

MICHAEL JOHANN

# EINFÜHRUNG IN DOCKER

@MALAGANT

WWW.RAILS-EXPERTS.COM



# WAS IST DOCKER?





JAVA VS DOCKER

**JAVA IST:** WRITE ONE, RUN  
EVERYWHERE

**DOCKER IST:** SETUP ONCE, DEPLOY  
EVERYWHERE

CONTAINER PASSEN IMMER





CONTAINER PASSEN IMMER

„Docker is light weight, allows you to port applications across systems and hardware with ease while containing those applications and running them in their own secure sandboxed environments“

- GANDALF DER GRAUE

DOCKER CONTAINER

HABEN IHRE EIGENE  
SANDBOX



EINFACH EINZUSETZEN

CONTAINER SIND  
VERZEICHNISSE UND  
KÖNNEN EINFACH  
KOPIERT WERDEN

→ Labor & Delivery

→ 401 - 429

→ Maternity Waiting

→ Nursery

→ Obstetric Center

→ Post Partum

→ Sleep Disorders Ctr.

← Outpatient Care Ctr.

← 430 - 443

A wide-angle photograph of a sunset over a calm sea. The sky is filled with soft, orange and yellow clouds. Several birds are silhouetted against the bright horizon. In the foreground, the dark blue surface of the water reflects the warm colors of the sky.

**BASIEREN AUF LINUX CONTAINERN (LXC)**

**BENÖTIGEN KEINE VIRTUALISIERUNG INFRASTRUKTUR**

**CONTAINER SIND LEICHTGEWICHTE**

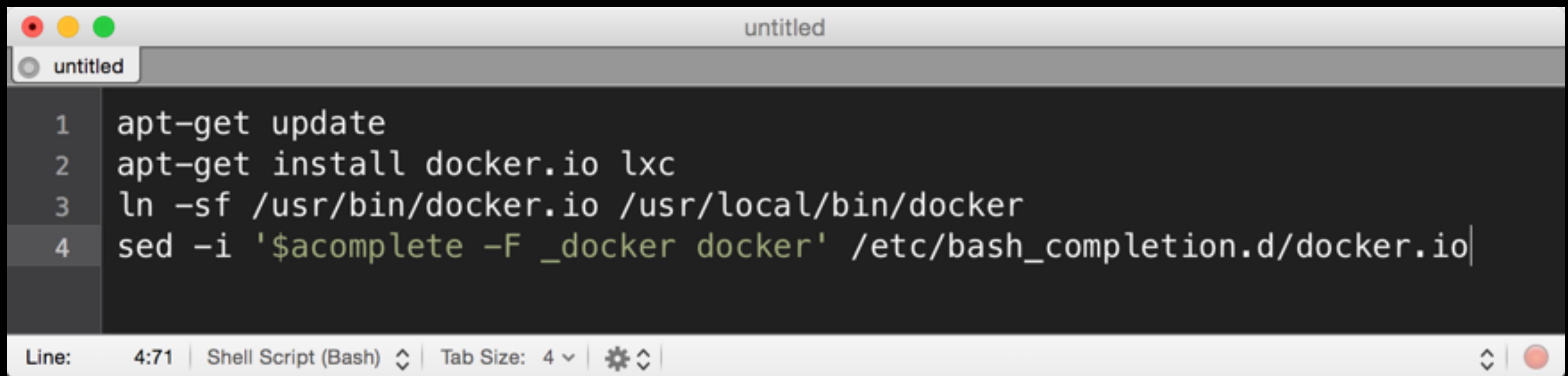


DOCKER CONTAINER SIND  
PORTABEL



INKREMENTELL  
INKREMENTELL  
INKREMENTELL  
INKREMENTELL  
INKREMENTELL

# INSTALLATION - UBUNTU 14.04

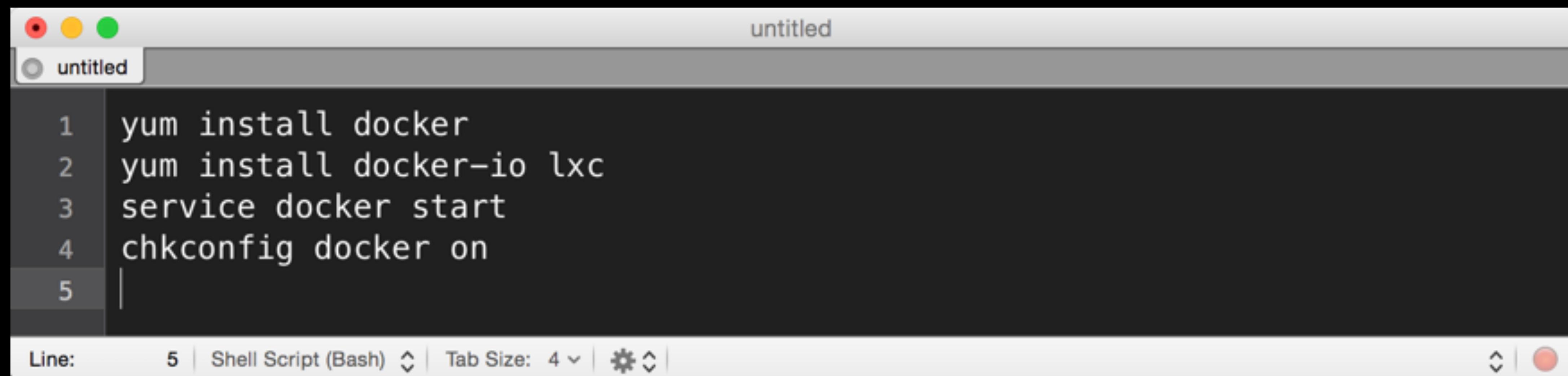


A screenshot of a terminal window titled "untitled". The window contains the following shell script code:

```
1 apt-get update
2 apt-get install docker.io lxc
3 ln -sf /usr/bin/docker.io /usr/local/bin/docker
4 sed -i '$acomplete -F _docker docker' /etc/bash_completion.d/docker.io
```

The terminal window has a dark background and light-colored text. The status bar at the bottom shows "Line: 4:71 | Shell Script (Bash) | Tab Size: 4 |" and includes standard terminal control buttons.

# INSTALLATION - CENTOS 7



A screenshot of a terminal window titled "untitled". The window contains the following shell script code:

```
1 yum install docker
2 yum install docker-io lxc
3 service docker start
4 chkconfig docker on
5 |
```

The terminal interface includes standard OS X-style window controls (red, yellow, green buttons) at the top left. The title bar says "untitled". The tab bar at the bottom shows "Line: 5 | Shell Script (Bash) | Tab Size: 4 |" with a gear icon for settings. A red circular button is visible on the right side of the window.

# INSTALLATION

## MACOS X

```
untitled  
untitled  
1 brew update  
2 brew tap phinze/homebrew-cask  
3 brew install brew-cask  
4 brew cask install virtualbox  
5  
6 brew install boot2docker  
7 boot2docker init  
8 boot2docker up  
  
Line: 5 | Shell Script (Bash) ▾ | Tab Size: 4 ▾ | ⚙ ▾ |
```



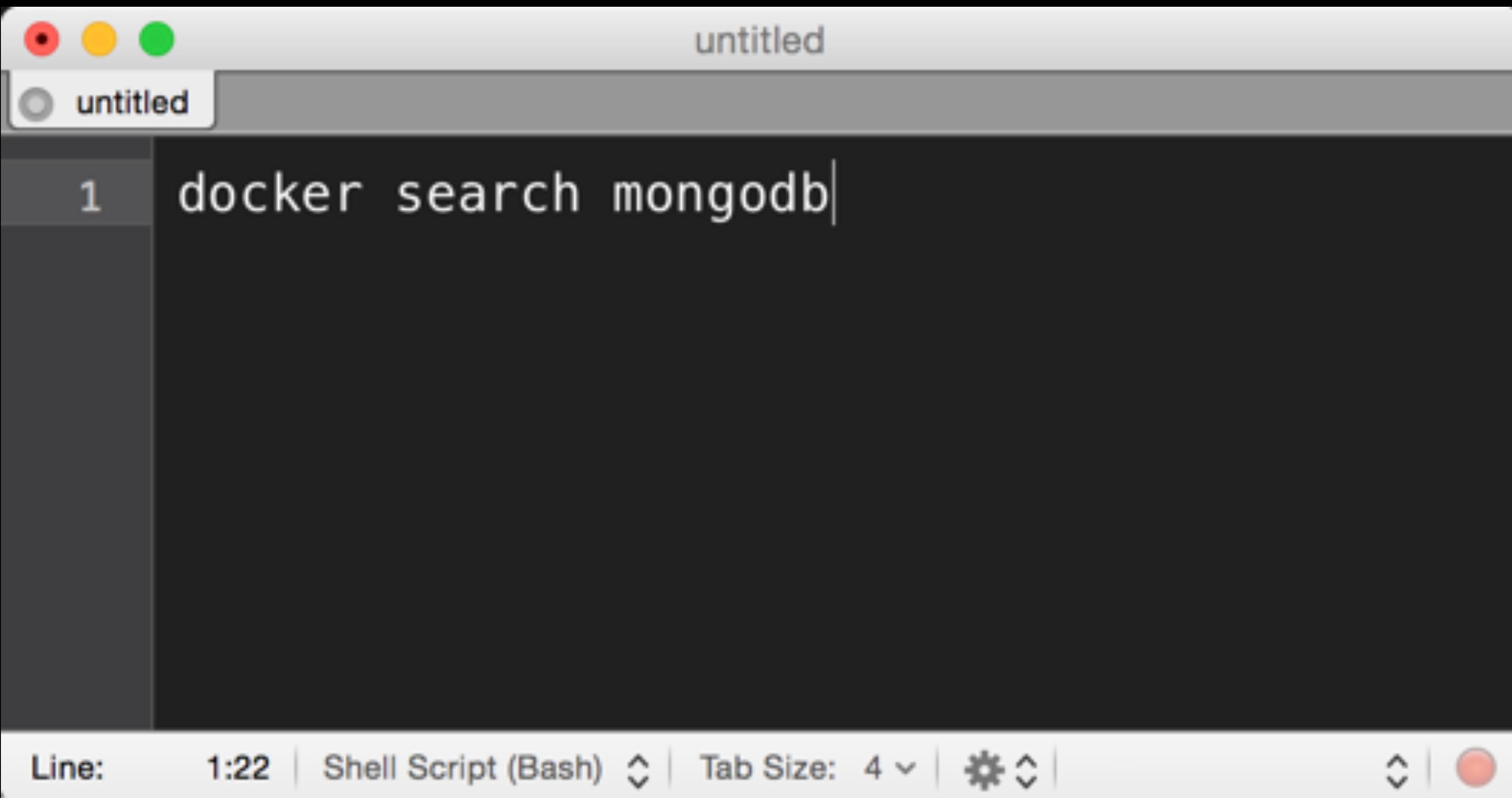
DEMO: DOCKER MACHINE

# DOCKER - DAS TOOL

```
1 mjohann@MBP15 ~ /private/etc/apache2 docker
2 Usage: docker [OPTIONS] COMMAND [arg...]
3   -H=[unix:///var/run/docker.sock]: tcp://host:port to bind/connect to or unix://path/to/socket to use
4
5 A self-sufficient runtime for linux containers.
6
7 Commands:
8   attach      Attach to a running container
9   build       Build an image from a Dockerfile
10  commit      Create a new image from a container's changes
11  cp          Copy files/folders from a container's filesystem to the host path
12  diff        Inspect changes on a container's filesystem
13  events      Get real time events from the server
14  export      Stream the contents of a container as a tar archive
15  history     Show the history of an image
16  images      List images
17  import      Create a new filesystem image from the contents of a tarball
18  info        Display system-wide information
19  inspect     Return low-level information on a container
20  kill        Kill a running container
21  load        Load an image from a tar archive
22  login       Register or log in to a Docker registry server
23  logout      Log out from a Docker registry server
24  logs        Fetch the logs of a container
25  port        Lookup the public-facing port that is NAT-ed to PRIVATE_PORT
26  pause       Pause all processes within a container
27  ps          List containers
28  pull        Pull an image or a repository from a Docker registry server
29  push        Push an image or a repository to a Docker registry server
30  restart     Restart a running container
31  rm          Remove one or more containers
32  rmi         Remove one or more images
33  run          Run a command in a new container
34  save         Save an image to a tar archive
35  search      Search for an image on the Docker Hub
36  start        Start a stopped container
37  stop        Stop a running container
38  tag          Tag an image into a repository
39  top          Lookup the running processes of a container
40  unpause     Unpause a paused container
41  version     Show the Docker version information
42  wait        Block until a container stops, then print its exit code|
```



