Short Note

docker Commands need sudo most of the time

Normaly docker-machine and docker-compose don't require sudo, but there may be exceptions

The CLI commands are the same on every device / system

The commands used here worked on Windows 10 and Zorin 12.3 (Ubuntu derivat) successfully

Simple Demo Applicaton

```python
@app.route("/")
def mainpage():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>no redis available</i>"

    ltime = time.asctime(time.localtime(time.time()))
    html = "<h3>Hello {name}!</h3>" \
        "<b>Hostname:</b> {hostname}<br/>" \
                "<b>OS:</b> {system} <br/>"\
        "<b>Visits:</b> {visits}<br/>" \
                "<b>Local Time:</b> {time}"
    return html.format(name=os.getenv("NAME",
"world"),system=os.getenv("OS", "Not Windows"),
hostname=socket.gethostname(), visits=visits, time=ltime)
```

```python
@app.route("/reset")
def resetCounter():
    try:
        value = redis.getset("counter", 0)
        return "Reset Done"
    except RedisError:
        value = "Reset not possible"

    return value
```

Dockerfile

#official Python runtime
FROM python:3.6-slim

#working directory is /app
WORKDIR /app

#copy everything from current directory to /app
ADD . /app

#install needed requirements
RUN pip install -r requirements.txt

#Make port 8080 available to outside
EXPOSE 8080

#Run myapp
CMD ["python", "myapp.py"]

Dockerfile is whitespace sensitive

FROM: The base image

WORKDIR: the directory to work in, commands are executed here

ADD: copy into the container, even URLs oder archives

COPY: just copy into container

RUN: to do setup, creates new layers

EXPOSE: ports that should be accessible from outside

ENTRYPOINT: Command when the container starts

CMD: Command when the container starts

LABEL: add meta data

ENV: set environment variable

VOLUME: create mount points

ARG: variables that can be set at build time

SHELL, HEALTHCHECK, STOPSIGNAL, ONBUILD, USER

Docker Terms and Wording

Container: lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it - defined by **Dockerfile**

Docker Compose

```yaml
version: "3"
services:
 web:
  build: .
  deploy:
   replicas: 5
   resources:
    limits:
     cpus: "0.1"
     memory: 50M
   restart_policy:
    condition: on-failure
  ports:
   - "8080:8080"
  volumes:
   - .:/app
  environment:
   - DEBUG=True
 redis:
  image: "redis:alpine"
```

- "Connect" / run multiple container together
- use image or build one
- Volumes: mount from host to container
  hostpath:containerpath

- Start: $ docker-compose up
- Stop: $ docker-compose stop
- Stop and Delete: $ docker-compose down

Docker Terms and Wording

Container: lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it - defined by **Dockerfile**

Service: just "containers in production"; different aspects of a applikation - defined by **docker-compose.yml**

Docker Swarms; Docker Stacks

```yaml
version: "3"
services:
 web:
   image: grimapps/simpleapp:1.0
   deploy:
     replicas: 5
     resources:
       limits:
         cpus: "0.1"
         memory: 50M
     restart_policy:
       condition: on-failure
   ports:
     - "8080:8080"
   networks:
     - webnet
networks:
 webnet:
```

```bash
#Swarm initialisation
$ sudo docker swarm init

# App Deploy
$ sudo docker stack deploy -c
docker-compose.yml myapp

#App Remove
$ sudo docker stack rm myapp

# Leave the swarm
$ sudo docker swarm leave --force
```

Docker Terms and Wording

Container: lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it - defined by **Dockerfile**

Service: just "containers in production"; different aspects of a applikation - defined by **docker-compose.yml**

Swarm: a group of machines that are running Docker and joined into a cluster. A machine in a swarm is a node. There are manager and worker nodes

Stack: group of interrelated services that share dependencies, and can be orchestrated and scaled together

Docker Machine; Create Swarms; Deploy Stacks

$ docker-machine create -d hyperv --hyperv-virtual-switch "myswitch" myvm1

$ docker-machine create -d virtualbox myvm1

$ docker-machine ssh myvm1 "docker swarm init --advertiese-addr <ip-address-myvm1>:2377"

