

X2Go in der Google Cloud, kostenlos und/oder
anonym – wie geht das?



TÜBIX, 2019-07-06



X2Go in der Google Cloud, kostenlos und/oder
anonym – wie geht das?

Vorstellung



Stefan Baur

Geschäftsführer der
BAUR-ITCS UG
(haftungsbeschränkt)

Vorstellung



Stefan Baur

X2Go-Projektkoordinator

X2Go-Eventplaner

X2Go-Lead-Evangelist

Plan für diesen Slot

- Was ist die Google Cloud Platform
- Was ist die Google Cloud Shell
(Requirements, Features, Limitations)
- Was Google will, dass man mit der Cloud Platform tut
- Was Google nicht bedacht hat: Hackers gonna hack
- Welche der Einschränkungen sind für uns relevant?
- Security-Maßnahmen
- Konfiguration und Start der Live-Demo
- Wieviel Aluhut hätten's denn gern?

Google-Werbung

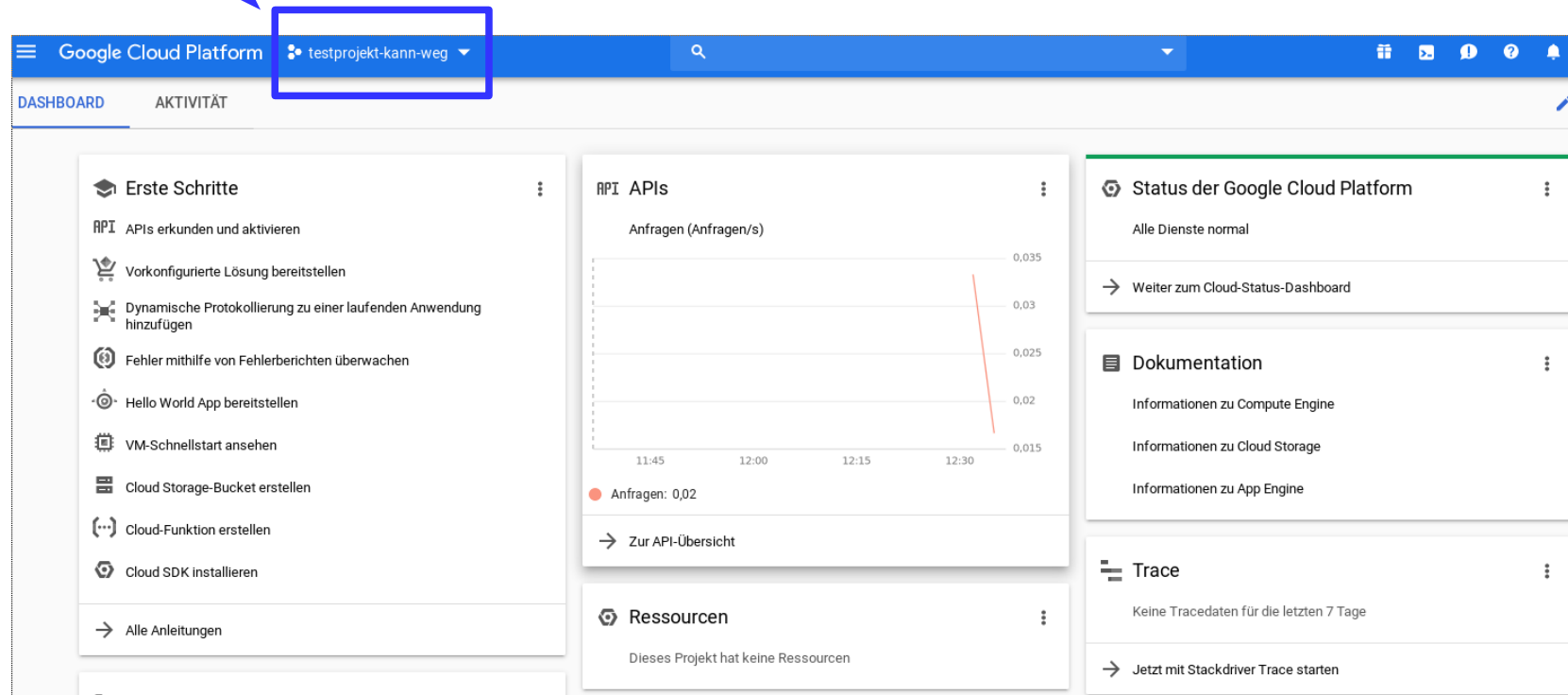
(unfreiwillig)



