



Just a simple cloud

Dokumentation für Experten

Inhaltsverzeichnis

1.	JUST A SIMPLE CLOUD FÜR EXPERTEN	4
1.1.	Einleitung	4
2.	GRUNDLAGEN ZUM BETRIEB EINES SERVERS	5
2.1.	Adressierung im Internet	5
2.2.	Private Internet-Anschlüsse	5
2.3.	Dynamisches DNS	6
2.4.	Firewall	6
2.5.	Relais-Server	7
3.	KONFIGURATION	8
3.1.	Grundlegende Vorgaben	8
3.2.	Konfiguration des RasPi Servers	9
3.2.1.	Abschnitt „General“	9
3.2.2.	Abschnitt „Projects“	10
3.2.3.	Server Backups	10
3.3.	Externe Festplatten	11
3.4.	FTP Server	11
3.5.	Konfiguration des Windows-Clients	12
3.5.1.	Abschnitt „General“	13
3.5.2.	Abschnitt „Projects“	13
3.6.	Zugriff auf Linux / Updates	14
4.	ANDROID APP	15
4.1.	Batterie-Optimierung	15
4.2.	Berechtigungen	15
4.3.	Teilen von Fotos / Videos	16
4.4.	Bereitstellen von Fotos / Videos	17
4.5.	Android Version 7 bis 10	18
4.6.	Android 11 und später	18
4.7.	Weitere Einstellungen	19

5.	RECHTE-VERWALTUNG	20
5.1.	Voraussetzungen	20
5.2.	Zugriffssteuerung	21
5.3.	Aufbau der XML-Dateien	22
5.4.	Home-Verzeichnis	22
5.5.	Administratoren	22
6.	TOOLS	23
6.1.	Passwort Generator (Windows)	23
7.	ANMERKUNGEN	24
7.1.	Passwort Generator (Linux)	25
7.2.	Server Remote (nur Windows)	26

1. Just a simple cloud für Experten

1.1. Einleitung

„Just a simple cloud“ (kurz „JASC“) bietet Ihnen die Möglichkeit, Ihre **Dateien** auf **Ihrem eigenen Raspberry Pi - Server** (kurz „RasPi“) zur Verfügung zu stellen und mit Hilfe der Clients von **überall auf der Welt verschlüsselt** darauf **zuzugreifen** und **auszutauschen**. Dabei wird sichergestellt, dass Ihre Dateien auf allen Clients immer synchron gehalten werden. Die Verbindung zwischen dem Server und den Clients ist jederzeit vollständig verschlüsselt, damit nur Sie Zugriff darauf haben. So behalten Sie die **volle Kontrolle über Ihre Daten** und müssen diese nicht einem Anbieter zur Verfügung stellen, bei dem Sie nicht wissen, wer außer Ihnen noch auf Ihre Dateien zugreift.

Diese Anleitung ergänzt die „Bedienungsanleitung Raspberry-Server & Clients“ und erläutert die detaillierte Konfiguration des gesamten Systems.

2. Grundlagen zum Betrieb eines Servers

Dieses Kapitel soll in die Grundlagen einführen, die zum Betrieb eines Servers im Internet nötig sind.

2.1. Adressierung im Internet

Jeder Server-Dienst, der im Internet erreichbar ist, wird über 2 Parameter angesprochen, die bekannt sein müssen, um ihn zu erreichen:

- Server-Adresse
- Server-Port

Die Server-Adresse wird in den meisten Fällen in einer für uns Menschen einfach lesbaren Form geschrieben, z.B. ist die Adresse unserer Webseite:

www.just-a-simple-cloud.com

Diese Darstellung lässt sich leicht merken. Für Computer muss sie jedoch in eine sogenannte IP-Adresse umgesetzt werden. Übersetzt man die obige Adresse in eine IP-Adresse, so erhält man:

212.72.175.201

Diese Umsetzung geschieht mit Hilfe des sogenannten „**Domain Name Service**“ (**DNS**), der dafür sorgt, dass beim Umzug eines Servers von einem Rechenzentrum in ein anderes, die IP-Adresse automatisch auf die des neuen Rechenzentrums aktualisiert wird. Grundsätzlich lässt sich als Adresse sowohl die Text-Form als auf die IP-Adresse gleichrangig verwenden.

Da auf einem Server in der Regel mehrere Dienste laufen (z.B. ein Web-Server und ein E-Mail-Server) braucht es zusätzlich einen **Port**, um unterscheiden zu können, welchen Dienst man ansprechen möchte. Die meisten Dienste laufen auf standardisierten Ports, z.B.:

- Web-Server unverschlüsselt (**http://**) Port 80
- Web-Server verschlüsselt (**https://**) Port 443
- E-Mail **POP3** Port 110

Eine genauere Beschreibung der Ports findet sich auf:

[https://de.wikipedia.org/wiki/Port_\(Protokoll\)](https://de.wikipedia.org/wiki/Port_(Protokoll))

2.2. Private Internet-Anschlüsse

Die Internet-Anschlüsse von gewerblichen Anbietern und Firmen sind fast immer festen IP-Adressen zugeordnet. Bei privaten Anschlüssen vergibt der Internet-Provider jedoch bei jeder neuen Einwahl in das Internet eine neue (zufällige) IP-Adresse. Selbst bei Anschlüssen, die scheinbar dauerhaft verbunden sind, finden zeitweise kurze Trennungen statt, gefolgt von einer sofortigen Neu-Einwahl und der Vergabe einer neuen IP-Adresse.

Betreibt man einen Server an einem privaten Anschluss, stellt sich also das Problem, dass sich die Internet-Adresse, unter dem der Server erreichbar ist, öfters ändert und diese Änderungen zunächst nicht bekannt sind.

2.3. Dynamisches DNS

Um trotzdem die sich ständig ändernden IP-Adressen eines privaten Anschlusses herausfinden zu können, nutzt man das sogenannte **dynamische DNS**.

