PRODUKTIVITÄT

Eigenes VPN bauen: kostenloser Cloud-Server in unter einer Stunde

Ein eigener VPN-Server bei Oracle Cloud kostet dauerhaft nichts und schützt deine Verbindung in fremden WLANs. Schritt für Schritt vom Konto bis zur ersten geprüften Verbindung – mit Emergent als KI-Helfer an deiner Seite und ohne eine einzige Stelle, an der du raten musst.

0 € pro Monat | **45 Min. Aufwand** | **Windows · macOS · iOS · Android** | **Keine Vorkenntnisse**

01 Warum überhaupt ein eigenes VPN

Du sitzt im Hotel-WLAN und loggst dich ins Online-Banking ein. Zwischen deinem Laptop und der Bank liegt ein Netzwerk, das jemand anderes betreibt und das du nicht kennst. Ein VPN legt einen verschlüsselten Tunnel darüber: Der WLAN-Betreiber sieht nur noch, dass Daten fließen, nicht mehr wohin.

Die meisten kaufen dafür ein Abo. Denselben Tunnel kannst du aber auch selbst betreiben – auf einem Server, der bei Oracle Cloud dauerhaft nichts kostet.

Der Unterschied ist nicht nur der Preis. Bei einem Anbieter läuft dein gesamter Datenverkehr über fremde Server, und du musst dem Versprechen glauben, dass nichts mitgeschrieben wird. Bei deinem eigenen Server weißt du es – weil du der Einzige bist, der Zugriff hat.

0 €	45 Min.	1 Land
PRO MONAT	AUFWAND	GRATIS

Was dieser Guide leistet

Er führt dich vom leeren Oracle-Konto bis zur geprüften Verbindung. Jeder Befehl steht fertig zum Kopieren, und über jedem Befehl steht ein Label, das dir sagt, wo genau du ihn einfügst. An den drei kniffligen Stellen – Länderwahl, Konfiguration, Fehlersuche – holst du dir mit fertigen Prompts Rückendeckung von **Emergent**, deinem KI-Helfer für diesen Aufbau. Was das ist und wie du es einrichtest, klärt direkt das nächste Kapitel.

KURZ VORWEG

Für Privatsphäre in fremden Netzen ist ein eigener Server hervorragend. Was er zuverlässig kann und wo seine Grenzen liegen, steht ehrlich in Kapitel 15 – damit du hinterher nicht enttäuscht bist.

02 Dein Helfer beim Aufbau: Emergent

Emergent ist eine KI-Plattform, mit der du per Chat Software baust und technische Fragen klärst. Du beschreibst in normalem Deutsch, was du vorhast, und bekommst konkrete Antworten, fertige Dateien und ehrliche Einschätzungen zurück. Genau dafür setzen wir es hier ein: an den Stellen, wo eine Anleitung allein nicht reicht, weil deine Situation eine eigene Antwort braucht.

So richtest du dir den Zugang ein

01 Öffne diesen Link: **app.emergent.sh/?utm_shift=redirect&via=gptmarlon**

02 Klicke auf **Sign up** und erstelle dir ein Konto – am schnellsten geht es mit deinem Google-Konto.

03 Bestätige die E-Mail, falls du dich mit einer E-Mail-Adresse angemeldet hast.

04 Klicke auf **New Chat**. In dieses Eingabefeld kopierst du später die Prompts aus diesem Guide.

Der Start ist kostenlos, für diesen Guide reicht der Gratis-Plan. Die drei Prompts findest du genau an den Stellen, wo du sie brauchst: bei der Länderwahl (Kapitel 05), bei der Konfiguration (Kapitel 10) und bei der Fehlersuche (Kapitel 16).

Zwei Orte, streng getrennt

In diesem Guide kopierst du zwei Arten von Text, und die dürfen nie durcheinandergeraten. Damit das nicht passiert, steht über jedem Block ein Label:

DIE VIER LABELS

IN EMERGENT Prompts – Anweisungen in normaler Sprache, die du in den Emergent-Chat einfügst.

POWERSHELL / TERMINAL Befehle für deinen eigenen Computer.

AUF DEM SERVER Befehle für den Server – dasselbe Fenster, aber erst nachdem du in Kapitel 07 verbunden bist.

IM BROWSER Klickwege auf einer Website, ganz normal mit der Maus.

WICHTIG

Emergent ist in erster Linie eine Plattform, die Anwendungen **baut**. Stellst du nur eine Frage, kann es passieren, dass sie ein Projekt anlegt, statt zu antworten. Deshalb beginnt jeder Prompt in diesem Guide mit „Antworte mir direkt hier im Chat, baue keine App“ – lösch diesen Satz nicht heraus. Und: Deine privaten Schlüssel haben in keinem Chat etwas verloren, bei keinem Anbieter.

03 Was du brauchst – und was es kostet

Ein Oracle-Cloud-Konto, einen Computer und etwa 45 Minuten. Terminal-Erfahrung ist nicht nötig – alle Befehle stehen fertig im Guide.

Die Kreditkarte ist Pflicht

Auch für das Gratis-Angebot. Oracle bucht einmalig rund einen Euro zur Prüfung und gibt den Betrag wieder frei. Debit-, Prepaid- und virtuelle Karten werden häufig abgelehnt – eine echte Kreditkarte funktioniert am zuverlässigsten, und die Rechnungsadresse sollte zum gewählten Land passen.

Was genau kostenlos bleibt

Oracle nennt das Paket **Always Free**: dauerhaft gratis, kein Testzeitraum, kein Ablaufdatum. Für dein VPN zählen daraus zwei Zahlen:

Server	Die kleinen Größen aus Kapitel 06 sind dauerhaft gratis. Ein privates VPN lastet sie nicht einmal ansatzweise aus.
Datenverkehr	**10 Terabyte pro Monat** sind frei. Zum Vergleich: Das sind über 3.000 Stunden Video in HD-Qualität.

Ob eine Person das VPN nutzt oder du zehn Geräte von Familie und Freunden anbindest, ändert daran nichts. Solange ihr zusammen unter den 10 TB im Monat bleibt, kostet es keinen Cent – und im normalen Alltag bleibt ihr weit darunter.

