

Jenkins protokoll

A__04 Jenkins Müller Protokoll

Arbeitsaufwand: 4h

Installation

Jenkins wurde mit docker-compose bereitgestellt. Im *Dockerfile* ist die Container Konfiguration und in *docker-compose.yml* die Deployment Konfiguration

docker-comose.yml

```
version: '3.3'
```

```
services:
```

```
  a04:
```

```
    build:
```

```
      context: .
```

```
    ports:
```

```
      - '8080:8080'
```

```
    container_name: A_04_Jenkins
```

Dockerfile

```
from jenkins/jenkins:lts
```

```
USER root
```

```
RUN apt-get update -y
```

```
RUN apt-get install python3 python3-pip -y
```

```
RUN pip3 install pylint
```

```
RUN apt-get install python3-nose python3-coverage -y
```

```
USER jenkins
```

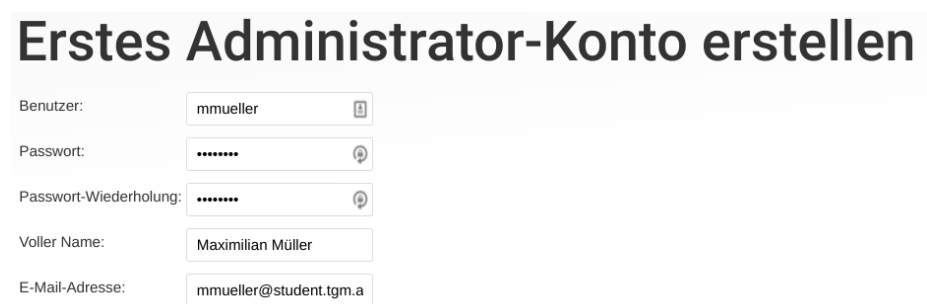
```
EXPOSE 8080
```

initial config

In der Konsole wird das Administrator Passwort angezeigt, mit dem man unter localhost:8080 die Konfiguration fortsetzen kann.

Bei der Frage welche Pakete installiert werden sollen, wurde die Option der vorgeschlagenen Pakete gewählt.

Danach wird ein Administratorkonto erstellt, wo nach einem Benutzernamen, Passwort, Vollen Namen und e-mail gefragt wird. Das kann aber auch übersprungen werden.



The screenshot shows a web form titled "Erstes Administrator-Konto erstellen". It contains five input fields with labels on the left: "Benutzer:" (containing "mmueller"), "Passwort:" (masked with dots), "Passwort-Wiederholung:" (masked with dots), "Voller Name:" (containing "Maximilian Müller"), and "E-Mail-Adresse:" (containing "mmueller@student.tgm.a"). Each input field has a small icon to its right, likely for password strength or validation.

Figure 1: Admin Konto

some

Auf der Weboberfläche kann man unter “**Jenkins verwalten**” “**Plugins verwalten**” auswählen und dort auf den Reiter **Verfügbar** alle verfügbaren Plugins sehen.

Um die Plugins “**Violations**” und “**Cobertura**” zu installieren muss der jeweilige Name des Plugins unter *Filter* eingegeben werden. Das Plugin wird mit einem Klick auf die Checkbox ausgewählt, der Button Installation und Neustart installiert dann alle ausgewählten Plugins.

Unter dem Reiter Installiert sind alle installierten Plugins aufgelistet, zusammen mit einer Option jenes zu Deinstallieren.

Repository

Für die Übung wurde ein Repository(https://github.com/mmueller-tgm/A_04_2018/) auf Github erstellt.

Filter:

Aktualisierungen

Verfügbar

Installiert

Erweiterte Einstellungen

Installieren	Name	Version
.NET-Entwicklung		
<input type="checkbox"/>	CCM Plug-in This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
<input type="checkbox"/>	FxCop Runner plugin	1.1
<input type="checkbox"/>	MSBuild Plugin Allows using MSBuild to build .NET projects	1.29
<input type="checkbox"/>	MSTest plugin Generates test reports for MSTest.	0.23
<input type="checkbox"/>	MSTestRunner plugin	1.3.0
<input type="checkbox"/>	NAnt Plugin	1.4.3
<input type="checkbox"/>	PowerShell plugin	1.3
<input type="checkbox"/>	Violation Comments to Bitbucket Server Plugin Finds violations reported by code analyzers and comments Bitbucket Server (or Stash) pull requests (or commits) with them.	1.64
<input type="checkbox"/>	Visual Studio Code Metrics Plugin	1.7
<input type="checkbox"/>	VSTest Runner plugin	1.0.5
<input type="checkbox"/>	Start Windocks Containers Allows for the creation and start of Windocks Containers	1.4
<input type="checkbox"/>	change-assembly-version-plugin	1.5.1
Agenten-Start und -Steuerung		
<input type="checkbox"/>	Amazon EC2 Container Service plugin	

Figure 2: plugins

Jenkins weitere config

Um Jenkins mit git zu verbinden muss bei Jenkins ein Neues Projekt erstellt werden.

