Bytecluster002 (alpha)

Als Vorbereitung zur Migration auf einen neuen Server wird die Konfiguration getestet. Dies soll hier dokumentiert werden.

Anforderungen

1. Webapplikationen
    1. Python
        1. