

# Docker on VDS

Aurelijus Banelis



# Aurelijus Banelis

Software developer  
[aurelijus.banelis.lt](http://aurelijus.banelis.lt)  
[aurelijus@banelis.lt](mailto:aurelijus@banelis.lt)





# Docker on VDS

Aurelijus Banelis





**You will learn**

**Why VDS? Why docker?**

**What is docker?**

**Is it possible? Why not?**

**Dev/prod and configs**

**Testing and debugging**

**Version control and state**

# Why VDS? Why docker?

Wanted new technologies on my old Virtual Dedicated Server

Why VDS?  
Why  
docker?

What is  
docker?




Is it  
possible?  
Why not?



















Dev/prod  
and configs

Testing and  
debugging

Version  
control and  
state



	<a href="#">CentOS 6</a> (64-bit)
	<a href="#">CentOS 7</a> (64-bit)
	<a href="#">Debian 7</a> (64-bit)

- ▼  etc
  - ▶  auginte
  - ▼  httpd
    - ▼  conf.d
      -  10-auginte-sites.conf
      -  20-default-sites.conf
      -  30-auginte-scala-site.conf
      -  40-auginte-php-site.conf
    - ▶  rc.d
    - ▶  ssh
      -  my.cnf
      -  php.ini
  - ▶  home
  - ▶  opt
  - ▶  root
  - ▶  var
  -  .gitignore
  -  README.md

# ⚡ HTTP/2 Test

Verify HTTP/2.0 Support

Online **HTTP/2 test** - Verify if your server or CDN supports HTTP/2.

URL




HTTP/2 Test Result [auginte.com](#)

Yeah! [auginte.com](#) supports **HTTP/2.0**. Share via:  Twitter

ALPN supported.

```
start-dev.sh
start-live.sh
start-stage.sh
```



	<a href="#">CentOS 6</a> (64-bit)
	<a href="#">CentOS 7</a> (64-bit)
	<a href="#">Debian 7</a> (64-bit)

```

  ▾ etc
    ▶ etc/auginte
    ▾ httpd
      ▾ conf.d
        ⚡ 10-auginte-sites.conf
        ⚡ 20-default-sites.conf
        ⚡ 30-auginte-scala-site.conf
        ⚡ 40-auginte-php-site.conf
      ▶ rc.d
      ▶ ssh
        📄 my.cnf
        📄 php.ini
      ▶ home
      ▶ opt
      ▶ root
      ▶ var
      📄 .gitignore
      📄 README.md
  
```

# ⚡ HTTP/2 Test

Verify HTTP/2.0 Support

Online **HTTP/2 test** - Verify if your server or CDN supports HTTP/2.

URL

HTTP/2 Test Result [auginte.com](#)

Yeah! [auginte.com](#) supports **HTTP/2.0**. Share via: [Twitter](#)

ALPN supported.

```
➤ start-dev.sh
```

```
➤ start-live.sh
```

```
➤ start-stage.sh
```





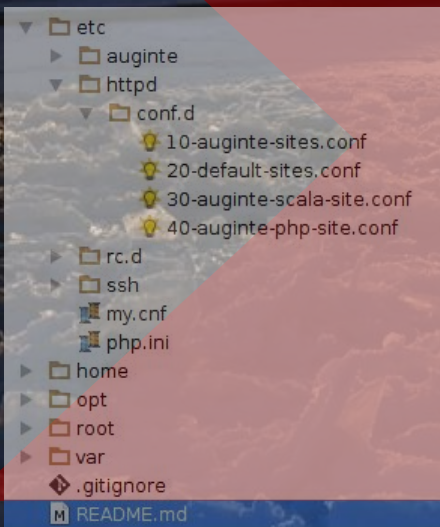
Markdown documentation

MD



Very smart bash script

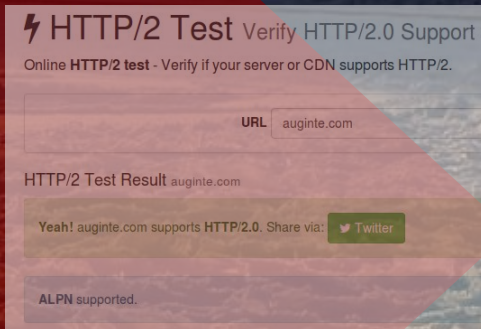
<input type="radio"/>	<a href="#">CentOS 6 (64-bit)</a>
<input checked="" type="radio"/>	<a href="#">CentOS 7 (64-bit)</a>
<input type="radio"/>	<a href="#">Debian 7 (64-bit)</a>



