



Codeanywhere – Quick Guide

Praktikum Betriebssysteme (INF4122)

20.02.2018, Jens Haußmann, M. Sc.



Inhalt

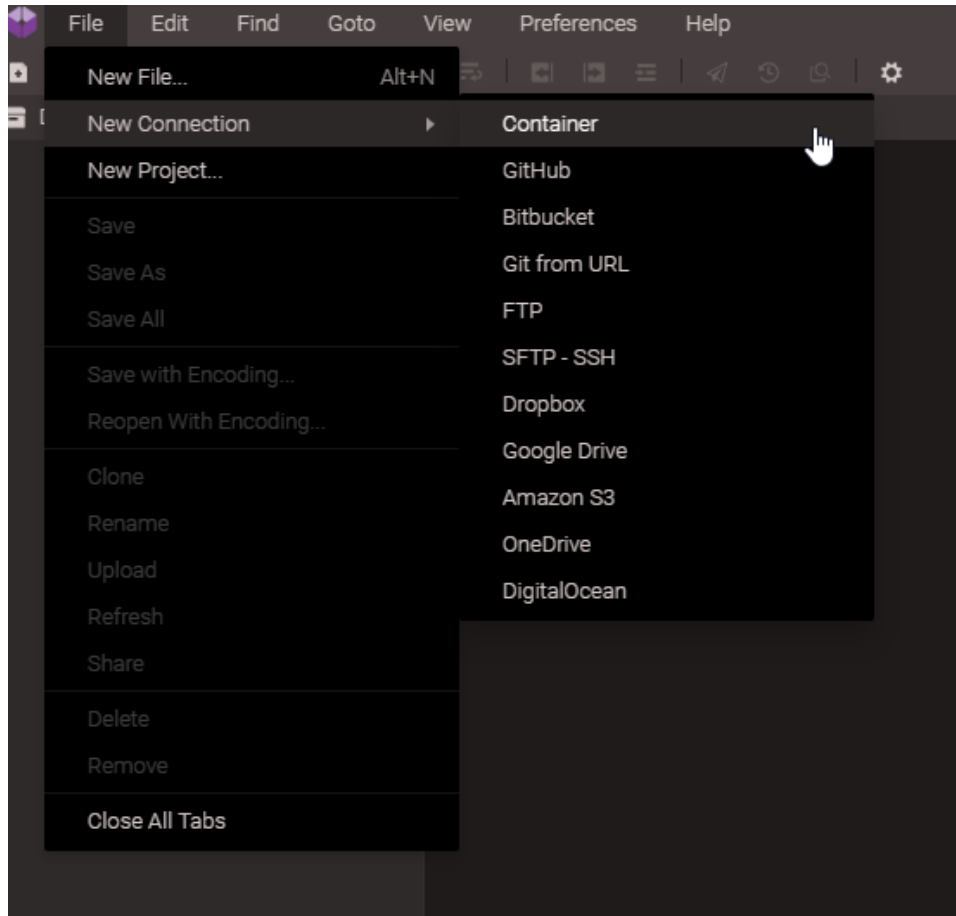
- Übersicht
- Container erstellen
- Web-Terminal aufrufen
- Eigenen SSH-Client nutzen



