

All in One - SmartNode Anleitung



SmartCash



Datum:

10. August 2018

SmartCash Client:

v1.2.4

Autor:

Azuuri

Inhaltsverzeichnis

1. Allgemeines	2
• Was sind SmartNodes	2
• Was benötige ich für eine SmartNode	2
• Wie kann ich durch das Betreiben einer SmartNode profitieren	2
2. VPS Server Erstellung	3
3. Desktop Node Client Setup und Konfiguration	4
4. Server Erstellung und SmartNode Konfiguration	5
5. Aufsetzen mehrerer SmartNodes	9
6. Überwachung meiner SmartNodes	11
7. Unterstützung beim Konfigurieren der SmartNode	11

1. Allgemeines

• Was sind **SmartNodes**

Eine SmartNode ist ein Server, auf dem eine SmartCash Node Client Wallet läuft, welche bestimmte Aufgaben für das SmartCash Network erfüllt. Eine Funktion, die durch SmartNodes ermöglicht wird ist InstantPa, dies speichert eingehende Transaktionen ab um die durch Node Kommunikation zu verifizieren. Der Empfänger bekommt seine SmartCash sofort und muss nicht auf den nächsten gemindten Block warten, welcher die Transaktion verarbeitet.

SmartNodes haben zu jeder Zeit Überblick über jede Adresse im Netzwerk, sodass sie ebenfalls über alle qualifizierten SmartRewards Adressen Bescheid wissen.

• Was benötige ich für eine **SmartNode**

Um eine SmartNode aufsetzen zu können, wird folgendes benötigt:

1. 10 000 SmartCash
2. Eine SmartCash Desktop Node Client Wallet, auf der deine Coins liegen
3. Einen VPS auf dem Ubuntu installiert ist und welcher 24/7 online ist

Die **10 000 SmartCash** kannst du auf einer der Börsen kaufen, auf denen SmartCash gehandelt wird, hier findest du einer vollständige Liste: [Exchange Liste](#)

Die **Desktop Node Client Wallet** kannst du dir hier für ein jeweiliges Betriebssystem herunterladen: [Desktop Node Client](#)

Der **VPS** muss mindestens diesen Anforderungen entsprechen: **1GB RAM, 20 GB Speicher** und eine **statische IPV4 Adresse** per Node. Näheres dazu im Abschnitt *Server Erstellung und SmartNode Konfiguration*.

• Wie kann ich durch das **Betreiben einer SmartNode** profitieren

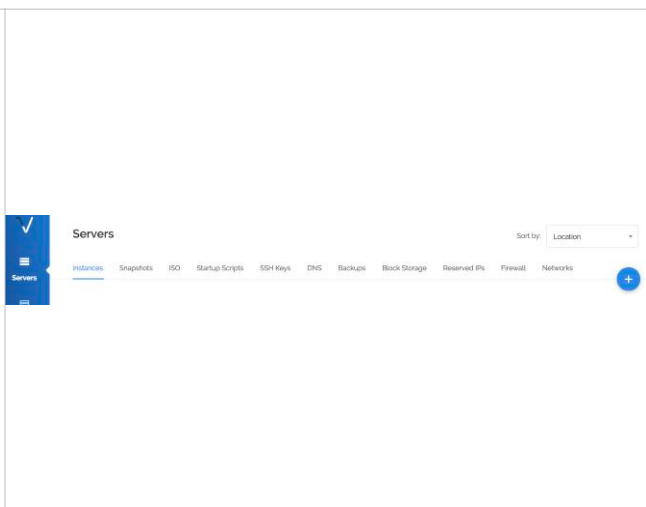
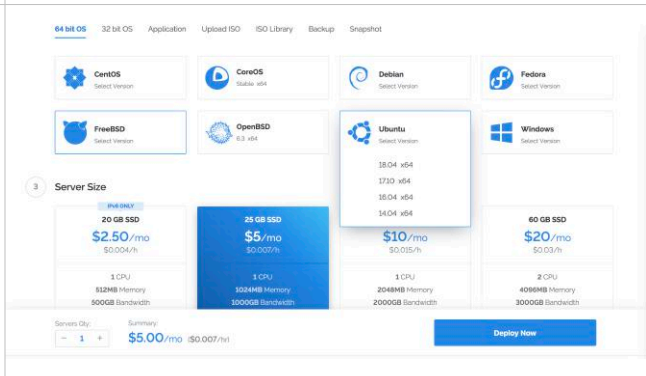
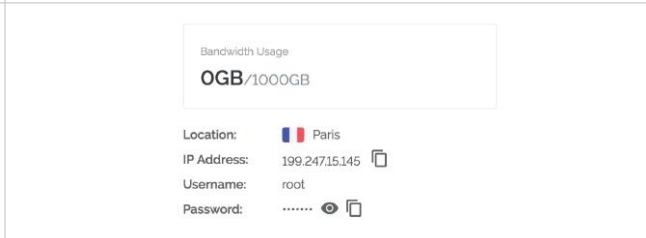
Da SmartNodes 24/7 online sein müssen und Aufgaben für das Netzwerk erfüllen, bekommen sie als Gegenleistung einen Teil des Block Rewards und zwar 10%.