Provision with puppet



Provision with saltstack



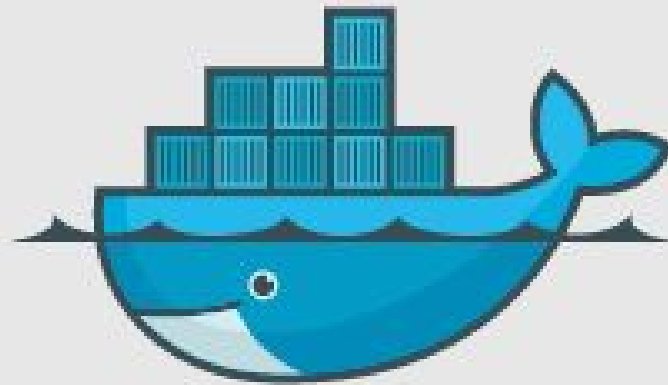


Markdown documentation



- CentOS 6 (64-bit)
- CentOS 7 (64-bit)
- Debian 7 (64-bit)

- etc
  - auginte
  - httpd
    - conf.d
      - 10-auginte-sites.conf
      - 20-default-sites.conf
      - 30-auginte-scala-site.conf
      - 40-auginte-php-site.conf
    - rc.d
    - ssh
      - my.cnf
      - php.ini
    - home
    - opt
    - root
    - var
    - .gitignore
    - README.md



# docker

### ⚡ HTTP/2 Test

Verify HTTP/2.0 Support

Online HTTP/2 test - Verify if your server or CDN supports HTTP/2.

URL:

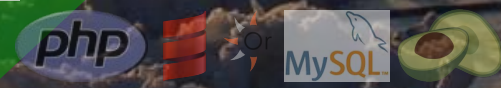
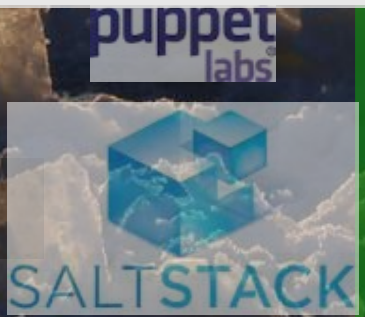
HTTP/2 Test Result: **auginte.com**

Yeah! auginte.com supports HTTP/2.0. Share via: [Twitter](#)

ALPN supported.

```
> start-dev.sh
> start-live.sh
> start-stage.sh
```

Provision with saltstack



# What is docker?

User-friendly containerization tools

Why VDS?  
Why  
docker?

What is  
docker?

Is it  
possible?  
Why not?

Dev/prod  
and configs

Testing and  
debugging

Version  
control and  
state



# What is docker?

BUILD, SHIP, RUN

Docker is the world's leading software containerization platform

Source: <https://www.docker.com/>



# One line to run

BUILD, SHIP, RUN

Docker is the world's leading software containerization platform

```
yum install php56w php56w-opca
```

```
curl -sS https://getcomposer.o
```

```
mv composer.phar /usr/local/bi
```

```
docker-compose up
```



# Isolated apps

CONTAINER ID	IMAGE
87d03fd32b5e	nginx:1.11.1-alpine
dda1f83d5add	aurelijusb/php:7.0.7-fpm
7c1668eeb487	java:8-jre
94a913518c62	java:8-jre
91be2fcad13d	mysql:5.7.12
643f37567fc1	orientdb/orientdb:2.1.5

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
459	458	nginx	S	17044	2%	0	0%	nginx:
458	1	root	S	15944	2%	0	0%	nginx:
467	0	root	S	1520	0%	0	0%	sh
1	0	root	S	1516	0%	0	0%	{run.sh
470	467	root	R	1516	0%	0	0%	top



# Tools: docker-compose

```
$ docker run --name some-nginx -v /some/nginx.conf:  
/etc/nginx/nginx.conf -d -p 8080:80 nginx
```

**docker-compose.yml**

```
services:  
  image: nginx  
  volumes:  
    - ./some/nginx.conf:/etc/nginx/nginx.conf  
  ports:  
    - "8080:80"
```



# Docker on VDS

Is it possible?  
Why not always?

