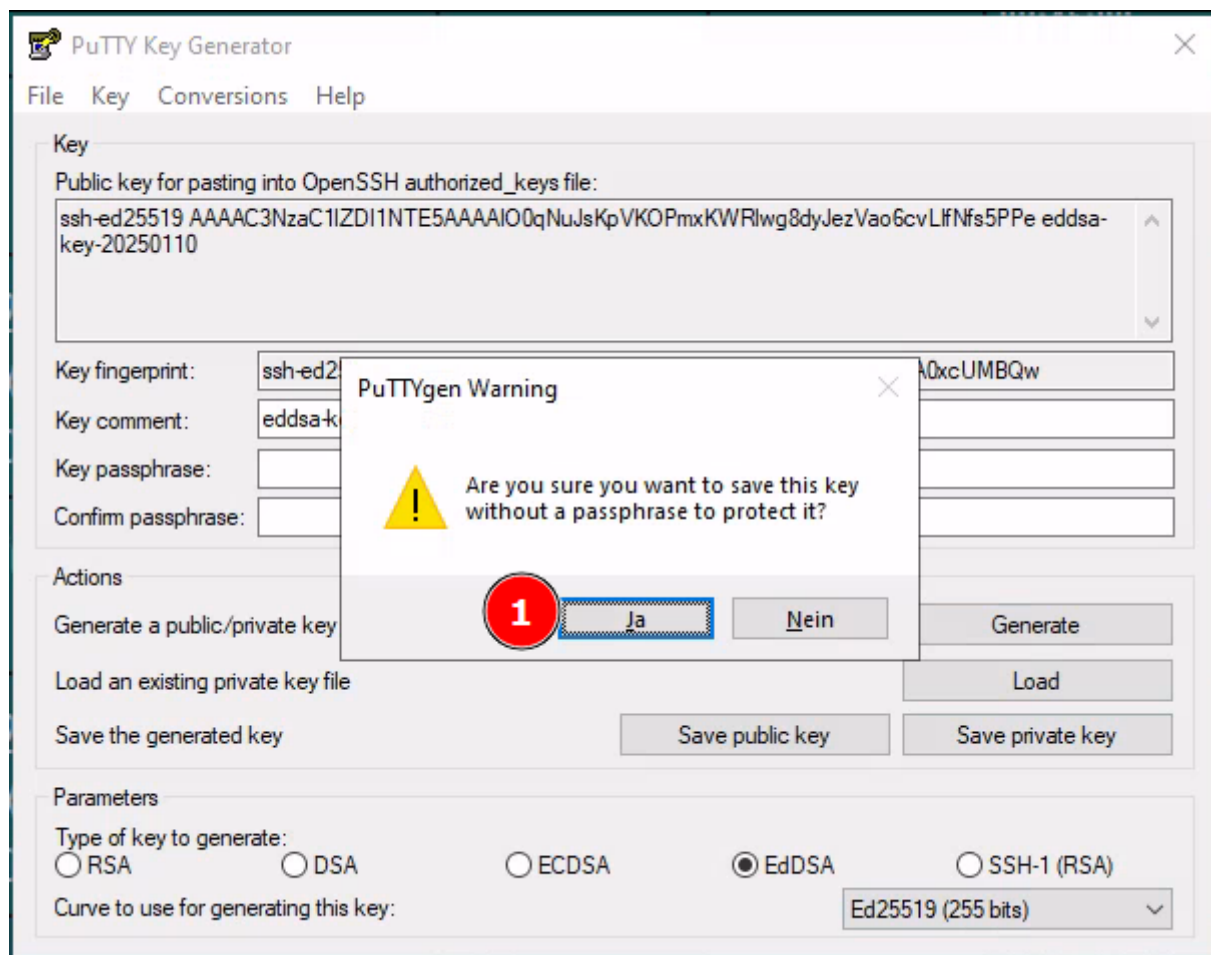


## Speichern des OpenSSH Private Keys

