



The Tooling Deep Dive

a **Hack & Söhne** Bootcamp

Learn how to use Git, Maven, Docker, IntelliJ and more!



#hackundsoehne

www.hackundsoehne.de



Who are we?



#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de



The Goal



#hackundsoehne

www.hackundsoehne.de



A question for you



#hackundsoehne

www.hackundsoehne.de



What the fuck?



#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de



Hack
&
Schne

Cheat Sheet



#hackundsoehne

www.hackundsoehne.de



Topics



#hackundsoehne

www.hackundsoehne.de





IDEs & Editors

IntelliJ, Atom, Sublime, Nano,
Vim



Shell

Explain the basics of the shell

Support

What to do when you are stuck
Gitter, Slack, etc.



#hackundsoehne

Package Managers

APT, Homebrew



Markdown

What is markdown and
how to use it

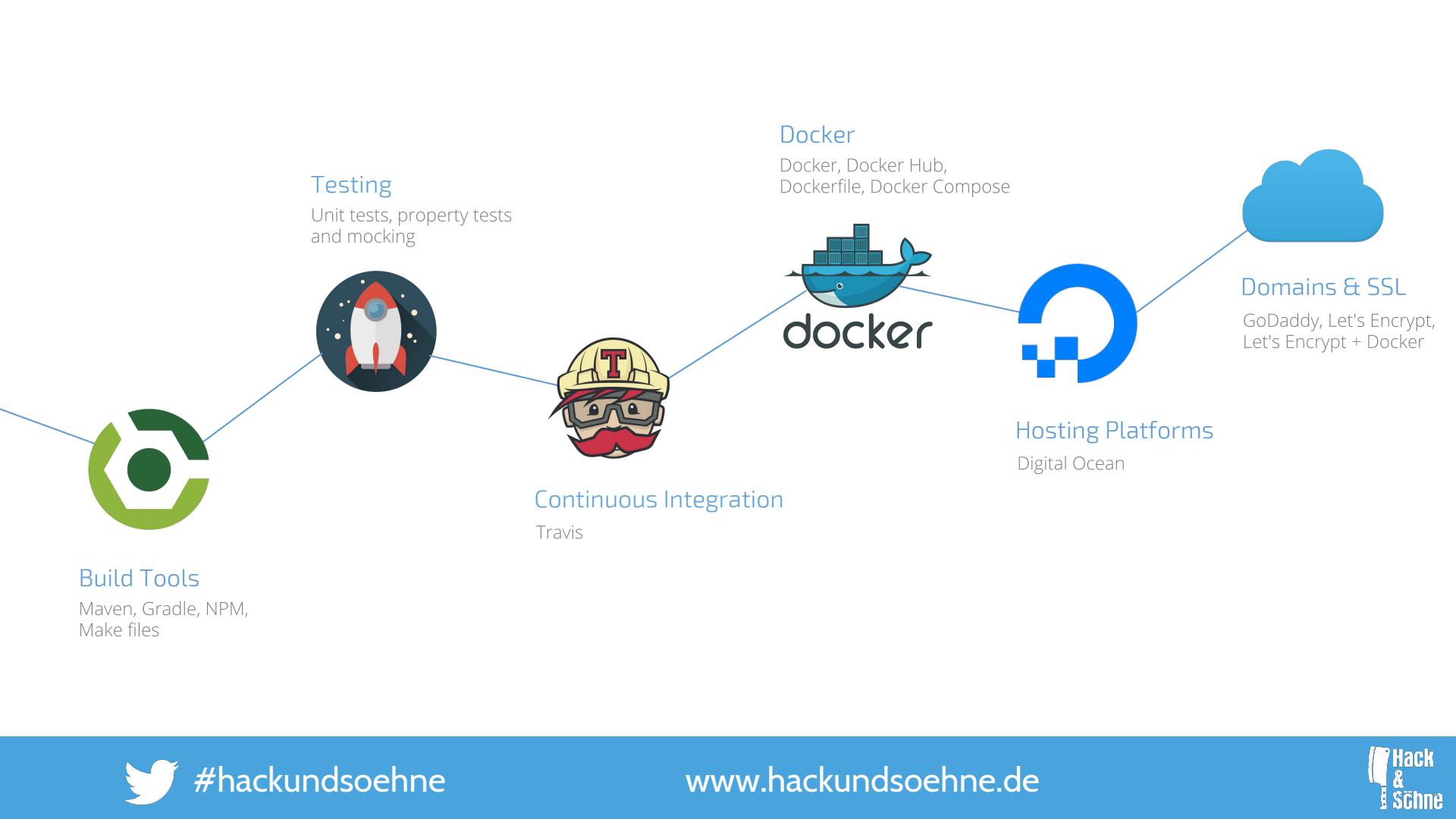


Git & GitHub

Pull requests, feature
branches, issues, GitHub
workflow, and more

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de





Support



#hackundsoehne

www.hackundsoehne.de



What to do when you are stuck?

Google! (thoroughly, **for a good hour**)

Ask stack overflow.

See if the project has some sort of chat, they often do, especially bigger projects.



#hackundsoehne

www.hackundsoehne.de



What chats are there

Gitter



IRC



Slack



Google groups



#hackundsoehne

www.hackundsoehne.de

Unboxing: Gitter



#hackundsoehne

www.hackundsoehne.de



...and Instagram!



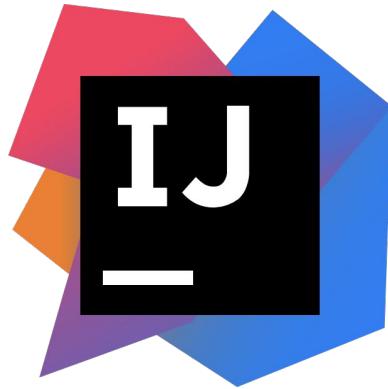
/hackundsoehne



#hackundsoehne

www.hackundsoehne.de





IDEs & Editors



#hackundsoehne

www.hackundsoehne.de



IDE vs Editor

IDE:

- IDE can debug
- IDE Setup (**knows context**)
- IDE is slow and big

Editor:

- Editor is very fast!
- No need to set up
- Editor is not made to debug



#hackundsoehne

www.hackundsoehne.de



Why IntelliJ is better than Eclipse

IntelliJ has that eclipse does not:

- Fast and smart contextual assistance
- Live Expression Debugger
- Amazing tool integration (Git, Maven, Gradle)
- Intuitive and Slick UI



#hackundsoehne

www.hackundsoehne.de



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "tiltmaster9000". The `Main` class is selected in the editor.
- Editor:** The code for `Main` is displayed. A red box highlights the declaration of `private final ActivityNotifier activityNotifier`. Below it, a code completion tooltip is shown for `activityNotifier.`, listing methods: `fire(Boolean event)`, `finalize()`, `notify()`, and `notifyAll()`.
- Toolbars:** Standard Java development tools like Run, Stop, and Debug are visible.
- Frames:** The Frames tool window shows the current stack trace:
 - "main" @1 in group ...
 - <init>:29, Main (xyz.tiltmaster)
 - main:42, Main (xyz.tiltmaster)
- Variables:** The Variables tool window shows the current state of variables:
 - this = {Main@1279}
 - activityNotifier = {ActivityNotifier@1281}
 - keyboardListener = {KeyboardListener@1282}
 - this.keyboardListener = {KeyboardListener@1282}
 - outputListener = {OutputListener@1283}
 - threadPool = {ThreadPoolExecutor@1284} "java.util.concurrent.ThreadPoolExecutor@4e9ba398[Running, pool size = 0, active threads = 0, queued tasks = 0, completed tasks = 0]"
 - this.outputListener = {OutputListener@1283}
- Result:** A tooltip for `activityNotifier` shows the result of the `fire` method: `{NullPointerException@1290} Method threw 'java.lang.NullPointerException' exception.`



#hackundsoehne

www.hackundsoehne.de



The screenshot shows the IntelliJ IDEA interface with the following details:

- Maven Projects** tool window on the right side, showing the project structure and available goals: Lifecycle (clean, validate, compile, test, package, verify, install, site, deploy), Plugins, and Dependencies.
- Code Editor** (Izou.java) showing the `pom.xml` file content. The file defines a Maven project with groupId `org.intellimate.izou`, version `1.15.1`, and packaging `jar`. It includes scm information for GitHub, properties for Log4j version `2.1`, and a license section for the Apache License, Version 2.0. The dependencies section lists Log4j API and Core.
- Project Structure** tool window on the left, showing the project's directory structure including `src/main/java`, `resources`, and `target` artifacts.
- Bottom Status Bar** showing the current time (27:27), file encoding (UTF-8), Git status (master), and other system information.

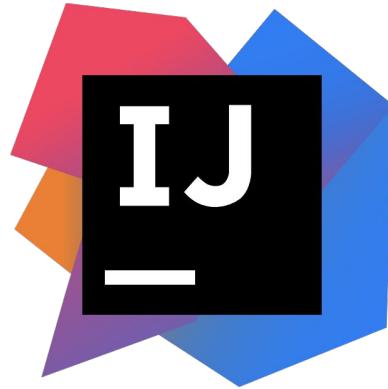


#hackundsoehne

www.hackundsoehne.de



Let's play: IntelliJ



#hackundsoehne

www.hackundsoehne.de



Let's play: IntelliJ

- IDE Setup (Project vs Module, Language)
- Editor-interactions (warnings, errors, navigation)
- VCS (Merge!)
- Code-generation
- Debugging (inline-stuff)
- History



#hackundsoehne

www.hackundsoehne.de



Useful Editors

- Sublime
- Atom
- VS Code



#hackundsoehne

www.hackundsoehne.de



Terminal Editors

- Nano
- Vim



#hackundsoehne

www.hackundsoehne.de



Nano

Exit: Ctrl x

Let's play:

```
:::  
iLE88Dj. :jD88888Dj:  
.LGitE888D.f8Gjjl8888E; .d8888b. 888b 888 888 888  
iE :8888Et. .G8888. d88P Y88b 8888b 888 888 888  
;i E888, :8888, 888 888 88888b 888 888 888  
D888, :8888: 888 888 88888b 888 888 888  
D888, :8888: 888 88888 888 Y88b888 888 888  
D888, :8888: 888 888 888 Y88888 888 888  
D888, :8888: Y88b d88P 888 Y8888 Y88b. .d88P  
888W, :8888: "Y88888P88 888 888 "I88888P"  
W88W, :8888:  
W88W: :8888: 88888b. 8888b. 88888b. .d88b.  
DGDD: :8888: 888 "88b "88b 888 "88b d88P" "88b  
:8888: 888 888 .d888888 888 888 888 888  
:W888: 888 888 888 888 888 888 Y88.88P  
:8888: 888 888 "Y888888 888 888 "Y88P"  
E888:  
tW88D
```



#hackundsoehne

www.hackundsoehne.de

Vim

The most asked question on Stack Overflow:

How to exit Vim.

Exit (on save): (esc) :x or Shift z z

Exit (without save): (esc) :q!



Let's play:



#hackundsoehne

www.hackundsoehne.de



Linters - what do they do?

- Show formatting discrepancies
- Show non-adherence to code standards and conventions
- Pinpoint to possible logical errors in your code



#hackundsoehne

www.hackundsoehne.de



Linters

Just as important as a book.

```
1  'use strict';
2  ● var foo = "bar";
3
4  ● fn(function (err) {});
```

Error foo is defined but never used (no-unused-vars) at line 2 col 5

Error foo is defined but never used (no-unused-vars) at line 2 col 5

Error Strings must use singlequote. (quotes) at line 2 col 11

Error "fn" is not defined. (no-undef) at line 4 col 1

Warning Expected error to be handled. (handle-callback-err) at line 4 col 4

Error err is defined but never used (no-unused-vars) at line 4 col 14



#hackundsoehne

www.hackundsoehne.de



What IDE for what Language?

Java: IntelliJ

Javascript: WebStorm

Python: PyStorm

Ruby: RubyMine

C/C++: Visual Studio, CLion

C#: Visual Studio, Rider (cross platform), MonoDevelop (for Unix)

Haskell: VS Code, Emacs





Shell



#hackundsoehne

www.hackundsoehne.de



Prozesse

- Bestehen aus
 - ID (PID)
 - Parent process
- Beinhaltet
 - UID
 - CWD (Current working directory)
- Bei interesse über weitere internals:
 - /proc/<pid>/* (z.B. /proc/<pid>/cwd)
 - Erlaubt Einsicht in weitere Prozess internals



#hackundsoehne

www.hackundsoehne.de



Der Shell process

- Normale Prozesse
 - zsh
 - bash
 - fish
- Erlaubt die Ausführung von anderer Software (z.B. ls cd etc.)
 - Überwachung des Prozesses
 - Interaktion mit dem Prozess
 - **CTRL+C CTRL+Z CTRL+D**
 - Ausgabe / Eingabe



#hackundsoehne

www.hackundsoehne.de



Der Shell process (2)

- Utils erlauben Exploration & Manipulation des Dateisystems
- Scripte
 - *sourcen* Ausführen der Kommandos eines files
 - Sinnvoll zum Aufsetzen eines Environments
 - `$ source myscript`
 - Ausführen eines files `./myscript`
 - Sinnvoll um Downloads auszuführen
 - `$./myscript`



Beispiel

```
#!/bin/sh
```

```
DIR=$1
```

```
FILE=$1
```

```
export ELM_THEME=default
```

```
mkdir -p $DIR || true
```

```
exec $@
```

```
kill -SIGUSR1 "$(pidof enlightenment_start)"  
gdb /usr/local/bin/enlightenment "$(pidof  
enlightenment)"
```



#hackundsoehne

www.hackundsoehne.de



Benutzerrechte

- Schränkt Verantwortung ein
 - z.B. im Filesystem
 - Bei System aufrufen
- Kann z.B. präventiv gegen fatale Fehler in einer software helfen
 - z.B.: <https://github.com/valvesoftware/steam-for-linux/issues/3671>
 - Fehlende Environment variable führt zum Aufruf von `rm -rf /`
 - Fehlende Benutzerrechte führen "nur" zur Löschung von "`~/.
~`"



Quelle:



#hackundsoehne

www.hackundsoehne.de



sudo

- Erweiterung der Benutzerrechte nach Passwortabfrage
- *sudo foobar -b*
 - Ausführung von *foobar -b*
 - Mit benutzerrechten von *root* (konfigurierbar)



!!!Shit can go wrong!!!!