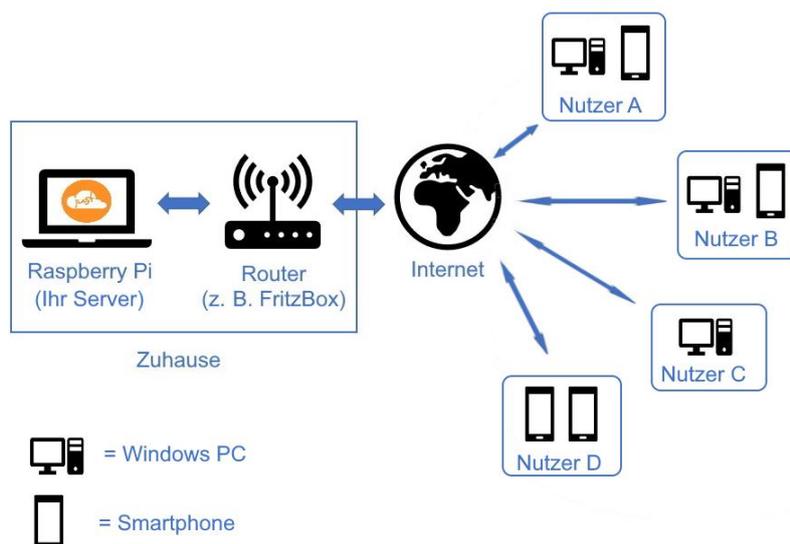
Die Idee dahinter ist ganz einfach: Ein Dienstleister führt eine Datenbank, in der er einen eindeutigen Namen des Anschluss-Inhabers mit der jeweils aktuellen IP-Adresse des Anschlusses verknüpft. Dazu läuft eine kleine Software auf einem Computer des Anschluss-Inhabers, die immer dann, wenn sich die IP-Adresse des Anschlusses ändert, diese Änderung der Datenbank sofort mitteilt.

Um also die IP-Adresse eines privaten Anschlusses herauszufinden, genügt es, die Datenbank des Dienstleisters abzufragen und man erhält immer die korrekte Adresse.

Bei Just a simple cloud ist diese Funktion bereits fest integriert. Sofern sie auf dem Server aktiviert wurde, wird die IP-Adresse automatisch an die Clients übertragen.

2.4. Firewall

Nahezu jeder Internet-Anschluss stellt über einen sogenannten **Router** eine Verbindung zwischen dem Internet und dem Netzwerk im eigenen Haus her.



Eine wesentliche Aufgabe des Routers ist es dabei, eine **Firewall** zur Verfügung zu stellen. Diese schützt vor unberechtigten Zugriffen aus dem Internet auf das häusliche Netzwerk, indem solche Anfragen blockiert werden.

Um einen Server „hinter“ einer Firewall zu betreiben, muss im Router zuvor eine entsprechende Ausnahmeregel („**Port-Forwarding**“) eingestellt werden. Diese Regel beschreibt, wie bestimmte eingehende Verbindungen aus dem Internet an einen bestimmten Server innerhalb des Hauses weitergeleitet werden.

Ohne eine solche Regel ist ein Betrieb eines Servers hinter einer Firewall im Normalfall nicht möglich.

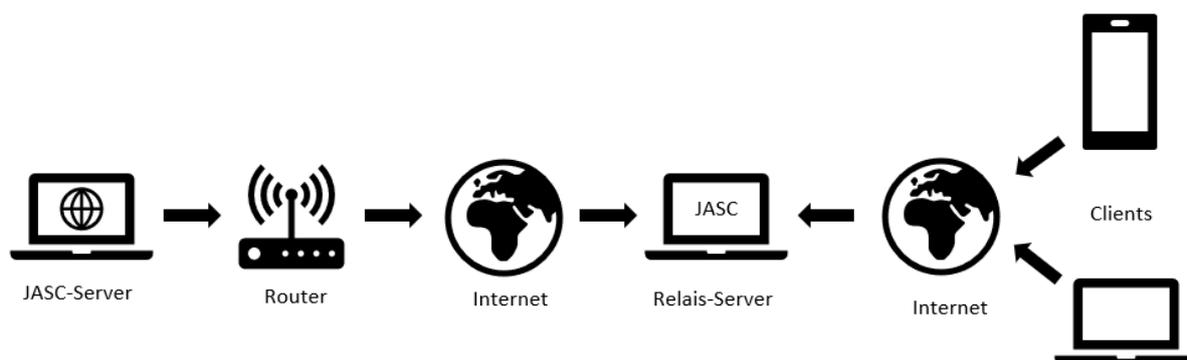
2.5. Relais-Server

Der **Relais-Server** von Just a simple cloud erübrigt sowohl das **dynamische DNS** als auch die **Konfiguration der Firewall** im Router.

Dazu baut der Server („**JASC-Server**“) eine **ausgehende Verbindung** über das Internet **zu unserem Relais-Server** auf.

Innerhalb des Relais-Server wird die **Verbindung** so lange **gehalten**, bis ein **Client** versucht, sich mit **seinem JASC-Server** zu verbinden. Der **Relais-Server vermittelt** dann die Daten zwischen dem JASC-Server und dem Client.

Dadurch wird weder die Server-Adresse des JASC-Servers oder sein Port benötigt und die Einrichtung des Port-Forwardings im Router entfällt ebenfalls.



Selbstverständlich ist dabei die gesamte Kommunikation verschlüsselt. Niemand hat einen Zugriff auf den Inhalt der Daten, die über den Relais-Server vermittelt werden.

3. Konfiguration

Sowohl der RasPi-Server als auch der Windows-Client werden über eine XML-Datei (im Unterverzeichnis „**Config**“) konfiguriert.

Die XML-Dateien werden bereits durch das Setup angelegt und mit den wichtigsten Einstellungen vorkonfiguriert. Dieses Kapitel erläutert, wie Sie selbst Hand anlegen können und die Konfiguration für Ihre Bedürfnisse optimieren können.

3.1. Grundlegende Vorgaben

Die XML-Dateien sind immer in zwei Abschnitte aufgeteilt:

- **General:** enthält grundlegende Einstellungen
- **Project:** enthält Einstellungen, die für das jeweilige Projekt relevant sind

Eine XML-Datei kann mehrere Projekte enthalten. Die Projekte sind dann in der XML-Datei durchnummeriert. D.h. das erste Projekt bildet den Abschnitt „Project 1“, das nächste „Project 2“. Die Nummerierung muss dabei aufsteigend und lückenlos sein.

Gleiches gilt, z.B. für den Abschnitt „User“ innerhalb eines Projektes.