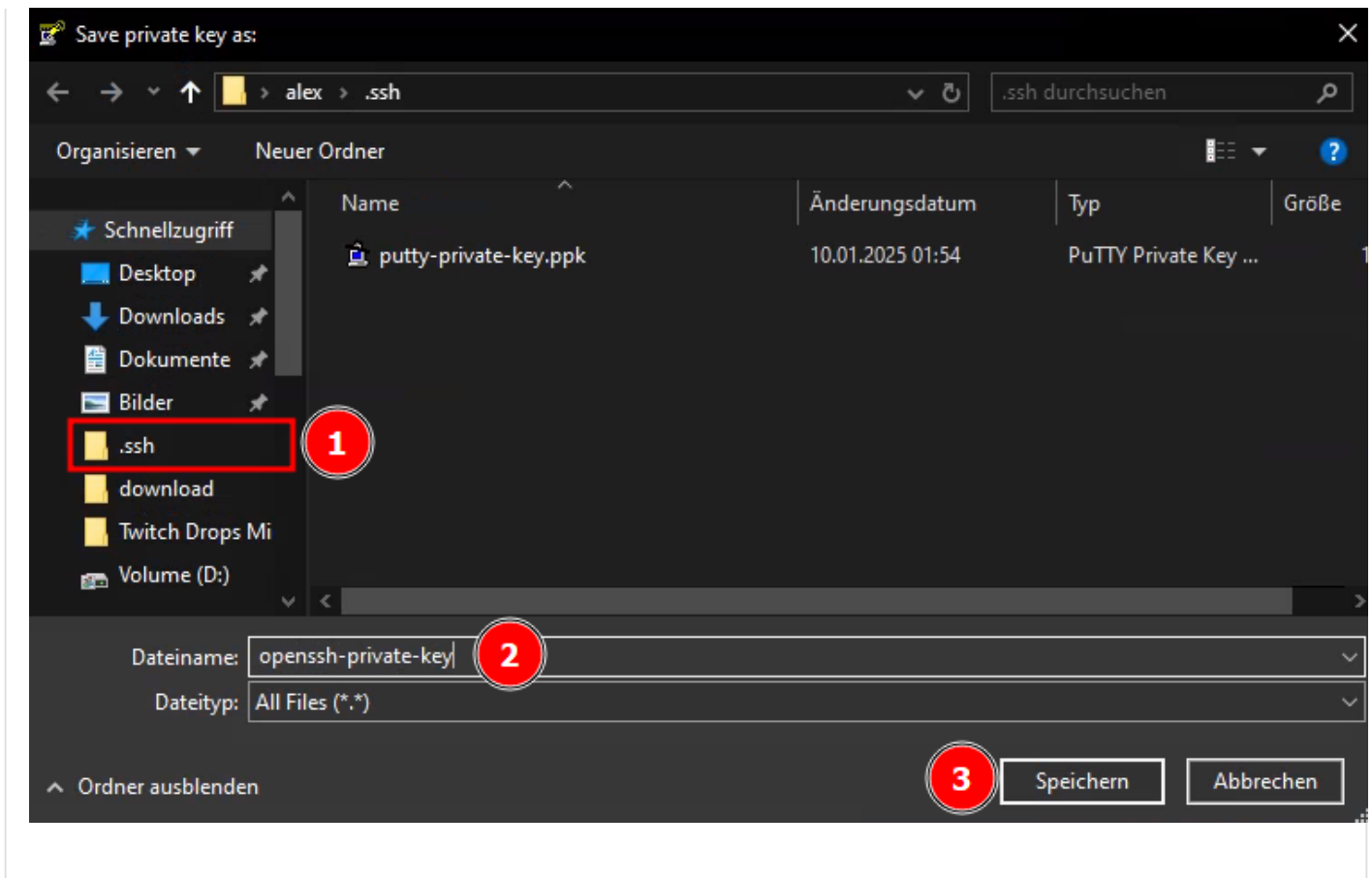
1. Auf Conversion klicken
2. Auf Export OpenSSH key klicken



