Kapitel 1

Schleifen

- Der Einsatz von with_* Ausdrücken
- Schleifen über Arrays
- Hashes (Dictionary) definieren und darüber iterieren

1.1 Das with_xxx

Das xxx steht für die Datentypen.

Beispiele:

```
with_items          Arrays
with_nested         Multidimensionale Arrays
with_dict           Hashes
with_fileglobs      Dateien mit "Patternmatch"
with_together       Sets von zwei Arrays
with_subelemnts     Hash-Subelement
with_sequence       Integer Reihe
with_random_choice  Zufallsauswahl
with_index_items    Array mit Index
```

1.2 Wordpress einrichten

Nur das runter laden von Wordpress reicht nicht aus.

Zutatenliste WP:

- Einen Webserver (nginx ist insatlliert)
- PHP
- MySQL-Datenbank und MySQL-Benutzer

PHP wird mit PHP5-FPM-Rolle auf den Webservern eingerichtet.

`ansible-galaxy init --init-path projects/roles/ php5-fpm`

```
- name: install php5-fpm and family
  apt:
    name: "{{ item }}"
  with_items: php5.packages
  notify:
   - restart php5-fpm service
```

Hier die Hash-Definition:

```
#filename: roles/php5-fpm/defaults/main.yml
#defaults file for php5-fpm
php5:
  packages:
    - php5-common
    - php5-curl
    - php5-mysql
    - php5-cli
    - php5-gd
    - php5-mcrypt
#    - php5-suhosin
    - php5-memcache
    - php5-fpm
  service:
    name: php5-fpm
```

1.3 Schleifen über ein Array

Siehe oben mit `with_items`

1.4 Zugriff auf Hashes

```
# filename: roles/php5-fpm/handlers/main.yml
# handlers file for php5-fpm
- name: restart php5-fpm service
  service: name="{{ php5['service']['name'] }}" state=restarted
```

1.5 Erstellen der DB und DB-User

Hier der Hash

```
#filename: group_vars/all
mysql_bind:  "{{ ansible_eth0.ipv4.address }}"
mysql:
  databases:
    fifalive:
      state: present
    fifanews:
      state: present
  users:
    fifa:
      pass: supersecure1234
      host: '%'
      priv: '*.*:ALL'
      state: present
```

1.6 Itteration über Hash

```
 - name: create mysql databases
   mysql_db:
     name: "{{ item.key }}"
     state: "{{ item.value.state }}"
   with_dict: "{{ mysql['databases'] }}"

 - name: create mysql users
   mysql_user:
     name: "{{ item.key }}"
     host: "{{ item.value.host }}"
     password: "{{ item.value.pass }}"
     priv: "{{ item.value.priv }}"
     state: "{{ item.value.state }}"
   with_dict: "{{ mysql['users'] }}"
```

1.7 Nginx virtual hosts

Wordpress wird als virtualer Host eingerichtet.

```
nginx:
  phpsites:
    fifanews:
      name: fifanews.com
      port: 8080
      doc_root: /var/www/fifanews
```

Der Zugriff in roles/nginx/tasks/configure.yml

```
 - name: create php virtual hosts
   template:
     src: php_vhost.j2
     dest: /etc/nginx/conf.d/{{ item.key }}.conf
   with_dict: "{{ nginx['phpsites'] }}"
   notify:
     - restart nginx service
```

In der Vorlage

```
# projects/roles/nginx/templates/php_vhost.j2
#{{ ansible_managed }}

server {
    listen   {{ item.value.port }};

  location / {
        root   {{ item.value.doc_root }};
        index  index.php;
    }

  location ~ .php$ {
        fastcgi_split_path_info ^(.+\.php)(.*)$;
        fastcgi_pass   backend;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  {{ item.value.doc_root }}
           $fastcgi_script_name;
        include fastcgi_params;
    }
}
upstream backend {
        server 127.0.0.1:9000;
}
```

Hinweis: Die Rolle muss noch für die Webserver eingetragen werden.

Testen: curl http://:8080