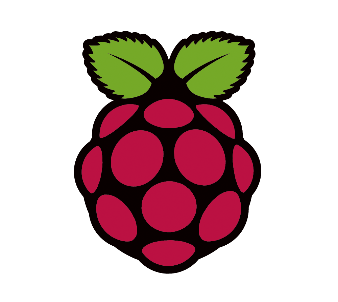
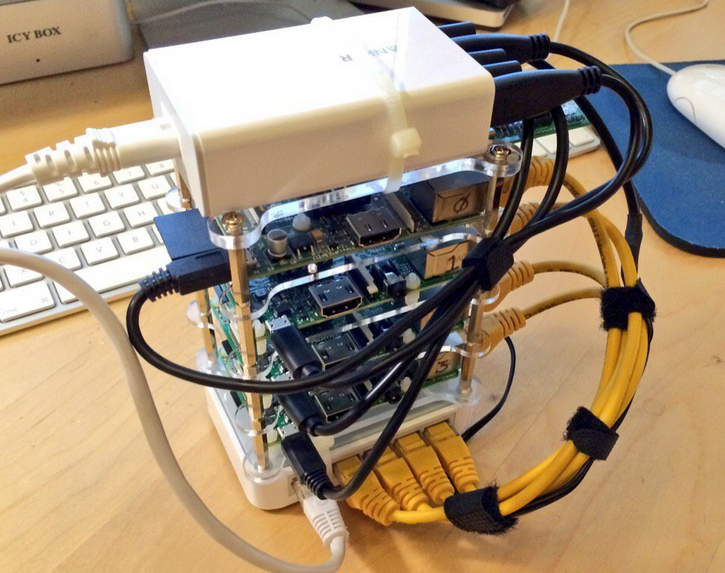
Lino. Thalmann

Lino.thalmann@gmail.com

Raspberry Pi Cluster

Ein Node JS Server mit 4 Raspberry PIs aufbauen





Inhalt

[1 Systemidee 1](#_Toc469315789)

[1.1 Ziel Mein Ziel ist es ein Raspberry Pi Cluster zu bauen, welches ich dann zum Node JS Server konfiguriere. Auf diesem Webserver sollte dann die Webapplikation von Jonas und Donato laufen. Wenn das geklappt hat und ich noch Zeit habe, möchte ich noch ein kleines Sub-Projekt machen. Es wäre cool wenn ein LCD Bildschirm der Status des Webservers und weitere Infos anzeigen würde. Doch dazu erst mehr, wenn mir noch Zeit bleibt. 1](#_Toc469315790)

[1.2 Management Summary 1](#_Toc469315791)

[1.3 Anforderungen 2](#_Toc469315792)

[1.4 Planung 2](#_Toc469315793)

[2.1 Vorbereitungsphase: 3](#_Toc469315794)

[2.1.1 Bauteile: 3](#_Toc469315795)

[2.1.2 Anleitungen 3](#_Toc469315796)

[Konzept Version 1 3](#_Toc469315797)

[Konzept Version 2 4](#_Toc469315798)

# 1 Systemidee

## 1.1 Ziel Mein Ziel ist es ein Raspberry Pi Cluster zu bauen, welches ich dann zum Node JS Server konfiguriere. Auf diesem Webserver sollte dann die Webapplikation von Jonas und Donato laufen. Wenn das geklappt hat und ich noch Zeit habe, möchte ich noch ein kleines Sub-Projekt machen. Es wäre cool wenn ein LCD Bildschirm der Status des Webservers und weitere Infos anzeigen würde. Doch dazu erst mehr, wenn mir noch Zeit bleibt.

## 1.2 Management Summary

Das Ziel ist es mehrere Einplatinen PCs (das sind minimale Computer, die nur aus einem Motherboard bestehen) zusammenschliessen, um ihre Leistungen zu addieren.

Nun kann man diesen Verbund als Webserver konfigurieren und darauf Websites hosten. Das spart die Kosten von externen Webhoster, die Hardware ist günstig und man hat einen eigenen Webserver direkt in Reichweite. Dieser kann selber gewartet, erweitert und verbessert werden, was ein grosser Vorteil darstellt.   
Da es sich um ein Cluster aus 4 Einpaltinen PCs handelt, fehlt es auch keineswegs an Leistung.

Bei Fragen melden Sie sich bei lino.thalmann@gmail.com

## 1.3 Anforderungen

Als Entwickler möchte ich eine Cluster aus 4 Raspberry Pi haben(REQ001). Dieses soll an ein System angeschlossen sein(REQ002).