Jeden zweiten Block werden jeweils 10 SmartNodes ausbezahlt, sodass jede Node 2% des BlockRewards pro Auszahlung bekommt.

Eine frisch aufgesetzte SmartNode muss erst einen vollständigen Payout Zyklus abwarten, bis sie sich auch einreihen darf.





Da SmartNode Adressen auch über 1000 SmartCash halten, sind sie SmartRewards berechtigt, was ebenfalls einen finanziellen Anreiz bietet ([Smartrewards.cc](https://www.smartrewards.cc)).

2. VPS Server Erstellung

<p>In dieser Anleitung benutzen wir vultr.com als Provider, dir steht es aber frei, einen anderen Provider deiner Wahl auszuwählen.</p> <p>Zuerst, gehe zu deren Website und erstellen einen Account. Im Beispiel für vultr benötigst du eine Kreditkarte, in weiterer Folge kannst du deine Zahlungen aber zu Beispiel auch mit Bitcoin abwickeln.</p> <p>Nun da der Account aktiviert ist, klicke auf das Blaue + um eine neue Serverinstanz zu erstellen.</p>	
<p>Zuerst wähle einen Standort, dieser kann frei ausgewählt werden.</p> <p>Als Servertyp wähle 64bit Ubuntu 16.04 oder 18.04. Als Servergröße wähle 1GB RAM für 5\$/Monat.</p> <p>Überspringe die nächste Auswahlmöglichkeiten und benenne deinen Server, zum Beispiel SmartNode01.</p>	
<p>Nun musst du einiges Minuten warten, bis der Server fertig installiert ist. Sobald dies geschehen ist, wirst du per Email informiert.</p>	

Nun da der Server einsatzbereit ist, gehe in die Details, dort findest du unter anderem die IP Adresse.

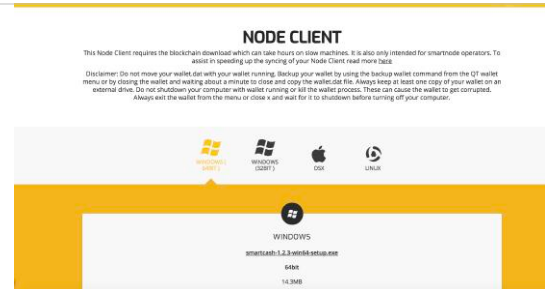
Kopiere diese IP Adresse und speichere sie irgendwo leicht zugänglich ab, da du sie für die nächsten Schritte im Desktop Node Client brauchen wirst.

Location:  Silicon Valley
 IP Address: 45.77.187.88 
 Username: root
 Password:  

3. Desktop Node Client Setup und Konfiguration

Lade dir den aktuellen Desktop Node Client für dein Betriebssystem herunter und installiere ihn:

<https://smartcash.cc/wallets/#nodeclient>



Nach der fertigen Installation, öffne deine Wallet und warte bis diese sowohl die Blockchain, als auch die SmartNode Daten fertig synchronisiert hat. Das Häkchen ganz rechts unten zeigt dir, wenn es soweit ist.

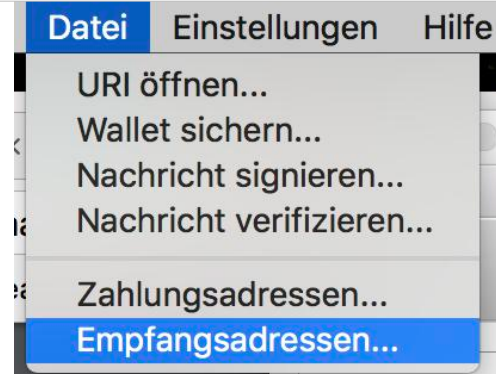


Nun erstelle dir eine neue Empfangsadresse, auf die du deine kürzlich erworbenen 10 000 SmartCash schicken kannst. Hierzu klickst du auf:

Datei -> *Empfangsadressen*


Klicke auf *Neu* und benenne sie z.B. SmartNode1.

