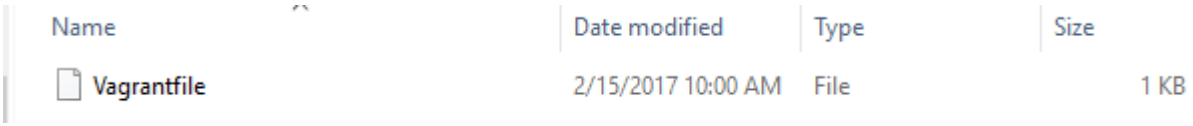
Vagrant box setup

Artur Malinowski (PGS Software)

CREATING CUSTOM VAGRANT BOX

Install Oracle VirtualBox and Vagrant

In any folder of your choice, create configuration file named `Vagrantfile`



Insert code fragment from below into `Vagrantfile`

```
Vagrant.configure("2") do |config|
     config.vm.box = "ubuntu/trusty64"
     config.vm.provision :shell, path: "provision.sh"
     config.vm.provider "virtualbox" do |vb|
          vb.customize ["modifyvm", :id, "--memory", "1024"]
     end
end
```

Some explanation:

```
config.vm.box = "ubuntu/trusty64"
```

tells Vagrant which box should be loaded, `ubuntu/trusty64` is the name of basic box provided by HashiCorp. If it is not available locally, Vagrant will download it from Vagrant's public box catalog (*https://atlas.hashicorp.com/boxes/search*) when we try to run our box.

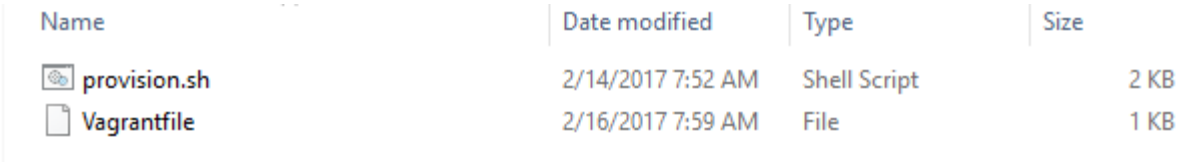
```
config.vm.provision :shell, path: "provision.sh"
```

points to shell script which will be executed after vagrant box is up and running. Content of `provision.sh` will be explained later on.

```
config.vm.provider "virtualbox" do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024"]
end
```

sets memory limit for vagrant box to 1024 MB

Create `provision.sh` in the same folder as `Vagrantfile`



Insert code fragment from below into `provision.sh`

```
#!/bin/sh -e
#POSTGRES
echo "Installing PostreSQL"
# PostgreSQL Version
PG_VERSION=9.4
export DEBIAN_FRONTEND=noninteractive
PG_REPO_APT_SOURCE=/etc/apt/sources.list.d/pgdg.list
# Add PG apt repo:
echo "deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main" >
"$PG_REPO_APT_SOURCE"
# Add PGDG repo key:
wget --quiet -O - https://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc | apt-key add -
# Update package list and upgrade all packages
apt-get update
apt-get -y upgrade
apt-get -y install "postgresql-$PG_VERSION" "postgresql-contrib-$PG_VERSION"
PG_CONF="/etc/postgresql/$PG_VERSION/main/postgresql.conf"
PG_HBA="/etc/postgresql/$PG_VERSION/main/pg_hba.conf"
PG_DIR="/var/lib/postgresql/$PG_VERSION/main"
# Edit postgresql.conf to change listen address to '*':
sed -i "s/#listen_addresses = 'localhost'/listen_addresses = '*'/" "$PG_CONF"
# Append to pg_hba.conf to add password auth:
echo "host    all             all             all                     md5" >> "$PG_HBA"
# Explicitly set default client_encoding
echo "client_encoding = utf8" >> "$PG_CONF"
service postgresql restart
echo "Successfully created PostgreSQL dev virtual machine."
echo "Cleaning for box exporting"
sudo apt-get clean
echo "Cleaning finished"
```

`provision.sh` script downloads and installs PostgreSQL version 9.4.
Script content will be explained in further version of this guide.