# SUCHE NACH VORHANDENEN IMAGES



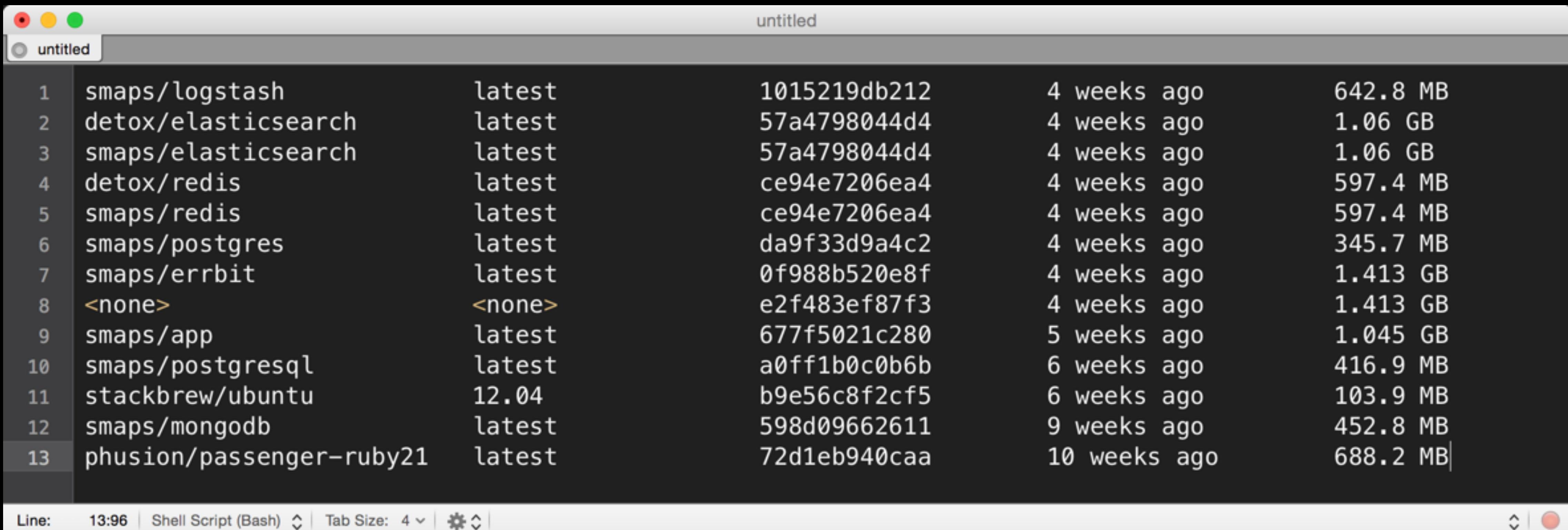
A screenshot of a terminal window titled "untitled". The window has three colored window controls (red, yellow, green) in the top-left corner. The title bar shows the word "untitled". Below the title bar, there is a tab labeled "untitled". The main area of the terminal contains a single line of text: "1 docker search mongodb|". At the bottom of the terminal window, there is a status bar with the following information: "Line: 1:22 | Shell Script (Bash) | Tab Size: 4 | ⚙️ | ⌂ | ⌂".

# EIN IMAGE VON DOCKER LADEN

```
mjohann@MBP15 ➤ /private/etc/apache2 ➤ docker pull ubuntu
Pulling repository ubuntu
463ff6be4238: Pulling dependent layers
195eb90b5349: Pulling dependent layers
822a01ae9a15: Pulling dependent layers
c4ff7513909d: Pulling dependent layers
3db9c44f4520: Download complete
75204fdb260b: Pulling dependent layers
c5881f11ded9: Pulling dependent layers
511136ea3c5a: Download complete
6cfa4d1f33fb: Download complete
af82eb377801: Downloading 23.24 MB/68.7 MB 1m22s
93c381d2c255: Downloading 4.328 MB/39.17 MB 7m6s
bac448df371d: Downloading 4.015 MB/39.93 MB 7m11s
1c9383292a8f: Downloading 5.812 MB/67.48 MB 9m10s
f127542f0b61: Downloading 4.459 MB/40.16 MB 6m54s
3af9d794ad07: Download complete
b7c6da90134e: Download complete
47dd6d11a49f: Pulling metadata
```

