



# KI-BUSTER

Praxiswissen für KI, Linux und sichere Automatisierung

KOSTENLOSES PDF-HANDOUT

## Codex CLI & Codex Remote Control on Linux

Installation, Konfiguration und sicherer Remote-Einsatz

Arch / CachyOS | Debian | Ubuntu | Fedora | Nobara | RHEL

```
$ curl -fsSL https://chatgpt.com/codex/install.sh | sh
$ codex login --device-auth
$ codex
> /permissions /status /review
# workspace-write + on-request = kontrollierter Start
```

Faktenstand: 18. August 2026 | OpenAI- und Linux-Originalquellen geprüft

<https://kibuster.braetter-int.de>

KI-Buster - verständlich, praxisnah, sicherheitsbewusst

# Codex CLI & Codex Remote Control on Linux

Praxis-Handout zur Installation, Konfiguration und Absicherung unter Arch/CachyOS, Debian, Ubuntu, Fedora, Nobara und Red Hat Enterprise Linux.

## Faktenstand

Dieses Handout wurde am 18. August 2026 gegen die aktuelle OpenAI-Dokumentation sowie offizielle Dokumentationen der Linux-Distributionen abgeglichen. Wo OpenAI eine Funktion als Preview oder Experimental kennzeichnet, wird das ausdrücklich genannt.

Codex CLI bringt Codex direkt ins Linux-Terminal. Der Agent kann in einem Projekt Dateien untersuchen, Änderungen vornehmen und lokale Werkzeuge aufrufen - innerhalb der Berechtigungen, die der Benutzer und die Codex-Sandbox erlauben. Die Linux-Installation selbst ist distributionsübergreifend, weil OpenAI für Linux einen Standalone-Installer bereitstellt. Die Vorbereitung von SSH und Basispaketen unterscheidet sich jedoch je Distribution.

## Wichtige Begriffe sauber trennen

### Codex CLI

Der Terminal-Client. Er läuft nativ unter Linux und ist für die Arbeit in lokalen oder per SSH erreichbaren Projekten gedacht.

### ChatGPT Desktop auf Linux

Seit 11. August 2026 gibt es laut OpenAI eine Linux-Desktop-Preview für unterstützte Ubuntu-, Debian- und Fedora-Desktop-Distributionen auf x64 und ARM64. Das ist nicht dasselbe wie die Remote-Host-Funktion.

### Remote / remote-control

Die dokumentierte Remote-Host-Steuerung nennt aktuell macOS- und Windows-Hosts. Der CLI-Befehl `codex remote-control` existiert zusätzlich, wird von OpenAI aber als Experimental geführt.

**Die wichtigste Konsequenz: Codex CLI auf Linux ist regulär nutzbar. Eine vollständige Linux-Remote-Host-Unterstützung sollte aber nicht behauptet werden, solange OpenAI sie in der Remote-Dokumentation nicht freigibt.**

# Inhaltsverzeichnis

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>Codex CLI &amp; Codex Remote Control on Linux</b>      | <b>2</b>  |
| Wichtige Begriffe sauber trennen . . . . .                | 2         |
| <b>1. Voraussetzungen und Vorprüfung</b>                  | <b>6</b>  |
| 1.1 Distribution und Architektur feststellen . . . . .    | 6         |
| 1.2 Was für Codex wirklich benötigt wird . . . . .        | 6         |
| 1.3 Distributionsmatrix . . . . .                         | 6         |
| <b>2. Offizielle Codex-CLI-Installation</b>               | <b>7</b>  |
| 2.1 Standalone-Installer von OpenAI . . . . .             | 7         |
| 2.2 Installation prüfen . . . . .                         | 7         |
| 2.3 Codex aktualisieren . . . . .                         | 7         |
| <b>3. Arch Linux und CachyOS</b>                          | <b>8</b>  |
| 3.1 Arch Linux vorbereiten . . . . .                      | 8         |
| 3.2 CachyOS vorbereiten . . . . .                         | 8         |
| 3.3 Codex installieren . . . . .                          | 8         |
| <b>4. Debian</b>                                          | <b>9</b>  |
| 4.1 Basispakete . . . . .                                 | 9         |
| 4.2 Optional: SSH-Server . . . . .                        | 9         |
| 4.3 Codex installieren . . . . .                          | 9         |
| 4.4 ChatGPT Desktop auf Debian . . . . .                  | 9         |
| <b>5. Ubuntu</b>                                          | <b>10</b> |
| 5.1 Basispakete . . . . .                                 | 10        |
| 5.2 Optional: SSH-Server . . . . .                        | 10        |
| 5.3 Codex installieren . . . . .                          | 10        |
| 5.4 ChatGPT Desktop auf Ubuntu . . . . .                  | 10        |
| <b>6. Fedora</b>                                          | <b>11</b> |
| 6.1 Basispakete . . . . .                                 | 11        |
| 6.2 Optional: SSH-Server . . . . .                        | 11        |
| 6.3 Codex installieren . . . . .                          | 11        |
| 6.4 ChatGPT Desktop auf Fedora . . . . .                  | 11        |
| <b>7. Nobara</b>                                          | <b>12</b> |
| 7.1 Was bei Nobara anders ist . . . . .                   | 12        |
| 7.2 Basispakete installieren . . . . .                    | 12        |
| 7.3 Optional: SSH-Server . . . . .                        | 12        |
| 7.4 Codex installieren . . . . .                          | 12        |
| 7.5 Linux-Desktop-Preview . . . . .                       | 12        |
| <b>8. Red Hat Enterprise Linux (RHEL)</b>                 | <b>13</b> |
| 8.1 Voraussetzung: funktionierende Repositories . . . . . | 13        |
| 8.2 Basispakete installieren . . . . .                    | 13        |

