

Auführen eines PHP-Scripts auf Google Appspot

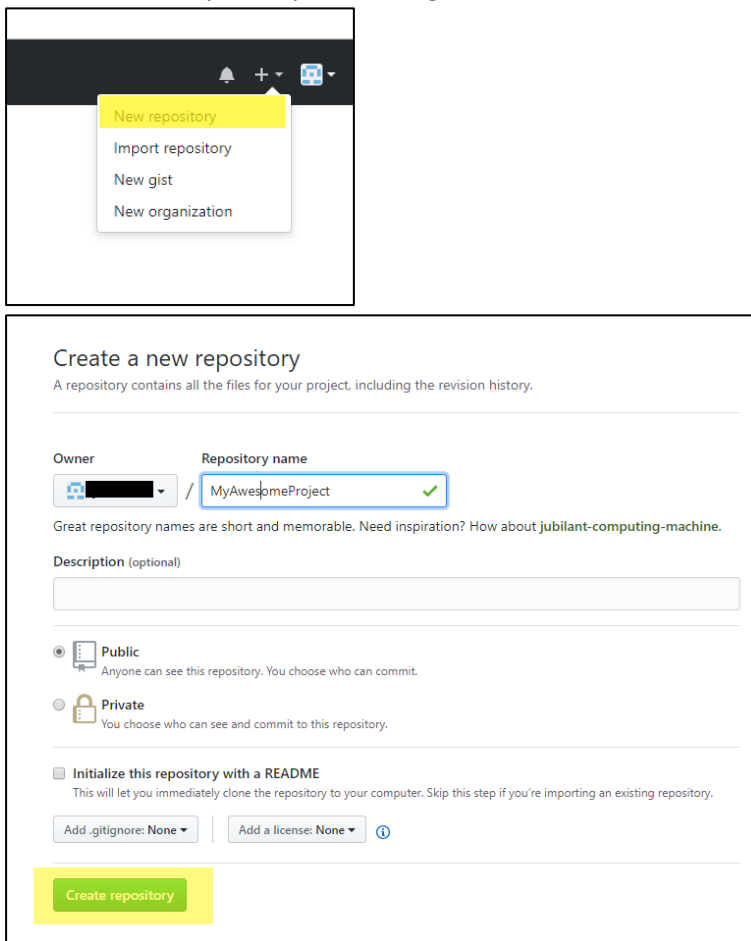
Voraussetzungen

- Google Account
- GitHub Account
- Git Grundlagen (aus der ersten mdse Vorlesung)

Teil 1

Im nachfolgenden möchten wir ein GitHub Repository anlegen, mit den Dateien, die wir später auf der Google Cloud Platform ausführen möchten

1. Legt ein neues Git Repository auf www.github.com an:



Create a new repository
A repository contains all the files for your project, including the revision history.

Owner **Repository name**
[User Avatar] / MyAwesomeProject ✓

Great repository names are short and memorable. Need inspiration? How about [jubilant-computing-machine](#).

Description (optional)
[Text Box]

☒ **Public**
Anyone can see this repository. You choose who can commit.

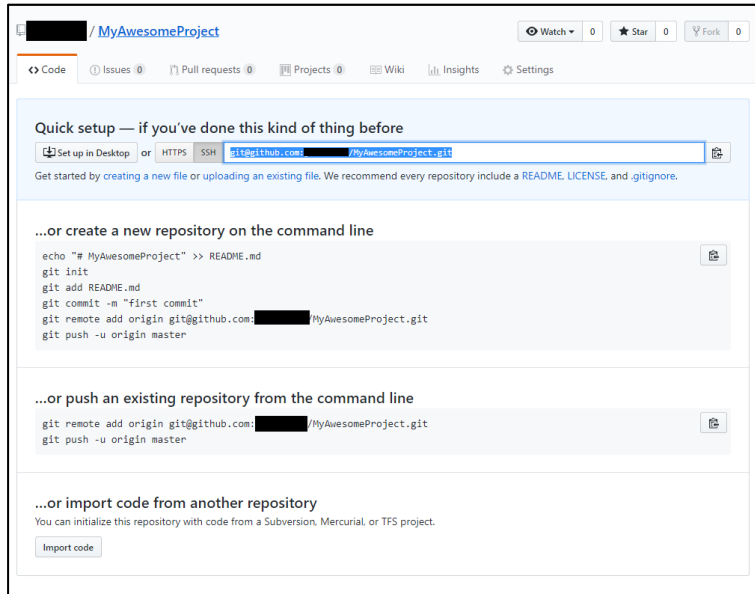
☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** Add a license: **None** ⓘ

