Kapitel 1

Trennung von Code und Daten – Variables, Facts und Templates

1.1 Vagrantfile

```
Vagrant.configure(2) do |config|

  # Falls vbguest-plugin bitte Guest nachladen
  if Vagrant.has_plugin?("vagrant-vbguest") then
    config.vbguest.auto_update = true
  end

  config.vm.provider "virtualbox" do |vb|
    # Display the VirtualBox GUI when booting the machine
    vb.gui = true
    # Customize the amount of memory on the VM:
    vb.memory = "768"
    vb.customize ["modifyvm", :id, "--cpuexecutioncap", "50"]
    vb.linked_clone = true
  end

  config.vm.box = "kraeml/ubuntu_de"
  config.ssh.insert_key = false

  config.vm.define "ctl" do | ctl |
    ctl.vm.hostname = "ctl"
    # Bitte in hosts eintragen.
    # 192.168.50.60 ubuntu_ais ubuntu_ais
    # Somit kann mit ping ubuntu_ais bzw. http://ubuntu_ais
    # aufgerufen werden.
    ctl.vm.network "private_network",
     ip: "192.168.50.60"

    # Das Gruene Netzwerk
```

```
    ctl.vm.network "private_network",
     ip: "192.168.60.254",
     virtualbox__intnet: "Gruen"

    # DHCP und WWW über das lokale Netz
    # Die DHCP Einstellungen werden übernommen
    ctl.vm.network "public_network",
     bridge: "eth0",
     use_dhcp_assigned_default_route: true

    # ctl.vm.provision "shell", path: "./provision/ctl/sshd_config.sh"
    ctl.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_ctl.rdf"
    end
    ctl.vm.synced_folder "projects/", "/home/vagrant/projects"
  end

  config.vm.define "loadbalancer" do | lb |
    lb.vm.hostname = "lb"
    #Das Gruene Netzwerk
    lb.vm.network "private_network",
     ip: "192.168.60.2",
     virtualbox__intnet: "Gruen"

    lb.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_lb.rdf"
    end
  end

  config.vm.define "web1" do | web1 |
    web1.vm.hostname = "web1"
    #Das Gruene Netzwerk
    web1.vm.network "private_network",
     ip: "192.168.60.11",
     virtualbox__intnet: "Gruen"

    web1.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_web1.rdf"
    end
  end

  config.vm.define "web2" do | web2 |
    web2.vm.hostname = "web2"
    #Das Gruene Netzwerk
    web2.vm.network "private_network",
     ip: "192.168.60.12",
     virtualbox__intnet: "Gruen"

    web2.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_web2.rdf"
    end
```

```
  end

  config.vm.define "web3" do | web3 |
    web3.vm.hostname = "web3"
    #Das Gruene Netzwerk
    web3.vm.network "private_network",
     ip: "192.168.60.13",
     virtualbox__intnet: "Gruen"

    web3.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_web3.rdf"
    end
  end
  # Die erste Datenbank
  config.vm.define "debian" do | debian |
    debian.vm.box = "debian/jessie64"
    debian.vm.hostname = "db"
    # Das Gruene Netzwerk
    debian.vm.network "private_network",
     ip: "192.168.60.21",
     virtualbox__intnet: "Gruen"

    debian.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_db.rdf"
    end
  end

  # Die zweite Datenbank
  config.vm.define "centos" do | centos |
    centos.vm.box = "bento/centos-6.7"
    centos.vm.hostname = "dbel"
    # Das Gruene Netzwerk
    centos.vm.network "private_network",
     ip: "192.168.60.22",
     virtualbox__intnet: "Gruen"

    centos.vm.provider "virtualbox" do |vb|
      vb.name = "ubuntu_ais_dbel.rdf"
    end
  end
end
```

1.2 Inventoryfile customhosts

```
#customhosts
#inventory configs for my cluster
[db]
```

```
192.168.60.21 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key
192.168.60.22 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key
[www]
192.168.60.11 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key
192.168.60.12 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key
192.168.60.13 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key

[lb]
192.168.60.2 ansible_ssh_user=vagrant ansible_ssh_private_key_file=/
   vagrant/insecure_private_key
```

1.3 Das Playbook site.yml

```
---
# This is a sitewide playbook
# filename: site.yml
- include: www.yml
- include: db.yml
```

db.yml

```
---
# Playbook for Database Servers
# filename: db.yml
- hosts: db
  remote_user: vagrant
  sudo: yes
  roles:
     - { role: mysql, mysql_bind: "{{  ansible_eth1.ipv4.address }}" }
```

www.yml

```
---
- hosts: www
  remote_user: vagrant
  sudo: yes
  pre_tasks:
     - shell: echo 'I":" Beginning to configure web server..'
```

```
  roles:
     - nginx
  post_tasks:
     - shell: echo 'I":" Done configuring nginx web server...'
```