Ein Land gratis – nicht mehrere

Bei der Anmeldung wählst du eine **Heimatregion**: das Rechenzentrum und damit das Land, aus dem dein Server kommt. Nur dort gilt das Gratis-Paket, und die Wahl ist endgültig. Ein zweiter Server in einem anderen Land läuft über dasselbe Konto **nicht** kostenlos. Ein Konto, ein Land: So bleibt es bei 0 €.

WICHTIG

Das Konto wandelt sich **nicht** von allein in ein Bezahlkonto um. Stößt du an eine Gratis-Grenze, wird gedrosselt statt abgebucht. Und: Sei während der Anmeldung mit keinem VPN verbunden – Oracle lehnt Registrierungen aus VPN- und Proxy-Netzen regelmäßig ab, und eine abgelehnte E-Mail-Adresse lässt sich nicht erneut verwenden.

CHECKLISTE VOR DEM START

Echte Kreditkarte bereit · VPN und Proxy ausgeschaltet · E-Mail-Adresse, die noch nie bei Oracle verwendet wurde · Entschieden, in welchem Land dein Server stehen soll (Kapitel 05 hilft)

04 Konto anlegen

Hier fängst du an: Öffne **oracle.com/cloud/free** im Browser und klicke auf **Start for free**. Die Anmeldung fragt mehr ab, als man für ein Gratis-Angebot erwartet, und lehnt einen Teil der Versuche ohne Erklärung ab. Mit diesen Schritten kommst du sauber durch:

IM BROWSER · ORACLE.COM/CLOUD/FREE

01 Klicke auf **Start for free** und trage deine E-Mail-Adresse ein – ein echtes, persönliches Postfach. Wegwerf-Adressen werden abgelehnt, und eine Adresse, die schon einmal ein Oracle-Konto hatte, lässt sich nicht wiederverwenden.

02 Wähle bei **Country/Territory** das Land, das deine Kreditkarte ausgestellt hat. Eine Abweichung ist einer der häufigsten stillen Ablehnungsgründe.

03 Bestätige den Link aus der E-Mail, die Oracle dir schickt.

04 Vergib ein Passwort. Vor- und Nachname wie auf der Karte hinterlegt; beim Firmennamen genügt „Privat“.

05 Wähle deine **Heimatregion** – die wichtigste Entscheidung der ganzen Anmeldung, denn sie ist endgültig. Lies erst das nächste Kapitel, bevor du hier klickst.

06 Trage die Kreditkarte ein. Oracle bucht rund einen Euro zur Prüfung ab und gibt ihn wieder frei.

07 Warte auf die E-Mail, dass dein Konto bereit ist. Das dauert zwischen wenigen Minuten und einer Stunde – vorher lässt sich die Oberfläche nicht sinnvoll benutzen.

Wenn die Anmeldung scheitert

Häufigste Ursachen in dieser Reihenfolge: aktives VPN, Karte aus einem anderen Land als angegeben, Prepaid- oder virtuelle Karte, bereits verwendete E-Mail-Adresse. Mit einer neuen Adresse und einer echten Kreditkarte klappt es meist im zweiten Anlauf.

05 Die wichtigste Entscheidung: das Land

Weil nur eine Region gratis ist, legst du mit ihr fest, als welches Land du im Netz erscheinst – dauerhaft.

Entfernung kostet Tempo

Frankfurt	rund 25 ms – praktisch nicht spürbar
Amsterdam, London	rund 30 ms – unauffällig
US-Ostküste	rund 110 ms – beim Surfen merkbar, für Video nutzbar
Tokio	rund 290 ms – in Videocalls deutlich störend

Ein Server im eigenen Land schützt deine Verbindung, verschiebt aber nichts – du erscheinst weiterhin als deutsche Verbindung. Willst du in einem anderen Land erscheinen, wähle es hier.

Unsicher? Dann ist das der Moment für den ersten Emergent-Prompt:

IN EMERGENT EINFÜGEN

PROMPT 01 — DIE RICHTIGE REGION WÄHLEN

```
Antworte mir bitte direkt hier im Chat. Baue keine App und lege kein
Projekt an - ich brauche nur eine Einschaetzung.

Ich richte nach einer Anleitung einen eigenen WireGuard-VPN-Server bei
Oracle Cloud ein und muss jetzt die Heimatregion auswaehlen. Diese Wahl
ist endgueltig, und kostenlos ist ausschliesslich diese eine Region.

Meine Situation:
- Ich wohne in: [LAND, STADT EINFÜGEN]
- Ich will das VPN vor allem fuer: [z. B. Sicherheit in fremden WLANs /
  im Ausland als mein Heimatland erscheinen / Zugriff auf mein Heimnetz]
- Videocalls oder Spiele ueber die Verbindung: [ja/nein]

Gib mir genau drei Dinge:
1. Die eine Region, die du empfiehlst - mit dem exakten Namen, wie er
   im Auswahlmenue von Oracle steht, im Format "Germany Central
   (Frankfurt)".
2. Eine Ausweichregion, falls meine erste Wahl ausgebucht ist.
3. In zwei Saetzen: warum diese Region - und was ich mir damit an
   Nachteilen einhandle.

Zwei Regeln fuer deine Antwort:
- Nenne keine konkreten Millisekunden-Werte, wenn du sie nicht sicher
  weisst. Sag stattdessen "nah", "mittel" oder "weit".
- Wenn dir eine Angabe von mir fehlt, frag nach, statt zu raten.
```

WICHTIG

Die Heimatregion lässt sich später nicht ändern – ein Wechsel bedeutet ein komplett neues Konto.

06 Lieber begleiten lassen? Emergent führt dich durch

Die nächsten Kapitel zeigen jeden Schritt einzeln. Wenn du lieber geführt werden möchtest, gib Emergent diesen Prompt: Er arbeitet einen Schritt pro Nachricht ab, wartet jeweils auf deine Ausgabe und prüft sie, bevor es weitergeht. Die beiden Eigenheiten dieses Setups, die REJECT-Regel und die Paketgröße, stecken bereits darin.