Beispiel:

```
<?xml version="1.0" ?>
<JustASimpleCloudServer_Config>
  <General>
    <!-- License: enter the license key for the server here -->
    <License>inactive</License>
    <!-- Email Administrator: enter the email address of the Admin here, it will be used for notifications -->
    <Email>MyName@MyDomain.com</Email>
    <!-- GlobalKey: used to build up the session key. The global key will be internally combined -->
    <!-- with an internal secret key to exchange the session key at login. -->
    <!-- The GlobalKey must be identical on the server and all clients all over your installation -->
    <!-- The GlobalKey is encrypted, use PasswordGenerator to create a new password -->
    <!-- GlobalKey: initial key is "hello" -->
    <GlobalKey>37c533c8f4ce84808c19b94296e3cf06</GlobalKey>
    <!-- Remote Access: must be activated here: -->
    <ActivateRemote>true</ActivateRemote>
    <!-- Remote Access Passwort: initial password is "hello" -->
    <!-- "PasswordRemote" contains a hash of the password -->
    <PasswordRemote>454f431ca3a68536f90106ba4ddd3dde</PasswordRemote>
    <!-- "PasswordRemote_Encrypted" contains the encrypted password -->
    <PasswordRemote_Encrypted>37c533c8f4ce84808c19b94296e3cf06</PasswordRemote_Encrypted>
    <!-- User privileges can be activated (= all folders are visible and writeable based on XML files) or de-activated (= default) -->
    <ActivateUserPrivileges>true</ActivateUserPrivileges>
    <!-- DNS service: if your server uses dynamic ip addresses, you can use our IP translation service: -->
    <ActivateDNSService>false</ActivateDNSService>
    <!-- Relay service: no need to configure port forwarding in your router, we will do that for you -->
    <ActivateRelayService>true</ActivateRelayService>
    <!-- Auto update: if activated, JASC will check once a day for updates and automatically install them -->
    <AutoUpdate>true</AutoUpdate>
    <!-- Enable webserver: the Apache webserver will be running anyway, but this flag controls, if the JASC server is connected
to Apache -->
    <EnableWebserver>true</EnableWebserver>
    <!-- shall the server automatically delete folders, that are empty (e.g. because the last file has been deleted from it)? -->
    <DeleteEmptyFolders>true</DeleteEmptyFolders>
    <!-- Where shall the server store a backup (if included in license)? -->
    <BackupFolder>./Backup</BackupFolder>
    <!-- Possible values for LogLevel: -->
    <!-- Error: critical errors preventing normal operation -->
    <!-- Warning: errors, which could have been repaired -->
    <!-- Info: all other informations -->
    <!-- IMPORTANT: The loglevel will not be updated, when you change it in the log dialog! -->
    <LogLevel>Error</LogLevel>
    <!-- LogUserActivity: indicates, if user log in/out, update of files shall be logged or not -->
    <LogUserActivity>true</LogUserActivity>
    <ScanIntervallUserrights>30</ScanIntervallUserrights>
    <TCPIPListenPort>50000</TCPIPListenPort>
  </General>
```

```
<Project1>
  <!-- Note: passwords are saved as salted hashes -->
  <Projectname>Testproject</Projectname>
  <Rootpath>ServerRoot/FirstProject</Rootpath>
  <Group1>
    <Groupname>Standard</Groupname>
  </Group1>
  <User1>
    <Username>Username</Username>
    <Realname>My Full Name</Realname>
    <!-- User password: initial password is "hello" -->
    <Password>37c533c8f4ce84808c19b94296e3cf06</Password>
    <Useremail>MyName@MyDomain.com</Useremail>
    <Group1>Standard</Group1>
    <IsAdmin>true</IsAdmin>
  </User1>
</Project1>
</JustASimpleCloudServer_Config>
```

Änderungen an den XML-Dateien werden erst nach einem Neustart des Servers / Clients übernommen.

3.2. Konfiguration des RasPi Servers

3.2.1. Abschnitt „General“

Der Abschnitt „General“ enthält die Grund-Einstellungen für den Server:

- **License:** enthält den Lizenz-Schlüssel für den Server, wird automatisch eingetragen. Bei Updates der Lizenz werden diese Änderungen ebenfalls automatisch übernommen.
- **Email:** enthält die Email-Adresse des Administrators des Servers
- **GlobalKey:** enthält ein Passwort, aus dem die Verschlüsselung für jede einzelne Verbindung zwischen Server und Client abgeleitet wird. Bei jeder Verbindung wird, basierend auf diesem Passwort, ein neuer „Sitzungsschlüssel“ ausgehandelt, mit dem die Kommunikation für diese eine Sitzung (= Verbindung zwischen Server und Client) verschlüsselt wird, bis sie beendet wird. Der Sitzungsschlüssel wird dabei mit Hilfe von Zufallszahlen erzeugt und stellt damit sicher, dass die Verbindung sicher ist.
- **ActivateRemote:** erlaubt den Zugriff mit dem Tool „Server Remote“ (siehe dort)
- **PasswordRemote:** das Passwort (in Form eines Hash) für die Verbindung mit dem „Server Remote“ Tool
- **PasswordRemote_Encrypted:** das Passwort (AES verschlüsselt) für die Verbindung mit dem „Server Remote“ Tool
- **ActivateUserPrivileges:** (de-) aktiviert die Behandlung von Benutzerrechten (siehe dort)
- **ActivateDNSService:** (de-) aktiviert den DNS-Service (siehe dort)
- **ActivateRelayService:** (de-) aktiviert den Relais-Service (siehe dort)
- **AutoUpdate:** (de-) aktiviert automatische Updates. Wenn diese Funktion aktiviert ist, prüft der Server einmal pro Tag (zu zufälligen Zeiten), ob ein Update verfügbar ist und installiert es ggf. automatisch
- **EnableWebserver:** (de-) aktiviert die Ausgaben und die Bedien-Oberfläche des Servers per Webserver
- **DeleteEmptyFolders:** (de-) aktiviert das automatische Löschen von leeren Verzeichnissen auf dem Server
- **BackupFolder:** gibt den Pfad für Server-Backups an
- **LogLevel:** gibt die Wichtigkeit der Log-Ausgaben an, die ausgegeben werden
- **LogUserActivity:** (de-) aktiviert die Log-Ausgaben von User-Aktivitäten
- **ScanIntervallUserrights:** Intervall in Minuten, nach dem die Benutzerrechte im Daten-Verzeichnis überprüft werden
- **TCPIPListenPort:** Auch wenn der Relais-Service nicht verwendet wird, ist der Server immer unter diesem Port erreichbar.

3.2.2. Abschnitt „Projects“

Es besteht die Möglichkeit (abhängig von der Lizenz des Servers), mehrere Projekte parallel anzulegen. Jedes Projekt hat dabei seinen eigenen Speicherbereich auf der SD-Karte des RasPi und ist vollständig von den anderen Projekten getrennt. Sie können also beispielsweise in einem Projekt geschäftliche Daten und in einem anderen Ihre privaten Fotos ablegen, ohne dass Zugriffe auf andere Projekte möglich sind.