Google Cloud Platform

Google Cloud Platform: Dashboard

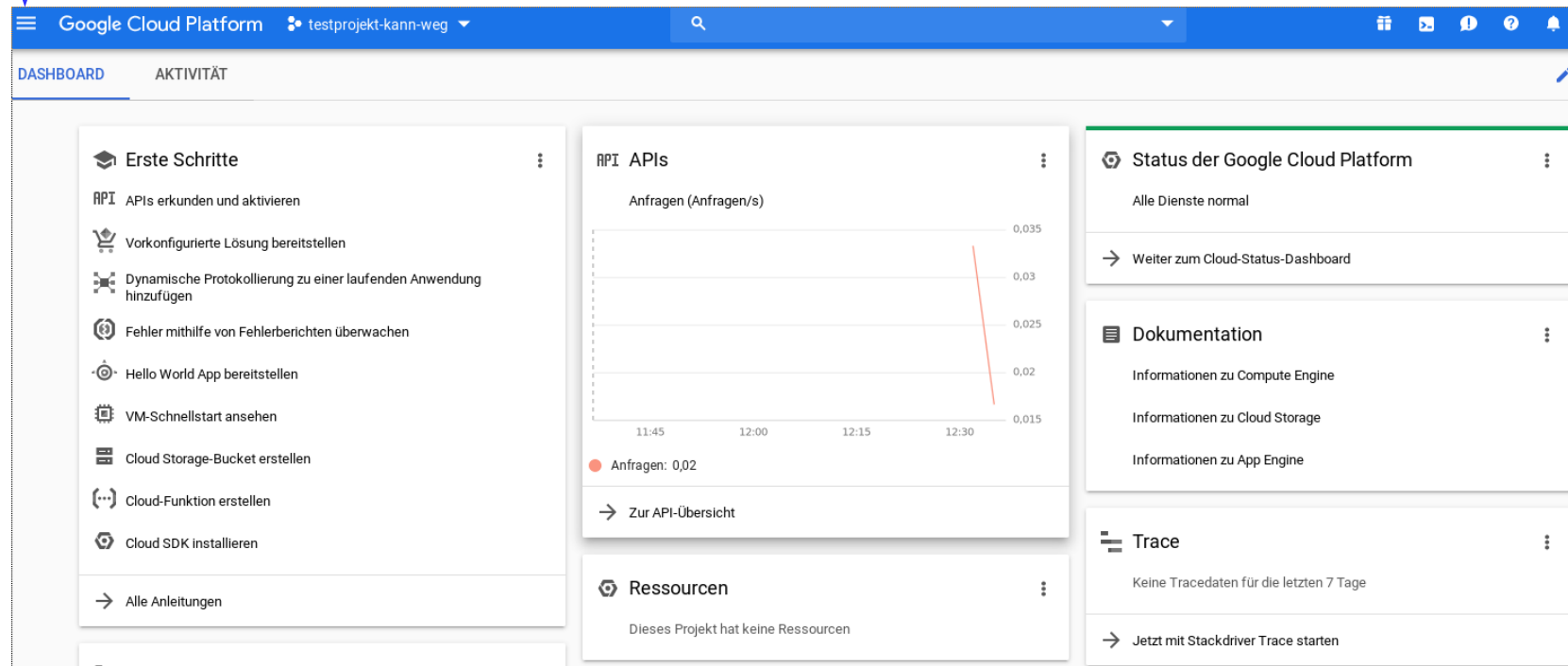
Kann mehrere Projekte verwalten,
pro Projekt eine *Kachelansicht*