IN EMERGENT EINFÜGEN

ZUSATZ-PROMPT — DEN TUNNEL SCHRITT FÜR SCHRITT AUFBAUEN

```
Antworte mir bitte direkt hier im Chat. Baue keine App und lege kein
Projekt an.

Begleite mich Schritt fuer Schritt beim Aufbau meines eigenen
WireGuard-VPN-Servers auf einer Oracle-Cloud-Instanz mit Ubuntu 22.04.

Mein Stand: [WAS SCHON STEHT, z. B. "Konto und Server angelegt,
SSH-Schluessel heruntergeladen" oder "noch gar nichts"]
Mein Computer: [WINDOWS ODER MAC]

So arbeitest du mit mir:
- Ein Schritt pro Nachricht, nicht mehr. Warte nach jedem Schritt auf
  meine Antwort.
- Gib mir pro Schritt genau einen Befehl zum Kopieren, dazu einen Satz,
  was er bewirkt, und sag mir, welche Ausgabe ich erwarten soll.
- Ich schicke dir die Ausgabe zurueck. Pruefe sie, bevor es weitergeht.
  Sieht sie falsch aus, sag das und nenne den naechsten Befehl zur
  Eingrenzung, statt einfach fortzufahren.
- Frag mich nie nach einem privaten Schluessel. Nutz Platzhalter wie
  SERVER_PRIVAT, die echten Werte trage ich selbst ein.

Diese Vorgaben sind nicht verhandelbar:
- Tunnelnetz 10.66.1.0/24, der Server ist 10.66.1.1, das erste Geraet
  ist 10.66.1.2.
- WireGuard auf UDP-Port 51820.
- MTU = 1380 in jeder Datei, Server wie Geraete. Nicht 1420: viele
  deutsche Anschluesse fahren DS-Lite, dort werden groessere Pakete
  stillschweigend verworfen.
- Die FORWARD-Regeln mit "-I FORWARD 1" einfuegen, niemals mit "-A".
  Grund: Die Ubuntu-Abbilder von Oracle haben am Ende der FORWARD-Kette
  eine REJECT-Regel. Angehaengte Regeln stehen dahinter und greifen nie.
  Der Tunnel verbindet sich dann, aber keine Webseite laedt.
- Der Port muss zusaetzlich in der Oracle Security List als Ingress-Regel
  offen sein: Quelle 0.0.0.0/0, Protokoll UDP, Port 51820. Erinnere mich
  daran, bevor wir zum ersten Mal testen.

Zum Schluss fuehrst du mich durch die Pruefung auf ipleak.net und sagst
mir, worauf ich bei angezeigter Adresse, DNS-Servern und IPv6 achten muss.

Fang mit Schritt 1 an.
```

WICHTIG

Kein Prompt richtet den Server allein ein. Emergent hat keinen Zugriff auf deine Instanz, es begleitet dich nur. Ausgeführt werden die Befehle von dir, im Terminal.

07 Server anlegen

Jetzt legst du deine Maschine an – bei Oracle heißt so ein gemieteter Server **Instance**. Alles in diesem Kapitel passiert im Browser.

IM BROWSER · CLOUD.ORACLE.COM

01 Melde dich auf **cloud.oracle.com** an und öffne links oben das Menü (☰).

02 Klicke auf **Compute** → **Instances** → **Create Instance**.

03 Vergib oben einen Namen, zum Beispiel `mein-vpn`.

04 Bei **Image** steht das Betriebssystem – die Grundsoftware, mit der der Server läuft. Wähle **Canonical Ubuntu 22.04**, ein kostenloses Linux. Du musst es nie direkt bedienen.

05 Bei **Shape** wählst du die Größe der Maschine: **VM.Standard.A1.Flex** mit 1 OCPU und 6 GB. Falls nicht verfügbar: **VM.Standard.E2.1.Micro** – ebenfalls dauerhaft kostenlos.

06 Bei **Add SSH keys** geht es um deinen Zugang: Ein SSH-Schlüssel ist eine kleine Datei, die wie ein fälschungssicherer Passwort-Ersatz funktioniert. Wähle **Generate a key pair for me** und klicke auf **Save private key**. Sie wird genau einmal angeboten – ohne sie kommst du später nicht auf den Server.

07 Klicke auf **Create** und warte, bis der Status auf **Running** steht.

08 Notiere dir die **öffentliche IP-Adresse** (Public IP) – die Hausnummer deines Servers im Internet. Du brauchst sie zweimal.

Wenn „Out of host capacity“ erscheint

Das ist keine Störung, sondern der Normalfall. Die kostenlosen ARM-Server sind chronisch ausgebucht. Wechsle die Shape auf **VM.Standard.E2.1.Micro**: ebenfalls dauerhaft kostenlos, anderer Kapazitätstopf, meist sofort verfügbar. Die 1 GB Arbeitsspeicher reichen vollkommen – die Verschlüsselung erledigt der Linux-Kernel.

> **TIPP**
>
> Der oft empfohlene Rat, eine andere Availability Domain zu probieren, hilft nur in Regionen mit mehreren davon. Etliche – Montreal etwa – haben genau eine.

08 Mit dem Server verbinden

Dein Server läuft. Um ihn einzurichten, verbindest du dich über das Eingabefenster deines Computers – unter Windows heißt es **PowerShell**, auf dem Mac **Terminal**. Das ist nichts Kompliziertes: ein Fenster, in das du Befehle als Text eintippst, statt mit der Maus zu klicken.

01 **Windows:** Startmenü öffnen, „PowerShell“ tippen, Enter. **Mac:** `cmd + Leertaste`, „Terminal“ tippen, Enter.

02 Kopiere den folgenden Befehl in das Fenster (Windows: Rechtsklick fügt ein, Mac: `cmd + V`), ersetze vorher die beiden Platzhalter, dann Enter.

POWERSHELL / TERMINAL · DEIN COMPUTER

```
ssh -i PFAD_ZUR_SCHLUESSELDATEI ubuntu@DEINE_SERVER_IP
```

Was da steht: `ssh` baut eine verschlüsselte Verbindung auf, `-i` zeigt auf deine Schlüsseldatei aus Kapitel 06 (unter Windows meist `C:\Users\DEINNAME\Downloads\ssh-key-....key`), und `ubuntu@...` sagt, wohin. Beim ersten Mal fragt das Fenster, ob du dem Server vertraust – tippe `yes` und Enter.