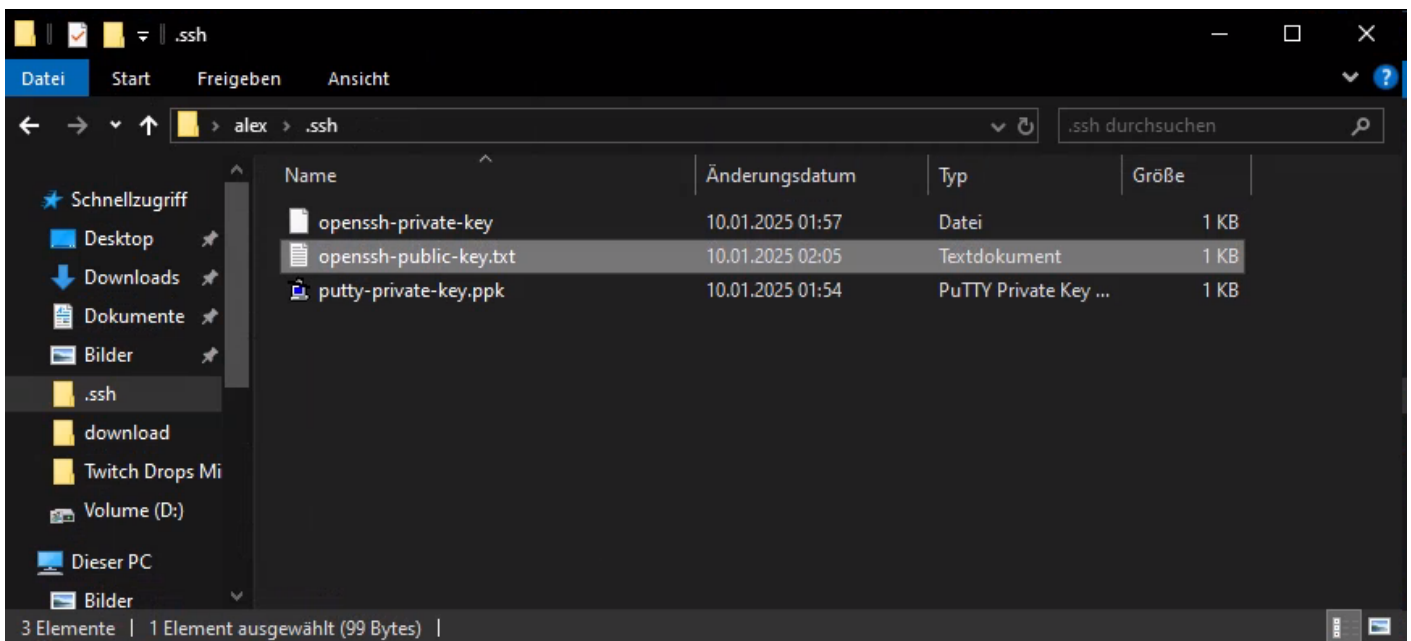
Dialog mit Ja bestätigen



1. Auf `.ssh` klicken
2. Bei Dateinamen `openssh-private-key` eintragen
3. Und auf `Speichern` klicken



Jetzt sollten im .ssh Ordner diese 3 Dateien liegen:



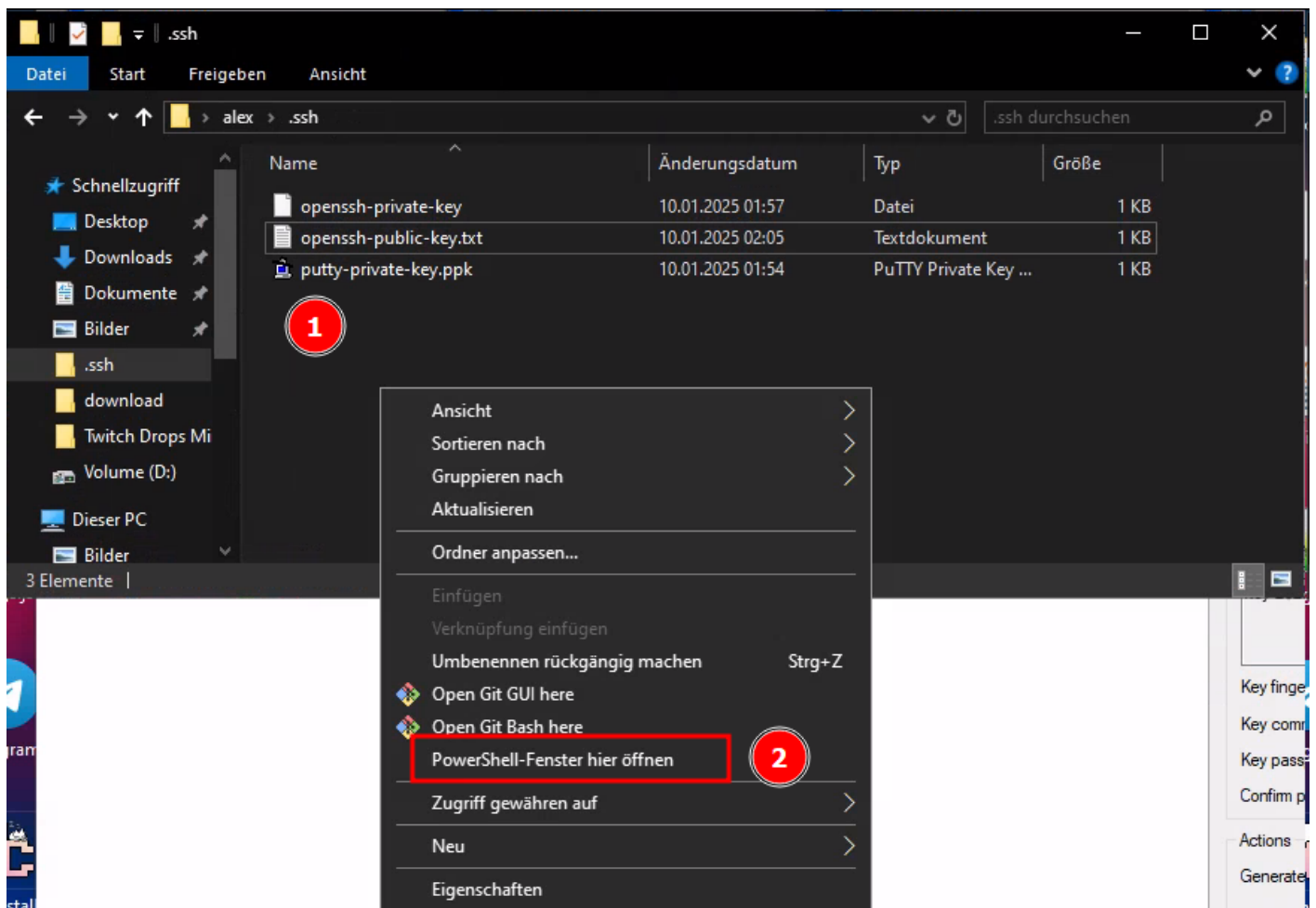
Den **putty-private-key.ppk** darf man nicht verlieren da aus ihm die anderen beiden Dateien wieder erstellt werden können.