---

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| 8.3 Optional: SSH-Server . . . . .                                 | 13        |
| 8.4 Codex installieren . . . . .                                   | 13        |
| 8.5 Corporate CA für Codex . . . . .                               | 13        |
| 8.6 Linux-Desktop-Preview . . . . .                                | 13        |
| <b>9. Anmeldung und Authentifizierung</b>                          | <b>14</b> |
| 9.1 Interaktiver Start . . . . .                                   | 14        |
| 9.2 Expliziter Login . . . . .                                     | 14        |
| 9.3 Headless-Systeme: Device Code . . . . .                        | 14        |
| 9.4 Anmeldung mit API-Key . . . . .                                | 14        |
| 9.5 Status und Logout . . . . .                                    | 14        |
| 9.6 Wo Credentials landen können . . . . .                         | 14        |
| <b>10. Codex konfigurieren</b>                                     | <b>15</b> |
| 10.1 Konfigurationsebenen . . . . .                                | 15        |
| 10.2 Sichere Startkonfiguration . . . . .                          | 15        |
| 10.3 Sandbox-Modi . . . . .                                        | 15        |
| 10.4 Netzwerkzugriff gezielt aktivieren . . . . .                  | 15        |
| <b>11. Projektregeln mit .codex und AGENTS.md</b>                  | <b>16</b> |
| 11.1 Projektbezogene config.toml . . . . .                         | 16        |
| 11.2 AGENTS.md . . . . .                                           | 16        |
| <b>12. Linux als SSH-Remote-System für Codex</b>                   | <b>17</b> |
| 12.1 Zielarchitektur . . . . .                                     | 17        |
| 12.2 Eigenen Benutzer verwenden . . . . .                          | 17        |
| 12.3 SSH-Key erzeugen . . . . .                                    | 17        |
| 12.4 SSH-Alias . . . . .                                           | 17        |
| 12.5 Entscheidender Test . . . . .                                 | 17        |
| <b>13. Codex Remote, Linux Desktop Preview und remote-control</b>  | <b>18</b> |
| 13.1 Was OpenAI aktuell als Remote-Host nennt . . . . .            | 18        |
| 13.2 Linux Desktop Preview ist eine andere Aussage . . . . .       | 18        |
| 13.3 Experimenteller CLI-Befehl remote-control . . . . .           | 18        |
| 13.4 Remote Control in verwalteten Umgebungen abschalten . . . . . | 18        |
| <b>14. Sichere Betriebsregeln für Administratoren</b>              | <b>19</b> |
| 14.1 Codex nicht als root starten . . . . .                        | 19        |
| 14.2 Git-Checkpoint vor größeren Aufgaben . . . . .                | 19        |
| 14.3 Development -> Staging -> Production . . . . .                | 19        |
| 14.4 sudo begrenzen . . . . .                                      | 19        |
| 14.5 Netzwerkzugriff nicht automatisch öffnen . . . . .            | 19        |
| 14.6 Fremde Repositories als untrusted behandeln . . . . .         | 19        |
| <b>15. Fehlerdiagnose</b>                                          | <b>20</b> |
| 15.1 codex: command not found . . . . .                            | 20        |
| 15.2 Login auf Headless-Systemen . . . . .                         | 20        |
| 15.3 TLS- oder Corporate-Proxy-Probleme . . . . .                  | 20        |