Auf diesem Cluster soll ein Node JS Server sein, auf dem der Entwickler seine Webapplikationen laufen lassen kann(REQ003).

Ausserdem muss der Entwickler Zugriff auf eine Kommandozeile haben, um den Server zu starten(REQ004),

Wenn eine Site auf dem Server läuft, muss man über eine IP auf diese zugreifen können(REQ005)

Und die Scripts auf dem Server stabil laufen (REQ006).

Die Daten müssen per USB auf den Webserver übermittelt werden (REQ007).

Der Status des Servers sollte auf einem externen LCD Bildschirmchen ablesbar sein, um die Benutzerfeundlichkeit verbessern(NTH001).

Es wäre gut, wenn die Daten alternativ auch per FTP auf den Server transferiert werden können (NTH002).

Eine Webseite braucht eine Datenbank. Diese könnte man leicht extern hosten. Nice to have wäre, wenn man diese auch auf diesem Server laufen lassen könnte(NTH003).

## 1.4 Planung

Vorbereitungsphase (1 Tag)

Schritt 1 → Wie ist ein Cluster hardwaretechnisch aufgebaut? Hardware besorgen

Schritt 2 → Wie ist ein Cluster softwaretechnisch aufgebaut? Software und Konfigurationsanleitung besorgen

Aufbauhase (5 Tage)

Schritt 1 → den einzelnen Pis IP zuordnen

Schritt 1 → Cluster aufbauen

Schritt 2 → einzelne Pis zum Cluster konfigurieren

Schritt 3 → Cluster zum Node JS einstellen

Schritt 4 → Testen REQ001 bis REQ004

Webphase (2 Tage)

Schritt 1 → Projekt auf Server laden

Schritt 2 → Veröffentlichung

Schritt 3 → Testen REQ5 bis REQ007

Sub-Projekt oder Puffer (3 Tage)

Schritt 1 → Planung

Schritt 2 → Umsetzung

Schritt 3 → Testen NTH001

Schritt 3 → Umsetzen und testen NTH002 und NTH003

Meilensteine: erfolgreicher Abschluss der verschiedenen Phasen

Genauer Zeitplan: Excel Dokument Zeitplanung

## 2.1 Vorbereitungsphase:

### 2.1.1 Bauteile:

* 4 Raspberry Pi Modell 3
* 4 (mind. 8GB) SD Speicherkarten mit Raspian
* Evtl. Gehäuse oder Stapler
* 5 Port Switch
* Stromversorgung für PIs
* 4 LAN Netzwerkkabel (möglichst kurz)

### 2.1.2 Anleitungen

Da der Raspberry Pi eine grosse Community hat, ist es einfach Anleitungen zu finden. Eine detaillierte Schritt für Schritt Anleitung für ein Cluster werde ich verwenden, sowie eine für einen Node JS Server auf einem PI. Diese Beiden Anleitungen werde ich verbinden, sodass ich daraus alles entnehmen kann was ich brauche.

Clustertutorial: http://makezine.com/projects/build-a-compact-4-node-raspberry-pi-cluster/

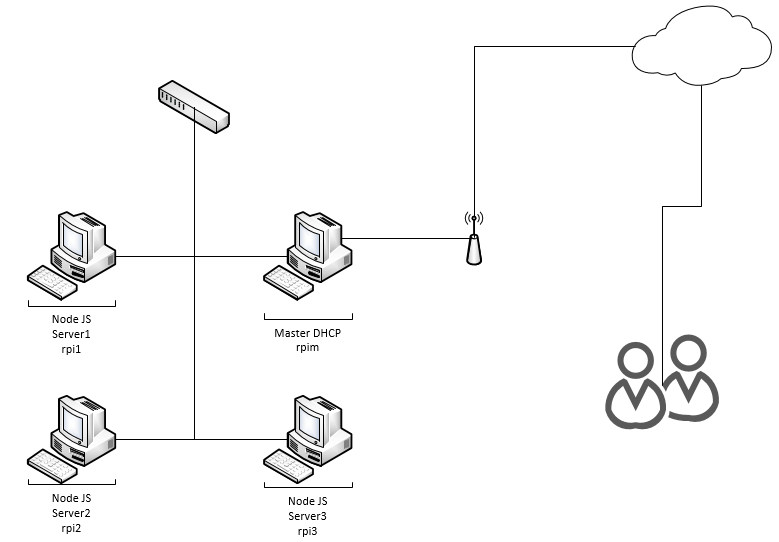
Node JS auf Raspberry Pi: <http://thisdavej.com/beginners-guide-to-installing-node-js-on-a-raspberry-pi/>