Create repository

2. Clont das Repository auf euren Computer



Im Terminal:

```
git clone git@github.com:MyName/MyAwesomeProject.git
```

3. Wechselt in das Repository und legt die Dateien index.php und app.yaml an

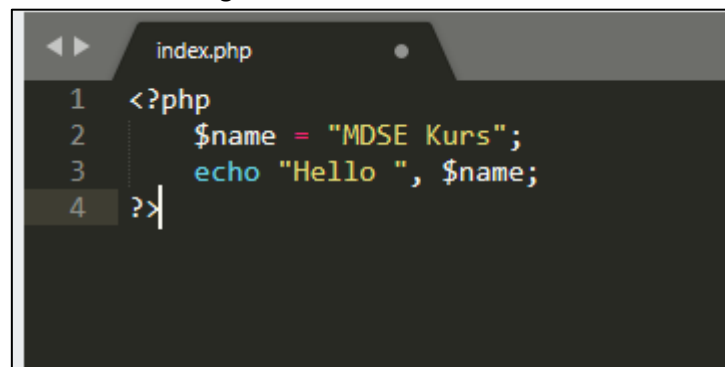
3.1. `touch index.php`

3.2. `touch app.yaml`

4. bearbeitet nun die Datei index.php

4.1. `nano index.php`

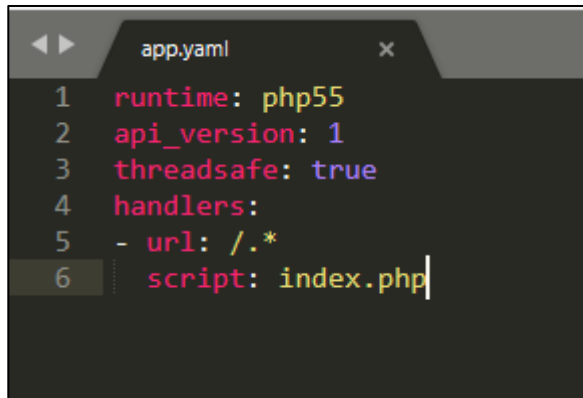
4.2. erstellt ein Hello World Programm in PHP



5. bearbeitet nun die Datei app.yaml

5.1. `nano app.yaml`

5.2. index.php muss indexiert werden, außerdem muss für Google die PHP Version erkennbar sein



```
1 runtime: php55
2 api_version: 1
3 threadsafe: true
4 handlers:
5 - url: /*
6   script: index.php
```

runtime:	Name der Laufzeitumgebung, bei PHP immer php55
api_version:	API Version für die die Anwendung geschrieben wurde (derzeit nur Version 1 verfügbar)
threadsafe:	kann die Anwendung mehrfach gleichzeitig ausgeführt werden falls nicht kann es zu Problemen führen
handlers:	Liste der Dateien und was sie beinhalten
url:	(zu handlers) Ordner der Dateien
script:	(zu handlers) Dateien, die beachtet werden sollen

6. Speichert die Dateien, fügt sie dem Branch hinzu, commitet sie und pusht sie

6.1. Speichern

6.2. `git add -A`

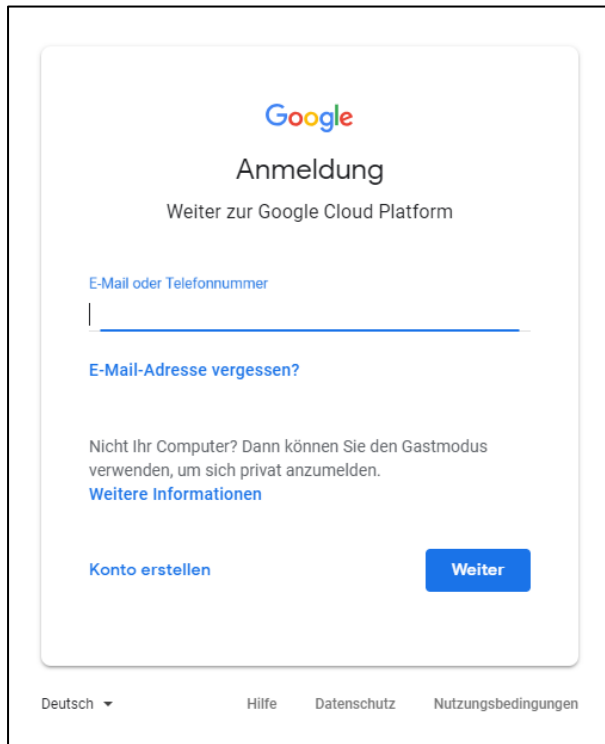
6.3. `git commit -m "added Hello World and Yaml file"`