$ docker-machine ssh myvm2 "docker swarm join --token <token> <ip-address-myvm2>:2377"

$ sudo docker stack deploy -c docker-compose.yml myapp

Docker Terms and Wording

Container: lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it - defined by **Dockerfile**

Service: just "containers in production"; different aspects of a applikation - defined by **docker-compose.yml**

Swarm: a group of machines that are running Docker and joined into a cluster. A machine in a swarm is a node. There are manager and worker nodes

Stack: group of interrelated services that share dependencies, and can be orchestrated and scaled together

Machine: (virtual) machine, that runs docker;

Docker Swarms; Docker Stacks

```yaml
visualizer:
  image: dockersamples/visualizer:stable
  ports:
    - "8081:8080"
  volumes:
    - "/var/run/docker.sock:/var/run/docker.sock"
  deploy:
    placement:
      constraints: [node.role == manager]
  networks:
    - webnet
```

```yaml
redis:
  image: redis
  ports:
    - "6379:6379"
  volumes:
    - "/home/docker/data:/data"
  deploy:
    placement:
      constraints: [node.role == manager]
  command: redis-server --appendonly yes
  networks:
    - webnet
```

myvm1
manager
0.963G RAM

● myapp_web
image : simpleapp:1.0@sha256:2917
tag : 1.0@sha256:2917f481f310b450
updated : 8/4 10:11
25b6212b98db44b632931104f7862
state : running

● myapp_web
image : simpleapp:1.0@sha256:2917
tag : 1.0@sha256:2917f481f310b450
updated : 8/4 10:11
c330a296aad5ef9f62d3e07a606669
state : running

● myapp_redis
image : redis:latest@sha256:6b9f93
tag : latest@sha256:6b9f935e89af00
cmd : redis-server,--appendonly,yes
updated : 8/4 10:16
d240f66796c7807c5ac9ecd6e1f5413
state : running

● myapp_visualizer
image : visualizer:stable@sha256:bc
tag : stable@sha256:bc680132f772c
updated : 8/4 10:11
a69f9c4ed8ebdad8ffd675bb624742
state : running

myvm2
worker
0.963G RAM

● myapp_web
image : simpleapp:1.0@sha256:2917
tag : 1.0@sha256:2917f481f310b450
updated : 8/4 10:11
66a7c5900ec69109f6671198841169
state : running

● myapp_web
image : simpleapp:1.0@sha256:2917
tag : 1.0@sha256:2917f481f310b450
updated : 8/4 10:11
f06f7785fd763820c4fd804525b5a7f
state : running

