

Exercise 9

Alexander Steinbrecher
asteinbr@stud.fh-frankfurt.de

December 19, 2013

1 Exercise - GAE-compatible Private Cloud Platform Service

This approach uses AppScale as PaaS under Windows 8.

First of all you have to fulfill the requirements for AppScale. These are

- Oracle VirtualBox
- Vagrant
- Cygwin with openssh, git, mingw-binutils
- msysgit with "Run Git from the Windows Command Prompt"

when the requirements are installed, you can start with opening vagrant. To do this, you need to open command shell (cmd.exe) or powershell (powershell.exe). Here you have to start vagrant via the full path:

```
PS C:\HashiCorp\Vagrant> .\bin\vagrant.bat
```

This command should give you the subcommands and options.

Afterwards you need to download the preconfigured Vagrantfile from https://s3.amazonaws.com/appscale_CDN/files/Vagrantfile into C:\HashiCorp\Vagrant\.

Then you're able to download the AppScale image via `.\bin\vagrant up`. This command is downloading an image with an OS (Ubuntu) with AppScale integrated and runs it within a VirtualBox environment.

When no errors occur you have to get SSH access via `.\bin\vagrant ssh`. In my case the output was

```
PS C:\HashiCorp\Vagrant> .\bin\vagrant.bat ssh
'ssh' executable not found in any directories in the %PATH% variable. Is an
SSH client installed? Try installing Cygwin, MinGW or Git, all of which
contain an SSH client. Or use the PuTTY SSH client with the following
authentication information shown below:
```

```
Host: 127.0.0.1
```

```
Port: 2222
```

```
Username: vagrant
```

```
Private key: C:/Users/Alexander/.vagrant.d/insecure_private_key
```

Now you need to transform the `insecure_private_key` with PuTTYgen to a PuTTY Key (*.ppk). After that you can connect with PuTTY to the virtual machine (with specifying the ppk-key file in PuTTY).

Within the terminal session you have to give your user a password:

```
login as: vagrant
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)
```

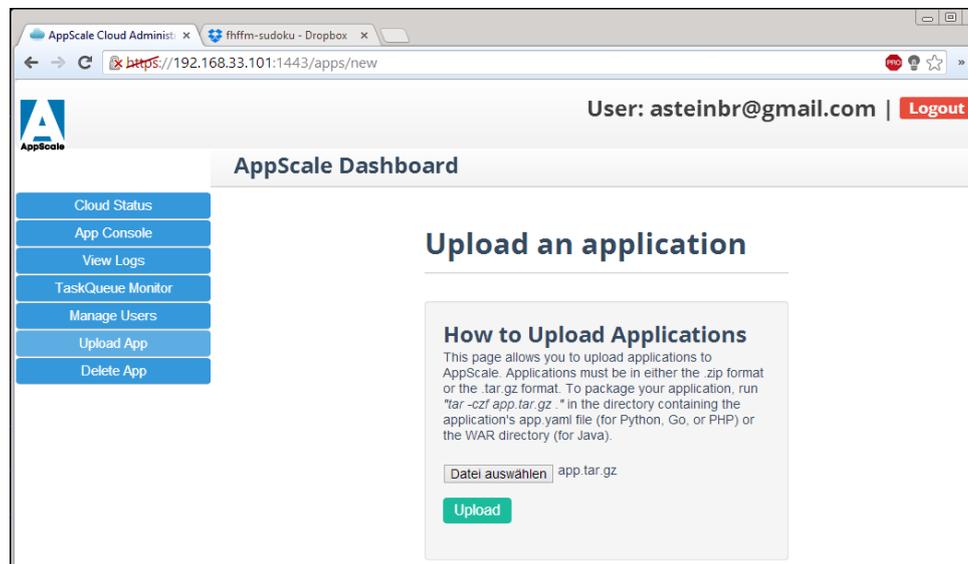
```
* Documentation:  https://help.ubuntu.com/
Welcome to your Vagrant-built virtual machine.
Last login: Wed Nov  6 20:41:13 2013 from 10.0.2.2
vagrant@precise64:~$ sudo su
root@precise64:/home/vagrant# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@precise64:/home/vagrant#
```

Then you have to navigate to your drive and download the preconfigured AppScale file:

```
root@precise64:/home/vagrant# cd
root@precise64:~# curl -O -L https://s3.amazonaws.com/appscale_CDN/files/AppScalefile
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 1912  100 1912    0     0   3009      0  --:--:--  --:--:--  --:--:--  4174
root@precise64:~#
```

Now you're able to start AppScale with `appscale up`. This will open the AppScale Service that you can access it over your web browser via `192.168.33.101:1080/status`.