WICHTIG

Meldet Windows **UNPROTECTED PRIVATE KEY FILE**, verweigert SSH den Schlüssel, weil zu viele Konten auf die Datei zugreifen dürfen. Ein Befehl in derselben PowerShell behebt das:

```
icacls PFAD_ZUR_SCHLUESSELDATEI /inheritance:r /grant:r "$env:USERNAME:(R)"
```

Auf dem Mac lautet er `chmod 600 PFAD_ZUR_SCHLUESSELDATEI`.

SELBST PRÜFEN

Geklappt hat es, wenn die Zeile vor deinem Cursor jetzt mit **ubuntu@…** beginnt. Ab diesem Moment landet alles, was du in dieses Fenster tippst, auf dem Server – dafür steht ab jetzt das Label „Auf dem Server“.

TIPP

Mac-Nutzer: Ab hier sind alle Befehle identisch mit Windows. Unterschiede gab es nur in diesem Kapitel – beim Pfad zur Schlüsseldatei und beim Schutzbefehl.

09 WireGuard installieren

WireGuard ist die eigentliche VPN-Software – sie baut den verschlüsselten Tunnel. Alle Befehle auf dieser Seite tippst du in das verbundene Fenster aus Kapitel 07.

Schritt 1 — Installation

AUF DEM SERVER

```
sudo apt update && sudo apt install -y wireguard
```

Schritt 2 — Schlüsselpaare erzeugen

AUF DEM SERVER

```
wg genkey | sudo tee /etc/wireguard/server.key | wg pubkey | sudo tee /etc/wireguard/server.pub
sudo chmod 600 /etc/wireguard/server.key

wg genkey | tee client.key | wg pubkey > client.pub
cat client.key client.pub
```

Das Fenster zeigt dir jetzt Zeichenketten an – deine Schlüssel. Notiere alle vier:

server.key	privater Schlüssel des Servers → Serverdatei (Kapitel 09)
server.pub	öffentlicher Schlüssel des Servers → dein Gerät (Kapitel 12)
client.key	privater Schlüssel deines Geräts → dein Gerät (Kapitel 12)
client.pub	öffentlicher Schlüssel deines Geräts → Serverdatei (Kapitel 09)

MERKE

Private Schlüssel verlassen niemals das Gerät, zu dem sie gehören. Getauscht werden ausschließlich die öffentlichen. Wer beides verwechselt, baut sich ein VPN, das jeder benutzen kann, der den Schlüssel sieht.

Schritt 3 — Name der Netzwerkkarte ermitteln

AUF DEM SERVER

```
ip -4 route show default
```

Der Name direkt hinter **dev** ist deine Netzwerkkarte, bei Oracle meist **ens3**. Notiere ihn – er gehört in die Konfiguration auf der nächsten Seite.

10 Serverkonfiguration

Jetzt schreibst du die Steuerdatei des VPN. Dafür öffnest du **nano**, einen einfachen Texteditor, der direkt im schwarzen Fenster läuft:

AUF DEM SERVER

```
sudo nano /etc/wireguard/wg0.conf
```

Es öffnet sich eine leere Datei. Füge den folgenden Block ein (Windows: Rechtsklick, Mac: `cmd + V`) und ersetze drei Dinge: `SERVER_PRIVATER_SCHLUESSEL` durch den Inhalt von server.key, `CLIENT_OEFFENTLICHER_SCHLUESSEL` durch client.pub – und `ens3` an allen Stellen durch den Namen deiner Netzwerkkarte, falls er anders lautet.

```
[Interface]
Address = 10.66.1.1/24
ListenPort = 51820
PrivateKey = SERVER_PRIVATER_SCHLUESSEL
MTU = 1380

PostUp = iptables -I FORWARD 1 -i wg0 -o ens3 -j ACCEPT
PostUp = iptables -I FORWARD 1 -i ens3 -o wg0 -m state --state RELATED,ESTABLISHED -j ACCEPT
PostUp = iptables -t nat -A POSTROUTING -s 10.66.1.0/24 -o ens3 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -o ens3 -j ACCEPT
PostDown = iptables -D FORWARD -i ens3 -o wg0 -m state --state RELATED,ESTABLISHED -j ACCEPT
PostDown = iptables -t nat -D POSTROUTING -s 10.66.1.0/24 -o ens3 -j MASQUERADE

[Peer]
PublicKey = CLIENT_OEFFENTLICHER_SCHLUESSEL
AllowedIPs = 10.66.1.2/32
```

Speichern mit **Strg + O** und Enter, schließen mit **Strg + X** – auch auf dem Mac mit Strg, nicht mit cmd.

WICHTIG

Die beiden FORWARD-Zeilen nutzen **-I FORWARD 1**, nicht **-A**. Die Ubuntu-Abbilder von Oracle enthalten am Ende dieser Kette eine Regel, die alles Weitergeleitete ablehnt. Mit -A landen deine Regeln dahinter und greifen nie: Der Tunnel verbindet sich einwandfrei, aber keine einzige Webseite lädt. Ein Fehlerbild, das leicht stundenlang in die falsche Richtung führt.

HINWEIS

`MTU = 1380` ist bewusst gesetzt und kein Tippfehler. Die Begründung steht in Kapitel 17 – bitte nicht auf den anderswo üblichen Wert 1420 ändern.

11 Lieber bauen lassen? Emergent erstellt die Dateien

Du kannst dir die Serverdatei und die Gerätedateien auch von Emergent zusammenstellen lassen – vor allem, wenn du mehrere Geräte anbinden willst. Der Prompt enthält alle Eigenheiten dieses Setups und lässt Emergent seine eigene Ausgabe am Ende überprüfen.

