

Vigthoria CLI Benutzerhandbuch

Vigthoria Labs

Januar 2026

Contents

Vigthoria CLI Benutzerhandbuch	2
KI-gesteuerter Terminal-Codierassistent	2
Inhaltsverzeichnis	2
Einführung	3
Was ist Vigthoria CLI?	3
Hauptfunktionen	3
Für wen ist das?	3
Voraussetzungen	3
Installation	4
Schnellinstallation (Empfohlen)	4
npm Globale Installation	4
Manuelle Installation	4
Installation überprüfen	4
Schnellstartanleitung	5
Schritt 1: Anmelden	5
Schritt 2: Status prüfen	5
Schritt 3: Mit dem Codieren beginnen	5
Schritt 4: Ihr erstes KI-Gespräch	5
Authentifizierungsstatus prüfen	6
Abmelden	7
Interaktiver Chat-Modus	7
Chat starten	7
Chat-Befehle	7
Beispiel-Chat-Sitzung	8
Dateioperationen	8
Datei bearbeiten	8
Code-Probleme beheben	9
Code erklären	9
Codegenerierung	9
Code aus Beschreibung generieren	9
Unterstützte Sprachen	10
Beispiele	10
Code-Review & Fehlerbehebung	10
Code-Review	10

Review-Ausgabe	10
Konfiguration	11
Interaktive Konfiguration	11
Einzelwerte setzen	11
Werte abrufen	11
Alle Einstellungen auflisten	11
Auf Standardwerte zurücksetzen	11
Projekt initialisieren	11
Verfügbare KI-Modelle	11
Modellauswahl	12
? Lokaler Modus mit Vigthoria Inference	12
Voraussetzungen	12
Verfügbare Native Vigthoria Modelle	12
Lokalen Modus verwenden	13
Lokale Modellzuordnung	13
Vorteile des Lokalen Modus	13
Systemanforderungen	13
Beispiel für Lokale Sitzung	14
Befehlsreferenz	14
Kernbefehle	14
Auth-Befehle	14
Konfig-Befehle	15
Arbeitsabläufe & Beispiele	15
Arbeitsablauf 1: Feature-Entwicklung	15
Arbeitsablauf 2: Code-Review vor dem Commit	15
Fehlerbehebung	16
Häufige Probleme	16
Alles zurücksetzen	16
FAQ	16
Allgemein	16
Technisch	17
Konto	17
Hilfe erhalten	17

Vigthoria CLI Benutzerhandbuch

KI-gesteuerter Terminal-Codierassistent

Version 1.0.0

(C) 2026 Vigthoria Labs

Inhaltsverzeichnis

1. [Einführung](#)
2. [Installation](#)
3. [Schnellstartanleitung](#)
4. [Authentifizierung](#)

5. [Interaktiver Chat-Modus](#)
 6. [Dateioperationen](#)
 7. [Codegenerierung](#)
 8. [Code-Review & Fehlerbehebung](#)
 9. [Konfiguration](#)
 10. [Verfügbare KI-Modelle](#)
 11. [Integration mit Vigthoria Coder](#)
 12. [Befehlsreferenz](#)
 13. [Arbeitsabläufe & Beispiele](#)
 14. [Fehlerbehebung](#)
 15. [FAQ](#)
-

Einführung

Was ist Vigthoria CLI?

Vigthoria CLI ist ein leistungsstarker Terminal-basierter Codierassistent, der die volle Leistungsfähigkeit der KI-Modelle von Vigthoria in Ihre Befehlszeile bringt. Egal ob Sie lokal, über SSH oder in einer Headless-Umgebung arbeiten - Vigthoria CLI bietet intelligente Code-Unterstützung, ohne das Terminal zu verlassen.