---

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| 15.4 SSH funktioniert, Codex Remote-Projekt aber nicht . . . . . | 20        |
| 15.5 Konfiguration wirkt nicht wie erwartet . . . . .            | 20        |
| <b>16. Schnellreferenz</b>                                       | <b>21</b> |
| Installation . . . . .                                           | 21        |
| Login . . . . .                                                  | 21        |
| API-Key . . . . .                                                | 21        |
| Start . . . . .                                                  | 21        |
| Wichtige interaktive Befehle . . . . .                           | 21        |
| Remote Control - Experimental . . . . .                          | 21        |
| Empfohlener Sicherheitsstart . . . . .                           | 21        |
| <b>17. Checkliste vor dem ersten produktiven Einsatz</b>         | <b>22</b> |
| <b>18. Quellen und Faktenbasis</b>                               | <b>23</b> |
| OpenAI / Codex . . . . .                                         | 23        |
| Linux-Distributionen . . . . .                                   | 23        |
| <b>Weiterlesen auf KI-Buster</b>                                 | <b>24</b> |

Tipp: Das PDF enthält Lesezeichen für die Hauptkapitel und Unterkapitel. In einem modernen PDF-Reader kannst du direkt dorthin springen.

# 1. Voraussetzungen und Vorprüfung

## 1.1 Distribution und Architektur feststellen

Bevor du Pakete installierst, prüfe, auf welchem System du arbeitest. Das verhindert, dass Paketmanager oder Dienstnamen verwechselt werden.

```
cat /etc/os-release
uname -m
uname -r
```

## 1.2 Was für Codex wirklich benötigt wird

- Internetverbindung für Installation, Anmeldung und Modellzugriff.
- `curl` für den offiziellen Standalone-Installer.
- `git` ist nicht zwingend für den Start, aber für sichere Entwicklungsworkflows dringend sinnvoll.
- CA-Zertifikate für TLS-Verbindungen.
- OpenSSH nur dann, wenn der Rechner als SSH-Client oder SSH-Ziel genutzt werden soll.
- Ein ChatGPT-Login oder alternativ ein OpenAI API-Key.

### Kein Node.js-Zwang

Fuer die in diesem Handout verwendete Linux-Installation ist Node.js nicht erforderlich. OpenAI stellt einen eigenen Standalone-Installer für macOS/Linux bereit.

## 1.3 Distributionsmatrix

| Distribution | Paketmanager               | SSH-Pakete                       | SSH-Dienst | Codex-Installation          |
|--------------|----------------------------|----------------------------------|------------|-----------------------------|
| Arch Linux   | pacman                     | openssh                          | sshd       | OpenAI Standalone-Installer |
| CachyOS      | pacman                     | openssh                          | sshd       | OpenAI Standalone-Installer |
| Debian       | apt                        | openssh-client / openssh-server  | ssh        | OpenAI Standalone-Installer |
| Ubuntu       | apt                        | openssh-client / openssh-server  | ssh        | OpenAI Standalone-Installer |
| Fedora       | dnf                        | openssh-clients / openssh-server | sshd       | OpenAI Standalone-Installer |
| Nobara       | dnf<br>(Paketinstallation) | openssh-clients / openssh-server | sshd       | OpenAI Standalone-Installer |
| RHEL         | dnf                        | openssh-clients / openssh-server | sshd       | OpenAI Standalone-Installer |

Der eigentliche Codex-Installationsbefehl ist auf allen hier behandelten Distributionen identisch. Unterschiede betreffen die Pakete rund um Git, SSH und die Systempflege.

## 2. Offizielle Codex-CLI-Installation

### 2.1 Standalone-Installer von OpenAI

OpenAI dokumentiert für macOS und Linux den folgenden Standalone-Installer. Derselbe Befehl wird auch zum Aktualisieren der CLI verwendet.