Google Cloud Platform: Dashboard

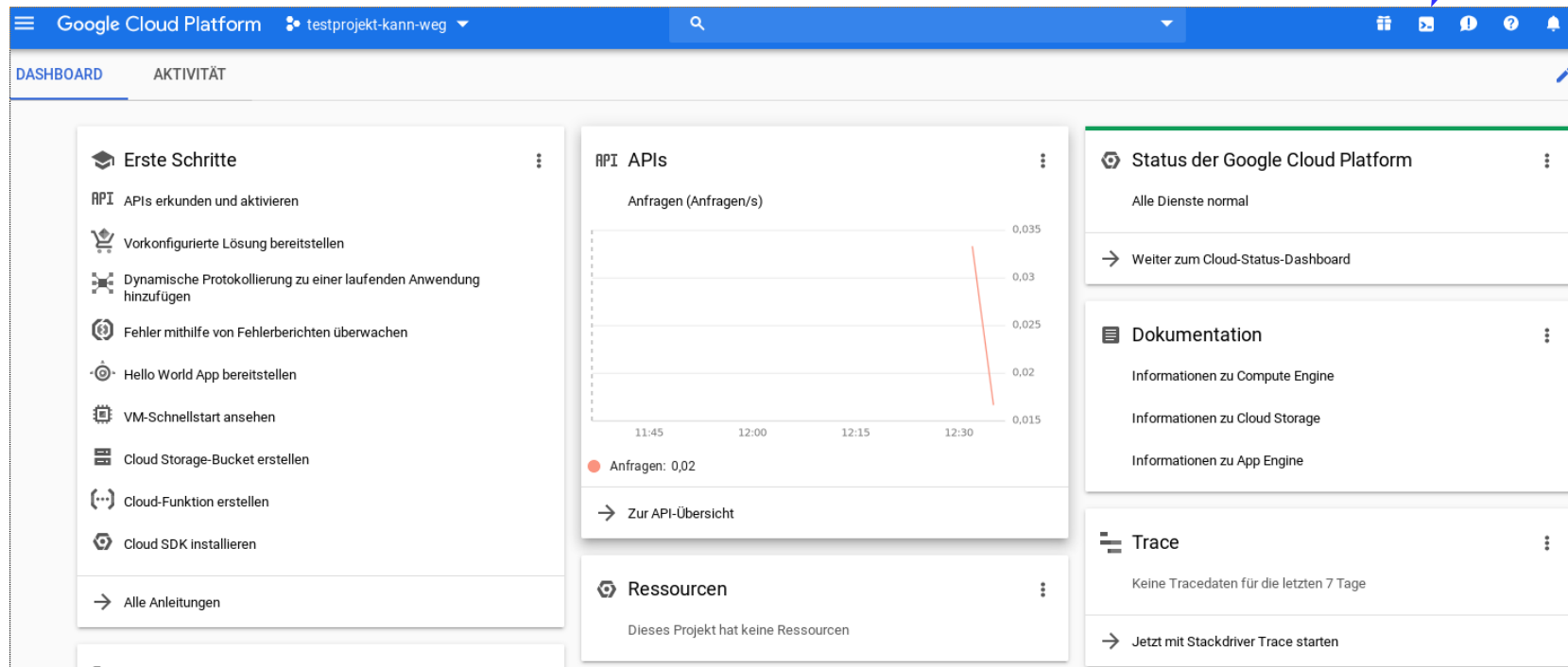
<https://console.cloud.google.com/home/dashboard>

Computing, Speicher, Netzwerk, Stackdriver, Tools,
Big Data, Künstliche Intelligenz, etc.



Google Cloud Platform: Dashboard

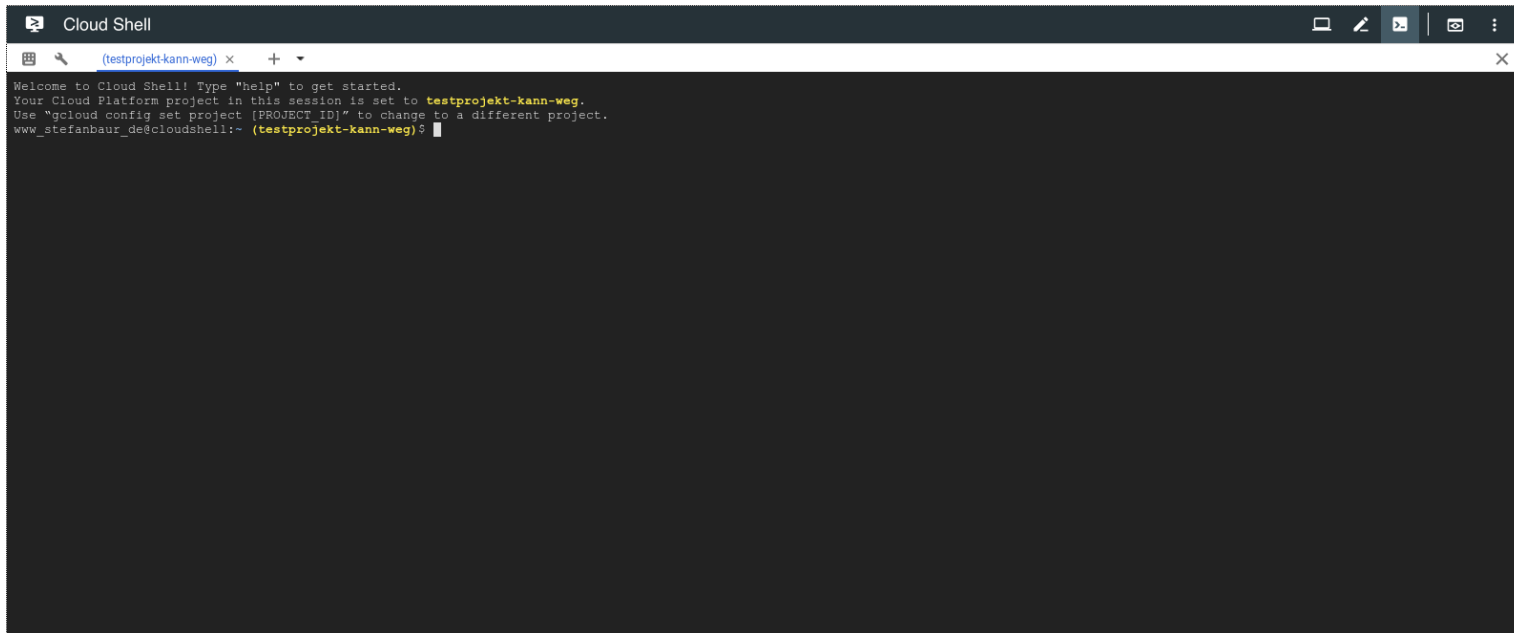
Google Cloud Shell ist aus dem Dashboard startbar





Google Cloud Shell – was ist das?