Hauptfunktionen

Funktion	Beschreibung
Interaktiver Chat	Natürlichsprachliche Codierunterstützung mit Kontextbewusstsein
Dateibearbeitung	KI-gesteuerte Codeänderungen mit Diff-Vorschau
Codegenerierung	Vollständigen Code aus Beschreibungen generieren
Code-Erklärung	Komplexen Code mit detaillierten Erklärungen verstehen
Fehlerbehebung	Automatische Problemerkennung und -behebung
Code-Review	Qualitätsanalyse mit umsetzbaren Vorschlägen
Multi-Modell-Unterstützung	Zugriff auf alle Vigthoria KI-Modelle je nach Abonnement
Projektkontext	Versteht Ihre Projektstruktur und Abhängigkeiten

Für wen ist das?

- **Backend-Entwickler** - Die hauptsächlich in Terminal-Umgebungen arbeiten
- **DevOps-Ingenieure** - Für Skripterstellung und Automatisierungsaufgaben
- **Remote-Arbeiter** - SSH-Sitzungen zu entfernten Servern
- **Power-User** - Die tastaturgesteuerte Arbeitsabläufe bevorzugen
- **CI/CD-Integration** - Automatisierte Codegenerierung und Review

Voraussetzungen

Komponente	Minimum	Empfohlen
Node.js	18.0.0	20.x LTS
npm	8.x	10.x
OS	Linux, macOS, Windows	Linux/macOS
Internet	Erforderlich	Stabile Verbindung
Terminal	Beliebig	iTerm2, Warp, Kitty

Installation

Schnellinstallation (Empfohlen)

```
curl -fsSL https://cli.vigthoria.io/install.sh | bash
```

Dieses Skript wird: 1. Die Node.js-Version prüfen 2. Vigthoria CLI global installieren 3. Shell-Vervollständigungen einrichten 4. Befehlsverknüpfungen erstellen

npm Globale Installation

```
npm install -g vigthoria-cli
```

Manuelle Installation

```
# Von Vigthoria Market klonen
git clone https://market.vigthoria.io/vigthoria/vigthoria-cli.git
cd vigthoria-cli
```

```
# Oder von Vigthoria Community
git clone https://community.vigthoria.io/repos/vigthoria-cli.git
cd vigthoria-cli
```

```
# Abhängigkeiten installieren
npm install
```

```
# Kompilieren
npm run build
```

```
# Global verknüpfen
npm link
```

Installation überprüfen

```
vigthoria --version
# Ausgabe: 1.0.0
```

```
vigthoria --help
# Zeigt alle verfügbaren Befehle
```

Schnellstartanleitung

Schritt 1: Anmelden

```
vigthoria login
```

Wählen Sie Ihre Anmeldemethode: - **E-Mail & Passwort** - Geben Sie Ihre Vigthoria-Kontodaten ein - **API-Token** - Verwenden Sie ein Token aus Ihrem Vigthoria Coder Dashboard - **Browser-Anmeldung** - OAuth-Flow (offnet Browser)

Schritt 2: Status prufen

```
vigthoria status
```

Dies zeigt: - Kontoinformationen - Abonnementplan - Verfügbare Modelle - API-Konnektivität

Schritt 3: Mit dem Codieren beginnen

```
# Interaktiven Chat starten
```

```
vigthoria chat
```

```
# Oder den kurzen Alias verwenden
```

```
vig c
```

Schritt 4: Ihr erstes KI-Gespräch

```
you > Erstelle eine Funktion zur E-Mail-Validierung in TypeScript
```

```
vigthoria >
```

Hier ist eine robuste E-Mail-Validierungsfunktion:

```
```typescript
function validateEmail(email: string): boolean {
 const emailRegex = /^[^s@]+@[^\s@]+\.\[^s@]+\$/;
 return emailRegex.test(email);
}

// Erweiterte Version mit detaillierter Validierung
interface EmailValidationResult {
 valid: boolean;
 reason?: string;
}

function validateEmailDetailed(email: string): EmailValidationResult {
 if (!email) {
 return { valid: false, reason: 'E-Mail ist erforderlich' };
 }

 if (email.length > 254) {
 return { valid: false, reason: 'E-Mail zu lang' };
 }
}
```