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
```

#### Sicherheitsgedanke

Ein per Pipe an sh ausgeführtes Installationsskript besitzt die Rechte des aktuellen Benutzers. In streng kontrollierten Umgebungen solltest du externe Installationsskripte vor dem Einsatz entsprechend der eigenen Change- und Softwarefreigabeprozesse prüfen.

### 2.2 Installation prüfen

```
command -v codex  
codex --help
```

Wenn die Shell den Befehl direkt nach der Installation nicht findet, öffne eine neue Login-Shell oder lade das passende Shell-Profil neu. Der Installer kann die Umgebung für den aktuellen Benutzer anpassen; ein harter, distributionsunabhängiger Installationspfad sollte deshalb nicht angenommen werden.

### 2.3 Codex aktualisieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
```

Danach erneut prüfen:

```
codex --help
```

## 3. Arch Linux und CachyOS

### 3.1 Arch Linux vorbereiten

Arch Linux verwendet pacman. Fuer Git, Download und SSH kannst du die benötigten Pakete aus den offiziellen Repositories installieren.

```
sudo pacman -Syu
sudo pacman -S --needed curl git openssh ca-certificates
```

Wenn der Rechner als SSH-Server dienen soll:

```
sudo systemctl enable --now sshd
systemctl status sshd
```

### 3.2 CachyOS vorbereiten

CachyOS basiert auf Arch Linux und verwendet ebenfalls pacman. Die Codex-CLI selbst wird nicht über ein CachyOS-spezifisches Paket installiert, sondern über den OpenAI-Installer.

```
sudo pacman -Syu
sudo pacman -S --needed curl git openssh ca-certificates
```

#### CachyOS-Hinweis

CachyOS bietet zusätzliche Werkzeuge und optimierte Repositories. Fuer die hier benötigten Standardpakete bleibt pacman ein gueltiger Installationsweg. Dieses Handout erfindet bewusst keinen inoffiziellen Codex-AUR- oder CachyOS-Sonderweg.

### 3.3 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
command -v codex
codex --help
```

#### Quelle

[ArchWiki: OpenSSH](#)  
[CachyOS Wiki](#)

## 4. Debian

### 4.1 Basispakete

```
sudo apt update
sudo apt install -y curl git openssh-client ca-certificates
```

### 4.2 Optional: SSH-Server

```
sudo apt install -y openssh-server
sudo systemctl enable --now ssh
systemctl status ssh
```

Debian dokumentiert openssh-client für den Client und openssh-server für den Server.

### 4.3 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
command -v codex
codex --help
```

### 4.4 ChatGPT Desktop auf Debian

#### Preview

OpenAI nennt Debian seit 11. August 2026 als unterstützte Distribution für die Linux-Desktop-Preview. Die Desktop-App ist für die Codex CLI nicht erforderlich. Die Remote-Host-Dokumentation nennt dennoch weiterhin macOS und Windows als Remote-Hosts.

#### Quelle

[Debian Wiki: SSH](#)

## 5. Ubuntu

### 5.1 Basispakete

```
sudo apt update
sudo apt install -y curl git openssh-client ca-certificates
```

### 5.2 Optional: SSH-Server

```
sudo apt install -y openssh-server
sudo systemctl enable --now ssh
systemctl status ssh
```

### 5.3 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
command -v codex
codex --help
```

### 5.4 ChatGPT Desktop auf Ubuntu

OpenAI führt Ubuntu in der Linux-Desktop-Preview auf. Die bereitgestellte Linux-Desktop-App kann Projekte, lokale Dateien und Codex verwenden. Für die hier beschriebene CLI-Installation bleibt sie optional.

#### Quelle

[Ubuntu Server Docs: OpenSSH Server](#)

## 6. Fedora

### 6.1 Basispakete

```
sudo dnf install -y curl git openssh-clients ca-certificates
```

### 6.2 Optional: SSH-Server

```
sudo dnf install -y openssh-server  
sudo systemctl enable --now sshd  
systemctl status sshd
```

### 6.3 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh  
command -v codex  
codex --help
```

### 6.4 ChatGPT Desktop auf Fedora

#### Preview

Fedora gehört laut OpenAI zu den Distributionen der Linux-Desktop-Preview. Die CLI bleibt davon unabhängig und funktioniert über den Standalone-Installer.

#### Quelle

[Fedora Documentation](#)

## 7. Nobara

### 7.1 Was bei Nobara anders ist

Nobara ist Fedora-basiert und stellt Pakete über DNF bereit. Für die Systemaktualisierung empfiehlt Nobara jedoch ausdrücklich den eigenen Nobara System Updater, weil dieser zusätzliche distributionsspezifische Schritte ausführt.

**Wichtig**

Verwende in diesem Handout `dnf` für die benötigten Pakete. Einen pauschalen `dnf upgrade`-Befehl als vollwertigen Ersatz für den Nobara System Updater gebe ich bewusst nicht an.