Kopiere dir diese Adresse nun.



Sende nun exakt 10 000 SmartCash an diese Adresse. Berücksichtige etwaige Gebühren, da nur eine Transaktion mit exakt 10 000 SMART als Pfand angenommen wird. Sobald deine Coins angekommen sind, sieht dies so aus -> (Transaktionen von Adressen innerhalb der Wallet werden nur als abgezogener Fee angezeigt.)

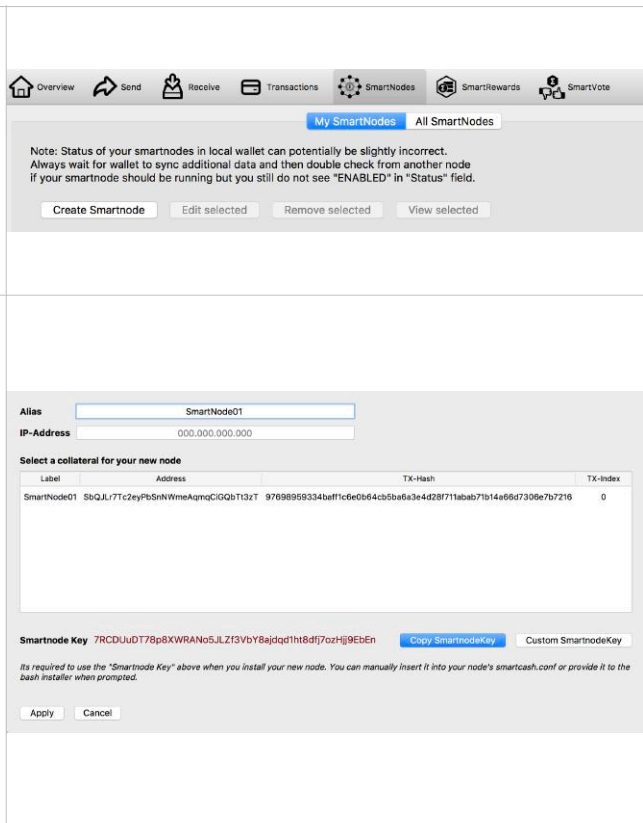
Letzte Transaktionen

 27.06.18 01:53 +10 000.00000000 SMART
 SmartNode1

Navigiere nun zum *SmartNodes* Reiter und klicke auf:

Create SmartNode

1. Wähle einen Namen für deine SmartNode, (z.B. SmartNode01).
2. Nun kopiere die vorher eben gespeicherte IP Adresse und füge sie im dazugehörigen Feld ein.
3. Die vorher gesendete Pfandtransaktion wird nun als collateral angezeigt, wähle sie aus.
4. Ein SmartNode Genkey wird ebenfalls automatisch für dich erstellt. Kopiere diesen und speichere ihn ebenfalls ab, da du ihn für die weiteren Schritte am Server benötigen wirst.
5. Klicke auf Apply und deine Arbeit innerhalb des Desktop Node Client ist erstmal erledigt.



4. Server Erstellung und SmartNode Konfiguration

Nun ist es Zeit, uns auf unseren wohin erstellen Server zu loggen und die SmartNode zu konfigurieren.

Vultr bietet eine integrierte Konsole an, von dieser rate ich jedoch ab, da copy&paste nicht unterstützt wird und einige Zeichen nur durch spezielle Tastenkombinationen zu erreichen sind.

Für Windows Nutzer empfiehlt sich PuTTY (Website Download), Mac Nutzer können dazu das integrierte Terminal nutzen.

Für die weitere Anwendung bitte beachtet, das Einfügen in PuTTY über einen Rechtsklick erfolgt, im Mac-Terminal kann man dies mit command+v lösen. Die Befehle sind für beide Varianten die selben.