```

const regex = /^[a-zA-Z0-9.!#$%&!*+/?^_`{|}~-]+@[a-zA-Z0-9](:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?$/;

if (!regex.test(email)) {
 return { valid: false, reason: 'Ungultiges E-Mail-Format' };
}

return { valid: true };
}

Authentifizierung

Anmeldemethoden

E-Mail & Passwort

```bash
vigthoria login
# Wählen Sie: E-Mail & Passwort
# Geben Sie Ihre E-Mail und Ihr Passwort ein
```

```

**API-Token** Holen Sie Ihr Token von: <https://coder.vigthoria.io/settings/api>

```

Interaktiv
vigthoria login
Wählen Sie: API Token
Fügen Sie Ihr Token ein

Oder direkt
vigthoria login --token IHR_API_TOKEN

```

**Umgebungsvariable** Für CI/CD oder automatisierte Umgebungen:

```

export VIGTHORIA_TOKEN="ihr-api-token"
vigthoria status # Verwendet das Umgebungstoken

```

**Authentifizierungsstatus prüfen**

```
vigthoria status
```

Ausgabe:

```
??? Kontostatus ???
```

Konto:

```
E-Mail: user@example.com
Benutzer-ID: usr_abc123
```

Abonnement:

Plan: PRO

Status: Aktiv

Lauft ab: 31.12.2026 (365 Tage)

Verfügbare Modelle:

- o vigthoria-fast - Schnelle Antworten (1.1B)
- o vigthoria-mini - Ausgewogen (3.8B)
- o vigthoria-code - Code-Spezialist (8B)
- o vigthoria-creative - Kreativ (9B)

API-Status:

Server: Online

Endpunkt: <https://coder.vigthoria.io>

## Abmelden

```
vigthoria logout
```

---

## Interaktiver Chat-Modus

Der Chat-Modus ist die primäre Methode zur Interaktion mit der Vigthoria KI. Er behält den Gesprächskontext bei und versteht Ihr Projekt.

### Chat starten

```
vigthoria chat
oder
vig c

Mit spezifischem Modell
vigthoria chat --model pro

Mit Projektkontext
vigthoria chat --project /pfad/zum/projekt
```

### Chat-Befehle

Verwenden Sie im Chat-Modus diese speziellen Befehle:

| Befehl        | Beschreibung                                |
|---------------|---------------------------------------------|
| /file <pfad>  | Dateiinhalt zum Gespräch hinzufügen         |
| /edit <pfad>  | Bearbeitung einer bestimmten Datei starten  |
| /diff         | Ausstehende Änderungen anzeigen             |
| /apply        | Ausstehende Änderungen auf Dateien anwenden |
| /model <name> | KI-Modell wechseln                          |
| /clear        | Gesprächsverlauf löschen                    |

| Befehl           | Beschreibung                |
|------------------|-----------------------------|
| /help            | Verfügbare Befehle anzeigen |
| /exit oder /quit | Chat-Modus beenden          |

## Beispiel-Chat-Sitzung

```
+-----+
| Vigthoria Chat |
| Modell: vigthoria-code |
| Plan: PRO |
| Projekt: /home/user/mein-projekt |
+-----+
```

Geben Sie Ihre Nachricht ein oder /help fur Befehle. Strg+C zum Beenden.

```
you > /file src/api.ts
? src/api.ts zum Kontext hinzugefugt (245 Zeilen)
```

you > Fuge Rate-Limiting zum Authentifizierungspunkt hinzufügen

```
vigthoria >
Ich werde Rate-Limiting zu Ihrem Authentifizierungsendpunkt hinzufügen...
```

[... detaillierte Antwort mit Code ...]