Geben Sie einen Element-Namen an

Bruch

» Pflichtfeld

- "Free Style"-Softwareprojekt bauen**
Dieses Profil ist das meistgenutzte in Jenkins. Jenkins baut Ihr Projekt, wobei Sie universell jedes SCM System mit jedem Build-Verfahren kombinieren können. Dieses Profil ist nicht nur auf das Bauen von Software beschränkt, sondern kann darüber hinaus auch für weitere Anwendungsgebiete verwendet werden.
- Maven-Projekt**
Dieses Profil baut ein Maven-Projekt. Jenkins wertet dabei Ihre POM Dateien aus und reduziert damit den Konfigurationsaufwand ganz erheblich.
- Pipeline**
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.
- Multikonfigurationsprojekt bauen**
Dieses Profil eignet sich sehr gut für Projekte mit zahlreichen Konfigurationen, die etwa in unterschiedlichen Umgebungen getestet oder plattformspezifisch gebaut werden müssen.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Figure 3: new project

Build

Um automatisch zu Builden, müsste ich meine Ports Exposed und meine IP Adresse bei GitHub Angeben, dies ist eine Sicherheitslücke und wurde deswegen nicht gemacht. In folgendem Bild ist das Feld, in dem ich meine URL eintragen würde.

Es hat nicht funktioniert das Bruch Beispiel zu builden, da er das Paket nicht gefunden hat, der Errorlog von Jenkins ist allerdings vorhanden.

Started by user Maximilian Müller

Building in workspace /var/jenkins_home/workspace/Bruch

```
> git rev-parse --is-inside-work-tree # timeout=10
```

Fetching changes from the remote Git repository

```
> git config remote.origin.url https://github.com/mmueeller-tgm/A_04_2018 # timeout=10
```

Fetching upstream changes from https://github.com/mmueeller-tgm/A_04_2018

```
> git --version # timeout=10
```

```
> git fetch --tags --progress https://github.com/mmueeller-tgm/A_04_2018 +refs/heads/*:refs/heads/*
```

Services / Manage Jenkins (Git plugin)

Test service

Install Notes

- Requires [Git Plugin v1.1.18](#), released 2012-04-27, and the "Poll SCM" build trigger needs to be enabled. (Though you can have it poll very infrequently, I recommend something like `H * / 3 * * *`)
- "Jenkins Url" is the base URL of your [Jenkins](#) server. For example: `http://ci.jenkins-ci.org/`. We will hit `/git/notifyCommit` under this URL. (See [the Git plugin wiki page](#) for more details.)

Details

[Jenkins](#) is a popular continuous integration server.

If you're using the standard [Jenkins Git plugin](#) to poll & check out your repository, you can quickly and easily switch to a push model using this service.

It will send a request to your Jenkins instance telling it about the repositories and branches that changed. Jenkins will then poll the repository and build if needed. See [push notification from repository](#) on the Jenkins wiki for information.

Jenkins url

☒ **Active**

We will run this service when an event is triggered.

Update service

Delete service

Figure 4: webhook

```

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision b93d9bc088a9f8ca85621353a3c9ee45863c0ee8 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b93d9bc088a9f8ca85621353a3c9ee45863c0ee8
Commit message: ":cry:"
> git rev-list --no-walk d3b5332a291b8d3060b9f02fae767bcb90dcf75b # timeout=10
[Bruch] $ /bin/sh -xe /tmp/jenkins5725257645905561998.sh
+ ls -la
total 72
drwxr-xr-x 1 jenkins jenkins 194 Jan 26 16:24 .
drwxr-xr-x 1 jenkins jenkins 10 Jan 26 16:10 ..
-rw-r--r-- 1 jenkins jenkins 5405 Jan 26 16:20 .coverage
drwxr-xr-x 1 jenkins jenkins 136 Jan 26 16:24 .git
-rw-r--r-- 1 jenkins jenkins 2 Jan 26 16:16 .pylintrc
-rw-r--r-- 1 jenkins jenkins 9361 Jan 26 16:10 Bruch.py
-rw-r--r-- 1 jenkins jenkins 35064 Jan 26 16:10 LICENSE
-rw-r--r-- 1 jenkins jenkins 12 Jan 26 16:10 README.md
-rw-r--r-- 1 jenkins jenkins 211 Jan 26 16:10 __init__.py
drwxr-xr-x 1 jenkins jenkins 132 Jan 26 16:16 __pycache__
-rw-r--r-- 1 jenkins jenkins 9361 Jan 26 16:24 bruch.py
drwxr-xr-x 1 jenkins jenkins 44 Jan 26 16:16 docs
-rw-r--r-- 1 jenkins jenkins 1701 Jan 26 16:20 nosetests.xml
drwxr-xr-x 1 jenkins jenkins 336 Jan 26 16:16 test
+ nosetests3 --with-xunit --all-modules --traverse-namespaces --with-coverage --cover-inclusi
E
=====
ERROR: Failure: ImportError (No module named 'bruch')
-----
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/nose/failure.py", line 39, in runTest
    raise self.exc_val.with_traceback(self.tb)
  File "/usr/lib/python3/dist-packages/nose/loader.py", line 418, in loadTestsFromName
    addr.filename, addr.module)
  File "/usr/lib/python3/dist-packages/nose/importer.py", line 47, in importFromPath
    return self.importFromDir(dir_path, fqname)
  File "/usr/lib/python3/dist-packages/nose/importer.py", line 94, in importFromDir
    mod = load_module(part_fqname, fh, filename, desc)
  File "/usr/lib/python3.5/imp.py", line 244, in load_module
    return load_package(name, filename)
  File "/usr/lib/python3.5/imp.py", line 216, in load_package
    return _load(spec)
  File "<frozen importlib._bootstrap>", line 693, in _load
  File "<frozen importlib._bootstrap>", line 673, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 673, in exec_module
  File "<frozen importlib._bootstrap>", line 222, in _call_with_frames_removed