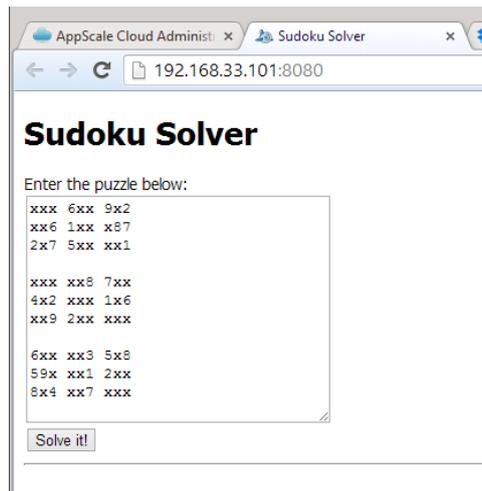
At the webpage you need to login with your given credentials during `appscale up`. Afterwards you can upload an existing application over the button `Upload App`.



After clicking `Upload`, the application is being uploaded on the AppScale instance.



Clicking on the deployed application opens it within the browser.



2 Exercise - Private Cloud Platform Service

After creating an account on <http://www.openshift.com> you are able to create your own applications.

Before creating the applications you have to specify a domain-name. Later it will be this format: <http://applicationname-domainname.rhcloud.com>. For example when you choose foobar1337 as a domain name, than your applications will have following url <http://app1-foobar1337.rhcloud.com> or <http://app2-foobar1337.rhcloud.com>. So the domain name is fixed.

After creating a domain name, you can add an application. OpenShift provides different types of applications:

1. PHP
2. JBoss
3. Tomcat
4. Jenkins
5. Python
6. Ruby
7. Node.js
8. etc.

For this example I choose PHP5.3. OpenShift offers different types of Gears, the size of the applications. Free accounts have only access to the small gear. When creating a application you have to specify a Public URL, which will be later the URL for this application, e.g. <http://php-asteinbr.rhcloud.com>.

After creating you will get some git commands which enables you to clone the repository to your local machine.

```
asteinbr@mbp ~/Development $ git clone
ssh://52b233f25973ca41b1000099@php-asteinbr.rhcloud.com/~/git/php.git/
```

Cloning into 'php'...

Warning: Permanently added the RSA host key for IP address '23.22.38.222' to the list of known hosts.

```

remote: Counting objects: 20, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 20 (delta 0), reused 20 (delta 0)
Receiving objects: 100% (20/20), 18.23 KiB | 0 bytes/s, done.
Checking connectivity... done
asteinbr@mbp ~/Development $ cd php
asteinbr@mbp ~/Development/php (master) $ ls
README.md  deplist.txt  libs          misc          php

```

You see the structure of the git repository. Under the folder php are the files located which can be accessed over the public url.

```

asteinbr@mbp ~/Development/php (master) $ cd php
asteinbr@mbp ~/Development/php/php (master) $ ls
health_check.php  index.php

```

To see that the service is working correctly you can create a info file for php.
Just create a new file with vim info.php with the content

```

<?php
phpinfo();
?>

```

Then add the file to git index via `git add info.php`, and commit the change with `git commit -m 'info file'`. And finally push it to the remote repository via `git push`.

```

asteinbr@mbp ~/Development/php/php (master) $ git push

```

```

Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 412 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Stopping PHP cartridge
remote: [Wed Dec 18 19:00:12 2013] [warn] PassEnv variable SHELL was undefined
remote: [Wed Dec 18 19:00:12 2013] [warn] PassEnv variable USER was undefined
remote: [Wed Dec 18 19:00:12 2013] [warn] PassEnv variable LOGNAME was undefined
remote: Waiting for stop to finish
remote: Building git ref 'master', commit df8fbb1
remote: Building PHP cartridge
remote: Preparing build for deployment
remote: Deployment id is b4b4ee2e
remote: Activating deployment
remote: Starting PHP cartridge
remote: Result: success
remote: Activation status: success
remote: Deployment completed with status: success
To ssh://52b233f25973ca41b1000099@php-asteinbr.rhcloud.com/~/git/php.git/
 87afb74..df8fbb1  master -> master

```

You also can add for every application databases to provide databases for your applications.