```
you > /edit src/api.ts
Welche Aenderungen mochten Sie vornehmen?
> Fuge den Rate-Limiting-Code hinzu, den du gerade gezeigt hast
```

```
[Diff-Vorschau]
Aenderungen anwenden? (J/n) j
? Aenderungen auf src/api.ts angewendet
```

## Dateioperationen

## Datei bearbeiten

```
vigthoria edit <datei> [optionen]
```

Optionen: `-i, --instruction <text>` - Bearbeitungsanweisung `-m, --model <model>` - Zu verwendendes KI-Modell

## Beispiele:

```
Interaktive Bearbeitung
vigthoria edit src/utils.ts

Mit Anweisung
vigthoria edit src/api.ts -i "Füge Fehlerbehandlung zu allen async-Funktionen hinzu"

Mit spezifischem Modell
vigthoria edit src/complex.ts -m pro
```

## Code-Probleme beheben

```
vigthoria fix <datei> [optionen]
```

Optionen: - -t, --type <typ> - Behebungstyp: bugs, style, security, performance - --apply  
- Behebungen automatisch ohne Bestätigung anwenden

Beispiele:

```
Bugs finden und beheben
vigthoria fix src/index.ts -t bugs
```

```
Sicherheitsprobleme mit Auto-Apply beheben
vigthoria fix src/auth.ts -t security --apply
```

```
Performance-Probleme beheben
vigthoria fix src/heavy-computation.ts -t performance
```

## Code erklären

```
vigthoria explain <datei> [optionen]
```

Optionen: - -l, --lines <bereich> - Zeilenbereich (z.B. 1-50) - -d, --detail <stufe> - brief, normal, detailed

Beispiele:

```
Gesamte Datei erklären
vigthoria explain src/algorithm.ts
```

```
Bestimmte Zeilen erklären
vigthoria explain src/utils.ts -l 50-100
```

```
Detaillierte Erklärung
vigthoria explain src/complex.ts -d detailed
```

---

## Codegenerierung

### Code aus Beschreibung generieren

```
vigthoria generate <beschreibung> [optionen]
```

Optionen: - -l, --language <sprache> - Zielsprache (Standard: typescript) - -o, --output <datei> - Ausgabedateipfad - -m, --model <model> - Zu verwendendes KI-Modell

## Unterstutzte Sprachen

- TypeScript / JavaScript
- Python
- Rust
- Go
- Java
- C# / C++
- Ruby
- PHP
- Swift
- Kotlin

## Beispiele

```
TypeScript-Code generieren
vigthoria generate "REST API Endpunkt fur Benutzerauthentifizierung"

Python mit Ausgabedatei generieren
vigthoria generate "Datenbank-Migrationsskript fur PostgreSQL" -l python -o migrate.py

Mit spezifischem Modell generieren
vigthoria generate "Hochleistungs-Caching-Schicht" -l rust -m pro
```

---

## Code-Review & Fehlerbehebung

### Code-Review

```
vigthoria review <datei> [optionen]
Optionen: - -f, --format <format> - text, json, markdown
```

### Review-Ausgabe

```
??? Reviewing: src/api.ts ???
Sprache: typescript | Zeilen: 245
```

```
Qualitatsbewertung: 78/100
????????????????????????????????????
```

```
??? Probleme (4) ???
```

```
? [security] Zeile 45: Potenzielle SQL-Injection-Schwachstelle
? [performance] Zeile 112: Unnotiges Re-Render in Schleife
? [style] Zeile 67: Magische Zahl sollte eine benannte Konstante sein
i [info] Zeile 189: Optionale Verkettung in Betracht ziehen
```

??? Vorschlage ???

1. Eingabeverifikation zur `createUser`-Funktion hinzufügen
  2. Rate-Limiting für öffentliche Endpunkte implementieren
  3. Umgebungsvariablen für Konfigurationswerte verwenden
  4. JSDoc-Kommentare zu exportierten Funktionen hinzufügen
- 

## Konfiguration

### Interaktive Konfiguration

`vigthoria config`

#### Einzelwerte setzen