Ein Projekt wird über die folgenden Parameter beschrieben:

- **Projectname:** eindeutiger Name des Projektes, über diesen Namen werden die Projekte auf den Clients identifiziert
- **EnableHomeFolder:** aktiviert die automatische Erstellung eines persönlichen Verzeichnisses im Verzeichnis „home“, siehe Kap. **Home-Verzeichnis**
- **Rootpath:** Pfad zum Stamm-Verzeichnis, das alle Dateien des Projektes enthält
- **Group:** eine Auflistung von Benutzer-Gruppen, siehe Kapitel „Rechte-Verwaltung“
- **IsAdmin:** gibt an, ob dieser Nutzer ein Administrator ist
- **User:** Parameter eines einzelnen Users:
 - **Username:** Benutzername, der zur Anmeldung am Server verwendet wird
 - **Realname:** (optional) Klarname des Nutzers
 - **Password:** Passwort des Nutzers (auf dem Server werden Passwörter als „gesalzene“ Hashes abgelegt)
 - **Useremail:** (optional) Email-Adresse des Nutzers
 - **Group:** eine Auflistung von Nutzergruppen, zu denen dieser Nutzer gehört, siehe Kapitel „Rechte-Verwaltung“

3.2.3. Server Backups

Sofern die Backup-Funktion über die Lizenz des Servers aktiviert ist, legt der Server jeweils um Mitternacht ein Backup aller Daten an. Dazu wird der gesamte Inhalt des „Rootpath“ (siehe oben) in einer ZIP-Datei gepackt.

Voraussetzung für das Anlegen des Backups ist, dass noch genügend Speicherplatz (in der Regel auf der verwendeten SD-Karte) zur Verfügung steht.

Das Backup wird für jeden Wochentag in einem eigenen ZIP-Archiv abgelegt, das den Namen des Wochentages im Dateinamen enthält. D.h. die Backups werden nach spätestens einer Woche überschrieben.

Den Pfad zum Backup können Sie in der XML-Konfiguration im Abschnitt **General** unter **BackupFolder** einstellen.

Steht nicht mehr genug Speicherplatz auf der SD-Karte für ein Backup zur Verfügung, wird dieses übersprungen.

Backups sollten unbedingt auf einer SSD oder einer externen Festplatte gespeichert werden.

3.3. Externe Festplatten

In der Grundeinstellung werden alle Dateien auf der SD-Karte Ihres RasPi gespeichert.

Wenn Sie **mehr Speicherplatz** benötigen, können Sie eine externe Festplatte an den USB-Anschluss Ihres RasPi anschließen und diese zur Ablage Ihrer Dateien verwenden.

Um eine externe Festplatte zu verwenden, **stoppen Sie zunächst den Server** über die Web-Oberfläche. Danach können Sie die Dateien auf die externe Festplatte transferieren. Passen Sie dann im Abschnitt „**Projects**“ der **Server XML-Konfigurationsdatei** (siehe Kap. 3) den Eintrag **Rootpath** entsprechend an. Als Letztes **starten Sie den RasPi** komplett neu.

Ebenso ist es auch möglich, **NAS-Laufwerke** o.ä. zu verwenden.

3.4. FTP Server

Der RasPi verfügt über einen vorkonfigurierten FTP-Server:

- Server Name = JASC-RasPi
- Port: 21
- Username: pi
- Passwort: raspberry

3.5. Konfiguration des Windows-Clients

Die Konfigurationsdatei des Windows Clients findet sich in folgendem Pfad:

%LOCALAPPDATA%\JASC_Client

Beispiel:

```
<?xml version="1.0" ?>
<!-- XML validator: https://www.w3schools.com/xml/xml_validator.asp -->
<JustASimpleCloudClient_Config>
  <General>
    <!-- ClientName: identifies this client on the server. Will be generated automatically during installation. -->
    <ClientName>8A473090B8B928B13EF9B1AF833D53C2C5104F46D39B871E4AF2E484AF890FF6</ClientName>
    <!-- GlobalKey: used to build up the session key. The global key will be internally combined -->
    <!-- with an internal secret key to exchange the session key at login. -->
    <!-- The GlobalKey must be identical on the server and all clients all over your installation -->
    <!-- The GlobalKey is encrypted, use PasswordGenerator to create a new key -->
    <GlobalKey>will be inserted here</GlobalKey>
    <!-- Relay service: no need to configure port forwarding in your router, we will do that for you -->
    <ActivateRelayService>true</ActivateRelayService>
    <!-- Relay service: enter machine identifier of server here -->
    <RelayServiceIdentifier>will be inserted here</RelayServiceIdentifier>
    <!-- The client will show ballon messages on important notifications, if allowed -->
    <ShowBallonMessages>true</ShowBallonMessages>
    <!-- The client will try to set the root folder icon to the JASC icon, if allowed -->
    <SetClientFolderIcon>true</SetClientFolderIcon>
    <!-- Disc space warning level: a warning will be raised once, when the available disc space on the RasPi will go below this threshold -->
    <DiscSpaceWarningLevel>40</DiscSpaceWarningLevel>
    <!-- Disc space error level: an error will be raised once, when the available disc space on the RasPi will go below this threshold -->
    <DiscSpaceErrorLevel>20</DiscSpaceErrorLevel>
    <!-- Possible values for LogLevel: -->
    <!-- Error: critical errors preventing normal operation -->
    <!-- Warning: errors, which could have been repaired -->
    <!-- Info: all other informations -->
    <!-- IMPORTANT: The loglevel will not be updated, when you change it in the log dialog! -->
    <LogLevel>Error</LogLevel>
    <ScanIntervallProjects>1</ScanIntervallProjects>
    <ServerListenPort>0</ServerListenPort>
    <ServerAddress>unused</ServerAddress>
  </General>
  <Project1>
    <Projectname>Test</Projectname>
    <Rootpath>/home/stefan/JASC/rootfolder/</Rootpath>
    <Username>Stefan</Username>
    <Userpassword>will be inserted here</Userpassword>
    <!-- Folder filters: add the complete name of folders to be filtered out -->
    <!-- This filter is just an example, change the settings to your needs! -->
    <FilterFolder1>Temp</FilterFolder1>
    <!-- File filters: add a string, to be searched for -->
    <!-- This filter is just an example, change the settings to your needs! -->
    <FilterFile1>.tmp</FilterFile1>
    <LinuxUsername>stefan</LinuxUsername>
  </Project1>
</JustASimpleCloudClient_Config>
```

3.5.1. Abschnitt „General“

Der Abschnitt „General“ enthält die Grund-Einstellungen zur Kommunikation mit dem Server:

- **ClientName:** Name des Clients, wird auf dem Server für Log-Ausgaben verwendet und unterliegt der maximalen Anzahl an Clients, die ein Server basierend auf seiner Lizenz unterstützt
- **GlobalKey:** Passwort für den Aufbau einer verschlüsselten Verbindung zum Server (siehe dort)
- **ActivateRelayService:** entscheidet, ob eine Verbindung zum Server über den Relais-Service oder eine direkte Verbindung (basierend auf „**ServerAddress**“ und „**ServerListenPort**“) aufgebaut werden soll
- **RelayServiceIdentifizier:** wenn der Relais-Service verwendet werden soll, ist hier die Maschinen-Kennung des Servers einzutragen
- **ShowBallonMessages:** (de-) aktiviert die Ausgabe von wichtigen Meldungen über Einblendungen in der Nähe des „System-Trays“ von Windows
- **SetClientFolderIcon:** wenn diese Einstellung aktiviert ist, wird im Windows-Explorer der „Rootpath“ eines jeden Projektes mit dem „Just a simple cloud“ Icon hervorgehoben
- **DiscSpaceWarningLevel:** wenn dieser Prozentsatz an freiem Speicher auf der SD-Karte des RasPi unterschritten wird, wird eine Warnung ausgegeben
- **DiscSpaceErrorLevel:** wenn dieser Prozentsatz an freiem Speicher auf der SD-Karte des RasPi unterschritten wird, wird ein Fehler ausgegeben
- **LogLevel:** gibt die Wichtigkeit der Log-Ausgaben an, die ausgegeben werden
- **ScanIntervallProjects:** gibt das Intervall in Minuten an, nach dem jeweils das Daten-Verzeichnis („Rootpath“) auf Änderungen überprüft wird und ggf. eine Synchronisation mit dem Server gestartet wird
- **ServerListenPort:** Port des Servers zur direkten Verbindung ohne den Relais-Service
- **ServerAddress:** Adresse des Servers zur direkten Verbindung ohne den Relais-Service

3.5.2. Abschnitt „Projects“

Auch beim Client können mehrere Projekte parallel konfiguriert werden:

- **Projectname:** eindeutiger Name des Projektes, über diesen Namen werden die Projekte auf dem Server identifiziert
- **Rootpath:** Pfad zum Stamm-Verzeichnis, das alle Dateien des Projektes enthält
- **Username:** Benutzername, der zur Anmeldung am Server verwendet wird
- **Userpassword:** Passwort des Nutzers (auf dem Client werden Passwörter AES verschlüsselt abgelegt)
- **FilterFolder:** Eine Liste von Verzeichnissen, die bei der Synchronisation mit dem Server ignoriert werden
- **FilterFile:** Eine Liste von Dateinamen bzw. Teilen von Dateinamen, die bei der Synchronisation mit dem Server ignoriert werden

3.6. Zugriff auf Linux / Updates

Die Partition der SD-Karte, die das Linux Root-System enthält, ist schreibgeschützt.

Um den Schreibschutz temporär aufzuheben, geben Sie die folgenden Kommandos auf der Kommando-Zeile ein:

- `sudo mount -o remount,rw /dev/root /`
- `sudo systemctl daemon-reload`

Die Root-Partition ist damit bis zum nächsten Booten schreibbar.

Um Linux-Updates einspielen zu können, müssen Sie temporär die Größe des „tmp“-Verzeichnisses vergrößern.

Geben Sie dazu das folgende Kommando ein:

- `sudo nano /etc/fstab`

Ändern Sie dann in den beiden folgenden Zeilen

```
tmpfs /tmp tmpfs defaults,noatime,nosuid,nodev,noexec,mode=0755,size=100M 0 0
tmpfs /var/tmp tmpfs defaults,noatime,nosuid,nodev,noexec,mode=0755,size=100M 0>
```

die Größe von 100MB auf 300MB.

4. Android App

Dieses Kapitel beschreibt die Hintergründe der App für Android.

4.1. Batterie-Optimierung

Das Betriebssystem Android verwaltet die Lebenszyklen aller Apps im Hintergrund. Als Lebenszyklus werden dabei die unterschiedlichen Zustände bezeichnet, die eine App einnehmen kann: sie kann z.B. zunächst vollständig beendet sein, wird dann gestartet und kommt in den Vordergrund. Danach wird sie (z.B. durch einen eingehenden Telefon-Anruf) in den Hintergrund verschoben, wo sie so lange bleibt, bis sie wieder vollständig beendet wird.

Falls die Ressourcen von Android (z.B. der Arbeitsspeicher) knapp werden, entscheidet das Betriebssystem Android, welche Apps beendet werden. Eine App hat darauf zunächst keinen Einfluss. Auch die App würde daher bei Bedarf durch das Betriebssystem vollständig beendet werden. Damit die Synchronisation im Hintergrund weiterlaufen kann, muss die App auf die sogenannten „Whitelist“ gesetzt werden. Apps auf dieser Liste dürfen auch bei begrenzten Ressourcen weiterlaufen. Dazu wird die Berechtigung „**REQUEST_IGNORE_BATTERY_OPTIMIZATIONS**“ benötigt, die beim ersten Start der App automatisch beantragt wird. Die App versucht dann selbstständig, sich auf die Whitelist einzutragen. Da jedoch jeder Hersteller von Smartphones das Betriebssystem anpasst, gibt es dafür kein absolut verlässliches Verfahren. Prüfen Sie daher nach der Installation, ob die App wirklich auf der Whitelist eingetragen ist.

Rufen Sie die Einstellungen von Android auf und navigieren Sie dann zu der „Akku-Optimierung“. Dort muss „Just a simple cloud“ als „nicht optimiert“ eingetragen sein!

4.2. Berechtigungen

Jede Android-App beantragt beim Betriebssystem unterschiedliche Berechtigungen, um auf kritische Funktionen zugreifen zu können. Grundsätzlich sollten diese Berechtigungen so restriktiv wie möglich gehalten werden. So darf beispielsweise die App nicht auf Ihre Kontakte zugreifen. Die folgenden Berechtigungen werden für eine korrekte Funktion der App benötigt:

- **CAMERA**: Zum Scannen eines QR-Codes
- **INTERNET**: Zugriff über das Internet auf den RasPi-Server
- **READ_EXTERNAL_STORAGE**: Zugriff auf Speicher, um von dort das Projektverzeichnis zu lesen
- **RECEIVE_BOOT_COMPLETED**: automatischer Start der App im Hintergrund, wenn das Handy neu gestartet wurde
- **REQUEST_IGNORE_BATTERY_OPTIMIZATIONS**: siehe oben
- **WRITE_EXTERNAL_STORAGE**: Zugriff auf Speicher, um das Projektverzeichnis dort abzulegen

4.3. Teilen von Fotos / Videos

In der Standard-Einstellung der App werden automatisch alle Fotos und Videos, die auf dem Smartphone gespeichert sind, importiert und synchronisiert.

Die Fotos werden in folgendem Pfad innerhalb der Projekt-Dateien abgelegt:

Smartphone/<Nutzername>/<Geräte-Name>

Sie können das automatische Importieren über die **Allgemeinen Einstellungen** aktivieren:



Um Fotos wieder zu entfernen, löschen Sie diese im **DCIM** Verzeichnis Ihres Smartphones.