Youtube: <https://youtu.be/eZ5uX-JJbyY>

# Konzept Version 1

Diese Version besteht aus einem DHCP und Master, der per WLAN (externe IP) mit dem Internet verbunden ist. Auf den restlichen drei Raspberrys läuft je ein Node JS Server. Der Zugriff erfolgt über den DHCP und die Aufgaben werden von den einzelnen Pis übernommen und aufgeteilt, um optimale Effizienz zu gewährleisten.   
Eine nötige Datenbank wird extern gehostet.  
Dieses Konzept ist nicht optimal, da die Nachteile den einzigen Vorteil (Effizienz) überwiegen:

1. Die Datenbank muss extern gehostet werden.  
2. Die einzelnen Node JS Server richtig zu konfigurieren, dass diese auch alles richtig aufteilen, ist äusserst schwierig und vor allem zeitaufwändig. Zeit, die wir nicht haben.   
https://nodejs.org/api/cluster.html  
3. Es ist in unserem Rahmen nicht nötig die Leistung von 3 Raspberrys für Node JS zu haben.



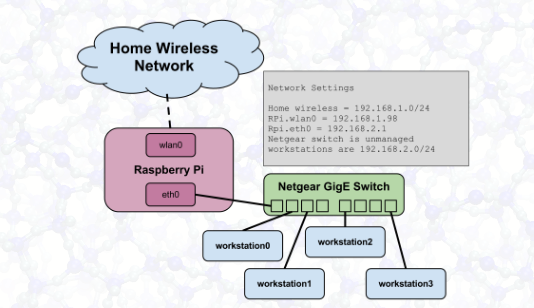
# 

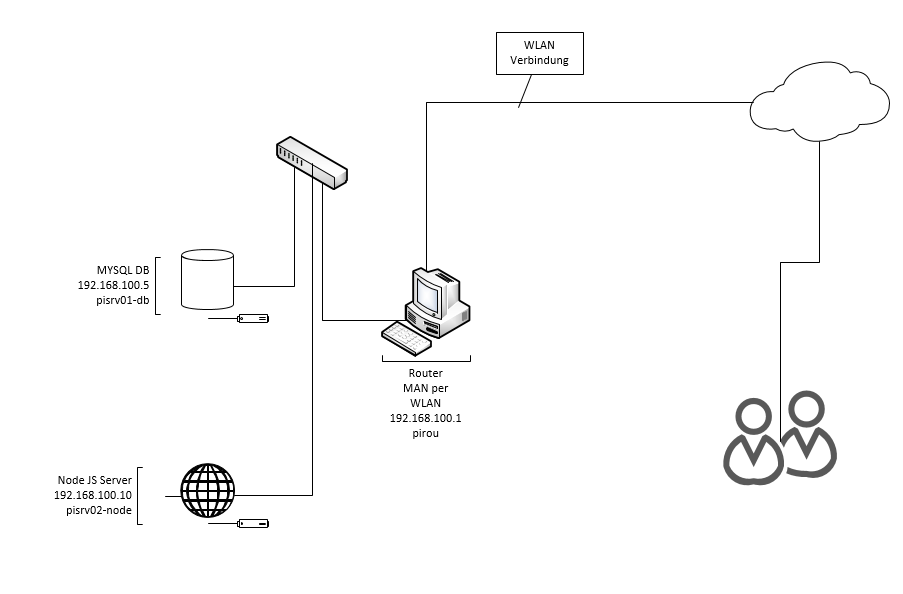
# Konzept Version 2

Die zweite Version ist kein Cluster. Es ist eine DB und ein Node JS Server einem Subnetz vereint, alles was die Webapplikation braucht.  
Der DHCP koordiniert alle die IP Zuweisungen, der Node JS Server verarbeitet die Scripts und der MySQL Server beherbergt die Datenbank. Die Kommunikation von Node JS Server und DB wird mit node-mysql gemacht (https://youtu.be/XuLRKMqozwA). Bei genügend Zeit kann der vierte Pi als Schnittstelle fungieren (mit einer Statusanzeige z.B. LED oder Bildschirm).

Vorteile:

1. Der ganze Node-JS-Clustering-Prozess muss nicht ausgeführt werden
2. Es sind dennoch dedizierte Server, die je genug Leistung haben.





(Keine akkurater Beschrieb)