#hackundsoehne

www.hackundsoehne.de



Datei Besitzer und Rechte

- *chmod* zum Ändern von Datei Rechten
 - chmod <optionen> <datei>
 - z.B. *chmod +x example.txt*
- *chown* zum Ändern der Datei Besitzer
 - chown <user>:<group> <datei>
 - z.B. *chown marcel:users example.txt*



Manuals

- Ungeklärte Fragen? <command> --help
- man <command>
- z.B. *man ls*

```
zsh
ls(1)                               User Commands
ls(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).  Sort entries alphabetically if none of
    -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .
Manual page ls(1) line 1 (press h for help or q to quit)
```



#hackundsoehne

www.hackundsoehne.de



Nützliche Applikationen

- *cd* - Change directory
 - Ändert CWD auf neues directory
- *sudo* - ändere aktuelle Ausführungsrechte zu *root*
- *ls* - zeige Verzeichnisinhalt des aktuellen CWD an
- *man* - Die FAQ für alles in der Shell
 - *\$ man ls*
- *rm* - Entfernung einer Datei oder eines Verzeichnisses
- *nano / vim* - Zum Editieren von Dateien





Package Managers



#hackundsoehne

www.hackundsoehne.de



What is a package manager?

```
apt-get install google-chrome-stable
```



#hackundsoehne

www.hackundsoehne.de



But.. why?

- No dependency chaos on your machine (old dependencies, etc.)
- All packages in one place!
- Easily install, update & remove software



#hackundsoehne

www.hackundsoehne.de



Advanced Packaging Tool (APT): Linux

apt-get install packagename: Used to install a package.

apt-get update: Resynchronize the package index files from their sources, so that information about new and updated packages is available.

apt-get upgrade: Used to install new versions of all packages currently installed on the system.

apt-get remove packagename: Will remove the binaries, but not the configuration or data files of the package.

apt-get purge packagename: Will remove *almost* everything regarding the package.



#hackundsoehne

www.hackundsoehne.de



Homebrew: Mac

Mostly the same as apt..

brew doctor: Check your Homebrew installation for common issues

All commands:

<http://zqpythonic.qiniucdn.com/data/20111223160257/index.html>



#hackundsoehne

www.hackundsoehne.de



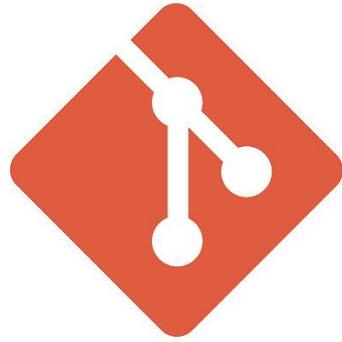
Install Homebrew for the Mac people



#hackundsoehne

www.hackundsoehne.de





Don't be a Git



#hackundsoehne

www.hackundsoehne.de



Terminologie um git

Commits

Branches

Remotes

Repositories



#hackundsoehne

www.hackundsoehne.de



Commits

- Besteht aus
 - Name (z.B. 716cc6be5ba4c7c5112750427281e448457cfe1a)
 - Beschreibung
 - Autor
 - Datum
 - Vorhergehender Commit
 - ...



#hackundsoehne

www.hackundsoehne.de



Branches

- Bestehend aus:
 - Verkettung von Commits
 - Namen



Aufbau eines Repositories



#hackundsoehne

www.hackundsoehne.de



Remotes

- Online Repository
- Austauschplattform von Zuständen
- *origin* Standard Remote name
- *git remote add <name> <URL>*
- *git pull <name>*



#hackundsoehne

www.hackundsoehne.de



Historie auf einem Remote

- Definiert durch eine feste geordnete Reihenfolge von Commits
- Diese sollte nicht gebrochen werden
- Kann erweitert werden durch Anfügen von neuen Commits



#hackundsoehne

www.hackundsoehne.de



Praxis time!



#hackundsoehne

www.hackundsoehne.de



Der Start

- *git init* - Initialisierung des Repositories
- *git clone <url>*- Download eines Repositories von einem Remote



#hackundsoehne

www.hackundsoehne.de



Unser Repository

<https://github.com/hackundsoehne/object-service.git>



#hackundsoehne

www.hackundsoehne.de



git branch <name>

Erstellt einen neuen Branch



#hackundsoehne

www.hackundsoehne.de



git checkout <name>

Wechsel in Zustand des Branches <name>

Remote Branches werden automatisch beim checkout
lokal erstellt
(z.B. *fix-docker-config*)



Commit it!

- Eine Änderung machen (z.B. in der README.md)
- Möglicherweise mit *git add* neue Dateien hinzufügen
- *git commit -a*
 - Öffnet einen Editor, dort die Commit Nachricht eingeben



#hackundsoehne

www.hackundsoehne.de



Nun wieder zusammenführen

Merge oder Rebase



#hackundsoehne

www.hackundsoehne.de

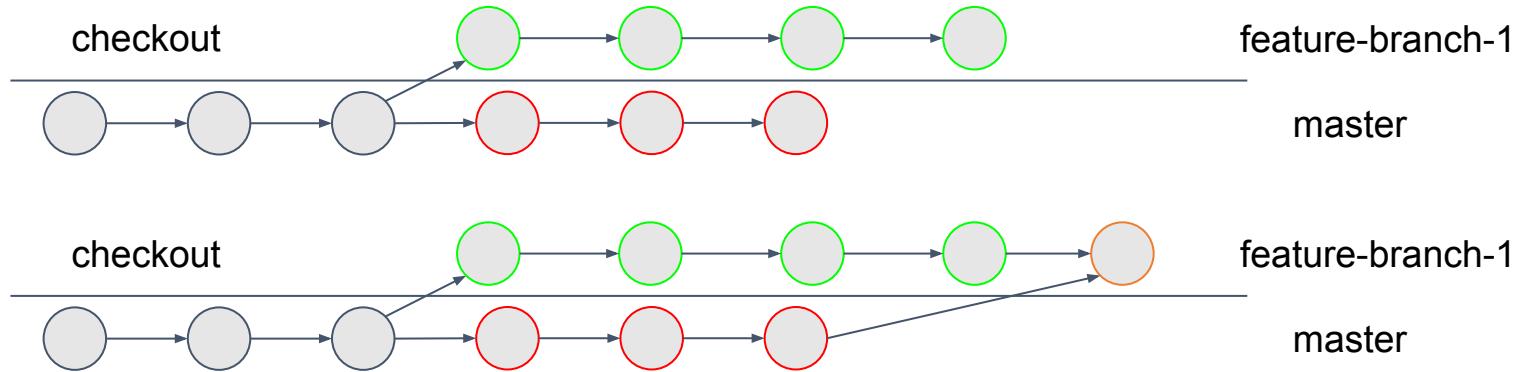


git merge

- Zueinander Führen von Branches
- *git merge B* (während wir auf dem master Branch sind)
 - Ermittlung aller Commits welche nur in B enthalten sind
 - Erneute Anwendung der Commits auf den master
 - Fast forward
 - *Squash* all der Commits in den zum merge Commit
 - No fast forward
- “merge commit” erstellt
 - enthält alle Änderungen von B



git merge master



#hackundsoehne

www.hackundsoehne.de



git rebase <name>

- Erkenne alle Commits die auf dem aktuellen Branch sind aber nicht im checkout Branch
- Wende diese Commits erneut auf den <name> Branch an
- Vorteile
 - Lineare Historie
- Nachteil
 - Historie muss u.U. neu geschrieben werden

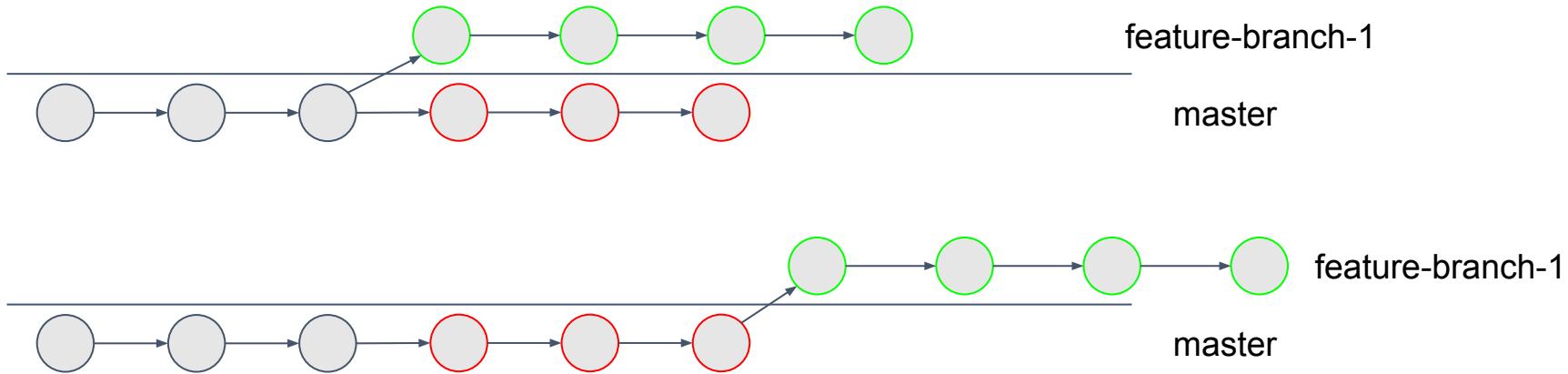


#hackundsoehne

www.hackundsoehne.de



git rebase <name> (2)



#hackundsoehne

www.hackundsoehne.de



Ausprobieren!

Branch *advanced-documentation* in master mergen!

git checkout master

git merge <der-name>



#hackundsoehne

www.hackundsoehne.de



Give me updates!

- *git pull* - Neue Commits werden vom Remote auf den lokalen Branch gelegt
- Verwendete Strategien
 - rebase
 - merge



#hackundsoehne

www.hackundsoehne.de



Who's the one to blame

- *git blame* - Zeigt den letzten Commit jeder Zeile einer Datei an
- Wer hat einen Bug verursacht ?
- Warum wurde dieser Code hinzugefügt



#hackundsoehne

www.hackundsoehne.de



git log

- Welche Änderungen wurde wo zuletzt gemacht
- Was sind die letzten Änderungen eines Autors



#hackundsoehne

www.hackundsoehne.de



Push it to the lim... origin

- *git push (remote)*
- Schickt die Differenz von lokalen Branch zu Remote an den Remote
- Alles was auf dem Remote ist kann nicht verändert werden
 - Es sei denn man pushed mit *-f* (NEIN NICHT TUN)



#hackundsoehne

www.hackundsoehne.de



Dokumentation

git help <command>

Schnelle Dokumentation zu weiteren Optionen und Kommandos



#hackundsoehne

www.hackundsoehne.de



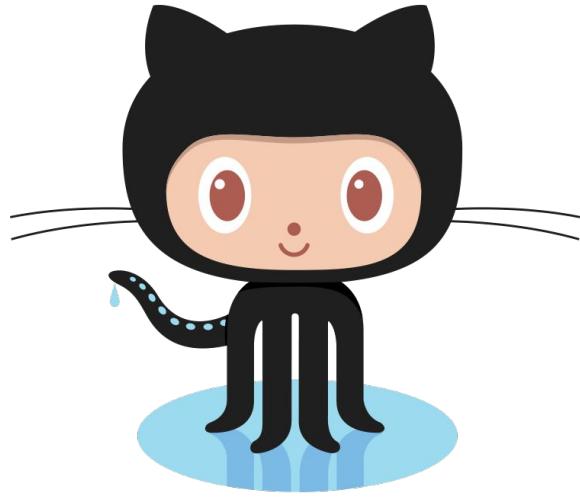
Weitere Kommandos

...

z.B.:

- git squash
- git cherry-pick
- git diff
- git revert
- git reset
- git status





GitHub



#hackundsoehne

www.hackundsoehne.de



Goal

To show you “the correct” GitHub workflow.



#hackundsoehne

www.hackundsoehne.de



All the important things:

- Organizations & Repos
- Feature branches
- Issues
- Pull requests
- Protecting branches
- Services
- GitHub Pages
- Personal Access Key



#hackundsoehne

www.hackundsoehne.de



Organizations & Repos

Exemplator organization demo:

<https://github.com/exemplator>



#hackundsoehne

www.hackundsoehne.de



Feature branches

Nasa feature branch demo:

<https://github.com/nasa/earthdata-search/network>



#hackundsoehne

www.hackundsoehne.de



Issues

PSE project (Crowd Control) issue demo:

<https://github.com/coolcrowd/object-service/issues/320>



#hackundsoehne

www.hackundsoehne.de



Pull requests

PSE project (Crowd Control) pull request demo:

<https://github.com/coolcrowd/object-service/pull/321>



#hackundsoehne

www.hackundsoehne.de



Protecting a branch



#hackundsoehne

www.hackundsoehne.de



Services

Connect services like Travis to your GitHub repo, but more on that later..



#hackundsoehne

www.hackundsoehne.de



GitHub Pages

Hack & Söhne tooling cheat sheet demo:

<https://github.com/hackundsoehne/tooling-cheatsheet>



#hackundsoehne

www.hackundsoehne.de



Personal Access

Via ssh (not here)

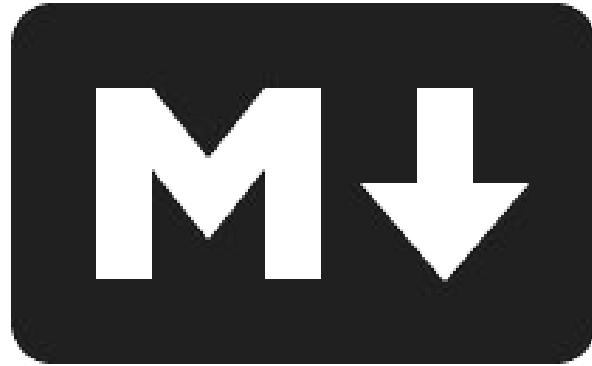
Via Personal Access Tokens



#hackundsoehne

www.hackundsoehne.de





Markdown



#hackundsoehne

www.hackundsoehne.de



Markdown structures Text



#hackundsoehne

www.hackundsoehne.de



```
1 # Heading
2
3 ## Sub-heading
4
5 ### Another deeper heading
6
7 Paragraphs are separated
8 by a blank line.
9
10 Two spaces at the end of a line leave a
11 line break.
12
13 Text attributes italic, *italic*, bold, **bold**,
`inline-code`.
14
15 Sometimes you want numbered lists:
16
17 1. One
18 2. Two
19 3. Three
20
21 Sometimes you want bullet points:
22
23 * Start a line with a star
24 * Profit!
25 * And if you have sub points, put two spaces before the dash or
star:
26   - Like this
27   - Yes, dashes work as well!
28
29 A [link](http://example.com).
```

Heading

Sub-heading

Another deeper heading

Paragraphs are separated by a blank line.

Two spaces at the end of a line leave a
line break.

Text attributes *italic*, *italic*, **bold**, **bold**, `inline-code`.

Sometimes you want numbered lists:

1. One
2. Two
3. Three

Sometimes you want bullet points:

- Start a line with a star
- Profit!
- And if you have sub points, put two spaces before the dash or star:
 - Like this
 - Yes, dashes work as well!

A [link](#).



```
1 There are many different ways to style code with GitHub's
2 markdown. If you have inline code blocks, wrap them in backticks:
3 `var example = true`. If you've got a longer block of code, you
4 can indent with four spaces:
```

```
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

```
if (isAwesome){
    return true
}

GitHub also supports something called code fencing, which allows
for multiple lines without indentation:

```
if (isAwesome){
 return true
}
```

And if you'd like to use syntax highlighting, include the
language:

```javascript
if (isAwesome){
 return true
}
```

```

There are many different ways to style code with GitHub's markdown. If you have inline code blocks, wrap them in backticks: `var example = true`. If you've got a longer block of code, you can indent with four spaces:

```
if (isAwesome){
    return true
}
```

GitHub also supports something called code fencing, which allows for multiple lines without indentation:

```
if (isAwesome){
    return true
}
```

And if you'd like to use syntax highlighting, include the language:

```
if (isAwesome){
    return true
}
```



The image shows a code editor window with two panes. The left pane is titled 'sample.md' and contains the following Markdown code:

```
1 First Header | Second Header
2 ----- | -----
3 Content from cell 1 | Content from cell 2
4 Content in the first column | Content in the second column
5
6
7 First Header | Second Header
8 :-----: | :-----:
9 Centered | Aligned Right
10 Content in the first column | Content in the second column
```

The right pane is titled 'Preview & sample.md' and displays the rendered table. It consists of two sections. The top section shows a table with two columns:

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

The bottom section shows a table with two columns:

| First Header | Second Header |
|-----------------------------|------------------------------|
| Centered | Aligned Right |
| Content in the first column | Content in the second column |



#hackundsoehne

www.hackundsoehne.de



The Benefits of Markdown

- Easy syntax
- Create rich text with plain text editor
- Plain Markdown has good readability
- Manage with Version Control Systems (git)



#hackundsoehne

www.hackundsoehne.de



Use Cases of Markdown

- GitHub
 - Readme
 - Wiki
 - Pull Request
 - Comments
- Stack Overflow
- Static Site Generators
- Slides, etc.



#hackundsoehne

www.hackundsoehne.de



Cheat Sheet



#hackundsoehne

www.hackundsoehne.de



Practice!

Write a short README.md
and try out preview in your Editor/IDE



#hackundsoehne

www.hackundsoehne.de





GitHub Pages demo!



#hackundsoehne

www.hackundsoehne.de





Package Managers & Build Tools



#hackundsoehne

www.hackundsoehne.de



But.. why?

- Managing multiple dependencies is a *huge mess*
- Building across multiple IDEs & platforms is a *huge mess*



#hackundsoehne

www.hackundsoehne.de



```
33 repositories {  
34     mavenCentral()  
35     mavenLocal()  
36 }  
37  
38 dependencies {  
39     compile 'org.jooq:jooq:3.7.1'  
40     compile 'org.jooq:jooq-meta:3.7.1'  
41     compile 'org.jooq:jool:0.9.7'  
42     compile 'mysql:mysql-connector-java:5.+'  
43     compile 'org.mybatis:mybatis:3.+'  
44     compile 'com.zaxxer:HikariCP:2.+'  
45     compile 'com.google.code.gson:gson:2.4'  
46     compile 'com.mashape.unirest:unirest-java:1.4.7'  
47     compile 'com.google.protobuf:protobuf-java:3.+'  
48     compile 'commons-validator:commons-validator:1.5.0'  
49     compile 'com.google.protobuf:protobuf-java-util:3.+'  
50     compile 'io.reactivex:rxjava:1.1.0'  
51     compile 'javax.mail:mail:1.4.7'  
52     compile 'org.apache.commons:commons-io:1.3.+'  
53     compile 'org.apache.commons:commons-lang3:3.4'  
54     compile 'org.apache.logging.log4j:log4j-api:2.5'  
55     compile 'org.apache.logging.log4j:log4j-core:2.5'  
56     compile 'org.apache.logging.log4j:log4j-slf4j-impl:2.0'  
57     compile 'org.yaml:snakeyaml:1.8'  
58     compile 'org.jooq:jooq-codegen:3.5.4'  
59     compile 'io.jsonwebtoken:jjwt:0.6.+'  
60     compile ('com.sparkjava:spark-core:2.3') {  
61         exclude group: 'org.slf4j', module: 'slf4j-simple'  
62     }  
63     compile 'org.joda:joda-money:0.11'  
64     testCompile 'org.mockito:mockito-core:1.+'  
65     testCompile 'junit:junit:4.12'  
66     runtime 'mysql:mysql-connector-java:5.+'  
67 }
```



So what does a build tool do?

Package manager:

- Dependency management

Build tool:

- Automated and system independent build lifecycles

- Testing
- Building
- Deploying
- ...



#hackundsoehne

www.hackundsoehne.de



So what do these tools **exactly** do?

1. Downloading dependencies.
2. Compiling source code into binary code.
3. Packaging that binary code.
4. Running tests.
5. Deploying to server.



#hackundsoehne

www.hackundsoehne.de



A quick example: **Maven**TM



#hackundsoehne

www.hackundsoehne.de



A few examples...

- Maven (**Java**)
- Gradle (**Java**)
- NPM (**Javascript**)
- CMake (**C/C++**)



#hackundsoehne

www.hackundsoehne.de



Maven - 2004

“Convention is good.”



#hackundsoehne

www.hackundsoehne.de



Maven - 2004

Pros:

- It works!
- Decent dependency management

Cons:

- XML (big and cumbersome)
- Huge overhead
- Not flexible



Need a programming language!



#hackundsoehne

www.hackundsoehne.de



Gradle - 2012

“Convention is good and so is flexibility.”



#hackundsoehne

www.hackundsoehne.de



Gradle - 2012

- No XML, instead Groovy DSL:
 - Much less boilerplate code
 - Config file much shorter and clearer
- Adopted by Google for Android



#hackundsoehne

www.hackundsoehne.de



Gradle - 2012

- No XML, instead Groovy DSL:
 - Much less boilerplate code
 - Config file much shorter and clearer
- Adopted by Google for Android



#hackundsoehne

www.hackundsoehne.de



build.gradle

```
1  group 'io.github.blobs'
2  version '1.0'
3
4  apply plugin: 'java'
5  apply plugin: 'application'
6
7  sourceCompatibility = 1.8
8  targetCompatibility = 1.8
9
10 mainClassName = 'io.github.blobs.Main'
11
12 repositories {
13     mavenCentral()
14 }
15
16 dependencies {
17     testCompile 'org.mockito:mockito-core:1.+'
18     testCompile group: 'junit', name: 'junit', version: '4.11'
19 }
20
21 jar {
22     manifest {
23         attributes 'Main-Class': mainClassName
24     }
25 }
26
```



NPM

Show by doing:



#hackundsoehne

www.hackundsoehne.de



CMake



#hackundsoehne

www.hackundsoehne.de



Let's play: Gradle



#hackundsoehne

www.hackundsoehne.de



Let's play: Gradle

- Groovy DSL basics
- Managing repositories and dependencies
- Configuring build tasks



#hackundsoehne

www.hackundsoehne.de



Some basic Gradle commands

gradle taskname: Execute task 'taskname'

gradle tasks: Show *most* available tasks

gradle tasks -all: Show *all* available tasks

gradle -b anotherfile.gradle taskname: Use another gradle file

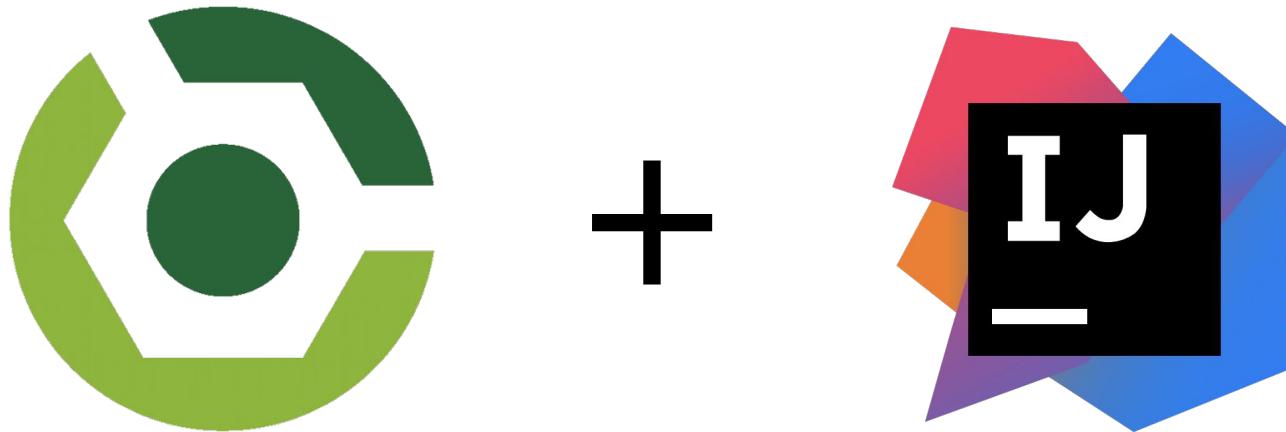


#hackundsoehne

www.hackundsoehne.de



Gradle IntelliJ Integration



#hackundsoehne

www.hackundsoehne.de



Overview of Package Managers & Build Tools

Java: Maven, Gradle

Javascript: NPM, Webpack, Rollup, Grunt, Gulp

Python: pip, PyBuilder

Ruby: RubyGem, Bundler, Rake

C/C++: Meson, CMake

C#: MsBuild

Haskell: Cabal, Stack





Testing



#hackundsoehne

www.hackundsoehne.de



But...why?



#hackundsoehne

www.hackundsoehne.de



We want to be confident in
our code, at all times.



#hackundsoehne

www.hackundsoehne.de



When is this helpful?



#hackundsoehne

www.hackundsoehne.de



Nov 15, 2015 – Apr 14, 2016

Contributions to master, excluding merge commits

Contributions: **Commits** ▾



LeanderK

588 commits / 129,470 ++ / 83,814 --

#1



lucaskrauss

480 commits / 86,608 ++ / 81,763 --

#2



marcelhollerbach

302 commits / 12,379 ++ / 4,238 --

#3



kelunik

206 commits / 20,510 ++ / 9,355 --

#4



redalertexport

#5



phimenk

#6



#hackundsoehne

www.hackundsoehne.de



[Code](#)[Issues 13](#)[Pull requests 2](#)[Projects 0](#)[Wiki](#)[Settings](#)[Insights](#)

Fix Continous Integration #328

[Edit](#)[Open](#)phimonk wants to merge 9 commits into `master` from `fix-ci-mysql`[Conversation 1](#)[Commits 9](#)[Files changed 5](#)[+10 -5](#)

phimonk commented an hour ago

Owner



This fixes errors with travis, gradle and tests that came from updated/changed/unavailable dependencies.

phimonk and others added some commits 2 days ago

- ci-test
- Fix outdated mysql install script for travis
- Add allow unauthenticated flag to apt install
- Try another install script that might work
- Use deprecated travis image
- gradle: fix codegen version ...
- ImageSimilarityTest: Ignore test case that failes because of outdated... ...
- Merge pull request #325 from coolcrowd/gradle ...
- Merge pull request #327 from coolcrowd/fix-test-duplicate-detection ...

bbbf2af

1370fbb

0195059

89838bb

ab6d600

37023f7

c23eff0

fe4c32a

cea823d

Reviewers



Suggestions

kelunik



lucaskrauss



LeanderK



Assignees



No one—assign yourself

Labels



None yet

Projects



None yet



#hackundsoehne

www.hackundsoehne.de



[Code](#)[Issues 14](#)[Pull requests 4](#)[Projects 0](#)[Wiki](#)[Insights ▾](#)

Changed mnist download source to web.archive.org, downloads also handle http 302 redirects #81

[Open](#)

avctrh wants to merge 9 commits into tensorflow:master from avctrh:webarchive

[Conversation 18](#)[Commits 9](#)[Files changed 1](#)[+19 -10](#)

avctrh commented on Mar 8

Contributor



MNIST downloads from Yann Lacun's website have been unreliable, leading to build (& CI) failures since this is part of Setup.hs in tensorflow-mnist-input-data.

I've made changes to Setup.hs to point to a webarchive.org cache of YL's site. This required code changes handle http 302 redirects, but still uses the HTTP library and doesn't add new dependencies.

Tested locally and resolves build failures due to MNIST downloads for me.

avctrh added some commits on Mar 6

[added osx dependency installation script](#) ✓ 7f1f9f2[added script to download mnist from webarchive since Yann LeCun's web... ...](#) ✓ ce8603f[requested changes from review](#) ✓ 554579b[whitespace](#) ✓ 4584a92**Reviewers**

blackgnezdo

**Assignees**

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone



#hackundsoehne

www.hackundsoehne.de



```
public class BadTests {
    @Test
    public void testCalculateSalary() {
        MockEmployee sample = new MockEmployee();
        sample.setLevel("Developer");
        DB db = new DBMock(sample);
        SalaryCalculator salaryCalculator = new SalaryQueries(db).getCalculator();
        assertEquals(55000, salaryCalculator.averageSalary());
    }
}
```



#hackundsoehne

www.hackundsoehne.de



```
@Test  
public void testImageSimilarity() {  
    Future<Image> imageAsync = loadImage("/test/image.jpg");  
    Image image = block(imageAsync);  
    Future<Integer> hashAsync = image.getHash();  
    int hash = block(hashAsync);  
    ...  
}
```



What are my options?



#hackundsoehne

www.hackundsoehne.de



1. Proving Correctness



#hackundsoehne

www.hackundsoehne.de



```
/*@ public behavior
@   ensures (myInt = \old(myInt) + 1);
@
@*/
public void addOneToMyInt() {
    myInt = myInt + 1;
}
```



2. Testing



#hackundsoehne

www.hackundsoehne.de



“Traditional” Testing

- The programmer must provide the sample
- The output is tested for one or more properties

Property based testing

- The programmer must specify properties
- The testing-library will now generate samples and test whether the property holds



#hackundsoehne

www.hackundsoehne.de



Testing Framework: **J**Unit



#hackundsoehne

www.hackundsoehne.de



Why use it?

- Simple
- Battle-tested
- Well-supported



#hackundsoehne

www.hackundsoehne.de



```
47
48 ➤    @Test
49    public void testExpectEmail() throws Exception {
50        Context context = prepareContext(data.getExperimentRecord().getIdExperiment(), data.getPla
51        Queries queries = prepareQuery(data, Optional.empty(), taskChooserAlgorithm: null);
52        View view = queries.getNext(context);
53        assertTrue(view.getType().equals(View.Type.EMAIL));
54    }
55 }
```



Run QueriesTest

All 13 tests passed – 5s 23ms

QueriesTest (edu.kit.ipd 5s 23ms)

- OK testStopgapAnswer 3s 819ms
- OK testCreative 72ms
- OK testInvalidQuality 93ms
- OK testProvideWorkerID 212ms
- OK testPreview 48ms
- OK testWrongCalibration 100ms
- OK **testExpectEmailFail** 86ms
- OK testExpectGetWorker! 127ms
- OK testStopgapAnswer 68ms

```
23:14:50.914 [main] DEBUG edu.kit.ipd.crowdcontrol.workerservice.query.Questions - handling no worker-id
23:14:50.916 [main] DEBUG edu.kit.ipd.crowdcontrol.workerservice.query.Questions - platform rlikf397379nq22hqa4a9brbdv returned nothing
23:14:50.918 [main] DEBUG org.jooq.tools.LoggerListener - Executing query : select `crowdcontrol`.`Platform`.`id_platform`, ...
23:14:50.918 [main] DEBUG org.jooq.tools.LoggerListener - -> with bind values : select `crowdcontrol`.`Platform`.`id_platform`, ...
23:14:50.919 [main] DEBUG org.jooq.tools.StopWatch - Query executed : Total: 0.587ms
23:14:50.921 [main] DEBUG org.jooq.tools.LoggerListener - Fetched result :
23:14:50.921 [main] DEBUG org.jooq.tools.LoggerListener - : +-----+
23:14:50.922 [main] DEBUG org.jooq.tools.LoggerListener - : |id_platform |name
23:14:50.922 [main] DEBUG org.jooq.tools.LoggerListener - : +-----+
23:14:50.922 [main] DEBUG org.jooq.tools.LoggerListener - : |rlikf397379nq22hqa4a9brbdv|v0t5tlcesv3giri3v5h
23:14:50.922 [main] DEBUG org.jooq.tools.StopWatch - Finishing : Total: 3.673ms, +3.086ms
```



#hackundsoehne

www.hackundsoehne.de



Demo time!



#hackundsoehne

www.hackundsoehne.de



JUnit Cheatsheet

- **testCompile('junit:junit:4.12')** - your gradle dependency
- **@Test** - to mark your tests with
- **@Test(expected = RuntimeException.class)** - if you expect them to throw Exceptions
- **import static org.junit.Assert.*** - to use the assertions
 - **assertTrue(x)** - whether x is true
 - **assertEquals(x, y)** - whether x equals y
 - ... (check Assert class)



#hackundsoehne

www.hackundsoehne.de



How many tests are enough?



#hackundsoehne

www.hackundsoehne.de



Mocking using **mockito**



#hackundsoehne

www.hackundsoehne.de



What is mocking?



#hackundsoehne

www.hackundsoehne.de



Example time!



#hackundsoehne

www.hackundsoehne.de



Mockito Cheatsheet

- **testCompile(org.mockito:mockito-core:2.+')** - your gradle dependency
- **import static org.mockito.Mockito.*** - to use the assertions
 - **mock(x)** - to mock x
 - **when(x).thenReturn(y)** - to fake functionality without input
Example: `when(mockedList.get(0)).thenReturn("first");`
 - **when(x.y(Mockito.anyInt())).thenAnswer()** - to fake functionality dynamically
 - ... (check Mockito class)



#hackundsoehne

www.hackundsoehne.de



Mock all the things?



#hackundsoehne

www.hackundsoehne.de



Test Driven Development (TDD)



#hackundsoehne

www.hackundsoehne.de



Test Driven Development

- Write your test(s) before you write your code!
- Can be a whole paradigm
- You don't have to buy all in
- Especially useful when working with new, complicated libraries



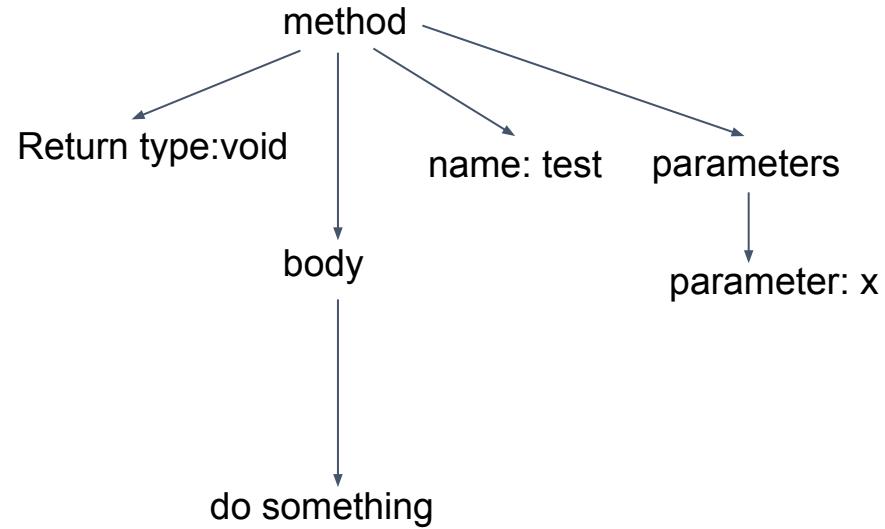
#hackundsoehne

www.hackundsoehne.de



AST

```
public void test(String x) {  
    doSomething();  
}
```



#hackundsoehne

www.hackundsoehne.de



Demo Time!



#hackundsoehne

www.hackundsoehne.de



JUnit Cheatsheet

- **testCompile('junit:junit:4.12')** - your gradle dependency
- **@Test** - to mark your tests with
- **@Test(expected = RuntimeException.class)** - if you expect them to throw Exceptions
- **import static org.junit.Assert.*** - to use the assertions
 - **assertTrue(x)** - whether x is true
 - **assertEquals(x, y)** - whether x equals y
 - ... (check Assert class)



#hackundsoehne

www.hackundsoehne.de



Mocha/Chai



#hackundsoehne

www.hackundsoehne.de



Mocha/Chai

Mocha => JUnit for Javascript, but without the asserts, provides other primitives (BDD)

Chai => the missing asserts



Behaviour Driven Development (BDD)



#hackundsoehne

www.hackundsoehne.de



Demo Time!



#hackundsoehne

www.hackundsoehne.de



Mocha Cheatsheet

- **npm install mocha** - your npm dependency
- **require('assert')** - to use asserts
- **describe('name', function)** - describe your use-case
- **it('description', function)** - your test



#hackundsoehne

www.hackundsoehne.de



Mocha/Chai Cheatsheet

- **npm install mocha** - your npm dependency
- **require('assert')** - to use asserts
- **describe('name', function)** - describe your use-case
- **it('description', function)** - your test



#hackundsoehne

www.hackundsoehne.de



Coverage



#hackundsoehne

www.hackundsoehne.de



Why do we care about coverage

- Convenient Metric whether your tests are thoroughly
- Provides a nice badge, looks good on paper



#hackundsoehne

www.hackundsoehne.de



Problems with Coverage

- Does not measure quality of your tests, only quantity
 - Many bugs result from the absence of code, not incorrect code
- Everybody focuses on getting 100% coverage
 - A lot of work
 - Tests tailored to the implementation
 - Diminishing returns



#hackundsoehne

www.hackundsoehne.de



Some general advice about testing software



#hackundsoehne

www.hackundsoehne.de



About testing

- Ask yourself: How much testing is necessary, how much testing can i afford?
- Listen to your tests, if they are a pain to write there is probably something wrong.
- Don't tailor your tests to your implementation
 - Implementations change
 - Do you really eliminate bugs?
- Outdated tests are a problem, never ever comment out tests!
- Write them early, a bit of discipline helps you in the long run
- Don't accept non-reproducible tests



#hackundsoehne

www.hackundsoehne.de



Questions?



#hackundsoehne

www.hackundsoehne.de





Continuous Integration



#hackundsoehne

www.hackundsoehne.de



Continuous Integration

- Automatisches *testing, building* und *deploying* für jeden *commit*

git push

VS

git push

Build

Execute Tests

Package and Upload

Test Again

Other annoying steps



#hackundsoehne

www.hackundsoehne.de



Anbieter

Travis CI: aus Berlin, gratis für Open Source Projekte, sehr gute Github Integration, gratis für Studenten



AppVeyor: gratis für Open Source Projekte, unterstützt Windows Builds



und weitere...



#hackundsoehne

www.hackundsoehne.de



Badges: So much winning!



README.md

These are "javascript bindings" to liblouis created by cross compiling [liblouis](#) using [emscripten](#). The Liblouis API written in C can be directly called using the `ccall` and `cwrap` functions provided by emscripten. As directly calling the C API is cumbersome, an additional API — called Easy API — is provided for most functions. This package supports NodeJS and browser environments.

[Easy-API @npm v0.3.0](#) [Easy-API @bower v0.3.0](#) [Latest C-API Build @npm v0.0.0-test.4](#) [Latest C-API Build @bower v0.0.0-test.4](#)

[Build of Easy-API](#) passing [Build of C-API](#) passing

Table of Contents

- 1. API Overview
 - i. Installation
 - ii. List of Available Liblouis Functions
 - iii. Compiling the Latest Version of Liblouis
 - iv. Testing a Liblouis Build
- 2. Usage Examples
 - i. Printing the Version Number Using the Easy API in the Browser
 - ii. Printing the Version Number By Directly Calling Liblouis in the Browser
 - iii. Printing the Version Number Using the Easy API in NodeJS



#hackundsoehne

www.hackundsoehne.de



Travis CI Einrichten

- Mit Github verknüpftes Konto erstellen
☒ <https://travis-ci.org/>
- Travis CI für das gewünschte *Repository* aktivieren
- Travis CI Konfigurationsdatei **.travis.yml** im *Root* des *Repositories* anlegen
- Distribution, Virtualisierungstyp und Programmiersprache auswählen

```
dist: trusty # Ubuntu 14.04
sudo: false # in Docker
language: node_js # mit NodeJS Tools vorinstalliert
# da keine Befehle spezifiziert sind, wird der Standardbefehl für Java Programme ausgeführt:
# npm test
```



#hackundsoehne

www.hackundsoehne.de



Travis CI Befehl anpassen

- Durchläuft mehrere Phasen, in welchen Befehle ausgeführt werden können:

*apt addons
cache components
before_install
install
before_script*

script
*before_cache
after_success
after_failure
before_deploy*

deploy
*after_deploy
after_script*

Syntax für *.travis.yml*

phase:
- Befehl
andere_phase:
- Befehl
- noch ein Befehl

- Exit Code der einzelnen Phasen entscheidet über *success*, *failure* oder *error*



#hackundsoehne

www.hackundsoehne.de



Travis CI Pitfalls

- Skripte im Repository mit chmod +x ausführbar machen
Befehle mit set -x anzeigen, \$? nur in Skripten verwenden
- Gefährliche Schritte (wie *Deploy*) für *Pull Requests* deaktivieren

```
if [ "$TRAVIS_PULL_REQUEST" != "false" -o "$TRAVIS_BRANCH" != "master" ]; then
    echo "Not publishing. Is pull request or non-master branch."
    exit 0
fi
```

- Passwörter und andere sensitive Informationen immer verschlüsselt anlegen
- Zeitnahe *Commits* können einen *Race Condition* im *Deployment* erzeugen!
- *Travis CI* führt *merged pull requests* aus, also wird *Travis CI* nicht bei *Merge Conflicten* ausgeführt!



Thesis Automatisch Kompilieren

Ziel: Jede Änderung des Latexquelltextes soll automatisch eine öffentlich zugängliche PDF erzeugen.