Wichtig: abhängig von Ihrer Lizenz werden Fotos und Videos wie folgt synchronisiert:

- Persönliche Lizenz: **nur Fotos**
- Professionelle Lizenz: **Fotos und Videos**

4.4. Bereitstellen von Fotos / Videos

Wenn an Ihrer Cloud mehrere Smartphones teilnehmen und alle ihre Fotos teilen, werden diese automatisch auf Ihr Smartphone heruntergeladen.

Die Fotos werden in folgendem Pfad innerhalb der Projekt-Dateien abgelegt:

Download/JASC

Sie können diese Funktion in den **Allgemeinen Einstellungen** aktivieren:



Diese Funktion ist **nur auf Android Smartphones mit Android 11** oder höher verfügbar.

4.5. Android Version 7 bis 10

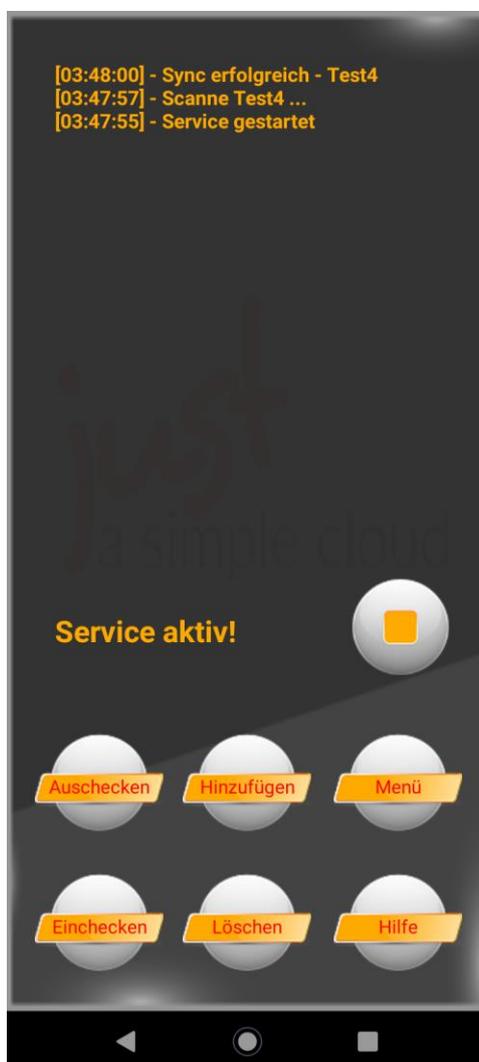
Bis zur Android Version 10 lässt Android den Zugriff auf das Dateisystem für alle Apps zu. Die App legt daher automatisch den folgenden Pfad an und speichert dort alle Dateien der Projekte:

Internal storage/JustASimpleCloud

Auf dem Haupt-Bildschirm der App finden sich daher keine weiteren Funktionen zum Zugriff auf die Dateien. Verwenden Sie stattdessen einen Datei-Explorer.

4.6. Android 11 und später

Ab Android Version 11 ist der Zugriff auf Dateien erheblich eingeschränkt worden. Es ist nicht mehr möglich, ein zentrales Verzeichnis anzulegen, auf das alle Apps Zugriff haben. Die App speichert daher alle Dateien in einem internen Verzeichnis. Um mit den Dateien arbeiten zu können, bietet die App zusätzliche Funktionen.



Auschecken: blendet eine Verzeichnis-Ansicht aller Dateien ein. Durch Klicken auf eine Datei, wird eine Kopie in folgendem Verzeichnis angelegt:

Interner Speicher/Dokumente/JASC/<Projektname>

Sie können die Datei in diesem Verzeichnis mit einer beliebigen App bearbeiten. Ausgecheckte Dateien werden in der Ansicht farblich hervorgehoben.

Wird eine Datei durch eine **Synchronisation** mit dem Server aktualisiert, wird **nur die Kopie im internen Speicher der App aktualisiert**, die bearbeitete Kopie bleibt erhalten.

Einchecken: blendet dieselbe Ansicht wie beim Auschecken ein. Durch Klicken auf eine Datei wird die zuvor ausgecheckte Datei wieder zurück in den internen Speicher kopiert und damit automatisch synchronisiert.

Hinzufügen: hiermit können Sie eine neue Datei dem internen Speicher hinzufügen. Zunächst wird ein Auswahl-Dialog angezeigt, mit dem die Datei ausgewählt werden kann, danach kann das Verzeichnis im internen Speicher ausgewählt werden, in das die Datei eingefügt werden soll.

Löschen: dient dem Löschen von Dateien im internen Speicher. Eine Datei kann in der Verzeichnis-Ansicht ausgewählt werden, um dann gelöscht zu werden.

4.7. Weitere Einstellungen

Bei manchen Smartphone-Herstellern ist es nötig, den automatischen Start der App beim Start des Smartphones manuell zu aktivieren. Dazu bitte die Einstellung "**Einstellungen -> Akku -> Autostart-Management -> App-Autostart**" verwenden."

5. Rechte-Verwaltung

5.1. Voraussetzungen

Die Rechteverwaltung benötigt folgende Voraussetzungen:

- In der Server-Konfiguration muss der Parameter „**ActivateUserPrivileges**“ aktiviert sein
- In der Server-Konfiguration muss ein sinnvoller Wert für „**ScanIntervallUserrights**“ eingetragen sein. Der Server scannt dann periodisch das gesamte Daten-Verzeichnis, dass in den Projekten unter „**Rootpath**“ eingetragen ist.
- Im Abschnitt „**Group**“ der Projekte muss mindestens ein Name für eine Nutzer-Gruppe eingetragen sein
- Jeder User, der einem Projekt zugeordnet ist, muss mindestens einen Eintrag für „**Group**“ haben. Dieser Eintrag muss dabei einem Gruppennamen des Projektes entsprechen.

Achtung; enthält ein Projekt **keine Angaben** zu „**Group**“ bzw. ist ein **User keiner Gruppe** zugeordnet, bedeutet dass, das er **weder Lese- noch Schreibrechte für das gesamte Daten-Verzeichnis** hat!

Beispiel:

```
<JustASimpleCloudServer_Config>
  <General>
    .... snipp ...
    <ActivateUserPrivileges>true</ActivateUserPrivileges>
    <ScanIntervallUserrights>30</ScanIntervallUserrights>
    .... snipp ...
  </General>
  <Project1>
    <Projectname>Testproject</Projectname>
    <Rootpath>ServerRoot/FirstProject</Rootpath>
    <Group1>
      <Groupname>Standard</Groupname>
    </Group1>
    <Group2>
      <Groupname>Admin</Groupname>
    </Group2>
    <Group3>
      <Groupname>Sale</Groupname>
    </Group3>
    <User1>
      <Username>John</Username>
      <Realname>John Doe</Realname>
      <Password>37c533c8f4ce84808c19b94296e3cf06</Password>
      <Useremail>john@doe.com</Useremail>
      <Group1>Standard</Group1>
      <Group2>Sale</Group2>
    </User1>
  </Project1>
</JustASimpleCloudServer_Config>
```

Für das “Testproject” wurden drei Gruppen definiert:

- Standard
- Admin
- Sale

Der Nutzer „John“ ist Mitglied in den Gruppen

- Standard
- Sale

jedoch nicht in der Gruppe „Admin“. D.h. Verzeichnisse, die der Admin-Gruppe vorbehalten sind, sind für John nicht sichtbar.