IN EMERGENT EINFÜGEN

PROMPT 02 — KONFIGURATIONSDATEIEN ERSTELLEN

```
Antworte mir bitte direkt hier im Chat. Baue keine App und lege kein
Projekt an - ich brauche fertige Textdateien zum Kopieren.

Ich richte einen WireGuard-VPN-Server auf Ubuntu 22.04 bei Oracle Cloud
ein und brauche die Konfigurationsdateien dafuer.

Meine Angaben:
- Name der Netzwerkkarte des Servers: [z. B. ens3]
- Tunnelnetz: 10.66.1.0/24, der Server selbst ist 10.66.1.1
- Anzahl Geraete, die ich anbinden will: [ANZAHL EINFÜGEN]

Diese Vorgaben sind nicht verhandelbar, halte dich exakt daran:
- MTU = 1380 in JEDER Datei, Server wie Geraete.
- Die beiden FORWARD-Regeln mit "-I FORWARD 1" einfuegen, niemals mit
  "-A". Grund: Die Ubuntu-Abbilder von Oracle haben am Ende der
  FORWARD-Kette eine REJECT-Regel. Angehaengte Regeln stehen dahinter
  und greifen nie.
- In jeder Geraetedatei: AllowedIPs = 0.0.0.0/0, ::/0
- In jeder Geraetedatei: DNS = 1.1.1.1
- In jeder Geraetedatei: PersistentKeepalive = 25
- Jedes Geraet bekommt eine eigene Adresse, beginnend bei 10.66.1.2.
- Setz fuer alle Schluessel Platzhalter ein, zum Beispiel
  SERVER_PRIVAT. Erzeuge keine echten Schluessel und frag mich auch
  nicht danach - die trage ich selbst ein.

Ausgabeformat, bitte genau so:
1. Die Serverdatei wg0.conf als ein einziger Codeblock, ohne
   Erklaertext dazwischen.
2. Danach pro Geraet eine eigene Datei als eigener Codeblock, benannt
   geraet1.conf, geraet2.conf und so weiter.
3. Danach eine Tabelle mit drei Spalten: Platzhalter | wo er hingehoert
   | woher ich den Wert bekomme.

Zum Schluss pruefst du deine eigene Ausgabe und bestaetigst mir jeden
dieser vier Punkte einzeln mit Ja oder Nein:
- Steht in jeder einzelnen Datei MTU = 1380?
- Steht in beiden FORWARD-Zeilen "-I FORWARD 1" und nicht "-A"?
- Hat wirklich jedes Geraet eine andere Adresse?
- Gibt es in der Serverdatei fuer jedes Geraet genau einen [Peer]-Block?

Wenn einer der Punkte nicht stimmt, korrigier ihn, bevor du mir
antwortest.
```

WICHTIG

Füge deine privaten Schlüssel niemals in einen Chat ein – bei keinem Anbieter. Die Ausgabe von `sudo wg show` ist dagegen unbedenklich: Sie zeigt private Schlüssel grundsätzlich nur als „(hidden)“.

12 Weiterleitung, Firewall, Start

Der Server muss Datenverkehr weiterleiten dürfen, der Port muss offen sein, und beides soll einen Neustart überstehen. Kopiere den Block ins verbundene Fenster:

AUF DEM SERVER

```
# Weiterleitung erlauben
echo "net.ipv4.ip_forward = 1" | sudo tee /etc/sysctl.d/99-wireguard.conf
sudo sysctl --system

# Port oeffnen und Regel dauerhaft speichern
sudo iptables -I INPUT 1 -p udp --dport 51820 -j ACCEPT
sudo apt install -y iptables-persistent
sudo netfilter-persistent save

# WireGuard starten, auch nach jedem Neustart
sudo systemctl enable --now wg-quick@wg0
sudo wg show
```

Zeigt **wg show** am Ende die Schnittstelle **wg0** mit Portnummer, läuft der Server.

Den Port zusätzlich bei Oracle öffnen

Oracle filtert ein zweites Mal, außerhalb des Servers. Ohne diesen Schritt kommt trotz laufendem WireGuard keine Verbindung zustande – der häufigste Grund, warum Anleitungen „nicht funktionieren“.

IM BROWSER · CLOUD.ORACLE.COM

01 Öffne das Menü (☰) und gehe zu **Networking** → **Virtual Cloud Networks**.

02 Klicke auf dein Netzwerk – es gibt nur eins – und dann in der linken Spalte auf **Security Lists**.

03 Öffne die **Default Security List** und klicke auf **Add Ingress Rules**.

04 Trage ein: **Source CIDR** 0.0.0.0/0 · **IP Protocol** UDP · **Destination Port Range** 51820. Dann speichern.

SELBST PRÜFEN

Läuft `sudo wg show` ohne Fehler und existiert die Ingress-Regel, ist die Serverseite fertig. Alles Weitere passiert auf deinem Gerät.

13 Computer verbinden

Der Server steht. Jetzt bekommt dein Computer seine eigene kleine Steuerdatei – danach ist es ein Klick.

Schritt 1 — Die Datei vpn.conf anlegen

Öffne einen Texteditor auf deinem Computer: unter Windows den **Editor** (Startmenü öffnen, „Editor“ tippen, Enter), auf dem Mac **TextEdit** (dort einmal **Format** → **Reines Textformat** wählen). Füge diesen Inhalt ein:

TEXTEDITOR · DEIN COMPUTER

```
[Interface]
PrivateKey = CLIENT_PRIVATER_SCHLUESSEL
Address = 10.66.1.2/32
DNS = 1.1.1.1
MTU = 1380

[Peer]
PublicKey = SERVER_OEFFENTLICHER_SCHLUESSEL
Endpoint = DEINE_SERVER_IP:51820
AllowedIPs = 0.0.0.0/0, ::/0
PersistentKeepalive = 25
```

Drei Platzhalter ersetzt du mit deinen notierten Werten: `CLIENT_PRIVATER_SCHLUESSEL` = Inhalt von client.key (Kapitel 08), `SERVER_OEFFENTLICHER_SCHLUESSEL` = Inhalt von server.pub (Kapitel 08), `DEINE_SERVER_IP` = die IP-Adresse aus Kapitel 06.