```
$ pdflatex ausarbeitung.tex  
$ bibtex ausarbeitung  
$ pdflatex ausarbeitung.tex  
$ pdflatex ausarbeitung.tex  
$ # irgendwie hochladen
```

Idee: In *Travis CI* kompilieren und auf *Github Pages* veröffentlichen.
(Analog auch für eine Webseite einsetzbar...)



#hackundsoehne

www.hackundsoehne.de



Thesis Automatisch Kompilieren Cont'd

sudo: required

dist: trusty

before_install: # get latex packages required by your thesis

- sudo apt-get -qq update && sudo apt-get install -y --no-install-recommends

texlive-latex-recommended

script: # make pdf, allow bibtex to contain errors

- mkdir build

- **pdflatex** -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex && (**bibtex** build/ausarbeitung || true) &&

- pdflatex** -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex &&

- pdflatex** -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex

after_success: # collect everything we want to publish/deploy

- mkdir deploy

- cp build/*.pdf deploy/

after_failure:

- cat build/ausarbeitung.log

deploy: # abuse deploy for github pages

provider: pages

github_token: \$GITHUB_TOKEN

local_dir: deploy

target_branch: pdfs

skip_cleanup: true

on:

branch: master

Linting & Testing der Konfiguration

Häufiges Problem: dutzende *Commits* beim Aufsetzen von *Travis CI*

- Einfache syntaktische Fehler durch *Linting* finden
🔗 <http://lint.travis-ci.org/>
- Für sudo: false verwendet *Travis CI* ein *Open Source Docker Image*
🔗 <https://quay.io/organization/travisci> für precise (Ubuntu 12.04)
🔗 <https://quay.io/organization/travisci> für trusty (Ubuntu 14.04)