### 7.2 Basispakete installieren

```
sudo dnf install -y curl git openssh-clients ca-certificates
```

### 7.3 Optional: SSH-Server

```
sudo dnf install -y openssh-server
sudo systemctl enable --now sshd
systemctl status sshd
```

### 7.4 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
command -v codex
codex --help
```

### 7.5 Linux-Desktop-Preview

OpenAI nennt für die Linux-Desktop-Preview explizit Ubuntu, Debian und Fedora. Nobara wird dort aktuell nicht als unterstützte Distribution genannt. Deshalb wird hier keine angeblich offizielle Nobara-Desktop-App-Installation erfunden.

#### Quelle

[Nobara: New User Guide](#)  
[Nobara: Updating Nobara](#)

## 8. Red Hat Enterprise Linux (RHEL)

### 8.1 Voraussetzung: funktionierende Repositories

Auf RHEL müssen die für das System vorgesehenen Red-Hat-Repositories erreichbar sein. In Unternehmensnetzen können außerdem Proxy-, CA- und Softwarefreigabevorgaben gelten.

### 8.2 Basispakete installieren

```
sudo dnf install -y curl git openssh-clients ca-certificates
```

### 8.3 Optional: SSH-Server

```
sudo dnf install -y openssh-server  
sudo systemctl enable --now sshd  
systemctl status sshd
```

### 8.4 Codex installieren

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh  
command -v codex  
codex --help
```

### 8.5 Corporate CA für Codex

Wenn ein Unternehmensnetz eine private Root-CA oder einen TLS-Proxy verwendet, dokumentiert OpenAI die Umgebungsvariable CODEX\_CA\_CERTIFICATE für ein PEM-Bundle:

```
export CODEX_CA_CERTIFICATE=/path/to/corporate-root-ca.pem  
codex login
```

### 8.6 Linux-Desktop-Preview

RHEL wird in OpenAIs aktueller Liste für die Linux-Desktop-Preview nicht explizit genannt. Daher beschreibt dieses Handout für RHEL nur die offiziell dokumentierbare Codex-CLI-Nutzung.

#### Quelle

[Red Hat Documentation](#)

## 9. Anmeldung und Authentifizierung

### 9.1 Interaktiver Start

Öffne ein Projektverzeichnis und starte Codex. Beim ersten Start bietet Codex die Anmeldung an.

```
cd ~/projects/mein-projekt
codex
```

### 9.2 Expliziter Login

```
codex login
```

### 9.3 Headless-Systeme: Device Code

OpenAI empfiehlt für Umgebungen, in denen der normale Browserfluss unpraktisch ist, die Device-Code-Anmeldung als Beta-Option:

```
codex login --device-auth
```

### 9.4 Anmeldung mit API-Key

```
export OPENAI_API_KEY="DEIN_API_KEY"
printenv OPENAI_API_KEY | codex login --with-api-key
```

#### API-Kosten

Bei API-Key-Authentifizierung gelten die entsprechenden API-Abrechnungs- und Zugriffsregeln. Ein ChatGPT-Abonnement und eine Plattform-API-Nutzung sind nicht dasselbe Abrechnungsmodell.

### 9.5 Status und Logout

```
codex login status
codex logout
```

### 9.6 Wo Credentials landen können

OpenAI dokumentiert `~/.codex/auth.json` als möglichen dateibasierten Auth-Cache. Ueber `cli_auth_credentials_store` kann zwischen `file`, `keyring` und `auto` gewählt werden.

#### Secret-Regel

`auth.json`, API-Keys und SSH-Private-Keys gehören weder in Git noch in Tickets, Chat-Prompts oder gemeinsam lesbare Projektordner.

# 10. Codex konfigurieren

## 10.1 Konfigurationsebenen

OpenAI dokumentiert mehrere Konfigurationsebenen. Für Linux sind besonders diese drei wichtig:

```
~/.codex/config.toml      # Benutzerkonfiguration
.codex/config.toml        # Projektkonfiguration
/etc/codex/config.toml    # Systemweite Unix-Konfiguration
```

Projektkonfigurationen werden aus Sicherheitsgründen nur aus Projekten geladen, denen du vertraust.

## 10.2 Sichere Startkonfiguration

```
approval_policy = "on-request"
approvals_reviewer = "user"
sandbox_mode = "workspace-write"
cli_auth_credentials_store = "auto"
allow_login_shell = false
web_search = "cached"

[sandbox_workspace_write]
network_access = false
```

### Warum diese Werte?

workspace-write erlaubt Arbeit im Workspace. on-request lässt Freigaben sichtbar werden. approvals\_reviewer = user hält den Menschen als Freigeber. network\_access = false verhindert ausgehenden Netzwerkzugriff innerhalb der workspace-write-Sandbox, solange du ihn nicht bewusst aktivierst.