Dann speichern: **Datei** → **Speichern unter**, als Dateinamen `"vpn.conf"` **mit Anführungszeichen** eingeben (so hängt Windows kein .txt an), bei Dateityp „Alle Dateien“ wählen.

WAS DIE WICHTIGSTE ZEILE MACHT

`AllowedIPs = 0.0.0.0/0, ::/0` schickt den gesamten Verkehr durch den Tunnel. Der zweite Teil steht für IPv6 – das neuere Adressformat des Internets, das parallel zum klassischen IPv4 läuft. Er gehört auch dann hinein, wenn dein Server kein IPv6 anbietet: Sonst läuft dieser Teil deines Verkehrs am VPN vorbei und verrät deine echte Adresse.

Schritt 2 — WireGuard-App installieren und Datei laden

01 Lade den offiziellen WireGuard-Client von **wireguard.com/install** herunter und installiere ihn.

02 Öffne WireGuard und klicke unten links auf **Tunnel importieren** (englisch: **Add Tunnel** → **Import tunnel(s) from file**) – gemeint ist damit schlicht: deine vpn.conf laden.

03 Wähle die gespeicherte **vpn.conf** aus.

04 Klicke auf **Aktivieren**. Windows und macOS fragen nach Administratorrechten – bestätigen, ein Netzwerk-Tunnel lässt sich ohne sie nicht anlegen.

14 Handy verbinden

Jedes Gerät braucht ein eigenes Schlüsselpaar und eine eigene Adresse – Geräte teilen sich nie einen Schlüssel.

Schritt 1 — Schlüssel für das Handy erzeugen

AUF DEM SERVER

```
wg genkey | tee handy.key | wg pubkey > handy.pub
cat handy.key handy.pub
```

Schritt 2 — Handy in der Serverdatei eintragen

Öffne die Serverdatei erneut mit `sudo nano /etc/wireguard/wg0.conf` und hänge ganz unten einen zweiten Peer-Block an – mit dem Inhalt von handy.pub und der nächsten freien Adresse:

AUF DEM SERVER · IN NANO

```
[Peer]
PublicKey = HANDY_OEFFENTLICHER_SCHLUESSEL
AllowedIPs = 10.66.1.3/32
```

Speichern (Strg + O, Enter), schließen (Strg + X), dann die Änderung übernehmen:

AUF DEM SERVER

```
sudo systemctl restart wg-quick@wg0
```

Schritt 3 — QR-Code erzeugen und scannen

Lege mit `nano handy.conf` eine Datei an – Inhalt wie die vpn.conf aus Kapitel 12, nur mit zwei Änderungen: `PrivateKey` = Inhalt von handy.key und `Address = 10.66.1.3/32`. Speichern, schließen, dann:

AUF DEM SERVER

```
sudo apt install -y qrencode
qrencode -t ansiutf8 < handy.conf
```

Installiere die **WireGuard-App** (App Store / Play Store), tippe auf **+** → **Aus QR-Code erstellen** und halte die Kamera auf den Bildschirm. Danach die Dateien vom Server löschen:

AUF DEM SERVER

```
rm handy.conf handy.key handy.pub
```

> **TIPP**
> Ein drittes Gerät? Gleiches Spiel mit der nächsten Adresse: 10.66.1.4, eigenes Schlüsselpaar, eigener Peer-Block.

15 Prüfen, ob es wirklich läuft

Aktiviere den Tunnel und öffne im Browser die Seite **ipleak.net** – ein kostenloser Test, der dir zeigt, wie du im Netz erscheinst. Drei Dinge liest du dort ab:

01 Ganz oben zeigt die Seite eine **IP-Adresse** mit Landesflagge. Das muss die Adresse deines Servers sein – die aus Kapitel 06, nicht deine heimische.

02 Das angezeigte **Land** muss zu deiner gewählten Region passen.

03 Scrolle zum Abschnitt **DNS Addresses**. Dort darf kein Eintrag deines eigenen Internetanbieters auftauchen – also nichts von Telekom, Vodafone oder o2. Stehen dort stattdessen Namen wie Cloudflare, ist alles richtig.

Der dritte Punkt ist der wichtigste. Ein VPN, das den Datenverkehr umleitet, aber die Namensauflösung beim Anbieter belässt, verrät ihm weiterhin jede besuchte Seite.

Dass der IPv6-Test „nicht erreichbar“ meldet, ist kein Fehler, sondern das gewünschte Ergebnis: Deine echte IPv6-Adresse bleibt verborgen.

SELBST PRÜFEN

Alle drei Punkte stimmen? Dann ist dein VPN fertig und sauber eingerichtet. Ab jetzt genügt ein Klick in der WireGuard-App, um den Tunnel ein- oder auszuschalten.

16 Was funktioniert – und was nicht

Hier trennt sich Werbung von Realität. Dein Server hat eine Adresse aus einem Rechenzentrum, und die ist als solche erkennbar.

Zuverlässig	Schutz in fremden WLANs · Aktivität vor dem Zugangsanbieter verbergen · Zugriff aufs Heimnetz · Surfen und Dienste im eingeloggten Zustand
Meistens	Dienste mit weltweit gleichem Angebot – etwa Plattformen, denen ihre Inhalte selbst gehören, und werbefinanzierte Angebote
Eingeschränkt	Dienste, die Inhalte pro Land lizenzieren – sie erkennen und sperren Rechenzentrums-Adressen häufig beim Abspielen

Die Regel dahinter ist simpel: Wer Inhalte pro Land lizenziert, ist vertraglich zum Sperren verpflichtet und investiert dauerhaft darin. Wer seine Inhalte weltweit selbst besitzt, hat keinen Grund dazu.

> **WICHTIG**
> Ein VPN macht nicht anonym. Dein Zugangsanbieter sieht nicht mehr, welche Seiten du aufrufst – der Dienst am anderen Ende sieht dich weiterhin, spätestens sobald du dich dort anmeldest.

Eine klare Grenze

Ob etwas technisch geht, ist die eine Frage – ob du es so nutzen darfst, die andere. Viele Dienste schließen das Umgehen von Ländersperren in ihren Nutzungsbedingungen aus, und je nach Dienst und Land kann es auch rechtlich unzulässig sein. Wo das der Fall ist, raten wir ausdrücklich davon ab – dieser Guide unterstützt eine solche Nutzung nicht.