Google Cloud Shell



```
Cloud Shell
(testprojekt-kann-weg) x + -
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to testprojekt-kann-weg.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
www_stefanbaur_de@cloudshell:~ (testprojekt-kann-weg)$
```

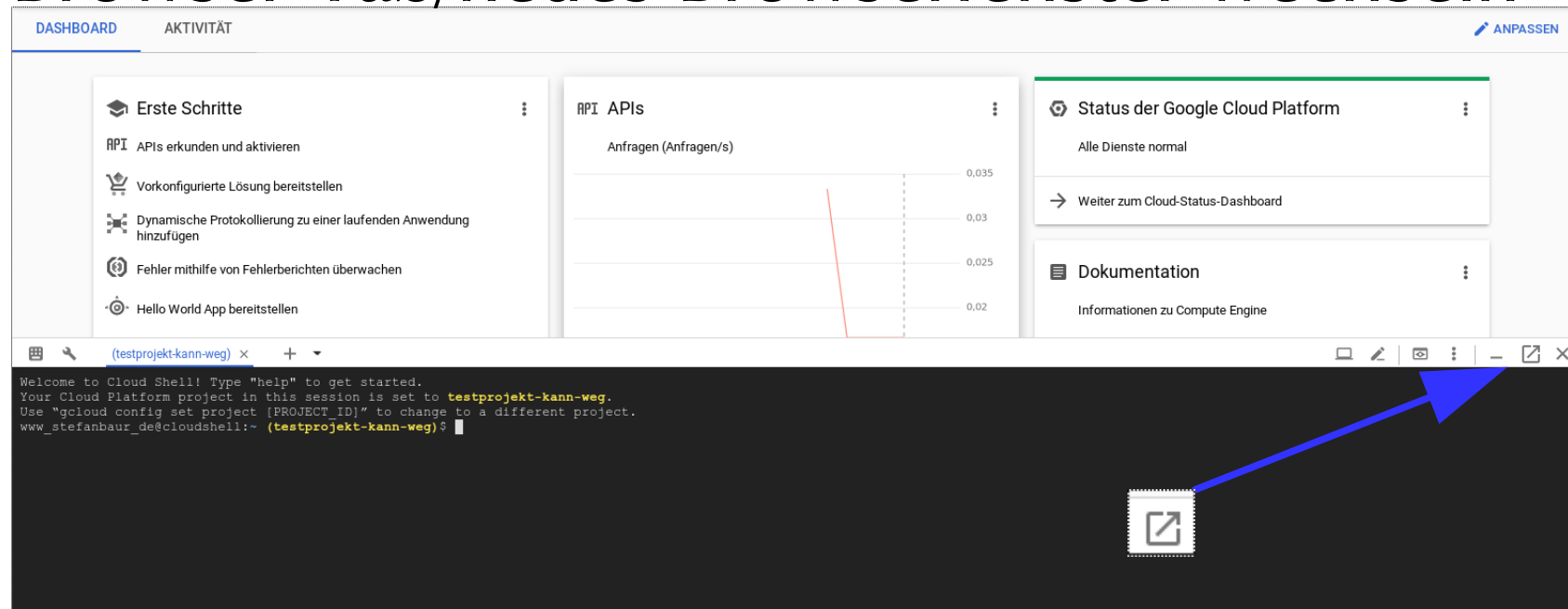
Laut Dokumentation: „Google Cloud Shell ist eine kostenlose Administrator-Maschine, mit der Sie Ihre Infrastruktur und Anwendungen auf der Cloud Platform browserbasiert per Befehlszeile verwalten können.“

Google Cloud Shell

Direkter Aufruf per

<https://console.cloud.google.com/cloudshell>

oder aus dem Dashboard per Button in neuen Browser-Tab/neues Browserfenster wechseln





Google Cloud Shell: Requirements, Features, Limitations

Google Cloud Shell – Requirements

- Nur eines:
 - braucht einen hinreichend *modernen* Browser mit HTML5-Unterstützung, genauer: vermutlich *Canvas-Unterstützung* und ähnliche Features
(also aktuelle Version von Firefox, Chrome, etc.)