5.2. Zugriffssteuerung

Die Zugriffsrechte werden auf Verzeichnis-Ebene vergeben. Für jedes Verzeichnis kann damit für jede in der Server-Konfiguration erstellten Gruppe definiert werden, ob sie Lese- bzw. Schreibrechte auf das Verzeichnis hat.

Die Rechtesteuerung wird dabei ebenfalls in XML-Dateien abgelegt, folgenden Aufbau haben:

~Verzeichnisname.xml

Beispiel:

Ansicht des **Rootpath** eines Projektes:

Name	Typ	Datum
 Folder1	Dateiordner	25.02.2021 11:12
 Folder2	Dateiordner	25.02.2021 11:12
 ~Folder1.xml	XML-Dokument	02.01.2021 18:07
 ~Folder2.xml	XML-Dokument	02.01.2021 18:07
 ~RootRights.xml	XML-Dokument	02.01.2021 18:07
 testfile.txt	Textdokument	20.05.2020 19:52

Zu den beiden Verzeichnissen „**Folder1**“ und „**Folder2**“ gibt es jeweils die XML-Dateien „~**Folder1.xml**“ und „~**Folder2.xml**“.

Die **Zugriffsrechte für den Rootpath** eines Projektes sind dabei immer in der Datei „~**RootRights.xml**“ abgelegt. Diese Datei muss immer existieren.

Falls für ein Verzeichnis **noch keine XML-Datei** existiert, wird sie beim nächsten periodischen Scan der Zugriffsrechte **automatisch angelegt**. Die Zugriffsrechte werden dabei aus den Rechten des übergeordneten Verzeichnisses übernommen.

5.3. Aufbau der XML-Dateien

Beispiel:

```
<?xml version="1.0" ?>
<!-- this file contains the user rights for the folder -->
<UserRights>
  <Group1>
    <Groupname>Standard</Groupname>
    <Visible>true</Visible>
    <Write>>false</Write>
  </Group1>
  <Group2>
    <Groupname>Sale</Groupname>
    <Visible>>false</Visible>
    <Write>>false</Write>
  </Group2>
  <Group3>
    <Groupname>Admin</Groupname>
    <Visible>true</Visible>
    <Write>true</Write>
  </Group3>
</UserRights>
```

Mitglieder der Gruppe „**Standard**“ haben Lese-Rechte für das Verzeichnis (d.h. sie können die Dateien und Unterverzeichnisse sehen), aber keine Schreib-Rechte.

Mitglieder der Gruppe „**Sale**“ haben keine Rechte für das Verzeichnis, d.h. sie können auch den Inhalt des Verzeichnisses nicht sehen.

Mitglieder der Gruppe „**Admin**“ haben volle Rechte für das Verzeichnis.

Ist eine **Gruppe in der XML-Datei nicht aufgeführt**, bedeutet dies, dass Mitglieder dieser Gruppe keine Rechte für das Verzeichnis haben (es sei denn, sie erhalten Rechte durch die Zugehörigkeit einer anderen Gruppe, die Teil der XML-Datei ist).

Die Gruppennamen müssen in der XML-Datei nicht in derselben Reihenfolge wie in der Server-Konfiguration angelegt sein.

5.4. Home-Verzeichnis

Wenn in der Projekt-Konfiguration des Servers (siehe Kap. **Abschnitt „Projects“**) die Eigenschaft „**EnableHomeFolder**“ auf „true“ gesetzt wurde, wird **automatisch** für jeden neu angelegten Nutzer ein **Unter-Verzeichnis in „home“-Verzeichnis angelegt**, dessen Name dem Nutzernamen entspricht. Dies gilt für:

- RasPi Server Ersteinrichtung
- Nutzer auf RasPi Server verwalten

Der Server stellt dabei sicher, dass **nur der jeweilige Nutzer Lese- und Schreibrechte** in diesem Unter-Verzeichnis hat und **kein anderer Nutzer**. Die Regelungen der Zugriffssteuerung haben hier keine Gültigkeit.

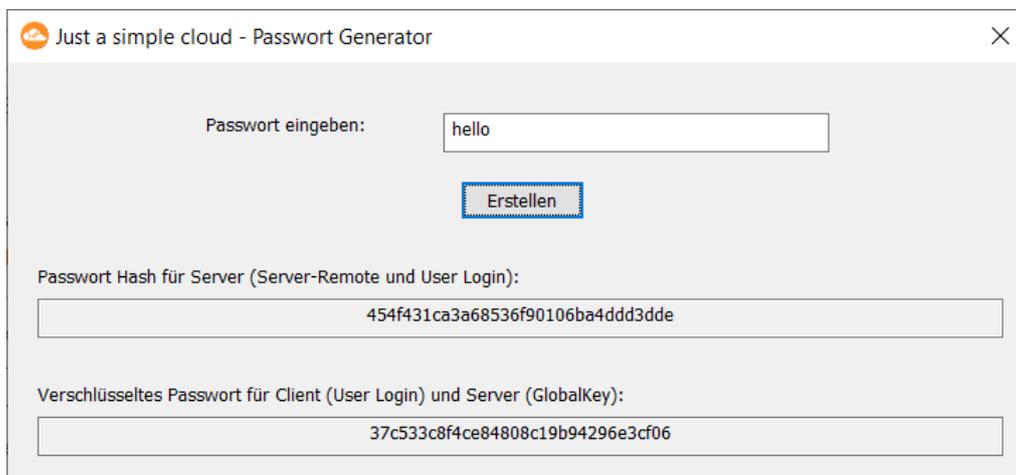
5.5. Administratoren

Administratoren (siehe „Bedienungsanleitung JASC-RasPi“, Kapitel „Nutzerverwaltung auf dem RasPi-Server“) sind von der **Rechte-Verwaltung ausgenommen**.

Das bedeutet, sie haben **in allen Verzeichnissen volle Lese- und Schreibrechte**, unabhängig von den Regelungen der Zugriffssteuerung oder den Home-Verzeichnissen.