Wofür er gedacht ist: deine Verbindung schützen, in fremden Netzen sicher arbeiten und dein eigenes Netz von unterwegs erreichen. Kostenlos, unter deiner Kontrolle, ohne Protokolle, die jemand anderes führt.

17 Wenn etwas klemmt

Wenn es hakt, rate nicht herum – füttere Emergent mit echten Ausgaben vom Server. Die beiden Befehle im Prompt tippst du im verbundenen Fenster ein (Kapitel 07) und kopierst ihre Ausgabe in die Platzhalter.

IN EMERGENT EINFÜGEN

PROMPT 03 — FEHLERSUCHE MIT ECHTEN AUSGABEN

```
Antworte mir bitte direkt hier im Chat. Baue keine App und lege kein
Projekt an.

Mein selbst gebauter WireGuard-VPN-Server bei Oracle Cloud (Ubuntu
22.04) funktioniert nicht richtig.

Was genau passiert: [FEHLER BESCHREIBEN - z. B. "verbunden, aber keine
Webseite laedt" oder "es kommt gar keine Verbindung zustande"]

Ausgabe von "sudo wg show":
[HIER EINFÜGEN]

Ausgabe von "sudo iptables -L FORWARD -n --line-numbers":
[HIER EINFÜGEN]

Mein Internetanschluss zu Hause: [z. B. Vodafone Kabel Deutschland]

So ist es eingerichtet: MTU 1380, Tunnelnetz 10.66.1.0/24, WireGuard
auf UDP-Port 51820, Ingress-Regel in Oracle ist angelegt.

Arbeite diese drei Punkte in genau dieser Reihenfolge ab und sag mir
zu jedem klar Ja oder Nein:
1. Zeigt "wg show" einen aktuellen Handshake? Wenn nein, liegt das
   Problem VOR dem Tunnel: Port, Firewall oder ein falscher
   Schluessel. Dann brauchen wir die anderen Punkte gar nicht.
2. Stehen meine ACCEPT-Zeilen in der FORWARD-Kette OBERHALB der
   REJECT-Zeile? Wenn sie darunter stehen, wird weitergeleiteter
   Verkehr verworfen.
3. Steigen bei "wg show" beide Uebertragungswerte oder nur einer?
   Wenn viel gesendet und kaum etwas empfangen wird, deutet das auf
   zu grosse Pakete hin, also auf die MTU.

Nenne mir danach die EINE wahrscheinlichste Ursache und genau einen
Befehl, mit dem ich sie bestaetige.

Wenn dir die Ausgaben dafuer nicht reichen, sag das offen und nenne
mir den naechsten Befehl, dessen Ausgabe du brauchst. Rate nicht.
```

TIPP

Die Ausgabe von `sudo wg show` kannst du bedenkenlos in den Chat kopieren – private Schlüssel zeigt sie grundsätzlich nur als „(hidden)“.

18 Fehler und Lösungen

Verbindung steht, nichts lädt	**MTU** auf 1380 setzen, notfalls 1280 – auf beiden Seiten. Zweiter Verdacht: FORWARD-Regeln stehen mit **-A** statt **-I** in der Serverdatei. Hintergrund: Manche Anschlüsse (etwa Vodafone Kabel) transportieren IPv4 innerhalb von IPv6, dadurch bleiben pro Paket weniger Byte übrig – zu große Pakete werden ohne Fehlermeldung verworfen.
Gar keine Verbindung	Ingress-Regel in der Oracle Security List prüfen (UDP 51820, Kapitel 11). Dann auf dem Server `sudo wg show` – erscheint **wg0**?
„Out of host capacity“	Shape auf **VM.Standard.E2.1.Micro** wechseln (Kapitel 06).
Handshake, aber kein Datenverkehr	Typisch für falsch einsortierte FORWARD-Regeln oder fehlendes `net.ipv4.ip_forward = 1` (Kapitel 11).
Nach Neustart tot	`sudo systemctl enable wg-quick@wg0` und `sudo netfilter-persistent save` ausgeführt?
Verbindung bricht ab	`PersistentKeepalive = 25` in der Gerätedatei prüfen – hält die Verbindung durch Router hindurch offen.

Nützliche Befehle

AUF DEM SERVER

```
sudo wg show                              # Status und letzter Handshake
sudo systemctl status wg-quick@wg0        # Laeuft der Dienst?
sudo journalctl -u wg-quick@wg0 -n 50     # Letzte Meldungen
sudo iptables -L FORWARD -n --line-numbers  # Reihenfolge der Regeln
```

Beim letzten Befehl müssen deine beiden ACCEPT-Zeilen **oberhalb** der REJECT-Zeile stehen. Tun sie das nicht, ist genau das die Ursache.

19 Die Oberfläche: Rahmen setzen

Dein VPN läuft. Was fehlt, ist Komfort: Für jedes neue Gerät musst du bisher Schlüssel im Terminal erzeugen und Text zusammenkopieren. Genau das packst du jetzt in eine kleine Web-Oberfläche, gebaut mit Emergent, ohne selbst zu programmieren. Die drei Prompts bauen aufeinander auf: Rahmen setzen, bauen lassen, nachprüfen.

IN EMERGENT EINFÜGEN

PROMPT 04 — RAHMEN SETZEN

```
Ich möchte mit dir eine Web-Oberfläche bauen, mit der ich die
Konfigurationsdateien für meinen eigenen WireGuard-VPN-Server verwalte.

Wichtig, bevor du irgendetwas baust: Eine Website im Browser kann keinen
VPN-Tunnel aufbauen, dafür braucht es Administratorrechte im Betriebssystem.
Baue deshalb keinen "Verbinden"-Knopf, der so tut, als würde er eine
Verbindung herstellen. Die Oberfläche soll Konfigurationen erzeugen und
verwalten. Verbinden tue ich anschließend mit der offiziellen WireGuard-App.

Stell mir jetzt Fragen, bis du verstanden hast, was ich brauche: wie viele
Geräte, welche Angaben ich schon habe, was auf dem Handy funktionieren muss.
Frag in kleinen Schritten, nicht alles auf einmal. Fang noch nicht an zu bauen.
```

WICHTIG

Eine Website im Browser kann keinen VPN-Tunnel aufbauen, dafür braucht es Administratorrechte im Betriebssystem. Ohne diesen Hinweis baut dir jede KI-Plattform bereitwillig einen „Verbinden“-Knopf, der nichts tut. Deshalb steht der Satz gleich im ersten Prompt.