Google Cloud Shell – Features I

- temporäre virtuelle Maschine (Docker-Image)
- Debian Stretch (9.x), 64-Bit
- Shell via Browser (https-geschützt)
- Authentisierung via Google-Konto, optional mit 2FA (Details dazu später)
- Praktisch volle root-Rechte über sudo (aber eben nur im Container)
- vorinstalliertes Google Cloud SDK und gängige Commandline-Tools, aber sonst nicht viel

Google Cloud Shell – Features II

- Voller Internetzugang über eine IPv4-Adresse aus dem privaten Adressbereich 172.16.0.0/12 (NAT)
→ auch ssh outbound geht, nicht nur Webkram
- 1 Core – Intel Xeon CPU @ 2.60GHz
- knapp 2 GB RAM, davon 0,8 GB im Leerlauf belegt
- grob 1 GB SWAP (auf einem zram-device)
- / hat ~ 6 GB frei, /home/account_name ~ 5 GB
- *Web Preview* – man bekommt ein Portforwarding (default 8080, aber änderbar) auf eine https-URL, unter der man seine Anwendung SSL-geschützt testen kann

Google Cloud Shell – Features III

- `/home/account_name` liegt auf einem quasi-persistenten Speicher, bedeutet:
 - mehr als 120 Tage lang keine VM gestartet
→ Homedir wird gelöscht
 - Vorher erhält man aber eine Warnung per Mail
 - Zähler lässt sich durch kurzes Einloggen resettet

Google Cloud Shell – Limitations I

- Nutzungslimit: 50 Stunden in einem 7-Tage-Korridor (Aktueller Verbrauch über Menü im Browser abfragbar)
- angeblich 1 GB outbound Traffic/Monat (Doku unklar), inbound dagegen wohl vom Volumen unbegrenzt
- 20 min Idle → Disconnect, 1h Disconnected → VM wird gelöscht
→ Konsequenz: Browserfenster besser nicht schließen, so lange man in der VM laufende Dinge nutzen will
- / ist nicht persistent → Maschine aus, Maschine weg, Daten/Programme/Konfiguration dort weg, nächster Neustart wieder frisches Google-Cloud-Shell-Template

Google Cloud Shell – Limitations II

- hat keine von außen erreichbare IP/DNS-Name (NAT)
- hat kein IPv6
- Beschränkungen durch die Docker-Virtualisierung:
modprobe fuse, tun, tap → geht alles nicht
 - keine Unterstützung für FUSE (Filesystem in UserSpace)
 - somit kein Google Drive zur Kapazitätserweiterung nutzbar
 - kein Mounten von Dateisystemen auf anderen Rechnern per sshfs

Google Cloud Shell – Limitations III

- keine Möglichkeit, andere Kernelmodule nachzuladen (zumindest nicht tun/tap → Kein OpenVPN)
- keine Swapfiles möglich → mehr RAM geht nicht
- *Web Preview*-Adresse ist nur von dem PC erreichbar, auf dem man per Google Account eingeloggt ist und die Web Shell im Browser geöffnet hat (Cookie)
→ Kann also nicht mal eben als öffentlicher Webserver genutzt werden, wirklich nur zum Test auf eigenem PC
- Copy-Paste in der Browser-Shell nur mit Ctrl-C/Ctrl-V
→ also genau das, was man an der Shell typischerweise NICHT erwartet

Google Cloud Shell – Limitations IV

- „Cloud Shell is intended for interactive use only. Non-interactive sessions will be ended automatically after a warning. Prolonged usage or computational or network intensive processes are not supported and may result in session termination without a warning.“
- Komplette Liste der Einschränkungen:
 - via <https://cloud.google.com/shell/docs/limitations>
 - via *help*-Kommando an der Konsole.

The background is a solid blue color. At the top, there are two stylized white clouds with dotted outlines. At the bottom, there is a decorative border consisting of a series of white, overlapping scalloped shapes.

Was Google will, dass man mit der
Cloud Platform tut

Was Google will: Cloud Platform

- breitgefächerte Nutzung der Cloud Platform für persistente VMs – aber auch für WebApps, KI-Kram, etc. (Hipsterscheiß eben ;-)) → es droht einem kein Ungemach, auch wenn man dauerhaft nur im *Free Tier* (<https://cloud.google.com/free/>) unterwegs ist
- und natürlich: EURE DATEN!
- denn auch für die Dienste im *Free Tier* muss man sich mit einer Kreditkarte auf <https://console.cloud.google.com/freetrial/> registrieren (auch Haftungsgründe → Follow the Money → Karten-ausgebende Bank ist verpflichtet, Identität zu prüfen)
- Lockangebot: 300 US-Dollar Startguthaben



Google-Werbung
(unfreiwillig)



Was Google nicht bedacht hat:
Hackers gonna hack

Hackers gonna hack I

- für die Cloud Shell allein braucht man keine Kreditkarte
- wer einen Google-Account hat, hat eine Cloud Shell
- wir sind root 🍆
- debootstrap lässt sich installieren
- chroot funktioniert
- mount --bind funktioniert
- wir haben genug Speicherplatz
- 1. Ansatz:
 - debootstrap stretch /home/account_name/\$SERVER
 - wenn fertig, chrooten und Services starten (ssh auf 222, da 22 schon vom Host belegt; x2goserver)