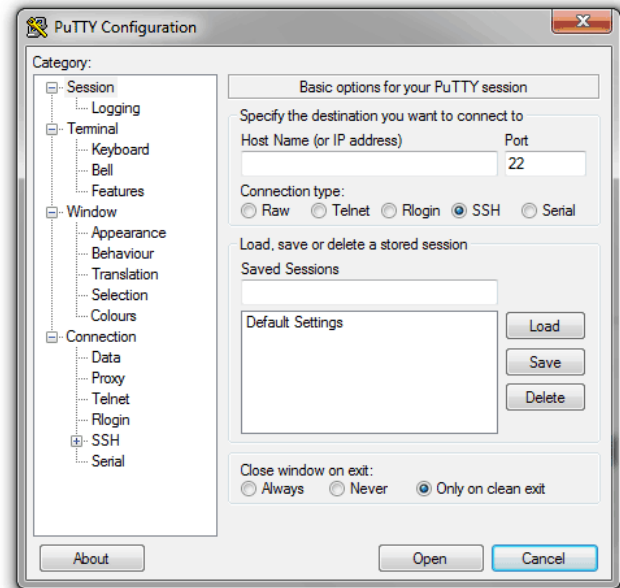
Login über PuTTY:

Im Feld IP Adresse fügen wir nun die IP-Adresse ein, auf der der eben erstellte Server läuft. Port bleibt erstmal 22 und Zugriffstyp wird SSH ausgewählt. Dann klicken wir auf Öffnen.

Im folgenden Feld wird nun gefragt unter welchem Benutzernamen man sich einloggen will. Hier schreiben wir `root` und bestätigen mit Enter.

Danach wird nach dem Passwort gefragt, hierzu gehen wir zurück in die Serverübersicht und kopieren uns das Passwort. Dieses fügen wir mit einem Rechtsklick in die PuTTY Konsole ein und bestätigen mit Enter. (ACHTUNG: Das Passwort wird NICHT angezeigt, ein Rechtsklick fügt es jedoch ein)

Nun sind wir mit dem Server verbunden.



Login über Mac Terminal:

Zuerst kopieren wir uns die IP Adresse unseres gerade erstellten Servers und im Mac Terminal schreiben wir:

```
ssh root@IPAdresseDesServers
```

Danach wird nach dem Passwort gefragt, hierzu gehen wir zurück in die Serverübersicht und kopieren uns das Passwort. Dieses fügen wir mit `cmd+v` im Terminal ein und bestätigen mit Enter. (ACHTUNG: Das Passwort wird NICHT angezeigt, `cmd+v` fügt es jedoch ein)

Nun sind wir mit dem Server verbunden.

```
Lukass-MBP:~ azuuri$ ssh root@199.247.15.145
root@199.247.15.145's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jul 1 01:08:17 UTC 2018

System load: 0.07          Processes:            82
Usage of /:   8.4% of 24.55GB   Users logged in:    0
Memory usage: 12%          IP address for ens3: 199.247.15.145
Swap usage:   0%

 * Meltdown, Spectre and Ubuntu: What are the attack vectors,
   how the fixes work, and everything else you need to know
   - https://ubu.one/u2kNow

0 packages can be updated.
0 updates are security updates.

root@SmartNodeTest:~#
```

Um die SmartNode nun Aufzusetzen schreiben wir (URL in einem, siehe Bild!):

```
wget https://  
raw.githubusercontent.com/SmartCash/  
smartnode/master/install.sh
```

Und danach:

```
bash ./install.sh
```

Direkt nach der Eingabe des zweiten Befehls, werden wir gewarnt, dass der Server rebooten wird, dies bestätigen wir mit Enter. Danach werden wir nach einem Custom Port gefragt, aus Sicherheitsgründen empfiehlt es sich den Standardport (22) zu wechseln, also schreibt hier eine beliebige Zahl hinein (<1000) und bestätigt mit Enter.

Nun werden wir nach dem SmartNode Genkey gefragt, den wir uns vorhin über den Desktop Node Client generiert haben und in unserer *smartnode.conf* Datei benutzt haben. Kopiert diesen und fügt in in PuTTY oder dem Terminal ein und bestätigt ebenfalls mit Enter.

Nun wird einiges installiert, nachdem dies abgeschlossen ist, müssen wir uns wie zuvor einloggen (bedenkt die Port Änderung solltet ihr eine vorgenommen haben).