20 Die Oberfläche: der Bauauftrag

Jetzt der eigentliche Auftrag. Das Format der Gerätedatei steht wörtlich im Prompt, damit Emergent nicht seine eigene Variante erfindet. Wichtig ist auch der Hinweis auf echte Curve25519-Schlüssel: Ein Modell erzeugt sonst gern Zufallstext, der wie ein Schlüssel aussieht und von WireGuard abgelehnt wird.

IN EMERGENT EINFÜGEN

PROMPT 05 — BAUAUFTRAG

```
Danke, jetzt bauen wir. Erstelle eine Web-Oberfläche mit diesen Funktionen:

1. Ich hinterlege einmalig meinen Server: öffentliche IP-Adresse, Port 51820
   und den öffentlichen Schlüssel des Servers.
2. Ich lege Geräte an (Laptop, Handy, Tablet). Jedes Gerät bekommt ein eigenes
   Schlüsselpaar und eine eigene Adresse im Netz 10.66.1.0/24, beginnend
   bei 10.66.1.2.
3. Für jedes Gerät erzeugst du eine Konfiguration in exakt diesem Format:

[Interface]
PrivateKey = privater Schlüssel des Geräts
Address = Adresse/32
DNS = 1.1.1.1
MTU = 1380

[Peer]
PublicKey = öffentlicher Schlüssel des Servers
Endpoint = Server-IP:51820
AllowedIPs = 0.0.0.0/0, ::/0
PersistentKeepalive = 25

4. Jede Konfiguration kann ich als .conf-Datei herunterladen und als QR-Code
   anzeigen. Der QR-Code muss exakt den Text der Konfiguration enthalten und
   sonst nichts, weil die WireGuard-App genau diesen Text erwartet.
5. Zusätzlich zeigst du mir pro Gerät den passenden Peer-Block für meine
   Serverdatei mit einem Kopier-Knopf.

Die Schlüssel müssen echte Curve25519-Schlüssel sein, wie WireGuard sie
verlangt: 32 Byte, Base64-kodiert, 44 Zeichen lang.
```

21 Die Oberfläche: nachprüfen

Der wichtigste der drei Prompts und der, den fast alle weglassen. Er lässt Emergent die eigene Umsetzung prüfen, Punkt für Punkt. Stimmt einer der fünf Punkte nicht, hast du eine Oberfläche, die überzeugend aussieht und unbrauchbare Konfigurationen erzeugt.

IN EMERGENT EINFÜGEN

PROMPT 06 — KONTROLLE

```
Bevor ich das benutze, prüfe deine eigene Umsetzung und antworte mir ehrlich
auf jeden Punkt einzeln:

1. Sind die erzeugten Schlüssel echte Curve25519-Schlüssel mit 44 Zeichen,
   oder ist es Zufallstext, der nur so aussieht?
2. Gehört zu jedem privaten Schlüssel der passende öffentliche? Erklär mir,
   wie du ihn ableitest.
3. Enthält der QR-Code exakt den Konfigurationstext, ohne zusätzliche Zeichen
   oder Umbrüche?
4. Verlässt irgendein privater Schlüssel mein Gerät, zum Beispiel durch
   Speichern auf einem Server oder Senden an eine Schnittstelle?
5. Gibt es irgendwo einen Knopf oder eine Anzeige, die einen aktiven
   VPN-Schutz behauptet, den die Seite gar nicht herstellen kann?

Wenn einer dieser Punkte nicht sauber ist, sag es mir klar und korrigiere ihn.
```

WICHTIG

Füge deine privaten Schlüssel niemals in einen Chat ein, weder bei Emergent noch anderswo. Lass dir Vorlagen mit Platzhaltern geben und setz die echten Werte erst lokal in die Datei ein.

22 Von der Oberfläche zum laufenden Tunnel

Die Oberfläche gibt dir pro Gerät zwei Dinge: die Gerätedatei und einen Peer-Block. Die Gerätedatei gehört auf dein Gerät, der Peer-Block auf den Server. Erst wenn beide an ihrem Platz sind, kommt eine Verbindung zustande. Emergent selbst fasst deinen Server nie an.

01 In der Oberfläche das Gerät anlegen. Lade die **.conf** herunter und kopier dir den **Peer-Block** daneben.

02 Auf dem Server die Serverdatei öffnen, den Peer-Block ganz unten anhängen, speichern mit Strg + O, schließen mit Strg + X.

03 WireGuard neu starten, sonst kennt der Server das neue Gerät nicht.

04 Auf deinem Computer die offizielle **WireGuard-App** öffnen, die .conf importieren und den Tunnel aktivieren.

05 Auf **ipleak.net** prüfen: deine Server-Adresse, kein Anbietername unter DNS, IPv6 „not reachable“.

AUF DEM SERVER

```
sudo nano /etc/wireguard/wg0.conf
sudo systemctl restart wg-quick@wg0
```

WICHTIG

Vergib die Adressen nur an einer Stelle. Hast du dein Handy vorher von Hand als `10.66.1.3` eingetragen, die Oberfläche zählt aber selbst ab `10.66.1.2` hoch, bekommen zwei Geräte dieselbe Adresse. Beide laufen dann einzeln, aber nie gleichzeitig. Trag bestehende Geräte in der Oberfläche nach oder lass sie erst ab der nächsten freien Adresse zählen.

SELBST PRÜFEN

Geklappt hat es, wenn `sudo wg show` auf dem Server für dein Gerät einen **letzten Handshake** anzeigt und beide Übertragungswerte steigen. Steigt nur der gesendete Wert, ist es die Paketgröße: `MTU` auf 1380, notfalls 1280.