```

```

File "/var/jenkins_home/workspace/Bruch/__init__.py", line 2, in <module>
    from bruch.Brch import *          # if this is'nt there sphinx fails to find the class even
ImportError: No module named 'bruch'

```

Name	Stmts	Miss	Cover	Miss
<hr/>				
/usr/lib/python3/dist-packages/nose/case.py	205	152	26%	5-28
/usr/lib/python3/dist-packages/nose/core.py	169	146	14%	5-32
/usr/lib/python3/dist-packages/nose/failure.py	25	15	40%	1-23
/usr/lib/python3/dist-packages/nose/importer.py	101	57	44%	7-29
/usr/lib/python3/dist-packages/nose/loader.py	348	310	11%	11-6
/usr/lib/python3/dist-packages/nose/plugins/base.py	195	191	2%	1-92
/usr/lib/python3/dist-packages/nose/plugins/capture.py	56	41	27%	12-5
/usr/lib/python3/dist-packages/nose/plugins/cover.py	153	149	3%	13-5
/usr/lib/python3/dist-packages/nose/plugins/debug.py	31	27	13%	7-31
/usr/lib/python3/dist-packages/nose/plugins/deprecated.py	16	12	25%	9-31
/usr/lib/python3/dist-packages/nose/plugins/doctests.py	220	212	4%	52-5
/usr/lib/python3/dist-packages/nose/plugins/errorclass.py	64	54	16%	95-5
/usr/lib/python3/dist-packages/nose/plugins/failedetail.py	20	17	15%	9-30
/usr/lib/python3/dist-packages/nose/plugins/isolate.py	34	32	6%	38-5
/usr/lib/python3/dist-packages/nose/plugins/logcapture.py	126	97	23%	18-3
/usr/lib/python3/dist-packages/nose/plugins/manager.py	265	216	18%	51-8
/usr/lib/python3/dist-packages/nose/plugins/multiprocess.py	490	480	2%	96-2
/usr/lib/python3/dist-packages/nose/plugins/prof.py	93	89	4%	13-5
/usr/lib/python3/dist-packages/nose/plugins/skip.py	22	18	18%	9-53
/usr/lib/python3/dist-packages/nose/plugins/testid.py	135	123	9%	95-5
/usr/lib/python3/dist-packages/nose/plugins/xunit.py	164	94	43%	42-6
/usr/lib/python3/dist-packages/nose/proxy.py	108	66	39%	18-2
/usr/lib/python3/dist-packages/nose/pyversion.py	132	126	5%	4-30
/usr/lib/python3/dist-packages/nose/result.py	131	98	25%	11-3
/usr/lib/python3/dist-packages/nose/selector.py	134	112	16%	9-34
/usr/lib/python3/dist-packages/nose/suite.py	396	238	40%	12-5
/usr/lib/python3/dist-packages/nose/util.py	351	294	16%	3-13
/usr/lib/python3/dist-packages/pkg_resources/_vendor/six.py	444	442	1%	23-5
/usr/lib/python3/dist-packages/pkg_resources/extern/__init__.py	35	32	9%	1-23
__init__.py	2	0	100%	
<hr/>				
TOTAL	4665	3940	16%	
<hr/>				

Ran 1 test in 0.002s

FAILED (errors=1)

Build step 'Execute shell' marked build as failure

Finished: FAILURE