● myapp_web
image : simpleapp:1.0@sha256:2917
tag : 1.0@sha256:2917f481f310b450
updated : 8/4 10:11
6861d6e1556d99960f9d077836f8a9
state : running

```
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> docker-machine create -
-driver azure --azure-subscription-id                              azvm1
Running pre-create checks...
(azvm1) Microsoft Azure: To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the
code D2S8QJJES to authenticate.
(azvm1) Registering subscription to resource provider.  ns="Microsoft.Compute" subs=
"
(azvm1) Registering subscription to resource provider.  ns="Microsoft.Network" subs=
"
(azvm1) Registering subscription to resource provider.  subs=                                    " ns="Microsoft.Storag
e"
(azvm1) Completed machine pre-create checks.
Creating machine...
(azvm1) Querying existing resource group.  name="docker-machine"
(azvm1) Creating resource group.  name="docker-machine" location="westus"
(azvm1) Configuring availability set.  name="docker-machine"
(azvm1) Configuring network security group.  name="azvm1-firewall" location="westus"
(azvm1) Querying if virtual network already exists.  name="docker-machine-vnet" rg="docker-machine" location="westus"
(azvm1) Creating virtual network.  name="docker-machine-vnet" rg="docker-machine" location="westus"
(azvm1) Configuring subnet.  name="docker-machine" vnet="docker-machine-vnet" cidr="192.168.0.0/16"
(azvm1) Creating public IP address.  name="azvm1-ip" static=false
(azvm1) Creating network interface.  name="azvm1-nic"
(azvm1) Creating storage account.  location="westus" sku=Standard_LRS name="vhdsceuhgzihdkxley0k1g42"
(azvm1) Creating virtual machine.  name="azvm1" location="westus" size="Standard_A2" username="docker-user" osImage="can
onical:UbuntuServer:16.04.0-LTS:latest"
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with ubuntu(systemd)...
Installing Docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: C:\Program Files\Doc
ker\Docker\Resources\bin\docker-machine.exe env azvm1
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks>
```
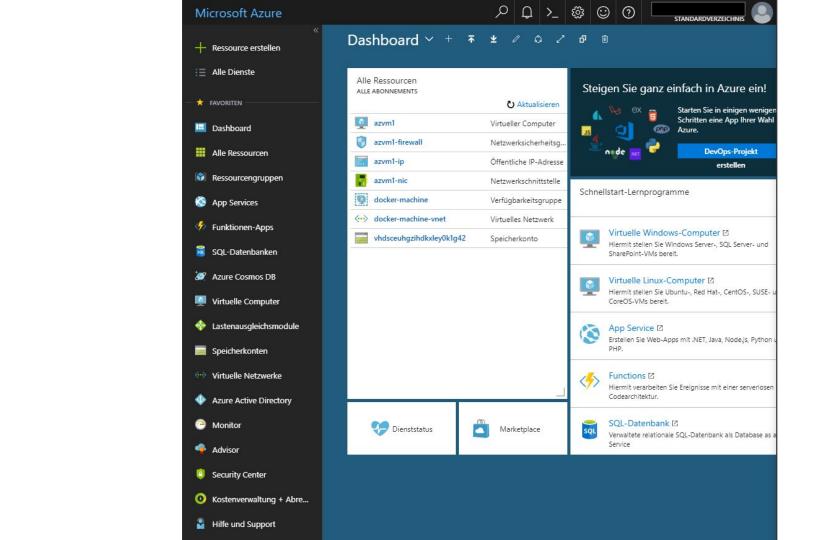
STANDARDVERZEICHNIS

Dashboard

+ Ressource erstellen

≔ Alle Dienste

★ FAVORITEN

▦ Dashboard

▦ Alle Ressourcen

▦ Ressourcengruppen

App Services

⚡ Funktionen-Apps

SQL-Datenbanken

Azure Cosmos DB

Virtuelle Computer

Lastenausgleichsmodule

Speicherkonten

Virtuelle Netzwerke

Azure Active Directory

Monitor

Advisor

Security Center

Kostenverwaltung + Abre...

Hilfe und Support

Alle Ressourcen
ALLE ABONNEMENTS

↻ Aktualisieren

azvm1		Virtueller Computer
azvm1-firewall		Netzwerksicherheitsg...
azvm1-ip		Öffentliche IP-Adresse
azvm1-nic		Netzwerkschnittstelle
docker-machine		Verfügbarkeitsgruppe
docker-machine-vnet		Virtuelles Netzwerk
vhdsceuhgzihdkxley0k1g42		Speicherkonto

Dienststatus

Marketplace

Steigen Sie ganz einfach in Azure ein!

Starten Sie in einigen wenigen Schritten eine App Ihrer Wahl Azure.

DevOps-Projekt
erstellen

Schnellstart-Lernprogramme

Virtuelle Windows-Computer ↗
Hiermit stellen Sie Windows Server-, SQL Server- und SharePoint-VMs bereit.

Virtuelle Linux-Computer ↗
Hiermit stellen Sie Ubuntu-, Red Hat-, CentOS-, SUSE- u CoreOS-VMs bereit.

App Service ↗
Erstellen Sie Web-Apps mit .NET, Java, Node.js, Python u PHP.