Possible on some VDS, but mine used bad kernel

Why VDS?  
Why  
docker?

What is  
docker?

Is it  
possible?  
Why not?

Dev/prod  
and configs

Testing and  
debugging

Version  
control and  
state

A wide, flat landscape covered in snow and ice under a hazy sky. The foreground is dominated by a vast field of broken, jagged ice chunks, likely from a frozen body of water. The ice is illuminated by a low sun, creating a warm, golden glow. In the background, a line of bare trees is visible against a pale, overcast sky. The overall scene is desolate and cold.

# One line to run

**docker-compose up**





# One line to run

**docker-compose up**

**If docker is installed**

\*Docker-compose is a docker wrapper, written in Python. There was less problems installing it, than docker itself



# Old distribution?

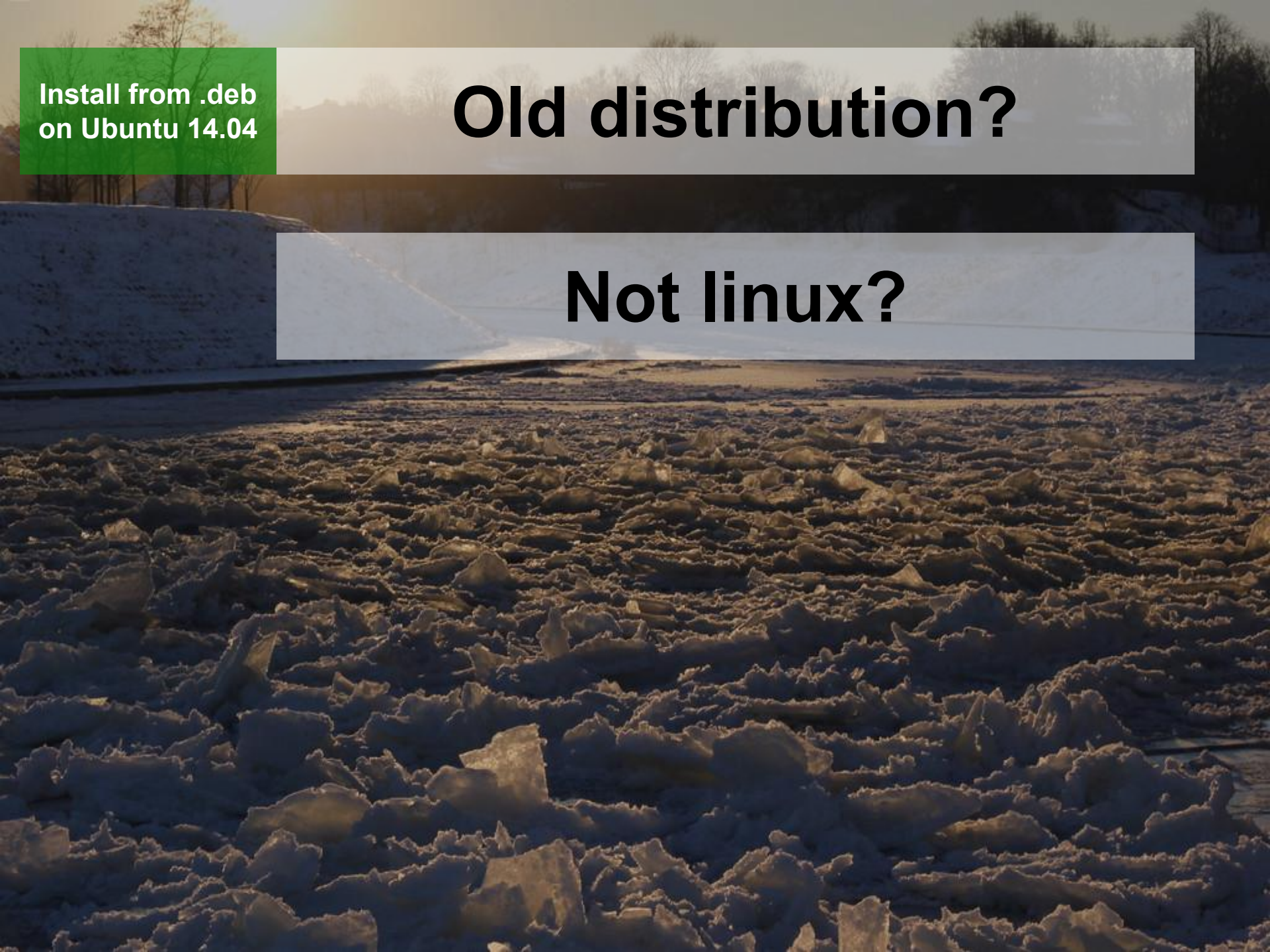




**Install from .deb  
on Ubuntu 14.04**

# **Old distribution?**

# **Not linux?**





Install from .deb  
on Ubuntu 14.04

**Old distribution?**

Native for  
OSX and Win

**Not linux?**

**Newest kernel**



**OpenVZ**

Your Linux kernel version 2.6.32-042stab113.12 can be unstable running docker.

**Please upgrade your kernel to 3.10.0.**



# Ended up on AWS





# Ended up on AWS

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	arn:aws:s3:::img-af0d52f4	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input type="checkbox"/>	<input type="checkbox"/>

Delete on Termination ⓘ

**Stable HDD**



# Ended up on AWS

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda	ami-af0d55fa	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypted

Delete on Termination ⓘ

NETWORK & SECURITY

- Security Groups
- Elastic IPs**
- Placement Groups
- Key Pairs
- Network Interfaces

<input checked="" type="checkbox"/>	Elastic IP	Allocation ID	Instance	Priv
<input checked="" type="checkbox"/>	52.50.220.71			

**Stable HDD**

**Static IP**

# Ended up on AWS

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	ami-af0d51fa	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input type="checkbox"/>	<input type="checkbox"/>

Delete on Termination ⓘ

NETWORK & SECURITY