## 10.3 Sandbox-Modi

read-only: Dateien lesen, aber keine normalen Änderungen ohne Freigabe.

workspace-write: Lesen, Schreiben und Routinebefehle innerhalb des Workspace.

danger-full-access: keine normalen Sandbox-Grenzen für Dateisystem und Netzwerk.

### OpenAI-Empfehlung

OpenAI beschreibt workspace-write zusammen mit on-request als den risikoärmeren lokalen Automatisierungs-Preset. Vollzugriff mit danger-full-access und never sollte nur in bewusst isolierten Umgebungen eingesetzt werden.

## 10.4 Netzwerkzugriff gezielt aktivieren

Wenn ein Projekt Paketdownloads oder Netzwerkzugriff für innerhalb der Sandbox ausgeführte Befehle braucht:

```
[sandbox_workspace_write]
network_access = true
```

Das sollte eine bewusste Projektentscheidung sein - nicht pauschaler Standard auf jedem System.

# 11. Projektregeln mit .codex und AGENTS.md

## 11.1 Projektbezogene config.toml

Eine Projektkonfiguration kann enger oder spezieller sein als die Benutzerkonfiguration.

```
mkdir -p .codex
$EDITOR .codex/config.toml
```

Beispiel:

```
approval_policy = "on-request"
sandbox_mode = "workspace-write"
```

## 11.2 AGENTS.md

In der interaktiven Codex CLI steht `/init` zur Verfügung, um eine AGENTS.md als Projektanweisung anzulegen.

```
/init
```

Beispiel für belastbare Regeln:

```
# Projektregeln
- Keine produktiven Deployments ohne ausdrückliche Freigabe.
- Keine Secrets ausgeben oder in Git schreiben.
- Vor Änderungen git status und vorhandene Tests prüfen.
- Fremde Änderungen im Working Tree bewahren.
- Datenbankmigrationen zuerst gegen Staging testen.
- Bei unklarer Ursache zuerst Diagnoseplan, dann Änderung.
```

### Wichtig

AGENTS.md ist eine Arbeitsanweisung an den Agenten, aber kein Ersatz für Linux-Dateirechte, Sandbox, sudo-Regeln, Git und Backups.

## 12. Linux als SSH-Remote-System für Codex

### 12.1 Zielarchitektur

OpenAI dokumentiert SSH-basierte Remote-Projekte. Die Desktop-App startet den Codex App Server über SSH mit der Login-Shell des Remote-Benutzers. Deshalb muss der Befehl `codex` dort im PATH liegen.

```
Client / Desktop Host
|
| SSH
v
Linux-Server oder Linux-Workstation
|
+-- Codex CLI
+-- Git / Build-Tools
+-- Projektdateien
```

### 12.2 Eigenen Benutzer verwenden

Auf Servern ist ein separater Benutzer sinnvoll. Beispiel für Debian/Ubuntu:

```
sudo adduser codex
sudo mkdir -p /srv/codex-projects
sudo chown codex:codex /srv/codex-projects
```

Auf Arch/Fedora/RHEL-Familien kann ein Benutzer beispielsweise mit `useradd` angelegt werden. Verwende dabei die für deine Distribution und Unternehmensrichtlinie übliche Benutzerverwaltung.

### 12.3 SSH-Key erzeugen

```
ssh-keygen -t ed25519
ssh-copy-id codex@server.example.com
```

### 12.4 SSH-Alias

```
Host codex-dev
  HostName server.example.com
  User codex
  IdentityFile ~/.ssh/id_ed25519
```

### 12.5 Entscheidender Test

```
ssh codex-dev 'command -v codex'
```

#### Remote-Voraussetzung

Wenn der letzte Befehl keinen Codex-Pfad liefert, kann die Desktop-App den Remote-App-Server nicht wie dokumentiert über die Login-Shell starten. Erst den PATH des Remote-Benutzers korrigieren, dann weiterarbeiten.

## 13. Codex Remote, Linux Desktop Preview und remote-control

### 13.1 Was OpenAI aktuell als Remote-Host nennt

Die aktuelle OpenAI-Seite zu Remote Connections sagt: Remote unterstützt Hosts, auf denen die ChatGPT-Desktop-App unter macOS oder Windows läuft. Gesteuert werden kann von iOS oder Android sowie - je nach Rollout - von einem weiteren Mac oder Windows-Rechner.