```
docker run --name travis-debug -dit quay.io/travisci/travis-ruby /sbin/init docker exec -it travis-debug bash -l  
su - travis
```

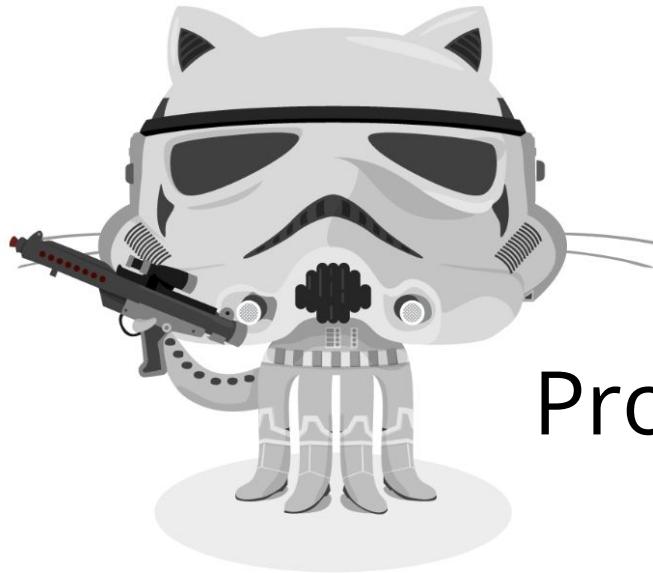
➤ Später mehr zu *Docker*...



#hackundsoehne

www.hackundsoehne.de





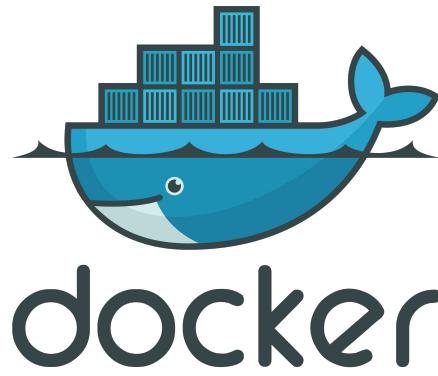
Protect and PR!



#hackundsoehne

www.hackundsoehne.de





Docker



#hackundsoehne

www.hackundsoehne.de



Deployment



#hackundsoehne

www.hackundsoehne.de



Why docker?



#hackundsoehne

www.hackundsoehne.de



Why docker?

- Easy if mastered once
- Everything is reproducible
 - Deployment is not dependent on server environment
 - Much cloud
 - Testing is easy
 - Shipping complex code
- Security



#hackundsoehne

www.hackundsoehne.de



This talk only
covers hobbyist
scenarios.



#hackundsoehne

www.hackundsoehne.de



Especially if you
store data, research
common security
measures!



#hackundsoehne

www.hackundsoehne.de



Docker internals



#hackundsoehne

www.hackundsoehne.de





By Vesahjr - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=20083285>



#hackundsoehne

www.hackundsoehne.de





By Vesahjr - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=20083285>



#hackundsoehne

www.hackundsoehne.de



VM

Host OS Kernel

Guest OS Kernel

Application

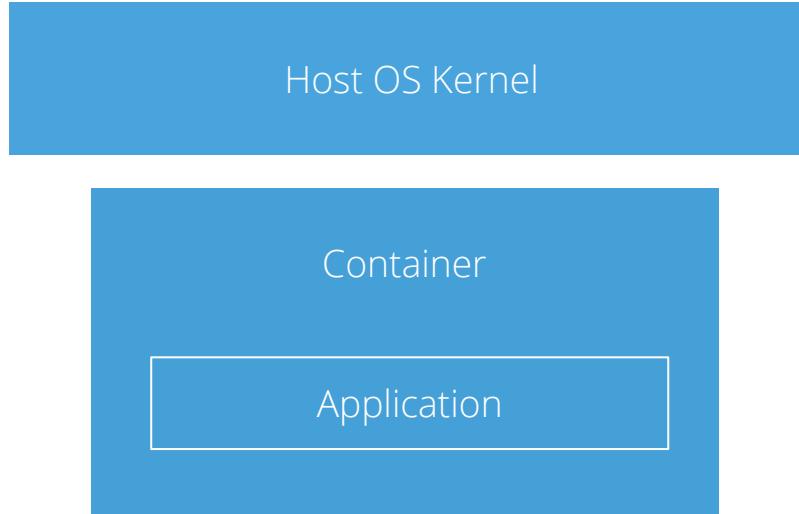


#hackundsoehne

www.hackundsoehne.de



Docker



#hackundsoehne

www.hackundsoehne.de



Cgroups

the kernel keeps track which processes belong together and what their capabilities are



#hackundsoehne

www.hackundsoehne.de



Docker Daemon

- powers docker
- manages the containers
 - sets up the cgroups
 - updates capabilities on change
 - keeps track of the containers
- simply interact via **docker xy**



#hackundsoehne

www.hackundsoehne.de



Virtualizations (overview)



#hackundsoehne

www.hackundsoehne.de



Dynamic file system - container

/certificates

/html

/java-code

...



./docker/server/certs

image

the parent

...



#hackundsoehne

www.hackundsoehne.de



Dynamic file system - overlayfs

/src

start.sh

/src

install.sh

/src

notify.sh

/src

start.sh

install.sh

notify.sh



Network

3 levels:

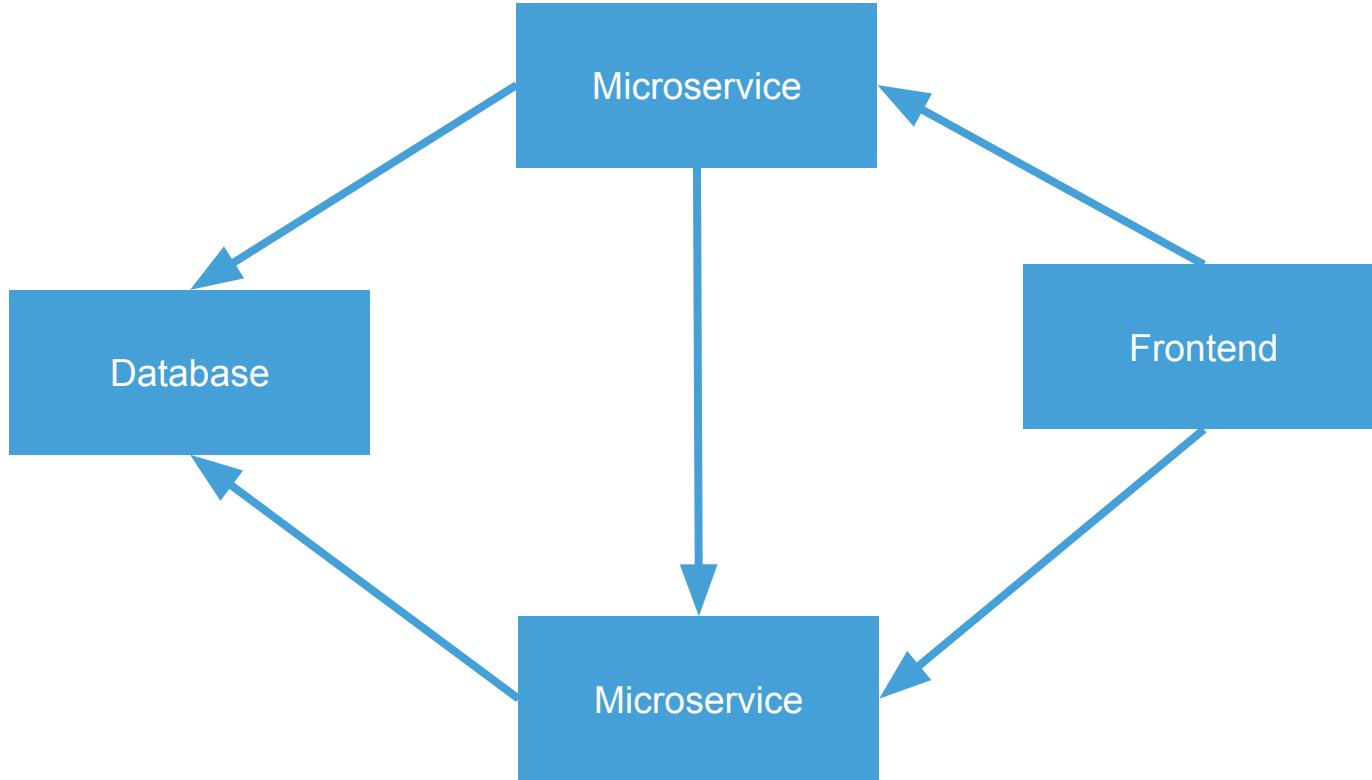
- exposed ports of the container
 - (on which ports should the container be reachable?)
- docker networks
 - (with which containers does the container communicate?)
- exposed ports of the host
 - (with which containers should the outside world be able to communicate?)



#hackundsoehne

www.hackundsoehne.de

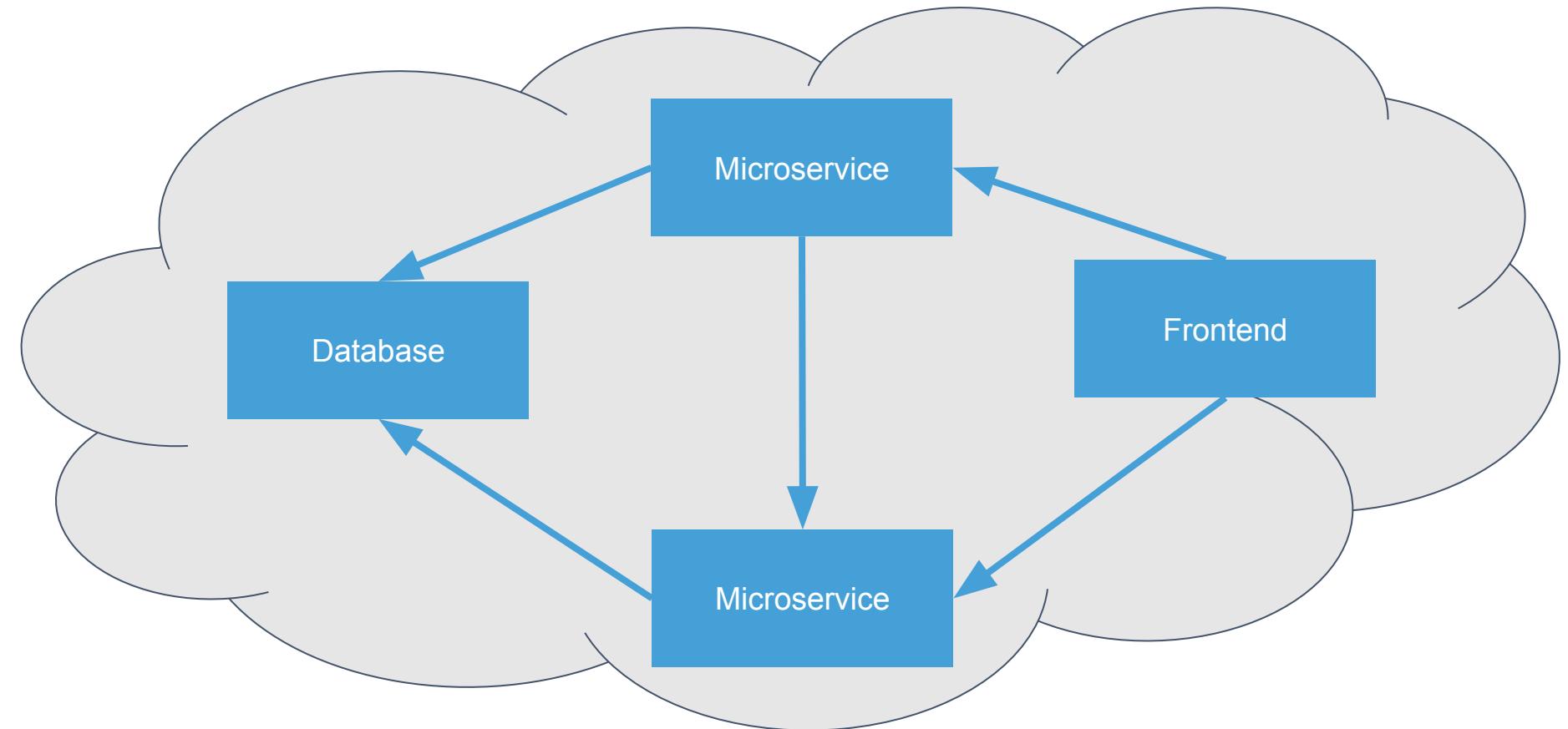




#hackundsoehne

www.hackundsoehne.de

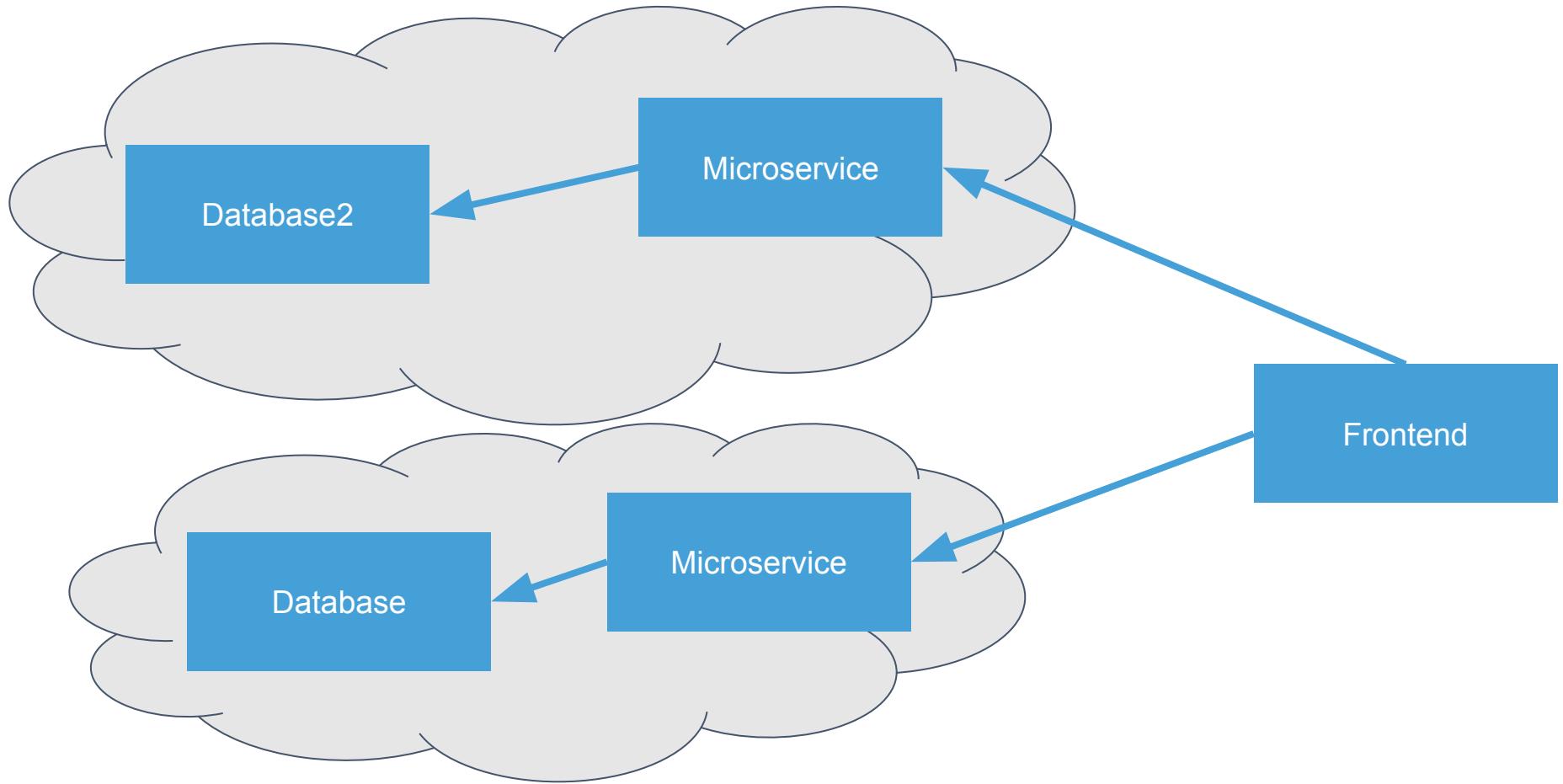




#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de



CPU Resources

ability to limit the containers

- CPU-share
- CPU-cores
- ram-access



#hackundsoehne

www.hackundsoehne.de



Docker Frontend



#hackundsoehne

www.hackundsoehne.de



Docker Engine



#hackundsoehne

www.hackundsoehne.de



Demo Time!



#hackundsoehne

www.hackundsoehne.de



installing docker

- clone **hackundsoehne/docker-assignments**
- instructions are in the **README.md** (best viewed in the browser)



#hackundsoehne

www.hackundsoehne.de



Commands Cheatsheet

- **docker build** - Build your docker image
- **docker push** - Push your docker image to a repository
- **docker pull** - Pull a docker image from a repository
- **docker run** - Run a docker image (image => container)
- **docker ps** - List all containers
- **docker rm** - Remove a container
- **docker stop** - Stop a container
- **docker logs** - see the logs of a container



Check your
containers -
they eat a lot of
disk space