# WELCHE IMAGES LIEGEN AUF DEM HOST?

## \$ DOCKER IMAGES

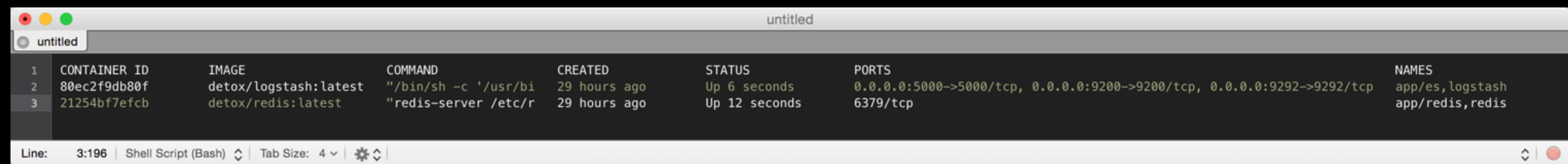


The screenshot shows a terminal window titled "untitled" with the command "\$ DOCKER IMAGES" entered. The output lists 13 Docker images along with their tags, commit IDs, creation times, and sizes. The images are from various repositories like smaps, detox, stackbrew, and phusion. The terminal interface includes standard Mac OS X window controls (red, yellow, green buttons) and a status bar at the bottom.

	Repository	Tag	ID	Created	Size
1	smaps/logstash	latest	1015219db212	4 weeks ago	642.8 MB
2	detox/elasticsearch	latest	57a4798044d4	4 weeks ago	1.06 GB
3	smaps/elasticsearch	latest	57a4798044d4	4 weeks ago	1.06 GB
4	detox/redis	latest	ce94e7206ea4	4 weeks ago	597.4 MB
5	smaps/redis	latest	ce94e7206ea4	4 weeks ago	597.4 MB
6	smaps/postgres	latest	da9f33d9a4c2	4 weeks ago	345.7 MB
7	smaps/errbit	latest	0f988b520e8f	4 weeks ago	1.413 GB
8	<none>	<none>	e2f483ef87f3	4 weeks ago	1.413 GB
9	smaps/app	latest	677f5021c280	5 weeks ago	1.045 GB
10	smaps/postgresql	latest	a0ff1b0c0b6b	6 weeks ago	416.9 MB
11	stackbrew/ubuntu	12.04	b9e56c8f2cf5	6 weeks ago	103.9 MB
12	smaps/mongodb	latest	598d09662611	9 weeks ago	452.8 MB
13	phusion/passenger-ruby21	latest	72d1eb940caa	10 weeks ago	688.2 MB