6.4. `git push origin master`

Teil 2

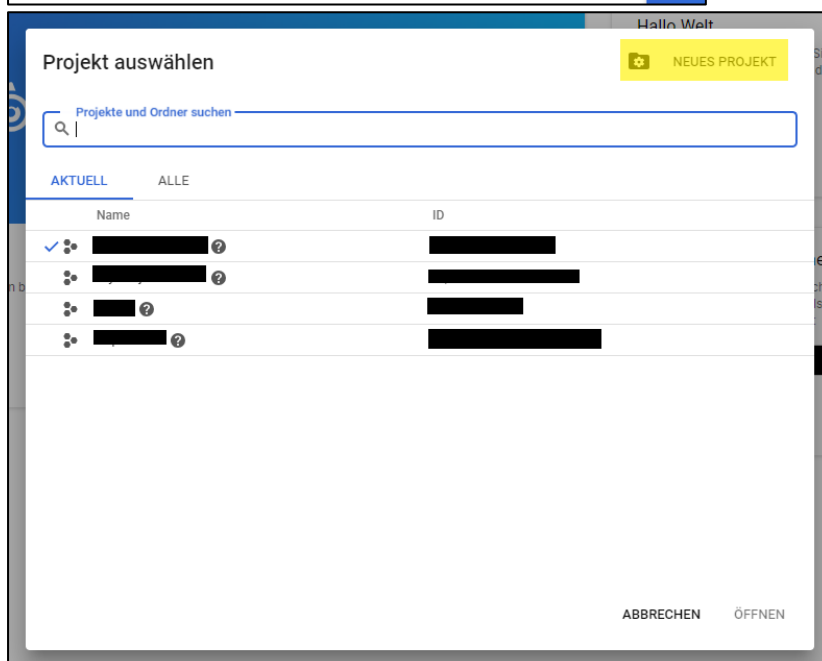
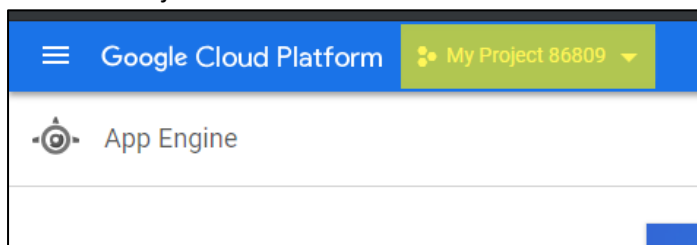
Hier möchten wir das erstellte git Repository nun bei Google importieren und ausführen

1. Geht auf die Seite www.appspot.com
2. Meldet euch mit eurem Google-Account an



The image shows the Google login page for the Google Cloud Platform. At the top is the Google logo, followed by the heading "Anmeldung" and the subtext "Weiter zur Google Cloud Platform". Below this is a text input field labeled "E-Mail oder Telefonnummer". A link "E-Mail-Adresse vergessen?" is positioned below the input field. A paragraph of text states: "Nicht Ihr Computer? Dann können Sie den Gastmodus verwenden, um sich privat anzumelden." followed by a link "Weitere Informationen". At the bottom left is a link "Konto erstellen" and at the bottom right is a blue button labeled "Weiter". The footer contains the language "Deutsch" with a dropdown arrow, and links for "Hilfe", "Datenschutz", and "Nutzungsbedingungen".

3. Klickt auf Projekt erstellen



4. Wählt einen Projektnamen und bestätigt

Google Cloud Platform

Neues Projekt

Warning: In Ihrem Kontingent sind noch 5 projects verfügbar. Fordern Sie eine Erhöhung an oder löschen Sie Projekte.
[Weitere Informationen](#)
[MANAGE QUOTAS](#)

Projektname *
MyAwesomeProject ?