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Elastic IP    Allocation ID    Instance    Priv

52.50.220.71

Stable HDD

Static IP

Similar price

t2.nano

1-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	\$0	\$3.65	\$0.005	29%	\$0.007 per Hour
Partial Upfront	\$28	\$1.17	\$0.005	31%	
All Upfront	\$41	\$0.00	\$0.005	33%	
3-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
Partial Upfront	\$49	\$1.10	\$0.003	51%	\$0.007 per Hour
All Upfront	\$83	\$0.00	\$0.003	54%	



# *Dev and prod* environments

Using multiple docker-compose.yml

Version  
control and  
state

Testing and  
debugging

Dev/prod  
and configs

Is it  
possible?  
Why not?

What is  
docker?

Why VDS?  
Why  
docker?

# Via multiple files

```
docker-compose  
-f docker-compose.yml  
-f docker-compose.stage.yml  
-f docker-compose.live.yml up -d
```

docker-compose.live.yml

## services:

### web:

#### environment:

- PHP\_AB\_LT\_HOST=aurelijus.banelis.lt
- JAVA\_WEBSITE\_HOST=auginte.com
- JAVA\_DEMOAPP\_HOST=demo.auginte.com

### php\_ab\_lt:

#### environment:

- DB\_PASSWORD=h2Jk\_PHP\_IMPRESS\_qlfEgtjeSFIA



# Update on run

docker-compose.yml

## volumes:

- ./my-nginx/run.sh:/custom/run.sh

## command:

- **"/custom/run.sh"**

Do not use `envsubst` - to not change `$uri`, `$document_root`, ...

run.sh

```
# ....  
sed -e "s#\${$IN}#\${OUT}#g" < ${CFG_TMP} > ${CFG_DEST}  
# ....  
nginx -g 'daemon off;'
```

# Availability

docker-compose.yml

**web:**

**ports:**

- "127.0.0.1:1380:80"
- "127.0.0.1:13443:443"

Ports not colliding  
with default  
applications

Friends in same WiFi  
will not see your work



# Availability

docker-compose.yml

**web:**

**ports:**

- "127.0.0.1:1380:80"
- "127.0.0.1:13443:443"

Ports not colliding  
with default  
applications

Friends in same WiFi  
will not see your work

docker-compose.stage.yml

**web:**

**ports:**

- "0.0.0.0:80:80"
- "0.0.0.0:443:443"

**restart:** always

Open to the users

Auto-restart,  
if application crashes

# Testing and debugging

Usually docker command-line

Version control and state

Testing and debugging

Dev/prod and configs

Is it possible?  
Why not?