Hackers gonna hack II

- Blöd: Changeroot mit X2GoServer, XFCE Desktop, LibreOffice, Firefox, Thunderbird, PDF-Viewer frisst ~ 50% unseres freien Homedir-Speicherplatzes
- 2. Ansatz:
 - Server vor jeder Nutzung jedes Mal per Skript neu bauen, in /\$SERVER → Mehr Platz im Home
 - /home unseres Changeroots nach /home/account_name/\$SERVER-home legen und per bind-mount einhängen → Vorteile:
 - Bleibt (semi-)persistent
 - Man sieht im chroot die Auslastung von /home

Hackers gonna hack III

3. Ansatz:

- Warum überhaupt das Changeroot, wenn man nach / installieren könnte?
 - Wir haben keine Kontrolle darüber, was Google im Template ändert → Server könnte von heute auf morgen nicht mehr funktionieren
 - Changeroot kann man notfalls auch anderswo installieren lassen (anderer Cloudanbieter, lokal)
 - Homedir für Changeroot ist sauber getrennt
- Wir zeigen heute nur den Ansatz mit Changeroot

Noch mal kurz Google-Werbung

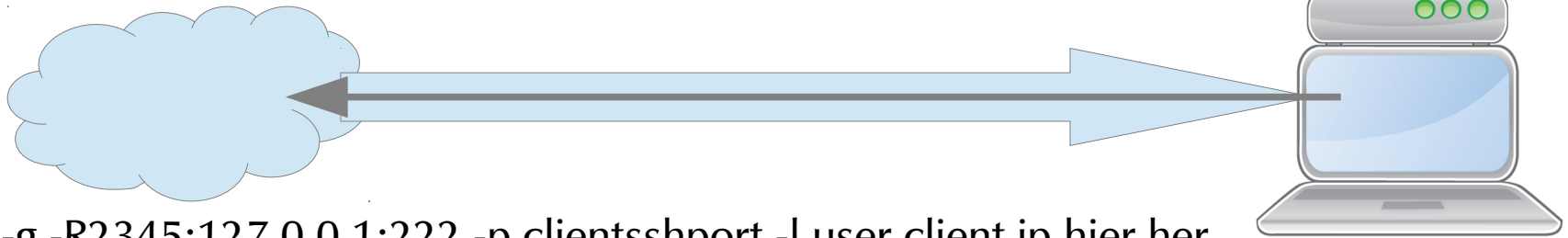
- Mit hinterlegten Kreditkartendaten kann man:
 - sein eigenes Docker-Image für die Cloud-Shell-Instanz speichern, muss also diesen Aufwand nicht treiben
 - einen richtigen kleinen persistenten Server im *Free Tier*-Modell betreiben, was die Sache wahrscheinlich noch komfortabler macht
→ <https://phillymesh.net/tag/f1-micro/> erklärt, wie das geht
- Ist aber dann nicht mehr anonym! :-)

Welche Einschränkungen sind für uns relevant?

- Hauptproblem: keine öffentliche IP → sshd läuft, aber wir kommen von außen nicht drauf
- Erfolglose Umgehungsversuche:
 - Web Preview → Portforwarding → Squid-Proxy? (SSH/X2GoClient funktioniert ja über Proxy) → Nein, weil wir das Session-Cookie nicht in den Client kriegen
 - keine tun/tap-Devices = kein VPN-Client → so kommen wir also auch nicht raus/drauf
- Aber: Wir können zwar nicht per VPN-Client *nach Hause telefonieren*, jedoch sehr wohl per ssh-Client, und ssh kann Ports forwarden, in beide Richtungen

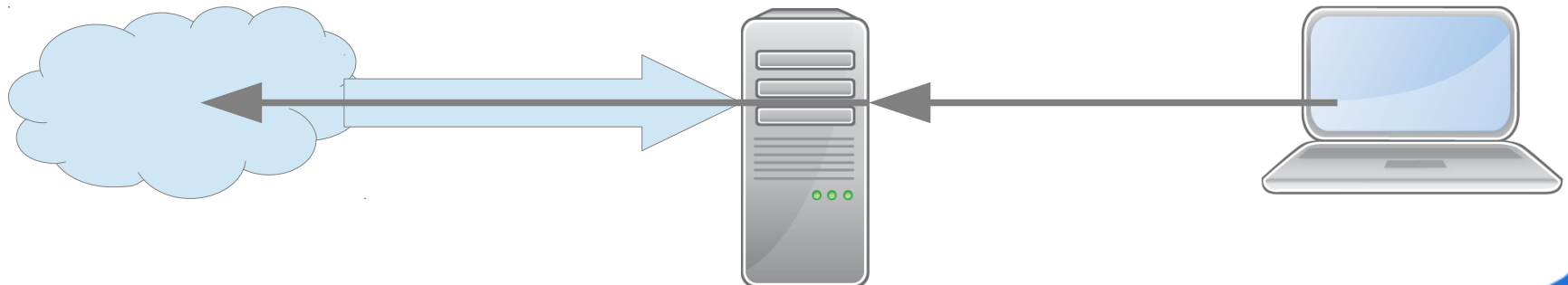
Wo telefonieren wir hin?