Weder der **putty-private-key.ppk** noch der **openssh-private-key** dürfen an andere Personen weitergegeben werden! Man gibt ja die Login Daten fürs Onlinebanking auch nicht weiter oder?

Wenn euch jemand nach eurem **SSH-Key** fragt gebt ihr entweder die **openssh-public-key.txt** weiter oder deren Inhalt

### 3. Schlüssel registrieren

1. Mit Shift+Rechtsklick das Erweiterte Kontext Menü aufrufen
2. PowerShell-Fenster hier öffnen auswählen



Folgendes in der PowerShell ausführen:

```
ssh-add .\openssh-private-key
```

```
Windows PowerShell
PS C:\Users\alex\.ssh> ssh-add .\openssh-private-key_
```

Wenn ihr folgende Ausgabe bekommt

```
Windows PowerShell
PS C:\Users\alex\.ssh> ssh-add .\openssh-private-key
Bad permissions. Try removing permissions for user: DESKTOP-AFH52IH\
\herter (S-1-5-21-605832379-3039829534-2830679314-1005) on file C:/U
sers/alex/.ssh/openssh-private-key.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: UNPROTECTED PRIVATE KEY FILE!           @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions for '.\openssh-private-key' are too open.
It is required that your private key files are NOT accessible by oth
ers.
This private key will be ignored.
```

Müsst ihr folgenden Befehl ausführen:

```
$p = ".\openssh-private-key"; icacls.exe $p /reset; icacls.exe $p /GRANT:R "$($env:USERNAME):(R)"; icacls.exe
$p /inheritance:r; ssh-add $p
```

Wenn ihr die folgende Ausgabe bekommt wurde der Schlüssel richtig geladen:

```
Windows PowerShell
PS C:\Users\alex\.ssh> $p = ".\openssh-private-key"; icacls.exe $p /reset; icacls.exe $p /GRANT:R "$($env:US
ERNAME):(R)"; icacls.exe $p /inheritance:r; ssh-add $p
Bearbeitete Datei: .\openssh-private-key
1 Dateien erfolgreich verarbeitet, bei 0 Dateien ist ein Verarbeitungsfehler aufgetreten.
Bearbeitete Datei: .\openssh-private-key
1 Dateien erfolgreich verarbeitet, bei 0 Dateien ist ein Verarbeitungsfehler aufgetreten.
Bearbeitete Datei: .\openssh-private-key
1 Dateien erfolgreich verarbeitet, bei 0 Dateien ist ein Verarbeitungsfehler aufgetreten.
Identity added: .\openssh-private-key (eddsa-key-20250110)
PS C:\Users\alex\.ssh>
```