Übersicht

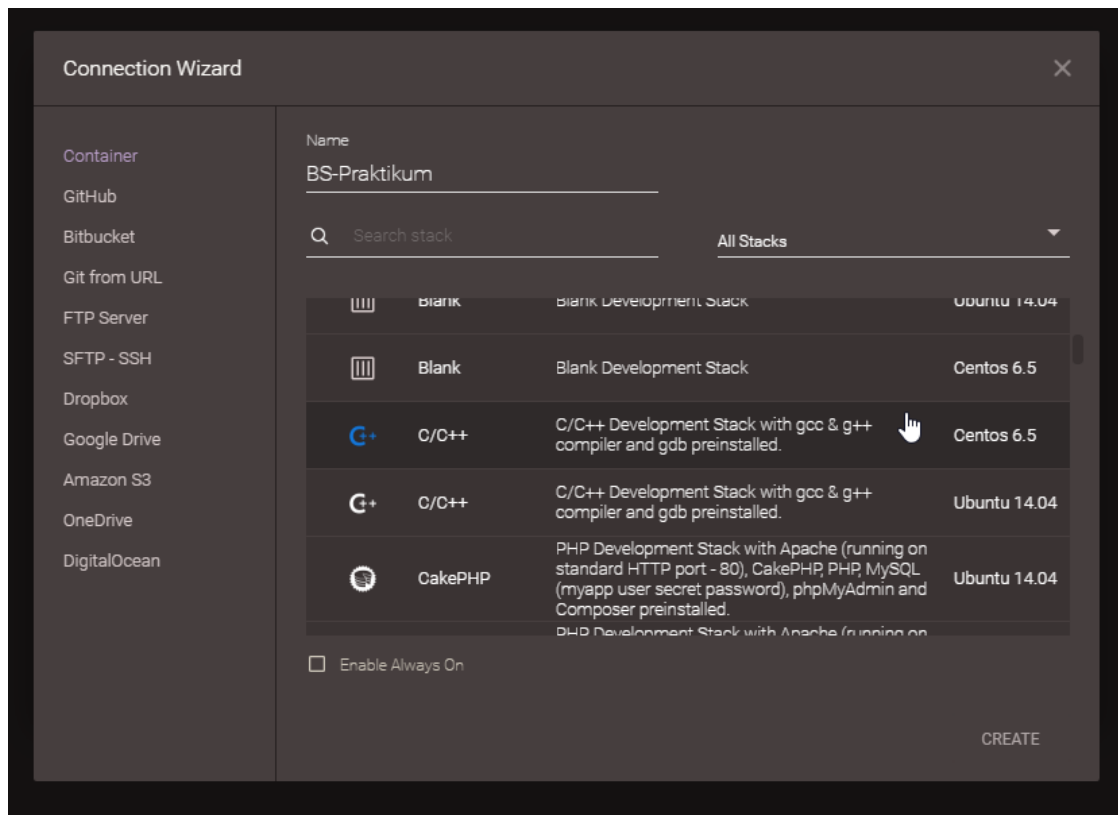
- <https://codeanywhere.com/>
- Pair Programming mit Live-Anzeige möglich
- Frei-Kontingent für 1 Projekt
 - > Beim Wechsel zwischen Java und C Aufgaben muss ein neuer Stack aufgesetzt werden. Aufgaben vor dem Löschen sichern!
- Alle abgegebenen Aufgaben müssen auf den verfügbaren Stacks der Container kompilieren!

Container starten



1) Neuen Container
im Web-Editor
erstellen

Container starten



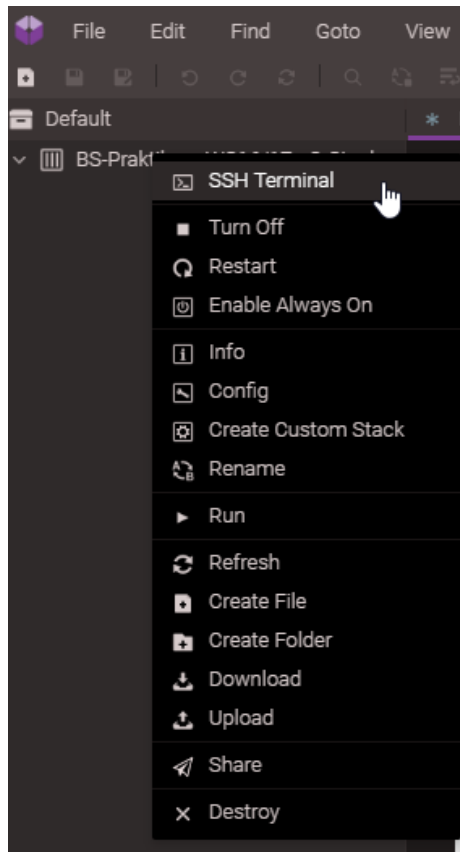
2) Name vergeben und Software-Stack auswählen:

Für „C“-Aufgaben:
C/C++ - Centos 6.5

Für „Java“-Aufgaben:
Java – Centos 6.5



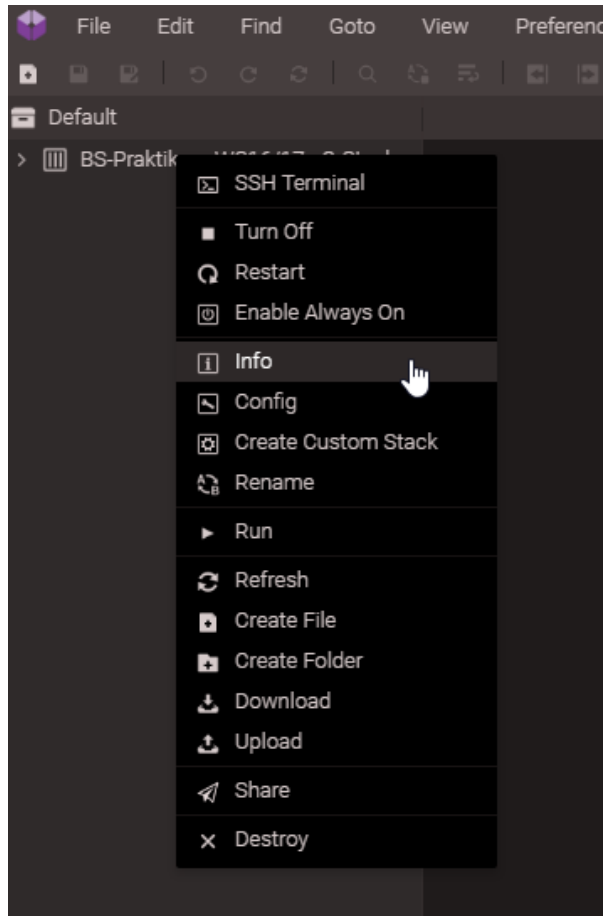
Web-Terminal verwenden



Rechtsklick auf Container -> Terminal



Eigenen SSH-Client verwenden



1) Rechtsklick auf Container -> Info



Eigenen SSH-Client verwenden

BS-Praktikum WS16/17 - C-Stack Container

C/C++ Development Stack with gcc & g++ compiler and gdb preinstalled.

This Codeanywhere Container comes with:

- 2GB of Disc Storage
- 256MB RAM (+ 512 MB swap)
- ~~SSH access~~
- SSH access on `host16.codeanyhost.com:49762`
- Access to all HTTP and Websocket ports

The operating system running on this Container is Centos 6.5 (64 bit). CentOS uses yum package manager. You can read more here: [yum](#)

To access an application running on your Container use the following link (ports 1024-9999 available):

```
| http://BS-Praktikum-WS16-17-C-Stack-gufiyusol715904.codeanyapp.com
```

To access your application over HTTPS, make sure your application is running on port 3000 and use the following link:

```
| https://BS-Praktikum-WS16-17-C-Stack-gufiyusol715904.codeanyapp.com
```

If the port is blocked by your firewall you can connect through the standard HTTP port: (replace XX with port you have specified in your app)

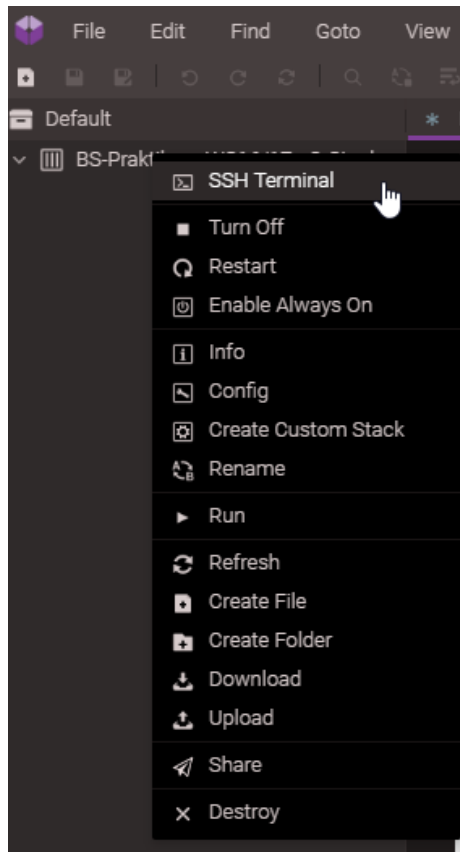
```
| http://port-XX.BS-Praktikum-WS16-17-C-Stack-gufiyusol715904.codeanyapp.com
```

2) IP und Port
notieren

Info:
Wenn keine
IP/Port
angezeigt
werden, dann
das Fenster
nochmal neu
öffnen.