```
vigthoria config --set model=vigthoria-code
vigthoria config --set theme=dark
vigthoria config --set autoApply=true
vigthoria config --set maxTokens=8192
```

#### Werte abrufen

```
vigthoria config --get model
Ausgabe: vigthoria-code
```

#### Alle Einstellungen auflisten

`vigthoria config --list`

#### Auf Standardwerte zurücksetzen

`vigthoria config --reset`

#### Projekt initialisieren

Projektspezifische Konfiguration erstellen:

```
cd /pfad/zum/projekt
vigthoria init
```

---

## Verfügbare KI-Modelle

| Modell                           | Grosse | Tokens | Optimal für                              | Plane               |
|----------------------------------|--------|--------|------------------------------------------|---------------------|
| <code>vigthoria-fast</code> 1.1B |        | 4K     | Schnelle Antworten,<br>einfache Aufgaben | Free, Pro,<br>Ultra |

| Modell             | Grosse | Tokens | Optimal fur                        | Plane            |
|--------------------|--------|--------|------------------------------------|------------------|
| vigthoria-mini     | 3.8B   | 8K     | Ausgewogene Leistung               | Free, Pro, Ultra |
| vigthoria-code     | 8B     | 16K    | Codegenerierung, Debugging         | Pro, Ultra       |
| vigthoria-creative | 16B    | 16K    | Kreatives Schreiben, Dokumentation | Pro, Ultra       |
| vigthoria-pro      | 32B    | 32K    | Komplexe Aufgaben, Architektur     | Ultra            |
| vigthoria-ultra    | 70B    | 64K    | Maximale Fähigkeiten               | Ultra            |

## Modellauswahl

```
Im Chat
/model code

Befehlszeile
vigthoria chat --model pro
vigthoria generate "..." --model ultra
```

---

## ? Lokaler Modus mit Vigthoria Inference

Der lokale Modus ermöglicht die Nutzung von Vigthoria CLI mit den **nativen Vigthoria LLMs** über den Vigthoria Inference Server. Dies gibt Ihnen Zugriff auf dieselben leistungsstarken Modelle wie in der Cloud, die lokal auf Ihrer Maschine laufen.

## Voraussetzungen

Installieren Sie Vigthoria Inference von Vigthoria Market:

```
Von Vigthoria Market klonen
git clone https://market.vigthoria.io/vigthoria/vigthoria-inference.git
cd vigthoria-inference

Oder von Vigthoria Community
git clone https://community.vigthoria.io/repos/vigthoria-inference.git
cd vigthoria-inference

Abhängigkeiten installieren
pip install -r requirements.txt

Vigthoria Inference Server starten (Port 8010)
python3 vigthoria_inference_server_v2.py
```

## Verfügbare Native Vigthoria Modelle

Vigthoria Inference bietet Zugriff auf alle nativen Vigthoria LLMs:

| Modell                    | Grosse | Beschreibung             |
|---------------------------|--------|--------------------------|
| vigthoria-mini-0.6b       | 0.6B   | Ultra-schnelle Antworten |
| vigthoria-fast-1.7b       | 1.7B   | Schnelles Streaming      |
| vigthoria-balanced-4b     | 4B     | Allzweck                 |
| vigthoria-code-v2-8b      | 8B     | Code-Spezialist          |
| vigthoria-creative-9b-v4  | 9B     | Kreatives Schreiben      |
| vigthoria-music-master-4b | 4B     | Musikproduktion          |

## Lokalen Modus verwenden

```
Chat mit Vigthoria Inference starten
vigthoria chat --local
vigthoria chat -l
```