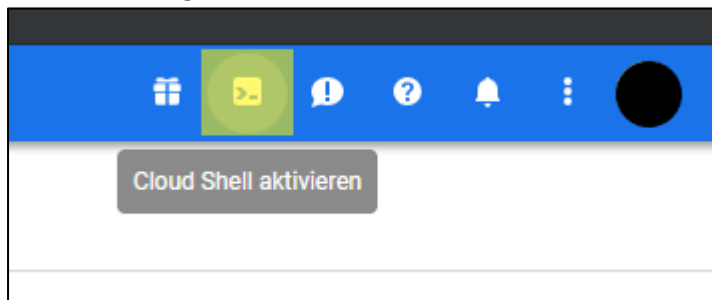
Project ID: myawesomeproject-222715. Sie kann später nicht mehr geändert werden.
[BEARBEITEN](#)

Speicherort *
Keine Organisation [DURCHSUCHEN](#)

Übergeordnete Organisation oder übergeordneter Ordner

[ERSTELLEN](#) [ABBRECHEN](#)

5. Öffnet die Google Cloud Shell



Unten im Fenster öffnet sich nun eine Linux Shell

6. Clont euer Git Repository

Wichtig! Clont per HTML und ohne die .git Endung!

```
(myawesomeproject-222715) x + v
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to myawesomeproject-222715.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
[redacted]@cloudshell:~ (myawesomeproject-222715)$ git clone https://github.com/[redacted]/MyAwesomeProject
```

7. Wechselt in das Git Repository

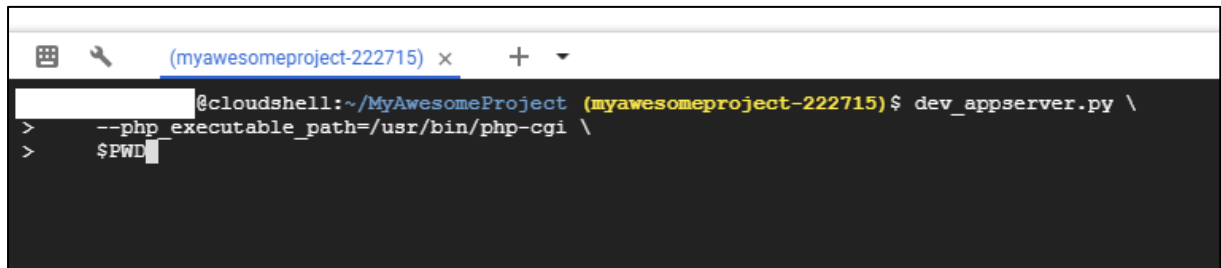
cd MyAwesomeProject

8. Guckt euch ein wenig in der Shell um

- Ihr könnt unter anderem die folgenden Befehle verwenden
 - ls
 - cd
 - nano
 - cat

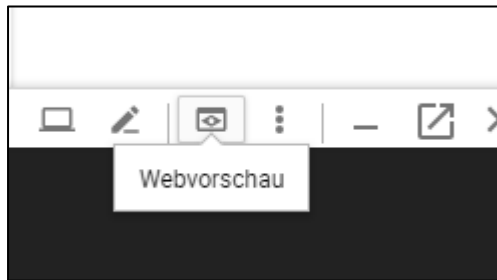
- git pull

9. Startet die Vorschau eurer Anwendung



```
(myawesomeproject-222715) $ dev_appserver.py \
--php_executable_path=/usr/bin/php-cgi \
$PWD
```

10. Wechselt in die Webbrowseransicht



11. Nun solltet ihr in dem neuen Fenster den ausgeführten PHP Code sehen können, der nun auf der Google Cloud Platform läuft. Ihr könnt das Projekt natürlich jederzeit erweitern und durch erneutes Pushen/Pullen den jeweils aktuellen Code ausführen.

Die Google Cloud Platform unterstützt neben PHP auch

- Node.js
- Python
- Go
- Java
- Ruby
- .NET

Wir haben uns hier jedoch für PHP entschieden, da die Konfiguration von PHP am einfachsten nachzuvollziehen ist, bei Java wäre dieser Aufwand deutlich größer.

12. Die Cloud Platform lässt sich außerdem auch mit IntelliJ verbinden, sodass ihr eure Projekte direkt in die Cloud Platform Deployen könnt.