6. Tools

6.1. Passwort Generator (Windows)



Just a simple cloud - Passwort Generator

Passwort eingeben:

Passwort Hash für Server (Server-Remote und User Login):

Verschlüsseltes Passwort für Client (User Login) und Server (GlobalKey):

Die Passwörter der Nutzer sind aus Sicherheitsgründen in den XML-Konfigurationsdateien nicht im Klartext gespeichert. In der Konfigurationsdatei für den Client wird das Passwort AES verschlüsselt gespeichert, in der Datei des Servers wird ein „gesalzenen“ Hash des Passworts abgelegt. Wenn Sie neue Passwörter setzen wollen (weil z.B. der Nutzer sein Passwort vergessen hat), unterstützt Sie dieses Tool bei der Erzeugung der verschlüsselten Einträge bzw. des Hash. Tragen Sie einfach das neue Passwort im Klartext ein und kopieren Sie dann die generierten Werte direkt in die entsprechende Konfigurationsdatei.

7. Anmerkungen

Dritt-Software

„VNC Viewer“ und „balenaEtcher“ sind Produkte von Dritt-Anbietern und Markenrechte des jeweiligen Anbieters.

7.1. Passwort Generator (Linux)

Starten Sie das JASC-Setup mit folgendem Kommando:

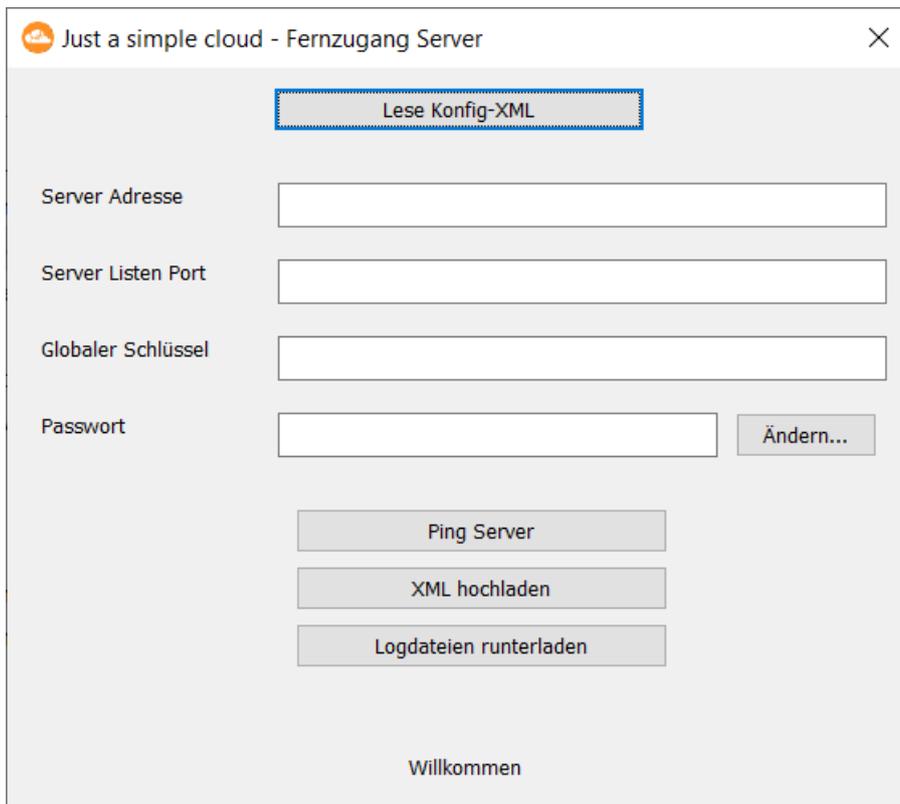
```
jascsetup password '<Passwort>'
```

Verwenden Sie **immer einfache Anführungsstriche** für das Passwort!

Die Passwörter der Nutzer sind aus Sicherheitsgründen in den XML-Konfigurationsdateien nicht im Klartext gespeichert. In der Konfigurationsdatei für den Client wird das Passwort AES verschlüsselt gespeichert, in der Datei des Servers wird ein „gesalzener“ Hash des Passworts abgelegt.

Wenn Sie neue Passwörter setzen wollen (weil z.B. der Nutzer sein Passwort vergessen hat), unterstützt Sie dieses Tool bei der Erzeugung der verschlüsselten Einträge bzw. des Hash. Tragen Sie einfach das neue Passwort im Klartext ein und kopieren Sie dann die generierten Werte direkt in die entsprechende Konfigurationsdatei.

7.2. Server Remote (nur Windows)



Als Hilfe zum Ausfüllen der Server-Parameter können Sie den Button „**Lese Konfig-XML**“ anklicken und die XML-Datei, die beim Anlegen ihres Nutzers erstellt wurde, verwenden.

Das Passwort für den Fernzugang zum Server können Sie mit dem Button „**Ändern...**“ neu setzen.

Die Kommunikation mit dem Server lässt sich mit der Funktion „**Ping**“ testen.

Mit diesem Tool können Sie außerdem die Konfiguration eines entfernten Servers aktualisieren. Dazu erstellen Sie lokal eine neue Version der Konfiguration und laden die Datei „JustASimpleCloudServer.xml“ direkt auf den Server hoch. Die folgenden Einstellungen aus dem Abschnitt „**General**“ der Konfiguration können aktualisiert werden:

- Email
- ActivateUserPrivileges
- ActivateDNSService
- ActivateRelayService
- LogLevel
- LogUserActivity
- ScanIntervallUserrights

Um die Dateien des Rechte-Managements zu aktualisieren, laden Sie eine XML-Datei mit den Änderungen hoch. Sie müssen dabei nicht alle Verzeichnisse einzeln hochladen, stattdessen werden die Verzeichnisse, die in der XML-Datei gelistet sind, automatisch aktualisiert.

Beispiel:

```
<UserRights>
  <Projectname>Testproject</Projectname>
  <Folder1>
    <Foldername>Rootfolder</Foldername>
    <Group1>
      <Groupname>Standard</Groupname>
      <Visible>true</Visible>
      <Write>true</Write>
    </Group1>
    <Group2>
      <Groupname>Sale</Groupname>
      <Visible>true</Visible>
      <Write>false</Write>
    </Group2>
  </Folder1>
  <Folder2>
    <Foldername>Folder1\Subfolder</Foldername>
    <Group1>
      <Groupname>Standard</Groupname>
      <Visible>false</Visible>
      <Write>false</Write>
    </Group1>
  </Folder2>
</UserRights>
```

Bitte warten Sie nach einer Aktualisierung der Einstellungen mindestens einen weiteren Scan des Servers ab, damit die neuen Einstellungen wirksam werden.

Um Ihre Einstellungen zu überprüfen, können Sie die **Konfigurationsdatei und alle Logfiles des Servers herunterladen**.