**Ziel:** bei Programmstart (welches Programm?) wird auf neue, vorhandene Updates geprüft. Existieren diese, werden sie automatisch installiert

**Ähnliches Beispiel:** auf f-vm-ps-cobacti unter C:\SER\vm-shutdown (wird eine VM 14 Tage nicht genutzt, wird sie automatisch heruntergefahren)

**Programmiersprache:** Java,

* es wird daher aber eine Datei benötigt, die zwischen dem Java-Programm und der VM vermitteln kann

**Zu programmieren:** jar-Datei

**Beginn:**

1. Request Client: Ziel ist es, auf einem Client eine Nachricht schreiben zu können, die dann vom Server angezeigt wird. (siehe Eclipse!)
2. Hash-Werte einer beliebigen Datei vergleichen:

* Client generiert Hash-Wert seiner Datei
* Client schickt Hash-Wert an den Server
* Server generiert Hash-Wert seiner Datei
* Server vergleicht die beiden Hash-Werte gegeneinander
* Sind sie ungleich -> schickt Hash-Wert
* Sind sie gleich -> schickt nicht

1. Datei des Servers (als roher Bytestream, keine Textdatei!) auf den Client übertragen und dort speichern.

-> Als Tipp schau dir Files.readAllBytes an und anschließend sendest du diese

-> Oder: sowas wie new FileInputStream und benutzt dort die read methode und schreibst anschließend die gelesenen Bytes direkt in den Connection stream

Daraus resultierendes mögliches Update und Löschung der alten Version

**Benötigt:**

* ConfigServer und ConfigClient benötigt
* IP-Adresse des Servers & Applikations-Port

**Infos:**

**Client-Server-Modell:** Der Client kann auf Wunsch einen Dienst vom Server anfordern (z. B. ein [Betriebsmittel](https://de.wikipedia.org/wiki/Betriebsmittel_(Informatik))). Der Server, der sich auf demselben oder einem anderen Rechner im Netzwerk befindet, beantwortet die Anforderung (das heißt, er stellt im Beispiel das Betriebsmittel bereit).

-> [Client-Server-Modell – Wikipedia](https://de.wikipedia.org/wiki/Client-Server-Modell#:~:text=Das%20Client-Server-Modell%20%28auch%20Client-Server-Konzept%2C%20-Architektur%2C%20-System,Server%20gleichzeitig%20f%C3%BCr%20mehrere%20Clients%20arbeiten.&text=Das%20Client-Server-Modell%20%28auch%20Client-Server-Konzept%2C,f%C3%BCr%20mehrere%20Clients%20arbeiten.&text=%28auch%20Client-Server-Konzept%2C%20-Architektur%2C%20-System,Server%20gleichzeitig%20f%C3%BCr%20mehrere)

**Socket:** is an endpoint of communication between two computers using TCP. An endpoint is a combination of an IP adress and a port number. Sockets are used to send and receive data.

**TCP (Transmission control protocol):** is a connection oriented protocol meaning two computers form a connection before sending date. Every TCP connection can be identified by it`s two endpoints.

## **Socket Programming:** One socket (node) listens on a particular port at an IP, while other socket reaches out to the other in order to form a connection. The server forms the listener socket while the client reaches out to the server. Socket and Server Socket classes are used for connection-oriented socket programming.

## -> <https://www.edureka.co/blog/socket-programming-in-java/>

Ein Bild, das Text, Elektronik, Screenshot enthält.

Automatisch generierte Beschreibung

**Client Programm:** creates Socket on its end, attempt to connect socket to socket of server. Client and server communicate by writing to an output stream and reading from an input stream on the socket.

**Streams (Input/Output):** sequence of data. There are two types, a charater stream (read/write, filled with text files) und a byte stream (used für images or binary files)

**OutputStreamWriter & InputStreamReader:** bridges from byte streamst o character streams. The underlying byte streams are the OutputStream and InputStream (bridge from byte to character stream) from the socket respectively.

**Hashfunktion:** reduzieren Zeichen beliebiger Länge (z.B. unterschiedliche Passwörter) auf Zeichen fester Länge (hier immer 3 Zeichen), sie werden also in eine kleine, kompakte Form gebracht.

-> <https://www.dr-datenschutz.de/hashwerte-und-hashfunktionen-einfach-erklaert>

**Hashwert:**  stellt das Ergebnis dar, welcher mittels einer Hashfunktion berechnet wurde. Man definiert eine feste Länge, wie lang ein Hashwert immer sein darf. Oft wird der Hashwert als eine hexadezimale Zeichenkette codiert d.h. der Hashwert besteht aus einer Zahlen und Buchstaben-Kombination zwischen 0 - 9 sowie A - F