1.4 Rolle nginx

meta/main.yml

```
---
dependencies:
  - { role: base}
```

tasks/main.yml

```
---
# This is main tasks file for nginx role
 - include: install.yml
 - include: configure.yml
 - include: service.yml
```

tasks/install.yml

```
---
 - name: add official nginx repository
   apt_repository: repo='deb http://nginx.org/packages/ubuntu/ lucid
      nginx'
 - name: install nginx web server and ensure its at the latest version
   apt: name=nginx state=latest force=yes
```

tasks/configure.yml

```
---
 - name: create default site configurations
   template: src=default.conf.j2 dest=/etc/nginx/conf.d/default.conf
      mode=0644
   notify:
    - restart nginx service
 - name: create home page for default site
   copy: src=index.html dest=/usr/share/nginx/html/index.html
```

```
---
 - name: start nginx service
   service: name=nginx state=started
```

defaults/main.yml

```
---
#file: roles/nginx/defaults/main.yml
nginx_port: 80
nginx_root: /usr/share/nginx/html
nginx_index: index.html
```

files/index.html

```
<html>
  <body>
    <h1>Ole Ole Ole </h1>
    <p> Welcome to FIFA World Cup News Portal</p>
   </body>
</html>
```

handlers/main.yml

```
---
- name: restart nginx service
  service: name=nginx state=restarted
```

templates/default.conf.j2

```
server {
    listen       {{ nginx_port }};
    server_name  {{ ansible_hostname }};

    location / {
        root   {{ nginx_root }};
        index  {{ nginx_index }};
    }
}
```

Rolle mysql

tasks/main.yml

```
---
# This is main tasks file for mysql role
# filename: roles/mysql/tasks/main.yml

# Load vars specific to OS Family.
- include_vars: "{{ ansible_os_family }}.yml"
  when: ansible_os_family != 'Debian'

- include: install_RedHat.yml
  when: ansible_os_family == 'RedHat'

- include: install_Debian.yml
  when: ansible_os_family == 'Debian'

- include: configure.yml
- include: service.yml
```

tasks/install_RedHat.yml

```
---
# filename: roles/mysql/tasks/install_RedHat.yml
- name: install mysql server
  yum:
    name: "{{ mysql_pkg }}"
```

tasks/install_Debian.yml

```
---
# filename: roles/mysql/tasks/install_Debian.yml

  - name: install mysql server
    apt:
      name: "{{ mysql_pkg }}"
      update_cache: yes
```

tasks/configure.yml

```
---
```

```
# filename: roles/mysql/tasks/configure.yml
 - name: create mysql config
   template: src="my.cnf.j2" dest="{{ mysql_cnfpath }}" mode=0644
   notify:
    - restart mysql service
 ```

tasks/service.yml

 ```yaml
---
# filename: roles/mysql/tasks/service.yml
 - name: start mysql server
   service: name="{{ mysql_service }}" state=started
```

defaults/main.yml

```
---
mysql_user: mysql
mysql_port: 3306
mysql_datadir: /var/lib/mysql
mysql_bind: 127.0.0.1
mysql_pkg: mysql-server
mysql_pid: /var/run/mysqld/mysqld.pid
mysql_socket: /var/run/mysqld/mysqld.sock
mysql_cnfpath: /etc/mysql/my.cnf
mysql_service: mysql
```

handlers/main.yml

```
---
# handlers file for mysql
- name: restart mysql service
  service: name="{{ mysql_service }}" state=restarted
```

templates/main.yml

```
# Notice: This file is being managed by Ansible
# Any manual updates will be overwritten
# filename: roles/mysql/templates/my.cnf.j2

[mysqld]
user         = {{ mysql_user | default("mysql") }}
```

```
pid-file      = {{ mysql_pid }}
socket        = {{ mysql_socket }}
port          = {{ mysql_port }}
datadir       = {{ mysql_datadir }}
bind-address = {{ mysql_bind }}
```

vars/main.yml

```
---
# vars file for mysql
```

vars/RedHat.yml

```
---
# RedHat Specific Configs.
# roles/mysql/vars/RedHat.yml
mysql_socket: /var/lib/mysql/mysql.sock
mysql_cnfpath: /etc/my.cnf
mysql_service: mysqld
mysql_bind: 0.0.0.0
```

1.5 Rolle base

tasks/main.yml

```
---
# essential tasks. should run on all nodes
 - name: creating devops group
   group: name=devops state=present
 - name: create devops user with admin previleges
   user: name=devops comment="Devops User" uid=2001 group=devops
 - name: install htop package
   action: apt name=htop state=present update_cache=yes
```