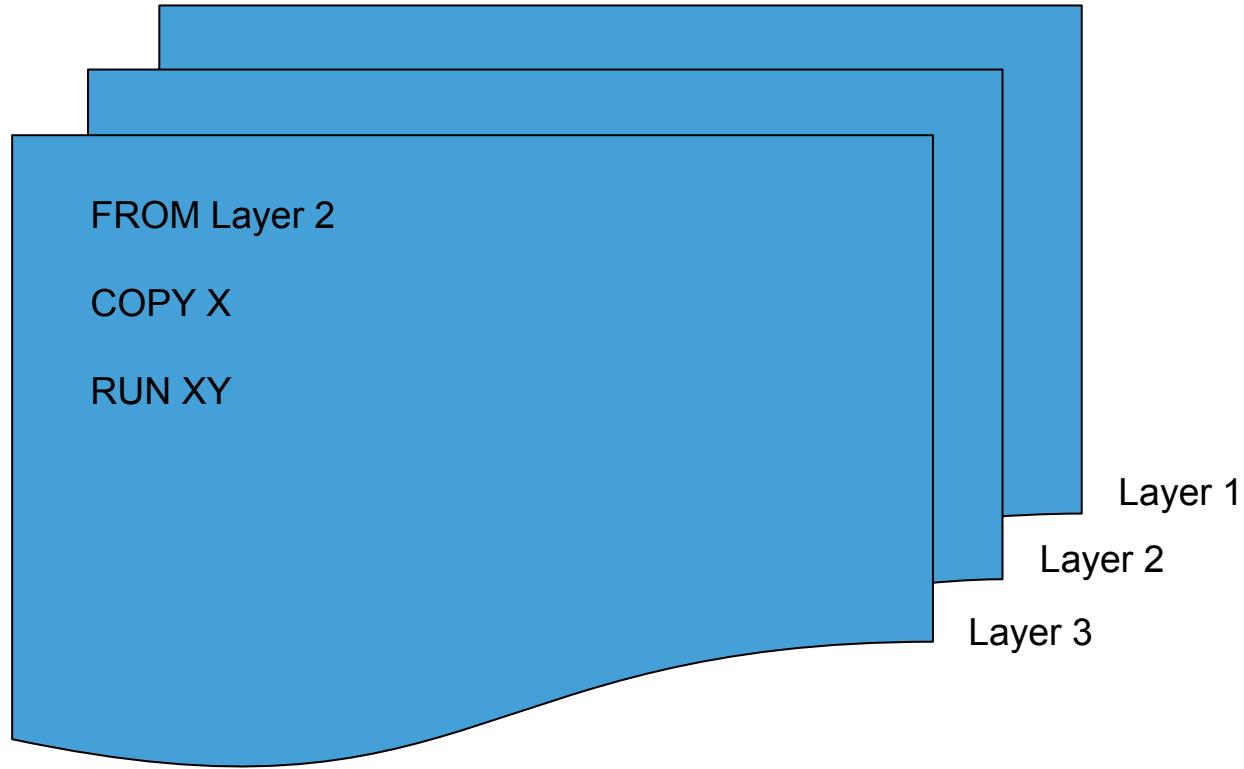
Dockerfile



#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de



Demo Time!



#hackundsoehne

www.hackundsoehne.de



Dockerfile Cheatsheet

- **FROM X** - the parent image
- **COPY X Y** - copies files from X (your local filesystem) to Y (image)
- **WORKDIR X** - changes the workdir to X
- **RUN X** - runs X during image-creation time
- **EXPOSE X** - opens the ports X from the docker-container
- **FROM X** - the parent image
- **CMD X** - runs THE COMMAND X when the container starts
- **ENTRYPOINT X** - runs x when the container starts



#hackundsoehne

www.hackundsoehne.de



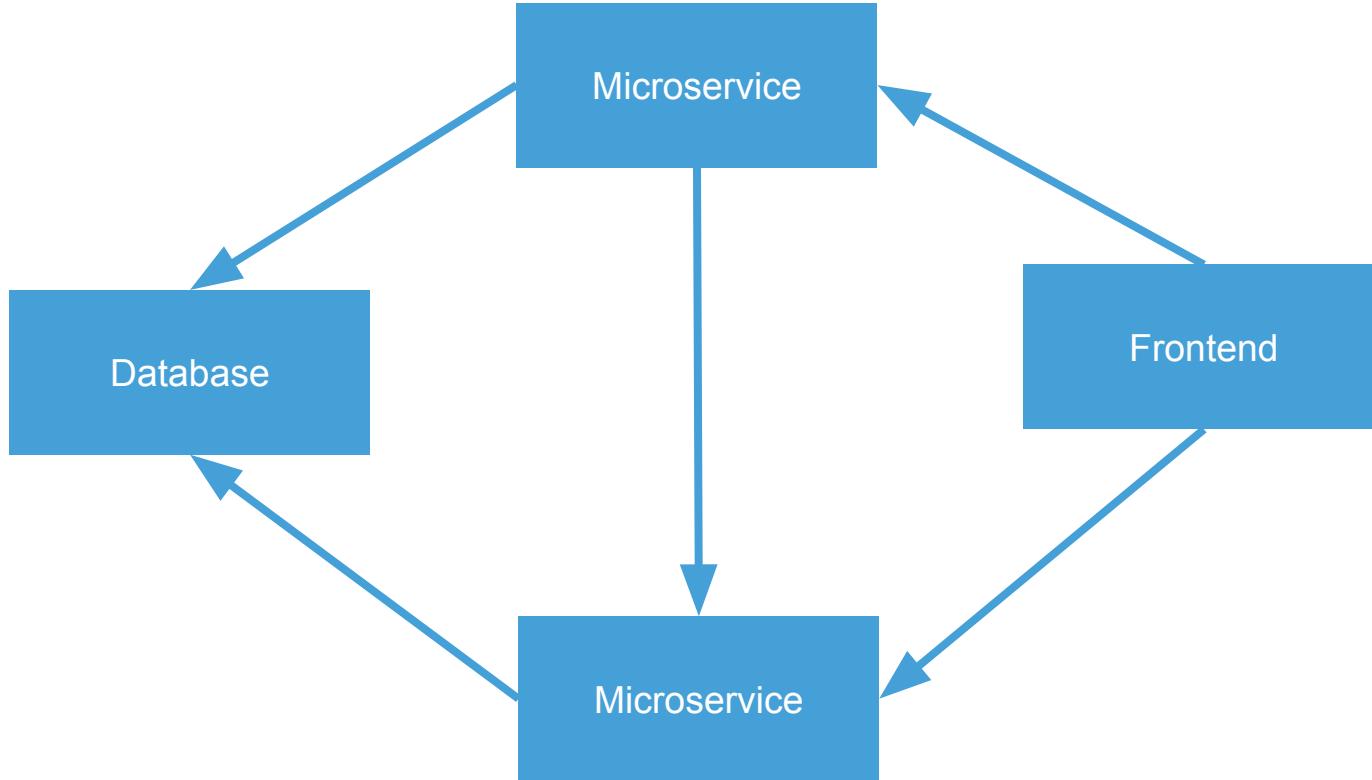
Docker Compose



#hackundsoehne

www.hackundsoehne.de





Docker compose to the rescue!



#hackundsoehne

www.hackundsoehne.de



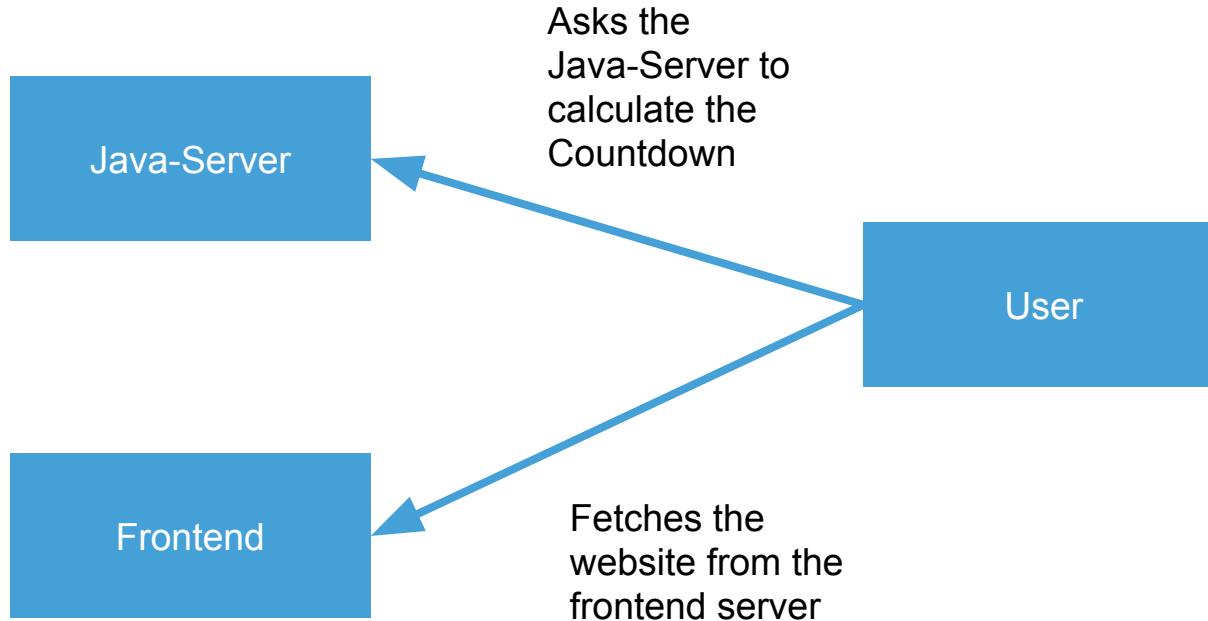
Demo time



#hackundsoehne

www.hackundsoehne.de





Docker Compose Cheatsheet

YAML - Syntax (**significant indentations with tabs**)

Group:

- Item1
- item2

Item: value



#hackundsoehne

www.hackundsoehne.de



Docker Compose Cheatsheet

version: '3' - needs to be on the top

services: - list your services

name: - the name of your service

image: "redis:alpine" - the source of your image

ports:

- **"8080:80"** - the open ports and the mapping

(HOST:CONTAINER)

environment:

- **KEY:VALUE** - sets the environment variable KEY to the value VALUE



#hackundsoehne

www.hackundsoehne.de



Docker Volumes

- the file-system is virtualized, but we still want to map local directories into docker container.
- useful for databases (you want to have the database-file in a local directory) etc.
- Docker Volumes to the rescue!
- They are used to map local directories dynamically into the container



#hackundsoehne

www.hackundsoehne.de



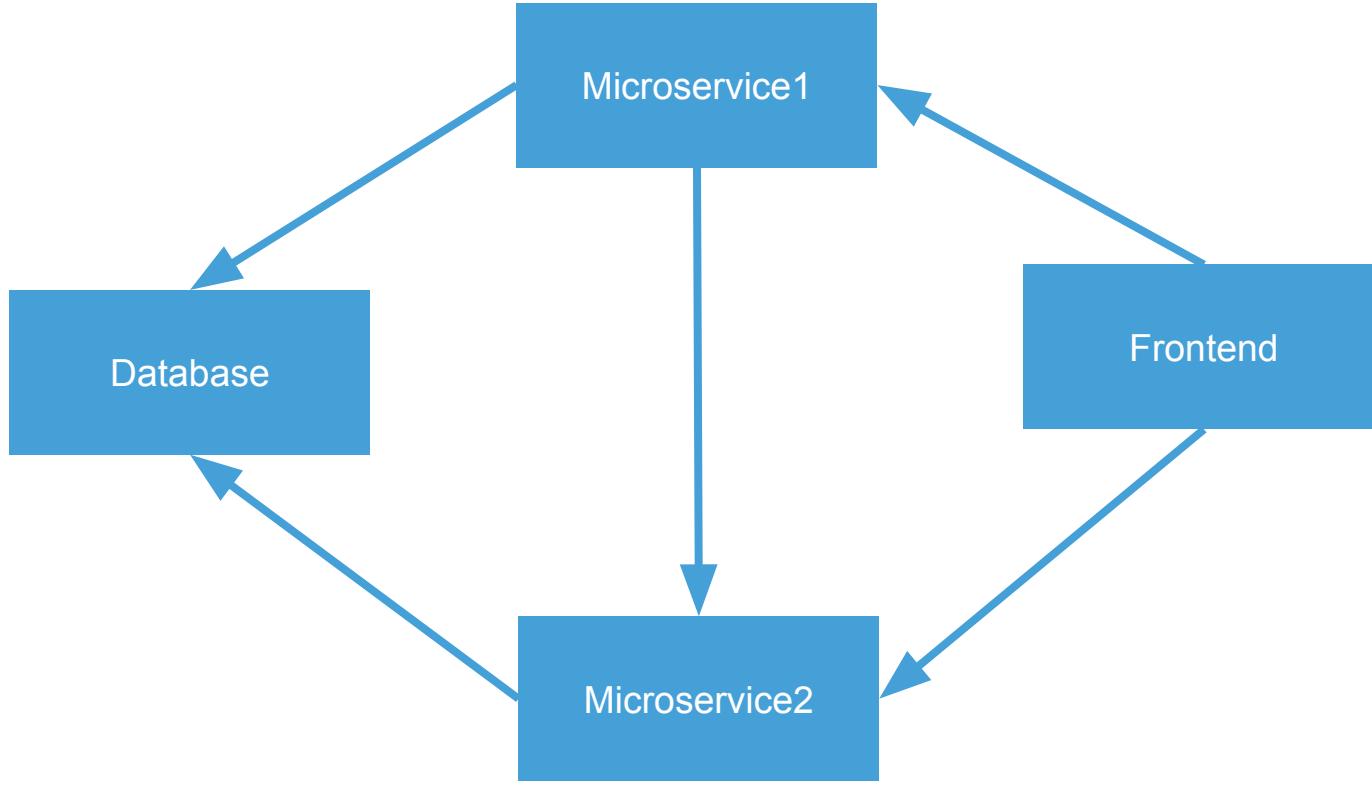
How to beautifully link two services?

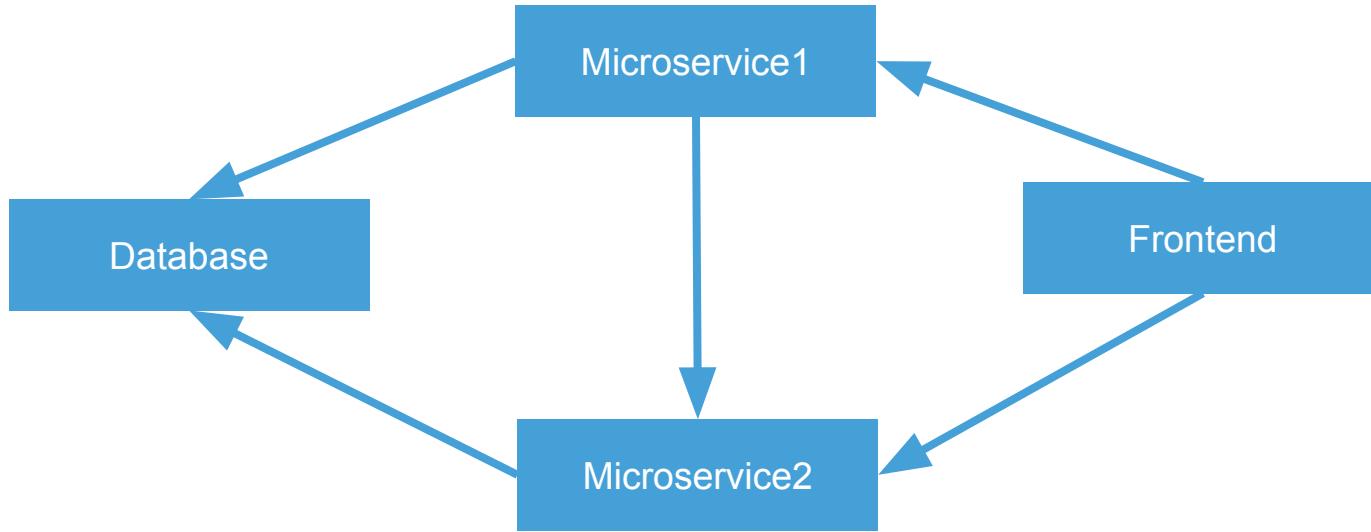


#hackundsoehne

www.hackundsoehne.de







ms1:

link:

- ms2:microservicetwo

 <http://microservicetwo:8080/calculate>



#hackundsoehne

www.hackundsoehne.de



Docker Compose - links

web:

links:

- **db:database** - SERVICE:ALIAS
- **ws:workerservice**

example usage: <http://workerservice:80/list/customers>



#hackundsoehne

www.hackundsoehne.de



Docker Compose - networks

web:

networks:

- **frontend**
- **backend**



#hackundsoehne

www.hackundsoehne.de



Docker Compose Cheatsheet

version: '3' - needs to be on the top

services: - list your services

name: - the name of your service

image: “**redis:alpine**” - the source of your image

ports:

- “**8080:80**” - the open ports and the mapping (HOST:CONTAINER)

networks: - assign the container to explicit networks, if left out it is added to the default network

- **backend**

- **frontend**

links: - provides explicit links to services

- **db:database**



#hackundsoehne

www.hackundsoehne.de



(Assignment)



#hackundsoehne

www.hackundsoehne.de



Docker ecosystem



#hackundsoehne

www.hackundsoehne.de



Docker ecosystem: nginx-proxy

auto-generates the nginx reverse proxy file based on the config file.

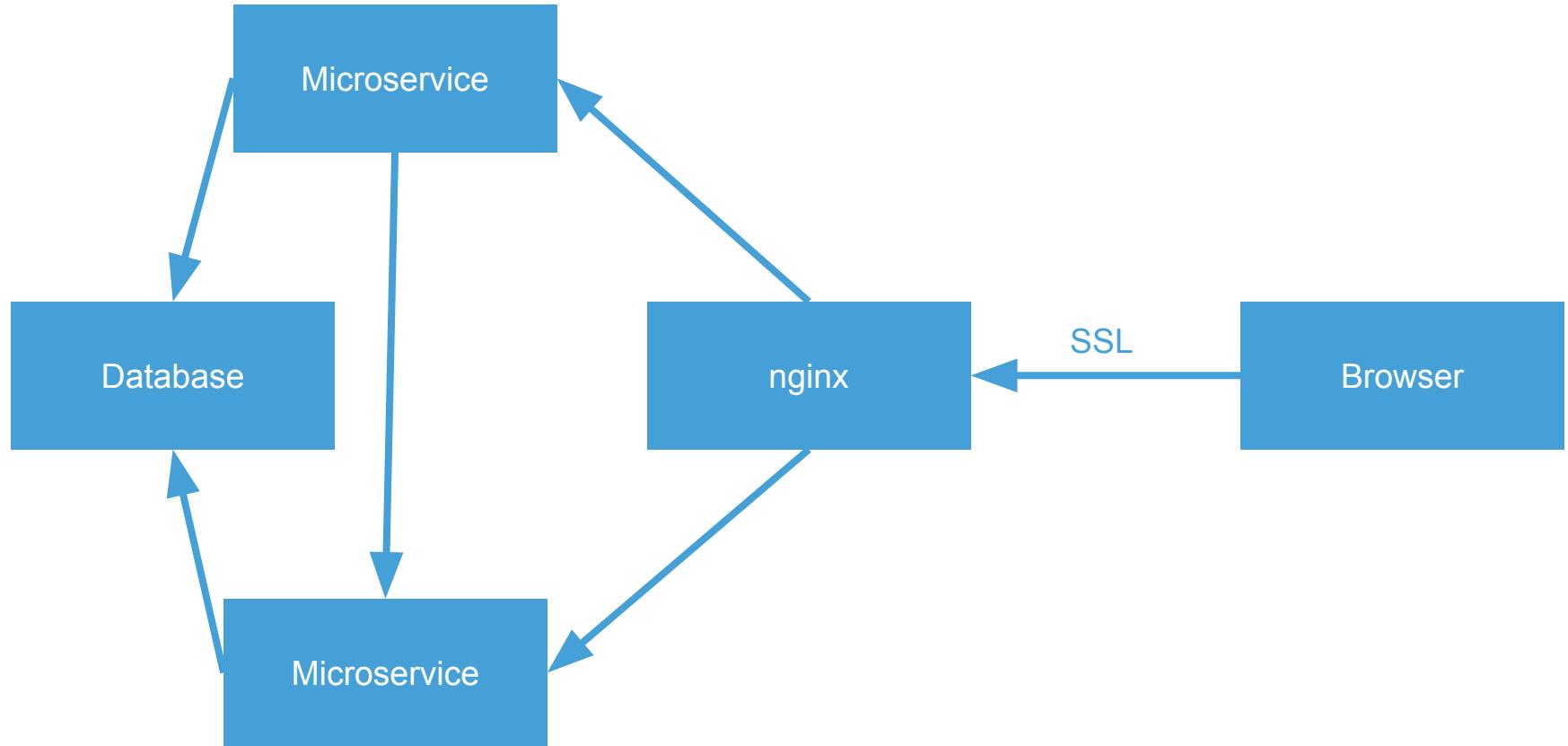
needs docker-daemon socket!



#hackundsoehne

www.hackundsoehne.de





#hackundsoehne

www.hackundsoehne.de



Docker ecosystem: letsencrypt-nginx-proxy-companion

automatically installs & configures SSL-certificates and
communicates with nginx-proxy



#hackundsoehne

www.hackundsoehne.de



Docker ecosystem: watchtower

automatically updates your images.
needs docker daemon socket!

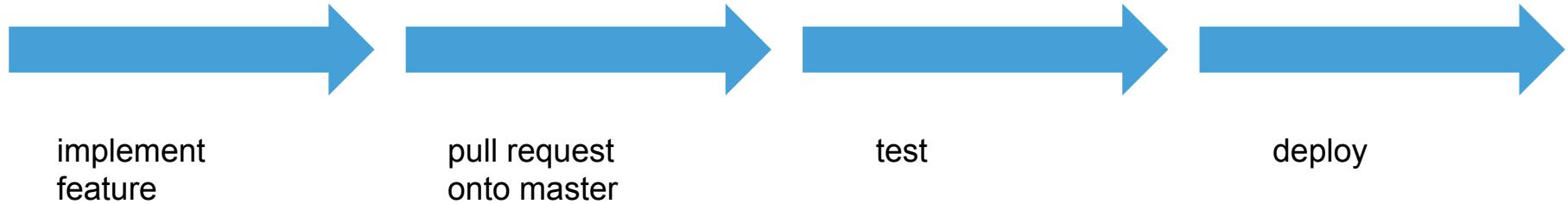


#hackundsoehne

www.hackundsoehne.de



Automate all the things!