What is docker?

Why VDS?  
Why docker?



# docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
87d03fd32b5e	nginx:1.11.1-alpine	"/custom/run.sh"	9 days ago	Up 9 days	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 127.0.0.1:1380->80/tcp, 127.0.0.1:13443->443/tcp	web
dda1f83d5add	aurelijusb/php:7.0.7-fpm	"/custom/run.sh"	9 days ago	Up 9 days	9000/tcp	php_ab_lt
7c1668eeb487	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8091->8091/tcp	java_demoapp
ef501b38749c	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Exited (0) 9 days ago		mysql_ab_lt_provision
94a913518c62	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8081->8081/tcp	java_website
91be2fcad13d	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Up 9 days	3306/tcp	mysql_ab_lt
643f37567fc1	orientdb/orientdb:2.1.5	"server.sh"	9 days ago	Up 9 days	127.0.0.1:24241->2424/tcp, 127.0.0.1:24801->2480/tcp	orientdb_demoapp

## STATUS

Up 9 days

Exited (0) 9 days ago

## PORTS

3306/tcp

## NAMES

mysql\_ab\_lt

mysql\_ab\_lt\_provision



# docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
87d03fd32b5e	nginx:1.11.1-alpine	"/custom/run.sh"	9 days ago	Up 9 days	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 127.0.0.1:1380->80/tcp, 127.0.0.1:13443->443/tcp	web
dda1f83d5add	aurelijusb/php:7.0.7-fpm	"/custom/run.sh"	9 days ago	Up 9 days	9000/tcp	php_ab_lt
7c1668eeb487	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8091->8091/tcp	java_demoapp
ef501b38749c	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Exited (0) 9 days ago		mysql_ab_lt_provision
94a913518c62	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8081->8081/tcp	java_website
91be2fcad13d	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Up 9 days	3306/tcp	mysql_ab_lt
643f37567fc1	orientdb/orientdb:2.1.5	"server.sh"	9 days ago	Up 9 days	127.0.0.1:24241->2424/tcp, 127.0.0.1:24801->2480/tcp	orientdb_demoapp

## STATUS

Up 9 days

Exited (0) 9 days ago

## PORTS

3306/tcp

## NAMES

mysql\_ab\_lt

mysql\_ab\_lt\_provision

# docker logs -f mysql\_ab\_lt\_provision

```
[ec2-user@ ~]$ docker logs -f mysql_ab_lt_provision
### 2016-07-10T19:11:01.468547336+0000 ### Waiting for MySQL to be ready
ERROR 2003 (HY000): Can't connect to MySQL server on 'mysql_ab_lt' (111)
### 2016-07-10T19:11:01.558166589+0000 ### MySQL not available yet. Sleeping
Database
```



# docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
87d03fd32b5e	nginx:1.11.1-alpine	"/custom/run.sh"	9 days ago	Up 9 days	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 127.0.0.1:1380->80/tcp, 127.0.0.1:13443->443/tcp	web
dda1f83d5add	aurelijusb/php:7.0.7-fpm	"/custom/run.sh"	9 days ago	Up 9 days	9000/tcp	php_ab_lt
7c1668eeb487	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8091->8091/tcp	java_demoapp
ef501b38749c	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Exited (0) 9 days ago		mysql_ab_lt_provision
94a913518c62	java:8-jre	"/custom/run.sh"	9 days ago	Up 9 days	127.0.0.1:8081->8081/tcp	java_website
91be2fcad13d	mysql:5.7.12	"docker-entrypoint.sh"	9 days ago	Up 9 days	3306/tcp	mysql_ab_lt
643f37567fc1	orientdb/orientdb:2.1.5	"server.sh"	9 days ago	Up 9 days	127.0.0.1:24241->2424/tcp, 127.0.0.1:24801->2480/tcp	orientdb_demoapp