Eigenen SSH-Client verwenden



2) Rechtsklick auf Container -> Terminal



Eigenen SSH-Client verwenden

The screenshot shows a VS Code interface with a terminal window. The terminal prompt is `[cabox@box-codeanywhere workspace]$` and the command `less ../.ssh/id_rsa` is being entered. The terminal title bar indicates the workspace is `BS-Praktikum WS16/17 - C-Stack`.

3) Folgendes Kommando ausführen:
less ../.ssh/id_rsa



Eigenen SSH-Client verwenden

4) Angezeigten Key **komplett** kopieren...

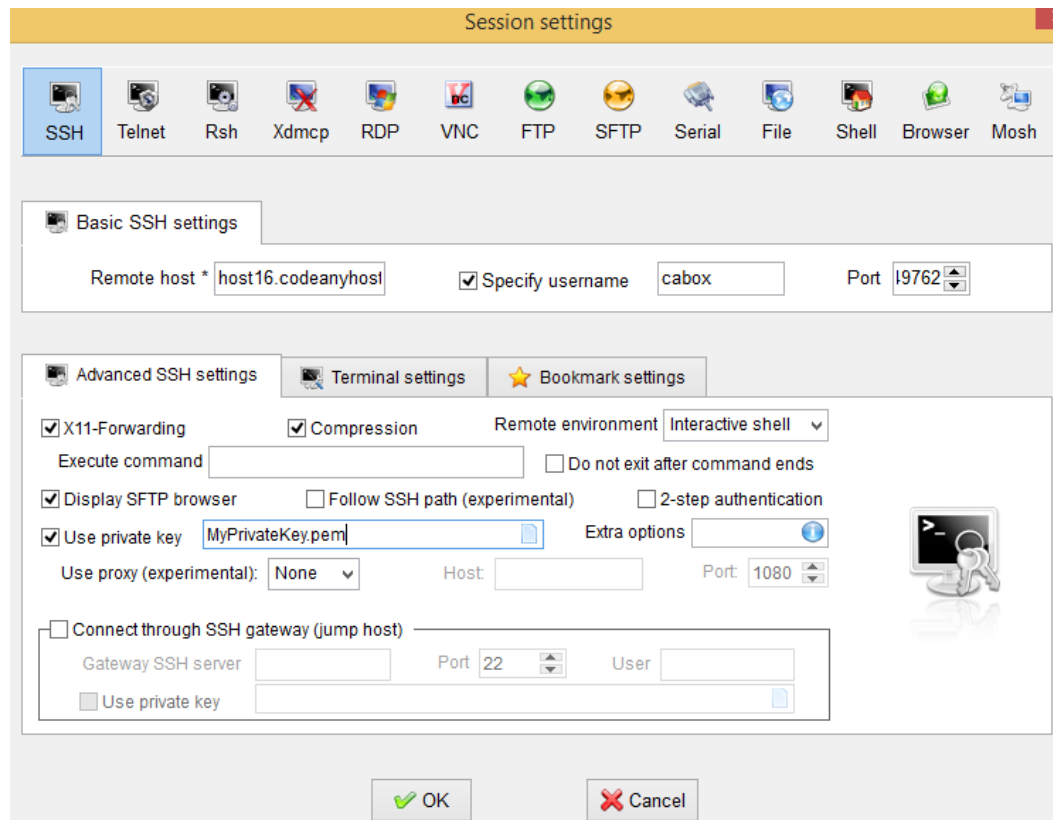
```
-----BEGIN RSA PRIVATE KEY-  
XXX  
XXX  
XXX  
-----END RSA PRIVATE KEY-----
```

5) ... und auf dem eigenen Computer als Textdatei speichern:

myPrivateKey.pem



Eigenen SSH-Client verwenden



6) In bevorzugtem SSH-Client, z.B. MobaXterm, neue SSH-Verbindung mit folgenden Daten erzeugen:

Remote host:

Zuvor notierte IP

Username:

cabox

Port:

Zuvor notierter Port

Private Key:

myPrivateKey.pem

ACHTUNG! IP und Port können sich bei jedem Login ändern!