Next step is to start box based on `Vagrantfile`. Open command line, go to folder with `Vagrantfile` and execute `vagrant up` command.
Screen after vagrant start up:

```
==> default:  * Starting PostgreSQL 9.4 database server
==> default:    ...done.
==> default: Setting up postgresql-contrib-9.4 (9.4.11-1.pgdg14.04+1) ...
==> default: Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
==> default:  * Restarting PostgreSQL 9.4 database server
==> default:    ...done.
==> default: Successfully created PostgreSQL dev virtual machine.
==> default: Cleaning for box exporting
==> default: Cleaning finished

D:\EDziennik\schooldaily\vagrant\base>
```

Now vagrant basic box is ready for packaging to final box.

In command line type in:

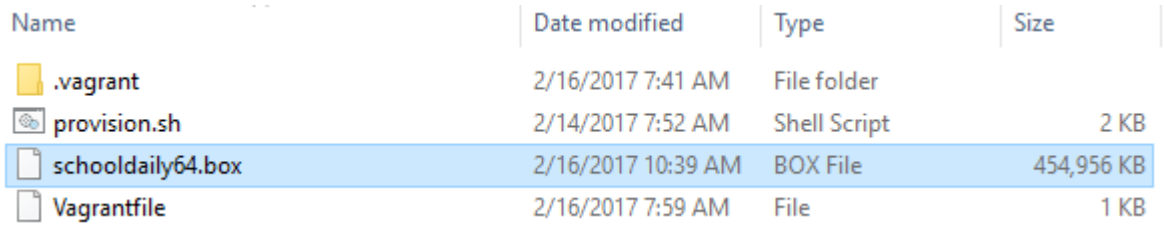
```
vagrant package –output schooldaily64.box
```

Packaging may take a while…
Screen after packaging:

```
D:\EDziennik\schooldaily\vagrant\base>vagrant package --output schooldaily64.box
==> default: Attempting graceful shutdown of VM...
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: D:/EDziennik/schooldaily/vagrant/base/schoolda
ily64.box

D:\EDziennik\schooldaily\vagrant\base>
```

Vagrant should create `schooldaily64.box` file:

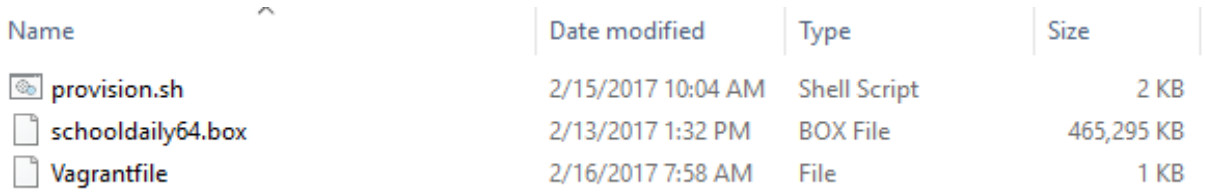


Name	Date modified	Type	Size
.vagrant	2/16/2017 7:41 AM	File folder	
provision.sh	2/14/2017 7:52 AM	Shell Script	2 KB
schooldaily64.box	2/16/2017 10:39 AM	BOX File	454,956 KB
Vagrantfile	2/16/2017 7:59 AM	File	1 KB

And that's it, our custom box with preinstalled PostgreSQL is ready to use.

STARTING CUSTOM VAGRANT BOX

In new empty folder create `Vagrantfile, provision.sh` and copy newly created `schooldaily64.box`.