#hackundsoehne

www.hackundsoehne.de



fast iterations,
immediate
feedback, chance
to immediately
f***k things up



#hackundsoehne

www.hackundsoehne.de



Demo time!



#hackundsoehne

www.hackundsoehne.de



Thank You



#hackundsoehne

www.hackundsoehne.de



Stay in touch



#hackundsoehne



www.hackundsoehne.de



/hackundsoehne



info@hackundsoehne.de



#hackundsoehne



/hackundsoehne



community.hackundsoehne.de

www.hackundsoehne.de





Hosting Platforms



#hackundsoehne

www.hackundsoehne.de





Da hosting platform.



#hackundsoehne

www.hackundsoehne.de



Free



Firebase

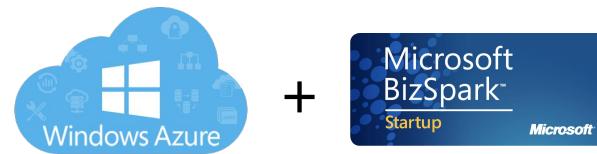
GitHub Pages



#hackundsoehne

Fast & Cheap

HOST1PLUS



Google Cloud Platform

www.hackundsoehne.de

Unboxing: Digital Ocean



#hackundsoehne

www.hackundsoehne.de





Domains & SSL Certificates

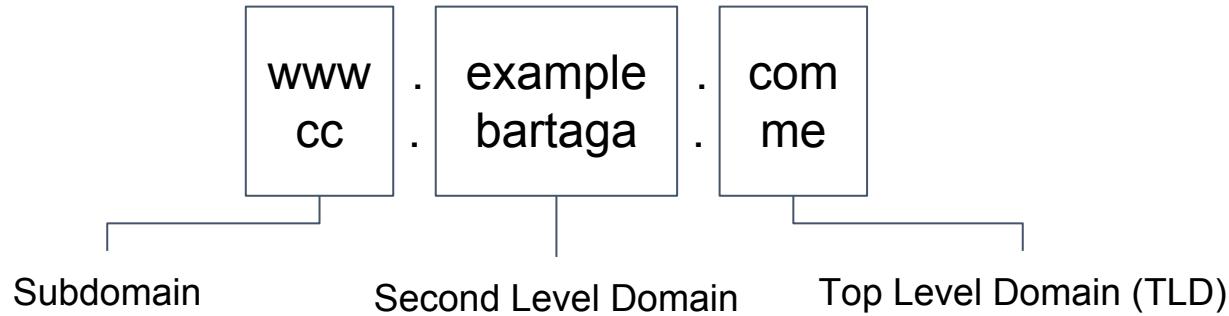


#hackundsoehne

www.hackundsoehne.de



Domain Name



→ Resolve the name to an IP

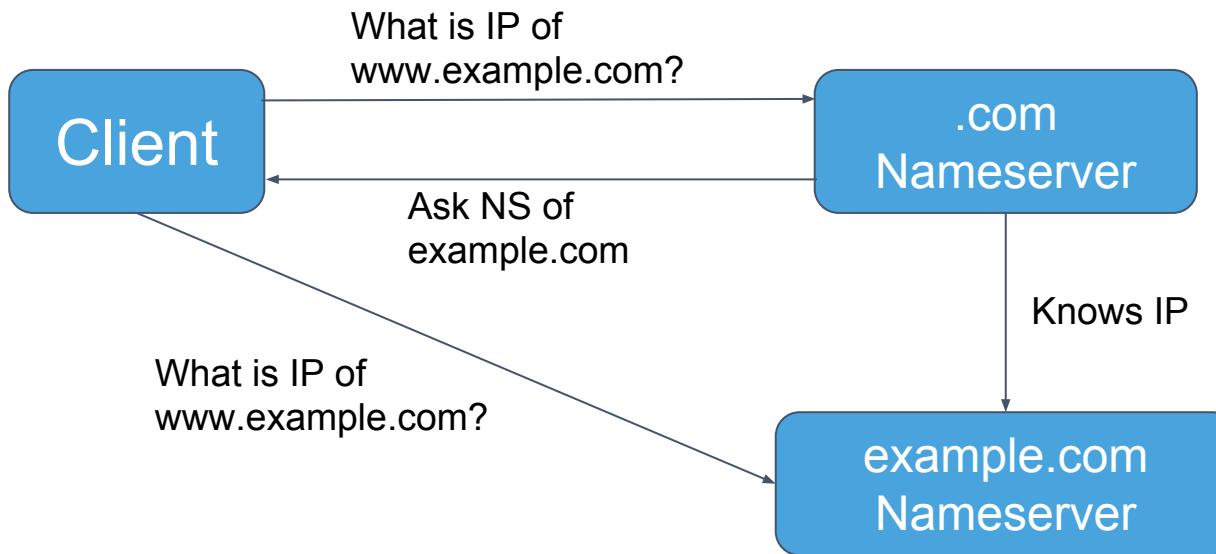


#hackundsoehne

www.hackundsoehne.de



Domain Name System - DNS



DNS Records

- **A:** Resolve name to IPv4 address
- **AAAA:** Same as above for IPv6
- **CNAME:** Alias from one name to another
- **MX:** Mail Exchange
- **NS:** Nameserver



#hackundsoehne

www.hackundsoehne.de



Domain Registrars



#hackundsoehne

www.hackundsoehne.de



Domain: bartaga.me

| Type | Host | Value | TTL | |
|--------------|------|--|-----------|---|
| A Record | cc | 138.68.82.53 | Automatic |  |
| CNAME Record | @ |  .myfritz.net. | 30 min |  |
| CNAME Record | home | bartaga.me. | 30 min |  |
| CNAME Record | ocds | bartaga.me. | 30 min |  |
| CNAME Record | www | bartaga.me. | 30 min |  |



#hackundsoehne

www.hackundsoehne.de

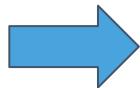


Dynamic DNS

Want a way to access hosts in local network



Usually don't have fixed public IP at home



Dynamically update the A record



#hackundsoehne

www.hackundsoehne.de



Typical Setup of DDNS

1. Get credentials from DDNS provider
2. Setup by either:
 - a. Entering DDNS credentials in router
 - b. Installing and setting up software client on a host
3. Forward/open ports to hosts behind the router
(e.g. Server, RaspberryPi)

Note: FRITZ!Box includes DDNS with free service “MyFritz!”.

Domains similar to 0d6abb0kftx9zj68.myfritz.net



#hackundsoehne

www.hackundsoehne.de



SSL/TLS

Secure Socket Layer/Transport Layer Security



#hackundsoehne

www.hackundsoehne.de



Secure connection is a must!

Unsecured connections allow to:

- Steal Login/Session
- Personal data
- Manipulate data

Current browsers give warnings or block device APIs for unsecured connections!

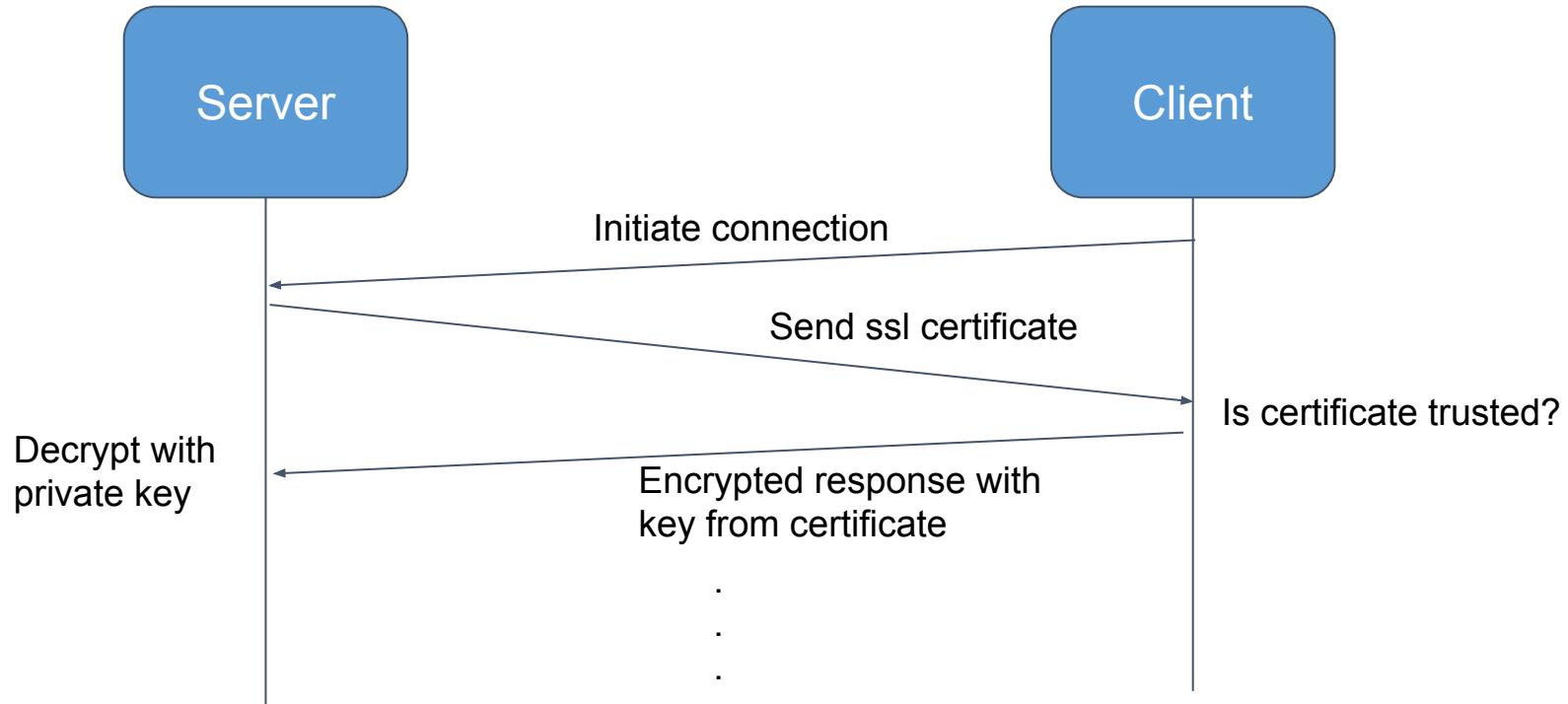


#hackundsoehne

www.hackundsoehne.de



SSL/TLS enables secure connection

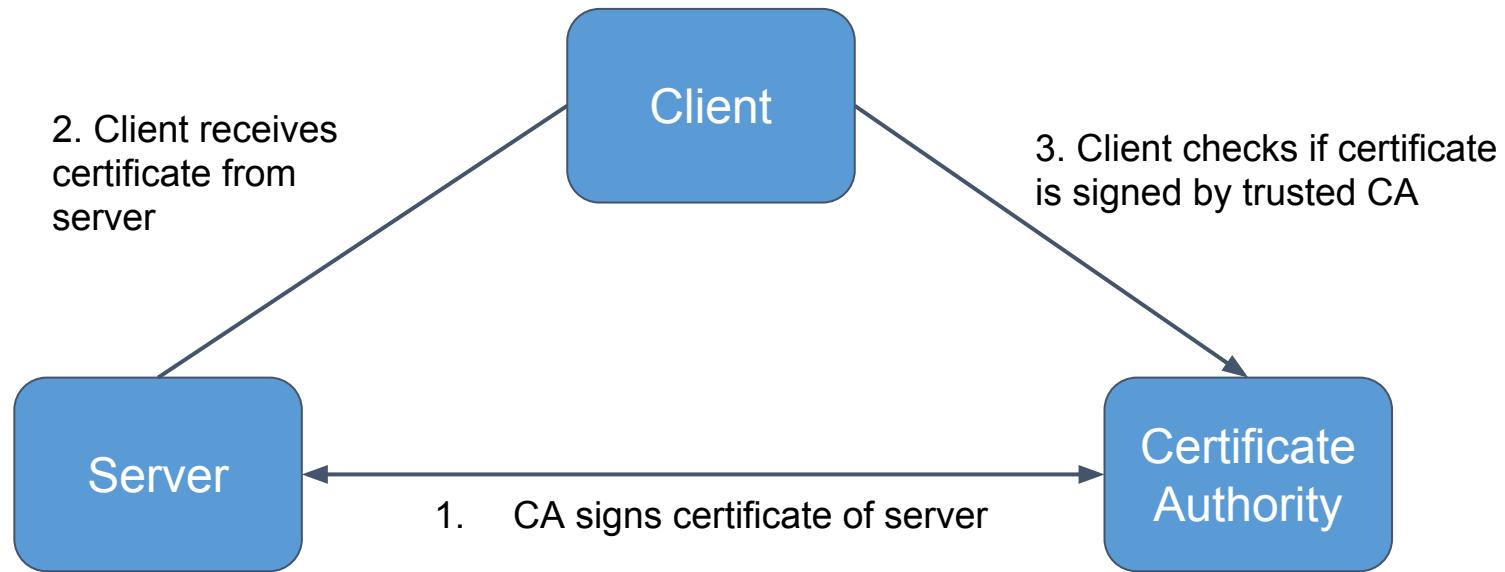


#hackundsoehne

www.hackundsoehne.de



CA gives trusted certificate



SSL Certificate in Practice

Excerpt from Nginx webserver config:

```
1 server {  
2     server_name home.bartaga.me;  
3     listen 443 ssl;  
4  
5     ssl_certificate      /etc/letsencrypt/live/home.bartaga.me/fullchain.pem;  
6     ssl_certificate_key  /etc/letsencrypt/live/home.bartaga.me/privkey.pem;  
7 }
```

Path to public keychain file

Path to private key file



#hackundsoehne

www.hackundsoehne.de



Free SSL Certificates



Let's Encrypt



#hackundsoehne

www.hackundsoehne.de



Thank You



#hackundsoehne

www.hackundsoehne.de



Stay in touch



#hackundsoehne



www.hackundsoehne.de



/hackundsoehne



info@hackundsoehne.de



#hackundsoehne



/hackundsoehne



community.hackundsoehne.de

www.hackundsoehne.de