Die Wallet wird nun die Blockchain synchronisieren, um dies zu verfolgen schreiben wir:

```
smartcash-cli getinfo
```

Den aktuellen Block findet man [hier](#). Solltet ihre eine geringe Anzahl an Connections bekommen schreibt:

```
rm ~/.smartcash/peers.*  
smartcash-cli stop  
smartcashd
```

```
root@SmartNodeTest:~# wget https://raw.githubusercontent.com/SmartCash/smartnode  
/master/install.sh  
--2018-07-01 01:11:00-- https://raw.githubusercontent.com/SmartCash/smartnode/m  
aster/install.sh  
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.120.1  
33  
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.120.  
133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 4431 (4.3K) [text/plain]  
Saving to: 'install.sh'  
  
install.sh      100%[=====] 4.33K  --.-KB/s   in 0s  
  
2018-07-01 01:11:00 (137 MB/s) - 'install.sh' saved [4431/4431]  
  
root@SmartNodeTest:~# bash ./install.sh  
WARNING: This script will reboot the server when it's finished.  
Press Ctrl+C to cancel or Enter to continue:  
Custom SSH Port(Enter to ignore):  
SmartNode GenKey: 7QY6C4bDi9hXSQnpL5CF2fBAFo3aQPDLKjv19JbFhJoKjqFgWre
```

```
{  
  "version": 1020400,  
  "protocolversion": 90026,  
  "walletversion": 130000,  
  "balance": 0.00000000,  
  "blocks": 566632,  
  "timeoffset": 0,  
  "connections": 6,  
  "proxy": "",  
  "difficulty": 157275.8255496097,  
  "testnet": false,  
  "keypoololdest": 1531334732,  
  "keypoolsize": 199,  
  "paytxfee": 0.00000000,  
  "relayfee": 0.00100000,  
  "errors": ""  
}
```


Die Zeit in der die Blöcke aufgeholt werden, nutzen wir, indem wir swap memory hinzufügen. Da SmartNodes auch mal über 1 GB kommen können, ist dies eine nicht notwendige, aber empfohlene Maßnahme und etwaigen Problemen vorzubeugen. Dazu geben wir folgende Befehle:

```
fallocate -l 4G /swapfile
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
cp /etc/fstab /etc/fstab.bak
echo '/swapfile none swap sw 0 0' | tee -a /etc/fstab
```

Dies fügt euch 4GB zusätzlich hinzu.

```
root@SmartNodeTest:~# fallocate -l 4G /swapfile
root@SmartNodeTest:~# chmod 600 /swapfile
root@SmartNodeTest:~# mkswap /swapfile
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
no label, UUID=36982b4c-9957-4126-afdf-2aa119035e2f
root@SmartNodeTest:~# swapon /swapfile
root@SmartNodeTest:~# cp /etc/fstab /etc/fstab.bak
root@SmartNodeTest:~# echo '/swapfile none swap sw 0 0' | tee -a /etc/fstab
/swapfile none swap sw 0 0
root@SmartNodeTest:~# sudo swapon --show
NAME      TYPE  SIZE USED PRIO
/swapfile file  4G 524K  -2
root@SmartNodeTest:~#
```

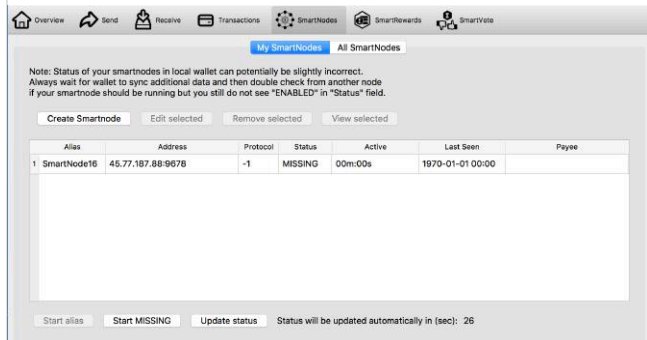
Sobald die Synchronisation nun vollendet ist, starten wir ebenfalls wieder unseren Desktop Node Client und lassen auch diesen synchronisieren.

Ist beides erledigt, checken wir am Server unseren SmartNode Status mit:

```
smartcash-cli smartnode status
```

Dieser gibt uns nun: *Not capable smartnode: Smartnode not in smartnode list*

Ein letzter Schritt ist nun nötig, dazu switchen wir zum Desktop Node Client, klicken auf **SmartNodes** und **My SmartNodes**. Dort wählen wir unsere Node aus und klicken unten aus **Start MISSING**.



Ihr werdet gefragt, ob ihr wirklich MISSING starten wollt, klickt auf Ja.

Danach sollte dieses Fenster aufpoppen und der Status der Node ändert sich auf PRE-ENABLED.

Dieser ändert sich später auf ENABLED, keine Sorge wenn dies nicht innerhalb von 30 Minuten passiert, es kann oftmals auch länger dauern.