- Variante 1: Auf unseren Client, wenn er eine öffentliche IP hat, bzw. per Portforwarding erreichbar ist



```
ssh -g -R2345:127.0.0.1:222 -p clientsshport -l user client.ip.hier.her
```

- Variante 2: Auf einen *Jump Host* (ssh-Proxy), auf den wir uns dann vom Client aus ebenfalls einwählen



Security-Maßnahmen I

- Egal ob Client oder Jump Host:
 - 2FA ist für SSH-Server im Internet Pflicht!
 - ein non-standard-Port ändert gar nichts daran, schon gar nicht wenn er die Zahlenkombination 22 enthält
 - fail2ban ändert nicht viel dran (Hacker kommt nicht so leicht rein, aber dafür wird man selbst auch schneller ausgesperrt)
 - Welche 2FA-Option man verwendet, ist zweitrangig
 - Keyfile
 - Token/App mit OTP-Codes
 - Empfehlung: Port 443 – ist häufig in Firewalls offen

Security-Maßnahmen II

- Für den Google-Cloud-X2Go-Server nach dieser Anleitung ist 2FA verzichtbar – aber, wie in der vorherigen Folie beschrieben, nicht für den Jump Host/Client) und nicht für einen Server im *Free Tier*
- Warum auf dem Server verzichtbar:
 - da nur auf 127.0.0.1 erreichbar (sshd_config)
- Ausnahme: Wenn auf dem Jump Host mehrere Benutzer sein können (z.B. FreeShell-Anbieter)
 - Alle User können auf den Remote-Port connecten
 - 2FA auch auf dem Server notwendig!

Client oder Jump Host?

- muss per öffentlicher IP erreichbar sein, entweder
 - direkt oder
 - per NAT-Portforwarding auf dem Router oder
- per Mullvad-VPN <https://mullvad.net/de/>
(bietet öffentliche IP mit Portforwarding)
- IP/DNS-Name muss bekannt sein
- Benutzerkonto muss vorhanden sein (nicht root)
- muss über laufenden SSH-Server verfügen
- Jump Host:
 - muss zusätzlich SSH-Portforwardings erlauben
 - Rechenleistung unkritisch, Raspi reicht theoretisch
 - FreeShell-Anbieter funktionieren eventuell auch

Die Praxis

- Installations- und Startskripte per Github auscheckbar:
<https://github.com/stefanbaur/google-cloud-x2go-server/>
- `git clone <obige URL>`
- `mv gopath gopath_old`
- `ln -s google-cloud-x2go-server/gopath gopath`
- Warum:
 - `~/gopath/bin` ist im von Google vorgegebenen Pfad für dieses Docker-Image
 - spart die Pfadangabe vor jedem Skriptaufruf
 - und man muss auch nicht nach `git pull` jedes Mal alles neu nach `/usr/local/(s)bin` kopieren

Server-Konfiguration und Installation

- `~/.gcs-x2go # google-cloud-server-x2go`

```
export SERVER_USE_ROOT=true
export SERVERNAME=demoserver
export USERNAME=demouser
export USERREALNAME="Max Mustermann"
export REMOTEPORT=443
export REMOTEUSER=client_jumphost_username
export REMOTESERVER=client_jumphost_ip_or_dns
createserver && startserver # los geht's
```

- Hinweis: X2GoServer ist in den stretch-backports hinreichend aktuell → X2Go-Repo einbinden unnötig

The background is a solid blue color. At the top, there are two stylized white clouds with dotted outlines. At the bottom, there is a decorative border consisting of a series of white, overlapping, upward-pointing curved shapes. In the center of the slide, the text "Live-Demo mit Jump Host" is written in a white, serif font.

Live-Demo mit Jump Host

Client-Konfigurationsunterschiede

- Grundsätzlich:
X2GoClient-Verbindung konfigurieren auf den frei gewählten SSH-Tunnel-Port und IP 127.0.0.1, sowie den Benutzernamen in der Changeroot-Umgebung, Sitzungstyp XFCE oder Published Applications
- Client hat selbst öffentliche IP:
 - keine Proxy-Einstellung
- Jump Host wird verwendet:
 - SSH-Proxy-Einstellung:
 - IP des Jump Hosts
 - Port des SSH-Servers auf dem Jump Host
 - Zugangsdaten (User/Keyfile) des Jump Hosts

Wie viel Aluhut hätten's denn gern?