## 4. Kontrolle

Zum Abschluss folgendes in der PowerShell ausführen zur Kontrolle: ssh-add -L und man sieht das der Schlüssel geladen ist:

```
Windows PowerShell
PS C:\Users\alex\.ssh> ssh-add -L
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO0qNuJsKpVKOPmxKWRlwg8dyJezVao6cvLIIfNfs5PPe eddsa-key-20250110
PS C:\Users\alex\.ssh>
```

Jetzt könnt ihr beim Login per SSH als Authentifizierung "Agent" verwenden so euer Public key im Server hinterlegt ist.





# Tweaks & Optimierung

# Windows 11 Erinnerung deaktivieren

Hiermit deaktiviert man den nervigen Hinweis doch bitte auf Windows 11 zu wechseln.

## Was ist gemeint



Um alle diese Meldungen zu deaktivieren sind nur 2 Einträge in der Windows Registry notwendig:

Alle Schritte müssen als **Administrator** ausgeführt werden!

## Als Script für die Powershell

Folgendes als update.ps1 speichern:

```
# Pfad zur Registry
$regPath = "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate"

# Prüfen, ob der Registry-Pfad existiert, falls nicht, wird er erstellt
if (-not (Test-Path $regPath)) {
    New-Item -Path $regPath -Force | Out-Null
}

# Setzen des REG_SZ Werts "ProductVersion" mit dem Wert "Windows 10"
New-ItemProperty -Path $regPath -Name "ProductVersion" -Value "Windows 10" -PropertyType String -Force

# Setzen des REG_DWORD Werts "TargetReleaseVersion" mit dem Wert "1"
New-ItemProperty -Path $regPath -Name "TargetReleaseVersion" -Value 1 -PropertyType DWord -Force

Write-Output "Registry-Werte erfolgreich gesetzt!"
```

Anschließend zuerst den Befehl ausführen:

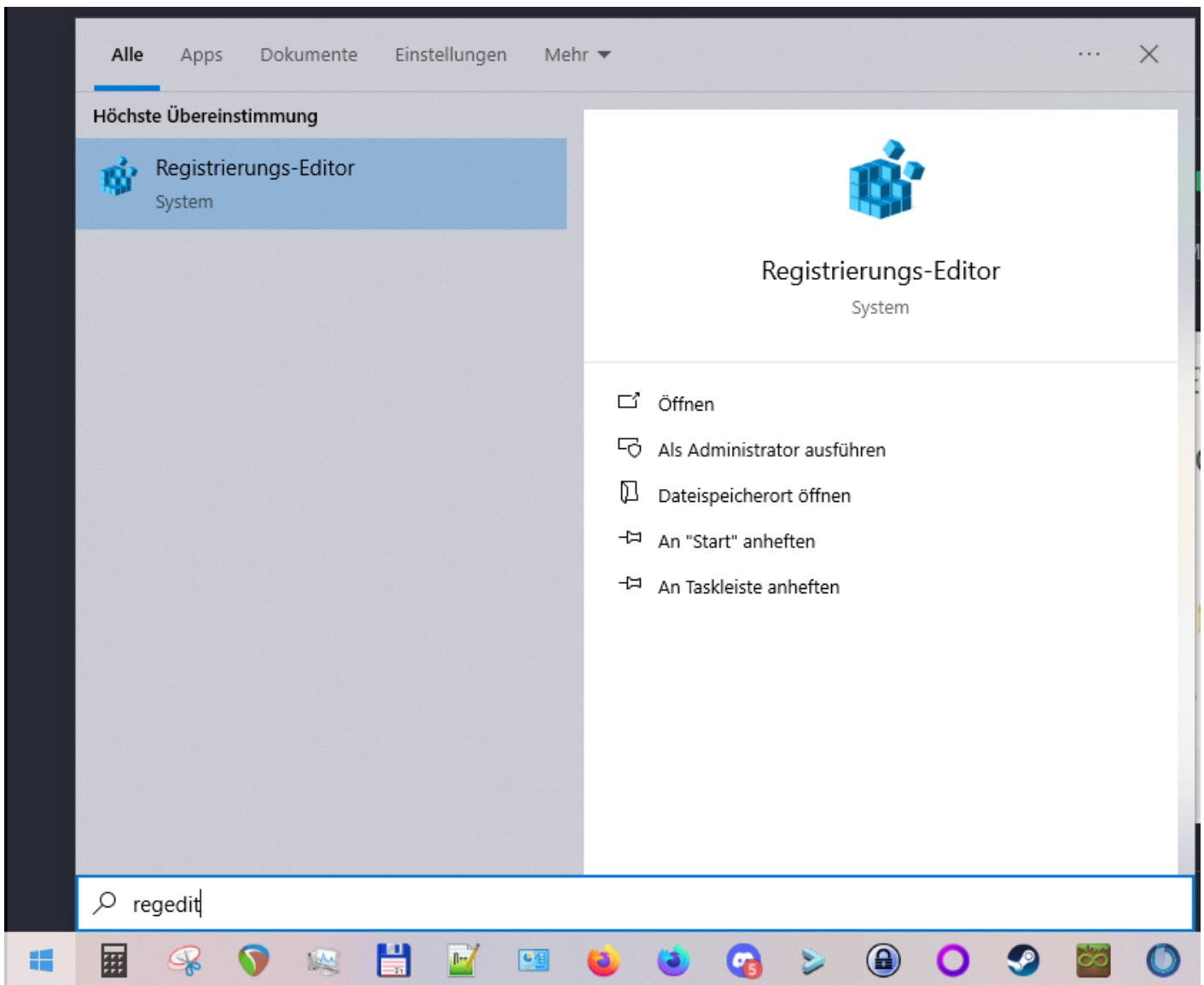
```
Set-ExecutionPolicy Unrestricted -Force
```