# WELCHE CONTAINER LAUFEN GERADE?

```
$ DOCKER PS
```

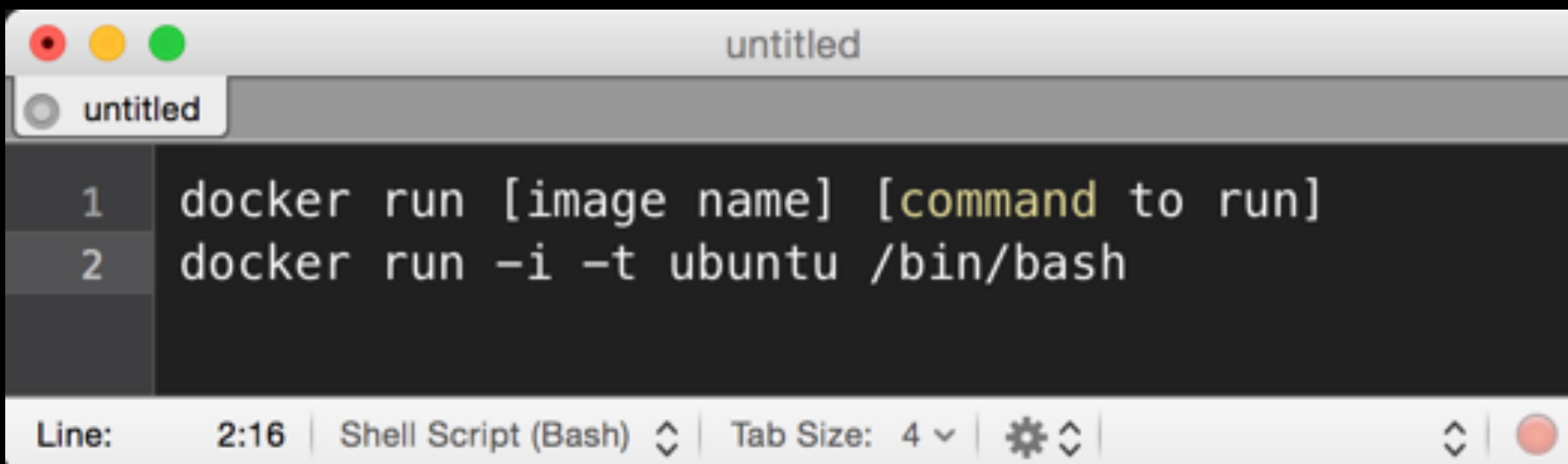


A screenshot of a terminal window titled "untitled". The window shows the output of the command \$ DOCKER PS. The output is as follows:

1	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2	80ec2f9db80f	detox/logstash:latest	"/bin/sh -c '/usr/bi	29 hours ago	Up 6 seconds	0.0.0.0:5000->5000/tcp, 0.0.0.0:9200->9200/tcp, 0.0.0.0:9292->9292/tcp	app/es,logstash
3	21254bf7efcb	detox/redis:latest	"redis-server /etc/r	29 hours ago	Up 12 seconds	6379/tcp	app/redis,redis

The terminal also displays the line number (Line: 3:196), the shell type (Shell Script (Bash)), and tab settings (Tab Size: 4). The status bar at the bottom right shows a red circular icon.

# CONTAINER STARTEN



A screenshot of a macOS terminal window. The window title is "untitled". The terminal has two tabs open, with the first tab, also titled "untitled", being active. The content of the terminal shows two lines of shell script:

```
1 docker run [image name] [command to run]
2 docker run -i -t ubuntu /bin/bash
```

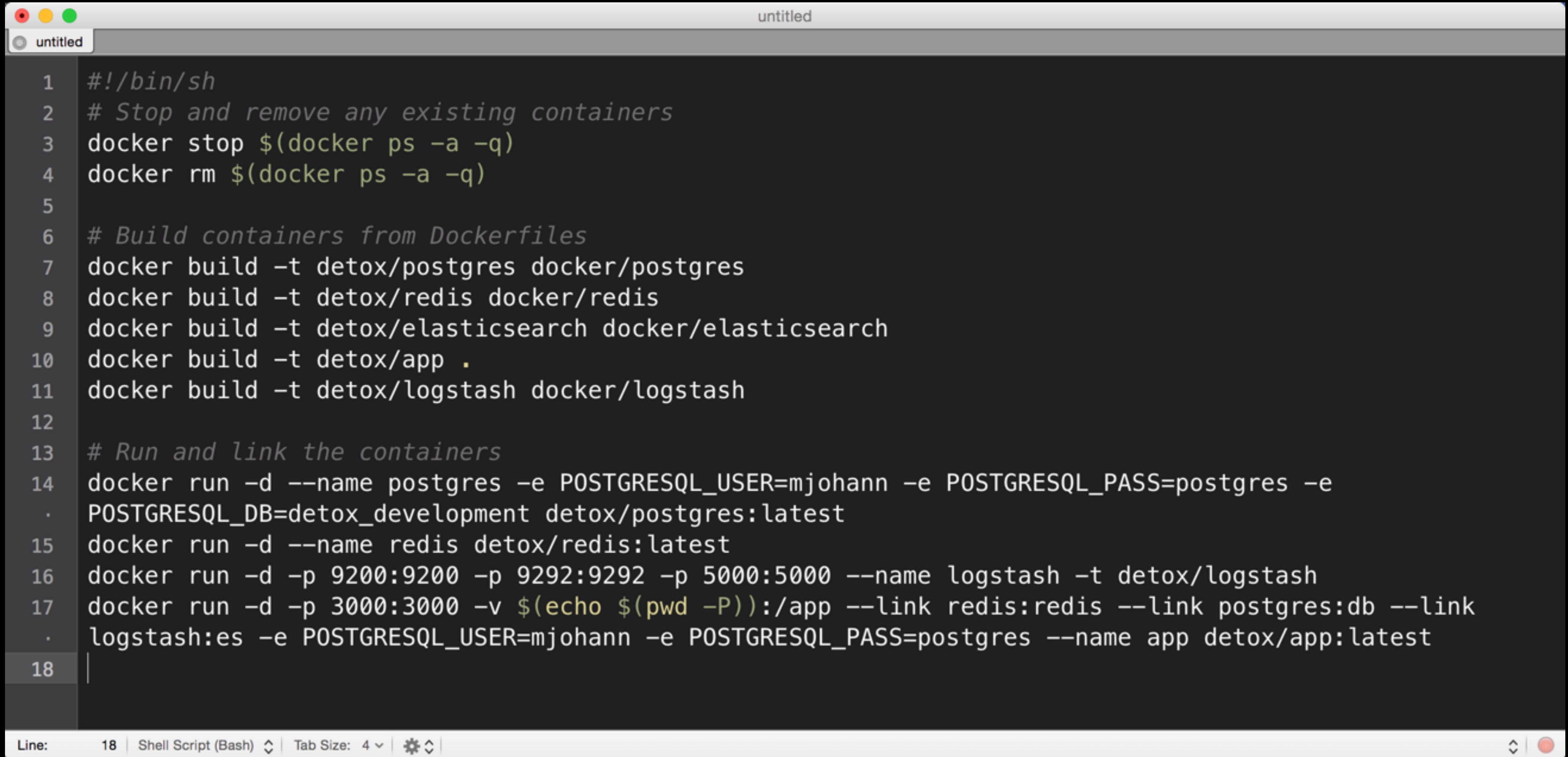
The terminal interface includes standard macOS-style controls (red, yellow, green buttons) at the top left, and a status bar at the bottom with "Line: 2:16", "Shell Script (Bash)", "Tab Size: 4", and other icons.