Name	Date modified	Type	Size
provision.sh	2/15/2017 10:04 AM	Shell Script	2 KB
schooldaily64.box	2/13/2017 1:32 PM	BOX File	465,295 KB
Vagrantfile	2/16/2017 7:58 AM	File	1 KB

Insert code fragment from below into `Vagrantfile`

```
Vagrant.configure("2") do |config|
     config.vm.box = "schooldaily64"
     config.vm.box_url = "file://schooldaily64.box"
     config.vm.provision :shell, path: "provision.sh"
     config.vm.network :forwarded_port, guest:5432, host:15432
end
```

```
config.vw.box = "schooldaily64"
```

sets name of registered vagrant box to use, if `schooldaily64` is not registered by vagrant then box will be imported from `config.vm.box_url`

```
config.vm.box_url = "file://schooldaily64.box"
```

points to box file which should be imported and registered by vagrant when there is no `schooldaily64` box available in vagrant local cache. It can also point to web resources i.e.:

```
config.vm.box_url = "https://atlas.hashicorp.com/ubuntu/boxes/trusty64"
```

```
config.vm.provision :shell, path: "provision.sh"
```

points to shell script that will be executed after vagrant box is up and running.

```
config.vm.network :forwarded_port, guest:5432, host:15432
```

forwards PostgreSQL default access port to be accessible from outside the VM on port `15432`

Insert code fragment from below into `provision.sh`

```
echo "Starting PostgreSQL"
APP_DB_USER=pgsuser
APP_DB_PASS=pgspass
# Database Name
APP_DB_NAME=schooldaily
print_db_usage () {
  echo " PostgreSQL database has been setup and can be accessed on port 15432"
  echo "  Host: localhost"
  echo "  Port: 15432"
  echo "  Database: $APP_DB_NAME"
  echo "  Username: $APP_DB_USER"
  echo "  Password: $APP_DB_PASS"
  echo ""
  echo "Admin access to postgres user via VM:"
  echo "  vagrant ssh"
  echo "  sudo su - postgres"
  echo ""
  echo "psql access to app database user via VM:"
  echo "  vagrant ssh"
  echo "  sudo su - postgres"
  echo "  PGUSER=$APP_DB_USER PGPASSWORD=$APP_DB_PASS psql -h localhost $APP_DB_NAME"
}

cat << EOF | su - postgres -c psql
-- Create the database user:
CREATE USER $APP_DB_USER WITH PASSWORD '$APP_DB_PASS';
-- Create the database:
CREATE DATABASE $APP_DB_NAME WITH OWNER=$APP_DB_USER
                              LC_COLLATE='en_US.utf8'
                              LC_CTYPE='en_US.utf8'
                              ENCODING='UTF8'
                              TEMPLATE=template0;
EOF

sudo su postgres
service postgresql restart
echo "Successfully started PostgreSQL."
print_db_usage
```

`provision.sh` creates new PostgreSQL user *pgsuser* with password *pgspass* and *schooldaily* database. It also restarts PostgreSQL and display some additional information about accessing it.

Now we can call `vagrant up`

```
D:\EDziennik\schooldaily\vagrant>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'schooldaily64' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Box file was not detected as metadata. Adding it directly...
==> default: Adding box 'schooldaily64' (v0) for provider: virtualbox
    default: Unpacking necessary files from: file://D:/EDziennik/schooldaily/vagrant/schooldaily64.box
    default: Progress: 100% (Rate: 121M/s, Estimated time remaining: --:--:--)
==> default: Successfully added box 'schooldaily64' (v0) for 'virtualbox'!
==> default: Importing base box 'schooldaily64'...
```

```
==> default: Starting PostgreSQL
==> default: CREATE ROLE
==> default: CREATE DATABASE
==> default:  * Restarting PostgreSQL 9.4 database server
==> default:    ...done.
==> default: Successfully started PostgreSQL.
==> default:  PostgreSQL database has been setup and can be accessed on port 15432
==> default:   Host: localhost
==> default:   Port: 15432
==> default:   Database: schooldaily
==> default:   Username: pgsuser
==> default:   Password: pgspass
==> default: Admin access to postgres user via VM:
==> default:   vagrant ssh
==> default:   sudo su - postgres
==> default: psql access to app database user via VM:
==> default:   vagrant ssh
==> default:   sudo su - postgres
==> default:   PGUSER=pgsuser PGPASSWORD=pgspass psql -h localhost schooldaily

D:\EDziennik\schooldaily\vagrant>
```

On final screen, there are some information about how to access PostgreSQL on our VM.