### 13.2 Linux Desktop Preview ist eine andere Aussage

Der Changelog vom 11. August 2026 nennt eine ChatGPT-Desktop-App-Preview für Ubuntu, Debian und Fedora auf x64 und ARM64. Das belegt eine Linux-Desktop-App, aber nicht automatisch die Remote-Host-Freigabe. Solange die Remote-Dokumentation Linux nicht als Host nennt, sollte man beides nicht gleichsetzen.

#### Praxisfolgerung

Nutze Linux für Codex CLI und SSH-Remote-Projekte. Bezeichne Linux nicht als offiziell unterstützten Remote-Host für die Smartphone-Steuerung, solange OpenAI dies in den Remote Connections nicht freigibt.

### 13.3 Experimenteller CLI-Befehl remote-control

```
codex remote-control
codex remote-control start
codex remote-control stop
codex remote-control pair
codex remote-control pair --json
```

OpenAI kennzeichnet `codex remote-control` als Experimental. Nach Start des Daemons kann `pair` einen kurzlebigen Pairing-Code erzeugen. Die JSON-Ausgabe enthält laut Dokumentation Felder wie `pairingCode`, `manualPairingCode`, `environmentId` und `expiresAt`.

#### Nicht verwechseln

`codex remote-control` ist kein Beleg dafür, dass jede Linux-Distribution als voll unterstützter ChatGPT-Remote-Host gilt. Die Funktion ist experimentell und die Produktverfügbarkeit kann sich separat entwickeln.

### 13.4 Remote Control in verwalteten Umgebungen abschalten

OpenAI dokumentiert für verwaltete Anforderungen den Schalter `allow_remote_control`. Administratoren können Remote Control damit deaktivieren:

```
allow_remote_control = false
```

## 14. Sichere Betriebsregeln für Administratoren

### 14.1 Codex nicht als root starten

**Regel**

Starte Codex im Normalfall mit einem Benutzer, dessen Rechte zur Aufgabe passen.  
`sudo codex` vergrößert den möglichen Schadensradius erheblich.

### 14.2 Git-Checkpoint vor größeren Aufgaben

```
git status
git add -A
git commit -m "Checkpoint before Codex task"
```

Nach der Agentenarbeit:

```
git status
git diff
```

### 14.3 Development -> Staging -> Production

Ein sinnvoller Betriebsweg für Agenten ist nicht "Prompt -> Production", sondern:

```
Codex
-> Development
-> Tests / Review
-> Git Commit
-> Staging
-> Freigabe
-> Production
```

### 14.4 sudo begrenzen

Wenn Codex administrative Leserechte oder einzelne Service-Aktionen benötigt, ist eine gezielte sudo-Regel besser als pauschales root. Die konkrete sudoers-Konfiguration muss an den Dienst und die Organisation angepasst und mit `visudo` validiert werden.

### 14.5 Netzwerkzugriff nicht automatisch öffnen

Ein Build kann Netzwerk brauchen, aber viele Analyseaufgaben nicht. Halte den Netzwerkzugriff der `workspace-write-Sandbox` standardmäßig aus und aktiviere ihn nur für Aufgaben, die ihn wirklich benötigen.

### 14.6 Fremde Repositorys als untrusted behandeln

Repositorys können Skripte, Build-Hooks, `AGENTS.md`, Projektkonfigurationen und andere Anweisungen enthalten. Bevor du einem Agenten Schreib- oder Ausführungsrechte gibst, prüfe Quelle, Inhalt und erwartete Build-Schritte.

# 15. Fehlerdiagnose

## 15.1 codex: command not found

```
command -v codex  
echo "$PATH"
```

Öffne anschließend eine neue Login-Shell. Bei Remote-Systemen immer zusätzlich den nicht-interaktiven SSH-Test ausführen:

```
ssh codex-dev 'command -v codex'
```

## 15.2 Login auf Headless-Systemen

```
codex login --device-auth
```

Wenn der Device-Code-Flow in der eigenen Umgebung nicht funktioniert, dokumentiert OpenAI als Fallback die Anmeldung auf einem Browser-System und das Übertragen des vorhandenen `~/.codex/auth.json` auf das Headless-System. Diese Datei ist ein Secret und muss entsprechend geschützt übertragen werden.