und zuletzt:

```
.\update.ps1
```

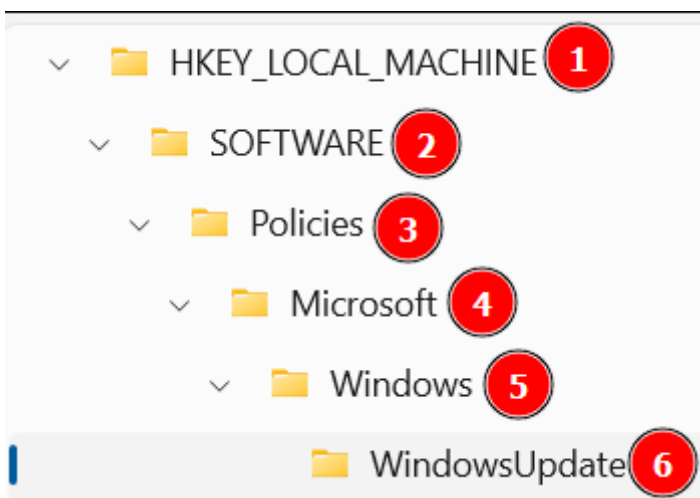
---

Über das Start Menü **regedit** suchen und starten.



Im **Registrierungseditor** diesen Pfad aufrufen:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate



## Neuen Eintrag **ProduktVersion** definieren

1. Rechtsklick und auf **Neu**

2. **Zeichenfolge** auswählen

3. Als **Namen** eintragen:

ProductVersion

4. Rechtsklick auf **ProductVersion**

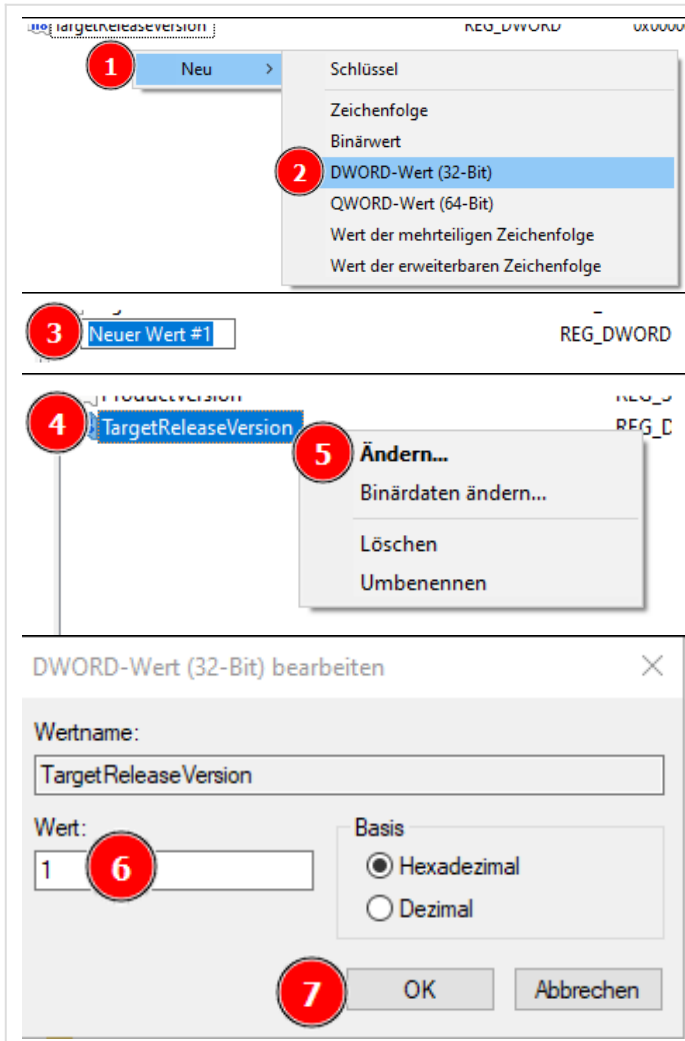
5. **Ändern...** auswählen

6. Bei **Wert** eintragen:

Windows 10

7. Mit **Ok** bestätigen

## Neuen Eintrag **TargetReleaseVersion** definieren



1. Rechtsklick und auf Neu
2. DWORD-Wert (32-Bit) auswählen
3. Als Namen eintragen:

TargetReleaseVersion

4. Rechtsklick auf TargetReleaseVersion
5. Ändern auswählen
6. Bei Wert eintragen:

1

7. Mit Ok bestätigen

# Netzwerktyp ändern

Um den Netzwerktyp (Öffentlich/Privat) von einer Netzwerkverbindung unter Windows zu ändern reichen die folgenden zwei Befehle:

Zuerst alle Netzwerkadapter ausgeben lassen:

```
get-netconnectionprofile
```

Beispiel Ausgabe:

```
PS C:\Windows\system32> get-netconnectionprofile

Name                : Hauptnetz 1
InterfaceAlias      : Hauptnetz
InterfaceIndex      : 4         2
NetworkCategory     : Private   3
DomainAuthenticationKind : None
IPv4Connectivity    : LocalNetwork
IPv6Connectivity    : NoTraffic
```

**1. Adapter Name**

**2. Adapter ID**

**3. Netzwerktyp (Öffentlich / Privat)**

Jetzt kann der entsprechende Adapter mit dem Befehl `set-netconnectionprofile -NetworkCategory ? -InterfaceIndex #` umgestellt werden.

Mit Werten aus der Beispiel Ausgabe wäre es dann zum ändern in ein privates Netzwerk:

```
set-netconnectionprofile -NetworkCategory Private -InterfaceIndex 4
```

und zum ändern in ein öffentliches Netzwerk:

```
set-netconnectionprofile -NetworkCategory Public -InterfaceIndex 4
```

# Bloatware loswerden

```
# Liste der App-Pakete, die deinstalliert werden sollen
$appstoRemove = @(
    "Microsoft.GetHelp",
    "Microsoft.Getstarted",
    "Microsoft.YourPhone",
    "Microsoft.People",
    "Microsoft.MixedReality.Portal",
    "Microsoft.Microsoft3DViewer",
    "Microsoft.Wallet",
    "Microsoft.BingWeather",
    "Microsoft.WindowsMaps",
    "Microsoft.WindowsSoundRecorder",
    "Microsoft.SkypeApp",
    "Microsoft.Office.OneNote",
    "Microsoft.MicrosoftStickyNotes",
    "Microsoft.WindowsAlarms",
    "Microsoft.MSPaint",
    "Microsoft.WindowsCamera",
    "Microsoft.Windows.Photos",
    "Microsoft.MicrosoftOfficeHub"
)

# Schleife über alle Apps
foreach ($app in $appstoRemove) {
    $package = Get-AppxPackage -Name $app -ErrorAction SilentlyContinue
    if ($package) {
        Write-Output "Deinstalliere $app..."
        Remove-AppxPackage $package.PackageFullName
    } else {
        Write-Output "$app ist nicht installiert oder bereits entfernt."
    }
}

Write-Output "Bereinigung abgeschlossen!"
```