Functions ↗
Hiermit verarbeiten Sie Ereignisse mit einer serverlosen Codearchitektur.

SQL-Datenbank ↗
Verwaltete relationale SQL-Datenbank als Database as a Service

```
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> docker-machine env azvm1
$Env:DOCKER_TLS_VERIFY = "1"
$Env:DOCKER_HOST = "tcp://13.91.101.69:2376"
$Env:DOCKER_CERT_PATH = "C:\Users\Fabian\.docker\machine\machines\azvm1"
$Env:DOCKER_MACHINE_NAME = "azvm1"
$Env:COMPOSE_CONVERT_WINDOWS_PATHS = "true"
# Run this command to configure your shell:
# & "C:\Program Files\Docker\Docker\Resources\bin\docker-machine.exe" env azvm1 | Invoke-Expression
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> & "C:\Program Files\Docker\Docker\Resources\bin\docker-machine.exe" env azvm1 | Invoke-Expression
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> docker swarm init
Swarm initialized: current node (s25opzg7pwzy1a6fsg23awir9) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3xrlgnr57pw6gcsi2s5p85kxlpyx20a5vjopa3mxwh28a1vd4c-10vmptsk3sg55rtnq1756f844 192.168.0.4:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks>
```

```
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> docker service ps myapp_web
ID              NAME            IMAGE                    NODE        DESIRED STATE       CURRENT STATE           ERROR           PORTS
uwniqtyk35dz    myapp_web.1     grimapps/simpleapp:2.0   azvm1       Running             Running 27 minutes ago
jbmo8g5npabi    myapp_web.2     grimapps/simpleapp:2.0   azvm1       Running             Running 27 minutes ago
h9wtqxqd2ids    myapp_web.3     grimapps/simpleapp:2.0   azvm1       Running             Running 27 minutes ago
cd838exuq570    myapp_web.4     grimapps/simpleapp:2.0   azvm1       Running             Running 27 minutes ago
yus28mrviedi    myapp_web.5     grimapps/simpleapp:2.0   azvm1       Running             Running 27 minutes ago
```

```
PS D:\Programme\Entwicklung\workspaces\workspaceDocker\DockerPlayground\GetStarted_Part5_Stacks> docker service ps myapp_web
ID              NAME            IMAGE                     NODE     DESIRED STATE     CURRENT STATE           ERROR     PORTS
m9tzuinpk8kx    myapp_web.1     grimapps/simpleapp:2.0    lovm1    Running           Running 5 minutes ago
n3ebyjvj62dq    myapp_web.2     grimapps/simpleapp:2.0    azvm1    Running           Running 5 minutes ago
vibwwj2bkohk    myapp_web.3     grimapps/simpleapp:2.0    lovm1    Running           Running 5 minutes ago
bwb6dw971eul    myapp_web.4     grimapps/simpleapp:2.0    lovm1    Running           Running 5 minutes ago
risvdj5o5it9    myapp_web.5     grimapps/simpleapp:2.0    azvm1    Running           Running 5 minutes ago
```

azvm1
manager
3.339G RAM

lovm1
worker
0.963G RAM

● myapp_visualizer
image : visualizer:stable@sha256:bc
tag : stable@sha256:bc680132f772c
updated : 8/4 20:47
cb3fba286e62c2bdb67ff9d082540d
state : running

● myapp_web
image : simpleapp:2.0@sha256:7d3
tag : 2.0@sha256:7d3c7bbccf2fce44
updated : 8/4 20:47
f08456478ea7f53021ed31dd665877
state : running

● myapp_redis
image : redis:latest@sha256:6b9f935
tag : latest@sha256:6b9f935e89af00
cmd : redis-server,--appendonly,yes
updated : 8/4 20:47
34e677baca43a207c0291c8f76b7976
state : running

● myapp_web
image : simpleapp:2.0@sha256:7d3
tag : 2.0@sha256:7d3c7bbccf2fce44
updated : 8/4 20:47
8c4937c08f5dafc68adeaa4ceb9942b
state : running