## 15.3 TLS- oder Corporate-Proxy-Probleme

```
export CODEX_CA_CERTIFICATE=/path/to/corporate-root-ca.pem  
codex login
```

## 15.4 SSH funktioniert, Codex Remote-Projekt aber nicht

- Mit `ssh host` prüfen, ob die Login-Shell funktioniert.
- Mit `ssh host 'command -v codex'` den PATH prüfen.
- Prüfen, ob der Remote-Benutzer Leserechte auf das Projekt hat.
- Prüfen, ob Git/Build-Werkzeuge, die der Task benötigt, auf dem Remote-System installiert sind.
- Keine experimentelle Remote-Control-Funktion als Ersatz für einen fehlerhaften SSH-Basiszugang verwenden.

## 15.5 Konfiguration wirkt nicht wie erwartet

Prüfe, ob eine projektlokale `.codex/config.toml`, ein Profil oder ein CLI-Flag die Benutzerkonfiguration übersteuert. Die Konfigurationshierarchie ist für die Fehlersuche entscheidend.

# 16. Schnellreferenz

## Installation

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
```

## Login

```
codex login
codex login --device-auth
codex login status
codex logout
```

## API-Key

```
export OPENAI_API_KEY="..."
printenv OPENAI_API_KEY | codex login --with-api-key
```

## Start

```
cd /pfad/zum/projekt
codex
```

## Wichtige interaktive Befehle

```
/init
/status
/permissions
/model
/review
```

## Remote Control - Experimental

```
codex remote-control
codex remote-control start
codex remote-control stop
codex remote-control pair
codex remote-control pair --json
```

## Empfohlener Sicherheitsstart

```
approval_policy = "on-request"
approvals_reviewer = "user"
sandbox_mode = "workspace-write"

[sandbox_workspace_write]
network_access = false
```

## 17. Checkliste vor dem ersten produktiven Einsatz

- Distribution und Benutzerkonto identifiziert.
- Codex mit dem offiziellen OpenAI-Installer installiert.
- `command -v codex` funktioniert.
- Authentifizierung getestet.
- `~/.codex/config.toml` angelegt und bewusst konfiguriert.
- Sandbox auf `workspace-write` oder restriktiver gesetzt.
- Approval Policy auf `on-request` oder strenger gesetzt.
- Keine pauschale `root`-Ausführung.
- Git-Repository und Rollback-Punkt vorhanden.
- Secrets nicht im Repository gespeichert.
- SSH-Remote-Benutzer hat nur die benötigten Rechte.
- Bei Remote-Projekten funktioniert `ssh host 'command -v codex'`.
- Experimental-Funktionen nicht als garantiert produktionsreif behandelt.
- Development/Staging von Production getrennt.

### Kurzform

Codex bekommt genau so viel Zugriff wie nötig - nicht so viel wie technisch möglich.

# 18. Quellen und Faktenbasis

Fuer dieses Handout wurden bevorzugt Originalquellen verwendet. Die folgenden Links sind im PDF anklickbar.

## OpenAI / Codex

[Codex CLI - Installation und Quickstart](#)  
[Authentication - Login, Device Code, API-Key, Corporate CA](#)  
[Config Basics - Konfigurationspfade und Prioritaet](#)  
[Configuration Reference - sandbox, approvals, Credential Store, Remote Control](#)  
[Sandbox - read-only, workspace-write, danger-full-access](#)  
[Remote Connections - Remote Hosts und SSH-Remote-Projekte](#)  
[Developer Commands - codex remote-control und codex login status](#)  
[Changelog - Linux Desktop Preview vom 11. August 2026](#)

## Linux-Distributionen

[ArchWiki - OpenSSH / sshd.service](#)  
[CachyOS Wiki](#)  
[Debian Wiki - OpenSSH](#)  
[Ubuntu Server Docs - OpenSSH](#)  
[Fedora Documentation](#)  
[Nobara New User Guide](#)  
[Nobara - System Update Hinweise](#)  
[Red Hat Enterprise Linux Documentation](#)

### Aktualitätshinweis

Codex entwickelt sich schnell. Besonders die Linux-Desktop-App, Remote und remote-control können sich nach dem Faktenstand dieses PDFs aendern. Fuer kritische Installationen immer die verlinkte OpenAI-Dokumentation gegenprüfen.

# Weiterlesen auf KI-Buster

Auf KI-Buster findest du weitere praxisnahe Anleitungen zu Linux, Codex, ChatGPT, MCP, KI-Agenten, Automatisierung und IT-Sicherheit.

## Website

<https://kibuster.braetter-int.de>

Dieses Handout ist für den kostenlosen Download vorgesehen. Du darfst den Link zur Downloadseite gerne weitergeben.

## KI-Buster

**Verständlich. Praxisnah. Sicherheitsbewusst. Direkt anwendbar.**