- völlige Anonymität ist schwer zu erreichen
- aber nicht unmöglich
- wir brauchen ein anonymes Google-Konto (*John Doe*)
 - Google will per SMS einen Bestätigungscode senden
 - nicht personalisierte Prepaid-SIM verwenden
 - In Liechtenstein gibt es die angeblich weiterhin
 - In Österreich dagegen bald nicht mehr
 - Extra Handy dafür anschaffen, nicht das eigene
 - Billigmodell oder gebraucht, gegen Cash
 - Nie den eigenen Internetzugang verwenden
 - Free-WiFi-Angebot oder UMTS der Prepaid-SIM
- Tipps/mehr Details in den 3 Folien am Ende

Paranoialevel: Aluhut Forte

- Bei UMTS-/FreeWifi-Nutzung eigenes Handy gar nicht erst mitführen, sondern daheim lassen (Bewegungsprofil)
- 2FA für den Google-Account nur dann einrichten, wenn auf dem Mobilgerät ein Token-Generator ohne Google-Account-Verknüpfung vorhanden ist
 - FreeOTP könnte funktionieren, ungetestet
 - Für nicht ganz so paranoide Leute reicht Google Authenticator (getestet: läuft auch ohne Netzanbindung; aber keine Garantie, dass er nicht doch mal nach Hause telefoniert)

Paranoialevel: Aluhut Extreme

- zum Anlegen des Google-Accounts und zum Einrichten/Starten des Servers nur ein Live-Linux, z.B. <https://tails.boum.org/>, von CD/DVD oder USB-Stick mit geprüftem Hardware-Schreibschutz verwenden
→ sauberer Browser, keine verräterischen Cookies/etc.
- nicht ganz so paranoide Leute benutzen in ihrem normalen Browser die „Private Browsing“-Funktion
- ein Angreifer könnte auch die Metadaten (Wer verbindet sich mit wem) auswerten versuchen
→ Risiko bei FreeShell als Jump Host etwas verringert:
 - Zuordnung, wer mit wem connected, schwieriger
 - Tipp: Client-Connect random verzögert durchführen

Ein bisschen X2Go-Werbung

- X2Go-Jahresevent „X2Go: The Gathering 2019“
 - 27.-29. September 2019, im Linuxhotel in Essen
 - Talks und Abendprogramm/Sightseeing
 - Alles zum Selbstkostenpreis
- Seit 2.7. wieder performanter Browser (PaleMoon 28.6)
- Upcoming features (Releasezeitraum Q3/2019):
 - X2Go-HTML5-Client → Full-Desktop im Browser
 - Full-Desktop-Unterstützung für Gnome3, KDE5/Plasma, 3D → Nennt sich X2Go-KDrive
 - Remmina-Plugin

Liste von FreeShell-Anbietern

- Diese Freeshell-Anbieter sollten alle in der Lage sein, einen Login per SSH und ein dabei angegebenes Portforwarding zu erlauben
 - Registrierung per E-Mail (Wegwerf-E-Mail nutzen):
 - <https://sdf.org/>
 - <https://www.xshellz.com/signup>
 - Registrierung, indem man ein ASCII-PONG gewinnen muss:
 - <http://bitcoinshell.mooco.com/>

Anonym nutzbare Internetzugänge

- Nicht das WLAN des Arbeitgebers!
- FreeWiFi-Angebote, die keine persönlichen Daten wie Handynummer wollen
 - Einige Cafés etc. bieten so was an
 - Freifunk (loggt gar nichts – nicht mal MAC-Adresse)!
→ selber einen Knoten betreiben!
- Mullvad VPN (siehe nächste Folie)
- UMTS-Verbindung über anonyme Prepaid-SIM (Liechtenstein ist kein EU-Land, aber im EWR
→ Roaming zu Inlandskonditionen)
 - SIM nicht in personalisierten UMTS-Stick stecken
 - Hotspot-Modus des anonymen Handys nutzen

Mullvad VPN

- Erlaubt Portforwarding von einer öffentlichen VPN-IP zur IP des Clients, siehe <https://mullvad.net/de/guides/> (Abschnitt *Port forwarding with Mullvad VPN*)
- Akzeptiert als anonyme Zahlungsmethoden:
 - Geldscheine in Briefumschlag
 - Bitcoin
 - Bitcoin Cash
- Kostet 5€/Monat
- Account anlegen:
<https://mullvad.net/de/account/create/>

The background is a solid blue color. At the top, there are two stylized white clouds with dotted outlines. At the bottom, there is a decorative border consisting of a series of white, overlapping, pointed shapes resembling a scalloped edge or a row of small mountains.

Live-Demo mit Jump Host

The background is a solid blue color. In the top left and top right corners, there are stylized white clouds with dotted outlines. At the bottom of the image, there is a decorative border consisting of a series of white, overlapping scalloped shapes.

Vielen Dank für euer Interesse!