## STATUS

Up 9 days

Exited (0) 9 days ago

## PORTS

3306/tcp

## NAMES

mysql\_ab\_lt

mysql\_ab\_lt\_provision

# docker logs -f mysql\_ab\_lt\_provision

```
[ec2-user@ ~]$ docker logs -f mysql_ab_lt_provision
### 2016-07-10T19:11:01.468547336+0000 ### Waiting for MySQL to be ready
ERROR 2003 (HY000): Can't connect to MySQL server on 'mysql_ab_lt' (111)
### 2016-07-10T19:11:01.558166589+0000 ### MySQL not available yet. Sleeping
Database
```

# docker exec -it web sh

```
[ec2-user@ ~]$ docker exec -it web sh
/ # cat /etc/nginx/conf.d/site.conf
server {
    listen      80;
```



# SSH & Port forwarding

```
ssh -i ~/.ssh/secret.pem -L 127.0.0.1:18081:127.0.0.1:8081 ec2-user@11.22.33.44
```





# SSH & Port forwarding

```
ssh -i ~/.ssh/secret.pem -L 127.0.0.1:18081:127.0.0.1:8081 ec2-user@11.22.33.44
```

## Insert bash in entrypoint script

```
[vagrant@localhost docker]$ docker-compose -f docker-compose.yml -f docker-compose.stage.yml run mysql_ab_lt_provision
```

```
### 2016-07-09T10:59:34.247761032+0000 ### Waiting for MySQL to be ready
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| ab_lt_stage |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
root@621911e712db: /#
```

# SSH & Port forwarding

```
ssh -i ~/.ssh/secret.pem -L 127.0.0.1:18081:127.0.0.1:8081 ec2-user@11.22.33.44
```

## Insert bash in entrypoint script

```
[vagrant@localhost docker]$ docker-compose -f docker-compose.yml -f docker-compose.stage.yml run mysql_ab_lt_provision
```

```
### 2016-07-09T10:59:34.247761032+0000 ### Waiting for MySQL to be ready
```

```
+-----+
| Database |
+-----+
| information_schema |
| ab_lt_stage |
| mysql |
| performance_schema |
| sys |
+-----+
root@621911e712db: /#
```

## Docker inspect


```
docker inspect -f '{{.Name}} - {{.NetworkSettings.Networks.docker_default.IPAddress }}' $(docker ps -aq)
```

```
/web - 172.18.0.8
/php_ab_lt - 172.18.0.7
/java_demoapp - 172.18.0.6
/mysql_ab_lt_provision -
/mysql_ab_lt - 172.18.0.3
/orientdb_demoapp - 172.18.0.4
/java_website - 172.18.0.2
```

Host:

Port:

Database:

 The following URL will be used to connect, edit to override:

```
MySQL - @172.18.0.3 [connected]
└─ information_schema: schema
└─ local: schema
  └─ ip_administrator: table
  └─ ip_email_queue: table
  └─ ip_inline_value_global: table
```



# Version control and state

`.gitignore` raw data, add fields for provisioning

Version  
control and  
state

Testing and  
debugging

Dev/prod  
and configs

Is it  
possible?  
Why not?

What is  
docker?

Why VDS?  
Why  
docker?

# Mount vs Build

**container\_name:** web

**image:** nginx:1.11.1-alpine

**volumes:**

- ../releases/php-ab-It/\$REV/install:/releases/php\_ab\_It/\$REV/install
- ../releases/php-ab-It/\$REV/lp:/releases/php\_ab\_It/\$REV/lp
- ../releases/php-ab-It/\$REV/Plugin:/releases/php\_ab\_It/\$REV/Plugin
- ../releases/php-ab-It/\$REV/Theme:/releases/php\_ab\_It/\$REV/Theme
- ../releases/php-ab-It/\$REV/vendor:/releases/php\_ab\_It/\$REV/vendor
- ../releases/php-ab-It/\$REV/admin.php:/releases/php\_ab\_It/\$REV/admin.php
- ../releases/php-ab-It/\$REV/favicon.ico:/releases/php\_ab\_It/\$REV/favicon.ico
- ../releases/php-ab-It/\$REV/index.php:/releases/php\_ab\_It/\$REV/index.php
- ../releases/php-ab-It/\$REV/robots.txt:/releases/php\_ab\_It/\$REV/robots.txt