```
Agent-Modus mit lokaler Inferenz
vigthoria agent --local
```

```
Mit spezifischem Vigthoria-Modell
vigthoria chat --local --model code
vigthoria chat -l -m balanced
```

## Lokale Modellzuordnung

Bei Verwendung von `--local` verbindet sich Vigthoria CLI mit Vigthoria Inference (Port 8010):

| Alias                 | Vigthoria Modell          | Grosse |
|-----------------------|---------------------------|--------|
| <code>mini</code>     | vigthoria-mini-0.6b       | 0.6B   |
| <code>fast</code>     | vigthoria-fast-1.7b       | 1.7B   |
| <code>balanced</code> | vigthoria-balanced-4b     | 4B     |
| <code>code</code>     | vigthoria-code-v2-8b      | 8B     |
| <code>creative</code> | vigthoria-creative-9b-v4  | 9B     |
| <code>music</code>    | vigthoria-music-master-4b | 4B     |

## Vorteile des Lokalen Modus

| Vorteil                | Beschreibung                       |
|------------------------|------------------------------------|
| <b>Datenschutz</b>     | Code verlässt nie Ihre Maschine    |
| <b>Native Modelle</b>  | Volle Leistung der Vigthoria LLMs  |
| <b>Offline</b>         | Funktioniert ohne Internet         |
| <b>Keine Auth</b>      | Keine Anmeldung erforderlich       |
| <b>Geschwindigkeit</b> | Geringe Latenz für lokale Inferenz |
| <b>Kostenlos</b>       | Keine API-Kosten                   |

## Systemanforderungen

| Modell        | VRAM benötigt | Empfohlene GPU |
|---------------|---------------|----------------|
| Mini (0.6B)   | 2 GB          | Jede GPU       |
| Fast (1.7B)   | 4 GB          | RTX 3060+      |
| Balanced (4B) | 8 GB          | RTX 3070+      |
| Code (8B)     | 16 GB         | RTX 4080+      |
| Creative (9B) | 20 GB         | RTX 4090/A6000 |

## Beispiel fur Lokale Sitzung

```
$ vigthoria chat -l -m code
```

[Lokaler Modus] Modell: vigthoria-code-v2-8b | Inferenz: localhost:8010

you > Schreibe eine Python-Funktion zum Umkehren eines Strings

vigthoria > Hier ist eine Python-Funktion zum Umkehren eines Strings:

```
def reverse_string(s: str) -> str:
 """Kehrt den Eingabestring um."""
 return s[::-1]
```

```
Alternative mit reversed()
```

```
def reverse_string_alt(s: str) -> str:
 return ''.join(reversed(s))
```

## Befehlsreferenz

## Kernbefehle

| Befehl                      | Alias | Beschreibung              |
|-----------------------------|-------|---------------------------|
| vigthoria chat              | vig c | Interaktiven Chat starten |
| vigthoria edit <datei>      | vig e | Datei mit KI bearbeiten   |
| vigthoria generate <beschr> | vig g | Code generieren           |
| vigthoria explain <datei>   | vig x | Code erklären             |
| vigthoria fix <datei>       | vig f | Code-Probleme beheben     |
| vigthoria review <datei>    | vig r | Code-Qualität prüfen      |

## Auth-Befehle

| Befehl                        | Beschreibung                   |
|-------------------------------|--------------------------------|
| <code>vigthoria login</code>  | Bei Vigthoria authentifizieren |
| <code>vigthoria logout</code> | Authentifizierung loschen      |
| <code>vigthoria status</code> | Kontostatus anzeigen           |

## Konfig-Befehle

| Befehl                                        | Beschreibung                        |
|-----------------------------------------------|-------------------------------------|
| <code>vigthoria config</code>                 | Interaktive Konfiguration           |
| <code>vigthoria config --list</code>          | Alle Einstellungen anzeigen         |
| <code>vigthoria config --set key=value</code> | Wert setzen                         |
| <code>vigthoria config --get key</code>       | Wert abrufen                        |
| <code>vigthoria config --reset</code>         | Auf Standardwerte zurücksetzen      |
| <code>vigthoria init</code>                   | Projektkonfiguration initialisieren |

## Arbeitsabläufe & Beispiele

### Arbeitsablauf 1: Feature-Entwicklung