BESCHREIBUNG VON IMAGES  
DOCKERFILES  
SIND SKRIPTE ZUM  
ERSTELLEN VON  
DOCKER IMAGES



# DEMO: DOCKERFILE

# CONTAINER VERLINKEN



The screenshot shows a terminal window titled "untitled" with a dark theme. It contains a shell script for managing Docker containers. The script starts by stopping and removing existing containers, then builds five new ones: detox/postgres, detox/redis, detox/elasticsearch, detox/app, and detox/logstash. Finally, it runs and links them together, setting environment variables for PostgreSQL and specifying port mappings and links between the containers.

```
#!/bin/sh
# Stop and remove any existing containers
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)

# Build containers from Dockerfiles
docker build -t detox/postgres docker/postgres
docker build -t detox/redis docker/redis
docker build -t detox/elasticsearch docker/elasticsearch
docker build -t detox/app .
docker build -t detox/logstash docker/logstash

# Run and link the containers
docker run -d --name postgres -e POSTGRESQL_USER=mjohann -e POSTGRESQL_PASS=postgres -e POSTGRESQL_DB=detox_development detox/postgres:latest
docker run -d --name redis detox/redis:latest
docker run -d -p 9200:9200 -p 9292:9292 --name logstash -t detox/logstash
docker run -d -p 3000:3000 -v $(echo $(pwd -P)):/app --link redis:redis --link postgres:db --link logstash:es -e POSTGRESQL_USER=mjohann -e POSTGRESQL_PASS=postgres --name app detox/app:latest
```

Line: 18 | Shell Script (Bash) | Tab Size: 4 | ⚙️ | ⌂ | ⏹

# 12 FACTOR APP 1/2

- I. Codebase: One codebase tracked in revision control, many deploys
- II. Dependencies: Explicitly declare and isolate dependencies
- III. Config: Store config in the environment
- IV. Backing Services: Treat backing services as attached resources
- V. Build, release, run: Strictly separate build and run stages
- VI. Processes: Execute the app as one or more stateless processes

# 12 FACTOR APP 2/2

- VII. Port binding: Export services via port binding
- VIII. Concurrency: Scale out via the process model
- IX. Disposability: Maximize robustness with fast startup and graceful shutdown
- X. Dev/prod parity: Keep development, staging, and production as similar as possible
- XI. Logs: Treat logs as event streams
- XII. Admin processes: Run admin/management tasks as one-off processes

# AUSBLICK



# ORCHESTRATION

- **dokku** - Docker powered mini-Heroku. The smallest PaaS implementation you've ever seen.
  - <https://github.com/progrium/dokku>
- **docker-compose**
- **docker-machine**
- **docker swarm**



„ES GIBT NOCH VIEL ZU TUN. ABER DER ANFANG IST GETAN.“

- UNKNOWN

