

Dokumentation Ansible

1 Einleitung

Bemerkungen

Diese Anleitung erläutert die Verwaltung unserer Ubuntu-Clients mit der Open-Source-Software Ansible (<https://www.ansible.com/>)

Ansible ist eine freie Software zur Verwaltung und Konfiguration einer großen Zahl an Clients. Ansible kombiniert Softwareverteilung, Ad-hoc-Kommando-Ausführung und Konfigurationsmanagement. Es verwaltet Netzwerkcomputer unter anderem über SSH und erfordert keinerlei zusätzliche Software auf dem zu verwaltenden System.

2 Grundlegende Konfiguration

2.1 Einrichtung und Konfiguration von Ansible

Die Installation von Ansible erfolgt einfach aus den Ubuntu-Paketquellen:

```
sudo apt install ansible
```

Damit sich Ansible auf den Client-Systemen anmelden kann, benötigt man einen ssh-Schlüssel, den man mit dem folgenden Befehl erstellt:

```
ssh-keygen -b 4096
```

Nach der Erstellung muss dieser Schlüssel auf die Clients kopiert werden, die verwaltet werden sollen. Dieses geschieht mit dem folgenden Befehl:

```
ssh-copy-id -i .ssh/id_rsa.pub benutzer@IP-Adresse
```

Dabei muss „benutzer“ durch den Benutzernamen ersetzt werden, unter dem Ansible laufen soll und zudem muss „IP-Adresse“ durch die korrekte IP-Adresse des Clients ersetzt werden. Nach der Verteilung des öffentlichen ssh-Schlüssels an die Clients müssen diese in die Datei /etc/ansible/hosts eingetragen werden. Es folgt ein Beispiel für unsere Roboter-AG-Laptops:

```
[RoboLaptops]
10.1.11.1
10.1.11.2
```

2.2 Ad-hoc-Kommando-Ausführung auf den Clients

Nachdem alle Clients, die verwaltet werden sollen, in die Datei `/etc/ansible/hosts` eingetragen und der öffentliche ssh-Schlüssel auf die Geräte übertragen wurde, kann man Ansible testen und einen Befehl auf allen Clients einer Gruppe (z. B. „RoboLaptops“) ausführen:

```
ansible RoboLaptops -u benutzer -a "shutdown -h now"
```

Dieses Kommando führt nun auf allen entsprechenden Geräten, die der Gruppe „RoboLaptops“ angehören, den Befehl „`shutdown -h now`“ aus und fährt die Geräte somit herunter. Zum Herunterfahren sind allerdings root-Privilegien erforderlich, weswegen der Befehl, so wie angegeben, nur funktioniert, wenn der Benutzer root ist.

2.3 Nutzung von „playbooks“

Damit man nicht immer einzelne Befehle eingeben muss, die dann auf allen Clients ausgeführt werden, gibt es sogenannte „playbooks“. Dabei handelt es sich um eine YAML-Datei, die die zu erreichende Konfiguration und Befehlsabfolge auf den Clients beschreibt. Die folgende Datei „`e31.yml`“ sorgt dafür, dass alle Clients im PC-Raum E31 die Paketquellen aktualisieren, alle verfügbaren Updates installieren und nicht mehr benötigte Abhängigkeiten entfernen:

```
---
- hosts: E31
  remote_user: root
  tasks:
  - name: Run the equivalent of "apt-get update" as a separate step
    apt:
      update_cache: yes
  - name: Update all packages to the latest version
    apt:
      upgrade: dist
      dpkg_options: 'force-confold,force-confdef'
  - name: Remove dependencies that are no longer required
    apt:
      autoremove: yes
```

Das „playbook“ wird mit dem folgenden Befehl aufgerufen:

```
ansible-playbook e31.yml
```

3 Installation einer Web-Oberfläche

Standardmäßig ist ansible eine Kommandozeilenprogramm. Es gibt aber unterschiedliche Web-Oberflächen, die eine Verwaltung der Clients auf eine komfortable Weise auch für Nutzerinnen und Nutzer ermöglicht, die sonst keine Erfahrung im Umgang mit Linux-Kommandozeilenprogrammen haben.

3.1 Installation von semaphore

Wir haben uns an unserer Schule für die Oberfläche semaphore entschieden (<https://github.com/ansible-semaphore/semaphore>).

Weite Einzelheiten folgen!