● myapp_web
image : simpleapp:2.0@sha256:7d3
tag : 2.0@sha256:7d3c7bbccf2fce44
updated : 8/4 20:48
62fbb2a21ad12213ef348c377389cb
state : running

● myapp_web
image : simpleapp:2.0@sha256:7d3
tag : 2.0@sha256:7d3c7bbccf2fce44
updated : 8/4 20:48
6112ec26601cc5fe6b5cbbbc7db55b
state : running

● myapp_web
image : simpleapp:2.0@sha256:7d3
tag : 2.0@sha256:7d3c7bbccf2fce44
updated : 8/4 20:48
3f113d06cb0cf791cf897d4e5b9074f
state : running

Eingangssicherheitsregeln

PRIORITÄT	NAME	PORT	PROTOKOLL	QUELLE	ZIELADRESSE	AKTION	
100	⚠ SSHAllowAny	22	TCP	Alle	Alle	✓ Zulassen	...
101	Port_8080	8080	Alle	Alle	Alle	✓ Zulassen	...
102	Port_8081	8081	Alle	Alle	Alle	✓ Zulassen	...
105	⚠ allowALL	Alle	Alle	Alle	Alle	✓ Zulassen	...
300	DockerAllowAny	2376	TCP	Alle	Alle	✓ Zulassen	...
65000	AllowVnetInBound	Alle	Alle	VirtualNetwork	VirtualNetwork	✓ Zulassen	...
65001	AllowAzureLoadBalancerInBound	Alle	Alle	AzureLoadBalancer	Alle	✓ Zulassen	...
65500	DenyAllInBound	Alle	Alle	Alle	Alle	✗ Ablehnen	...

Ausgangssicherheitsregeln

PRIORITÄT	NAME	PORT	PROTOKOLL	QUELLE	ZIELADRESSE	AKTION	
101	allowALLout	Alle	Alle	Alle	Alle	✓ Zulassen	...
65000	AllowVnetOutBound	Alle	Alle	VirtualNetwork	VirtualNetwork	✓ Zulassen	...
65001	AllowInternetOutBound	Alle	Alle	Alle	Internet	✓ Zulassen	...
65500	DenyAllOutBound	Alle	Alle	Alle	Alle	✗ Ablehnen	...

Öffentlicher Port-Range	Lokale IP-Adresse	Ziel-Port-Range	Protokoll	Löschen
2377-2377	192.168.0.11	2377-2377	TCP/UDP ⌄	☐
7946-7946	192.168.0.11	7946-7946	TCP/UDP ⌄	☐
4789-4789	192.168.0.11	4789-4789	TCP/UDP ⌄	☐
6379-6379	192.168.0.11	6379-6379	TCP/UDP ⌄	☐

Breite Unterstüzung

AWS Elastic Container Service

AWS Elastic Container Service for **Kubernetes**

AWS Elastic Container Registry

Azure Container Instance

Azure Container Service (AKS)

Azure Container Registry

Azure Service Fabric

Azure App Service

Azure Batch

Google **Kubernetes** Engine

Google Container Registry

Google Container Builder

Digital Ocean, Exoscale, OpenStack, Rackspace, IBM Softlayer, Heroku...

Well, that's s***

- load balancing after reboot of a swarm node doesn't work reliable
- Access a Remote-Host, from a different machine (that did not create the machine)
    - manually copy the certificates to the other computer
    - adjust hardcoded paths
    - sometimes small timeouts (using azure)
    - => it was not possible for me
    - => Kubernetes is a Alternative

Well, that's s***

- Compose, Service, Swarm, Stack
    - What is what, What does it do, When use what?
    - confusing for starters
- Sometimes unintuitive naming in the CLI

Weiterführende Interessante Artikel

Debug nodejs ohne Hostinstallation
https://blog.docker.com/2016/07/live-debugging-docker/

Why docker has not the best future
https://chrisshort.net/docker-inc-is-dead/

Docker Swarm / Stack Alternative: Kubernetes
https://kubernetes.io

Ausblick