Successfully started 1 smartnodes, failed to start 0, total 1

OK

Zum Schluss checken wir nun den Status der SmartNode am Server, **denn nur dieser gibt uns IMMER korrekte Information über den Zustand der Node.** Dazu schreiben wir wieder:

```
smartcash-cli smartnode status
```

Wenn du diesen Status erhältst ->

darf ich dich beglückwünschen, du hast soeben deine erste SmartNode erfolgreich aufgesetzt!!

```
"status": "Smartnode successfully started"
```

5. Aufsetzen mehrerer SmartNodes

Das Aufsetzen einer oder mehrerer weiterer SmartNodes erfolgt prinzipiell genauso, wie das Aufsetzen der ersten SmartNode. Das heißt:

- Exakt 10 000 SmartCash an eine neu generierte Empfangsadresse schicken.
- In der Desktop Client Console einen neuen Genkey generieren (*smartnode genkey*)
- Mit *smartnode outputs* die ID und den Index der Pfandtransaktion ermitteln.
- Die neuen Daten als weitere Zeile in der *smartnode.conf* Datei einfügen ->
- Einen weiteren Server anmieten und wie beim ersten Mal einrichten.
- Warten bis alles fertig synchronisiert ist und die Node über den Desktop Node Client mit *Start MISSING* starten.

Also, jede SmartNode benötigt einen eigenen Genkey und eine eigene IP, somit auch einen eigenen Server und eine weitere Zeile in der *smartnode.conf* Datei

```
SmartNode1 199.247.15.145:9678 70Y6C4bd19hXSQnPL5CF2f8AFo3aQPDLKjv19JbfHjokiaFgWre  
1874a8220a0de35e9f3351e821f86a7acf98de1f2ce0b954b5f1fea6b0d4f02c1a 0  
SmartNode2 34.545.34.145:9678 7G0OmI8hUnf75CF2f8AFo3aQPDLKjv19JbfHjokiaFgWre  
1456a8220a0de35e9f3351e821f86a7acf98de1f2ce0b954b5f1fea6b0d4f02c1a 1
```

Als Alternative für **Fortgeschrittene Benutzer**, bieten mehrere VPS Provider die Möglichkeit an, einen Snapshot vom aktuellen Server zu machen und diesen auf einen neuen zu kopieren.

Dies ermöglicht es uns, dass wir auf Serverseite nur die *smartcash.conf* Datei anpassen müssen und uns viele Schritte ersparen.

Auf Desktop Node Client Seite bleibt alles gleich, ihr benötigt eine neue 10 000 SmartCash Pfandtransaktion, einen neuen Genkey und die neue IP vom Server.

Habt ihr eure lokale *smartnode.conf* dementsprechend angepasst, loggt ihr euch in den neuen Server ein und schreibt:

```
nano ~/.smartcash/smartcash.conf
```

Dort passt ihr euren Genkey und die externe IP Adresse an und bestätigt das Ganze mit Control + X.

Danach tippt ihr:

```
smartcash-cli stop  
smartcashd
```

Damit eure Änderungen in Kraft treten. Nachdem ihr euren Desktop Client erneut geöffnet habt und dieser fertig synchronisiert hat, könnt ihr eure zweite SmartNode nun mit *Start MISSING* starten.

Snapshots

Instances

Snapshots

ISO

Add Snapshot

6. Überwachung meiner SmartNodes

Grundsätzlich lässt sich der Status einer SmartNode über den Server durch den Befehl `smartcash-cli smartnode status` immer feststellen. Da dies aber recht mühsam ist und auch das Öffnen der Wallet um eingehende Node Zahlungen im Überblick zu haben auf Dauer anstrengend ist, empfiehlt sich der **SmartNode Monitor Bot**.

Durch simples hinzufügen deiner SmartNode IP Adressen, informiert dich dieser über Statusänderungen, Nodezahlungen, deine Position in der Reihenfolge und etliche andere Dinge.

Der SmartNode Monitor Bot ist für Discord und Telegram User zu benutzen und eine Anleitung, wie man ihn benutzt findet man hier: [SmartNode Monitor Bot Guide](#)

7. Unterstützung beim Konfigurieren der SmartNode

Solltest du beim Anwenden dieser Anleitung bei bestimmten Schritten nicht weiter kommen, empfiehlt es sich unseren SmartCash Community Discord unter discord.smartcash.cc aufzusuchen und dort im Channel #smartnodes und #support um Hilfe zu bitten.