Explore Help



Explore Help



Explore Help

OFFICIAL REPOSITORY

nginx ☆

Last pushed: a month ago

OFFICIAL REPOSITORY

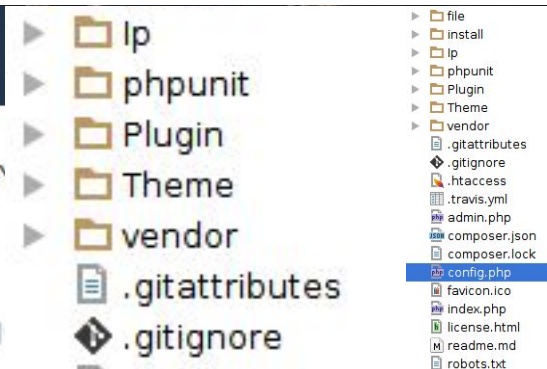
php ☆

Last pushed: 7 days ago

OFFICIAL REPOSITORY

mysql ☆

Last pushed: a month ago





# Mount vs Build

```
docker-php-ext-install mbstring pdo_mysql gd
```



Dashboard Explore Organizations

PUBLIC | AUTOMATED BUILD

aurelijusb/php 

Last pushed: a few seconds ago

Repo Info

Tags

Dockerfile

Build Details

Build Settings

Collaborators

Webhooks

Settings

Status	Tag	Created	Last Updated
 Building	7.0.7-fpm	3 minutes ago	a few seconds ago
 Success	latest	4 minutes ago	a few seconds ago

# Mount vs Build

```
Pulling php_ab_lt (aurelijusb/php:7.0.7-fpm)...
7.0.7-fpm: Pulling from aurelijusb/php
5c90d4a2d1a8: Already exists
357b76a49838: Downloading [=====>] 38.39 MB/77.34 MB
0e87614c69f0: Download complete
6120681ee5f7: Downloading [=====>] 22.54 MB/40.88 MB
80cc9c19278e: Download complete
d787b842ee97: Download complete
142d4ea9ea2b: Download complete
9fb5bc4b3e4f: Download complete
```

Configuration

System check

Database

Finish

## System check

**Nginx configuration required (?)**

Warning

**PHP version >= 5.3.3 (?)**

Ok

**PHP module "PDO" (?)**

Ok

**GD Graphics Library (?)**

Ok

**PHP sessions (?)**

Ok



# Data: dump/merge

**mysql\_ab\_it:**

**volumes:**

- /custom/data/mysql-ab-it/stage:/var/lib/mysql

**mysql\_ab\_it\_provision:**

**environment:**

- MYSQL\_ROOT\_PASSWORD=h2JfESFIA
- IMPORT\_DBNAME=ab\_it\_stage
- IMPORT\_FILE=/custom/dumps/aurelijusb\_ip.sql
- REPLACE\_FROM=aurelijus.banelis.lt
- REPLACE\_TO=aurelijus.banelis.stage

# Data: dump/merge

**mysql\_ab\_it:**

**volumes:**

- /custom/data/mysql-ab-it/stage:/var/lib/mysql

**mysql\_ab\_it\_provision:**

**environment:**

- MYSQL\_ROOT\_PASSWORD=h2JfESFIA
- IMPORT\_DBNAME=ab\_it\_stage
- IMPORT\_FILE=/custom/dumps/aurelijusb\_ip.sql
- REPLACE\_FROM=aurelijus.banelis.lt
- REPLACE\_TO=aurelijus.banelis.stage

```
bin/auginte-server -Dprovision=${PROVISION}
```



**You have learned**

**Why VDS? Why docker?**

**What is docker?**

**Is it possible? Why not?**

**Dev/prod and configs**

**Testing and debugging**

**Version control and state**

You have learned

That's it. Questions?

Why VDS? Why docker?

What is docker?

Is it possible? Why not?

Dev/prod and configs

Testing and debugging

Slides hosted using Docker:

Version control and state

<https://aurelijus.banelis.lt/presentations/nta-2016/Docker-on-VDS-v1.pdf>