```
1. Chat mit Projektkontext starten
cd ~/projekte/meine-app
vigthoria chat

2. Das Feature besprechen
you > Ich muss Benutzeroauthentifizierung mit JWT hinzufügen

3. Den Code generieren
vigthoria generate "JWT-Authentifizierungs-Middleware für Express" -o src/auth/middleware.ts

4. Den generierten Code überprüfen
vigthoria review src/auth/middleware.ts

5. Eventuelle Probleme beheben
vigthoria fix src/auth/middleware.ts -t security
```

### Arbeitsablauf 2: Code-Review vor dem Commit

```
Pre-Commit-Hook erstellen
#!/bin/bash
.git/hooks/pre-commit

for file in $(git diff --cached --name-only --diff-filter=ACM | grep '\.ts$'); do
 # Jede Datei überprüfen
 result=$(vigthoria review "$file" -f json)
 score=$(echo "$result" | jq '.score')
```

```
if ["$score" -lt 70]; then
 echo "Code-Review fehlgeschlagen fur $file (Punktzahl: $score)"
 exit 1
fi
done
```

---

## Fehlerbehebung

### Haufige Probleme

#### “Nicht authentifiziert” Fehler

```
Losung: Erneut anmelden
vigthoria login
```

“Abonnement abgelaufen” Warnung Ihr Abonnement muss erneuert werden. Besuchen Sie: <https://coder.vigthoria.io/subscription>

#### “API-Verbindung fehlgeschlagen” Fehler

```
API-Status prufen
vigthoria status

Alternativen Endpunkt versuchen
vigthoria config --set apiUrl=https://backup.coder.vigthoria.io
```

### Alles zurücksetzen

```
Konfiguration und Auth löschen
vigthoria logout
vigthoria config --reset

Neuinstallieren
npm uninstall -g vigthoria-cli
npm install -g vigthoria-cli
```

---

## FAQ

### Allgemein

**F: Ist Vigthoria CLI kostenlos?** A: Grundfunktionen sind kostenlos. Erweiterte Modelle erfordern ein Pro- oder Ultra-Abonnement.

**F: Funktioniert es offline?** A: Nein, eine Internetverbindung ist für KI-Operationen erforderlich.

**F: Wird mein Code in die Cloud gesendet?** A: Ja, Code wird zur KI-Verarbeitung an Vigthoria-Server gesendet. Wir speichern Ihren Code nicht dauerhaft.

## Technisch

**F: Welche Modelle kann ich verwenden?** A: Abhangig von Ihrem Abonnement. Prufen Sie mit `vigthoria status`.

**F: Kann ich es in CI/CD verwenden?** A: Ja, verwenden Sie die Umgebungsvariable `VIGTHORIA_TOKEN` zur Authentifizierung.

## Konto

**F: Kann ich mein Vigthoria Coder-Konto verwenden?** A: Ja, dasselbe Konto funktioniert fur CLI, Web-IDE und VS Code-Erweiterung.

**F: Wie kann ich mein Abonnement upgraden?** A: Besuchen Sie <https://coder.vigthoria.io/subscription>

---

## Hilfe erhalten

- **Dokumentation:** <https://docs.vigthoria.io/cli>
- **Discord-Community:** <https://discord.gg/vigthoria>
- **GitHub Issues:** <https://github.com/vigthoria/vigthoria-cli/issues>
- **E-Mail-Support:** [support@vigthoria.io](mailto:support@vigthoria.io) (Pro/Ultra-Abonnenten)

---

**Viel Spass beim Codieren mit Vigthoria CLI! ?**

*Mit ? erstellt von Vigthoria Technologie*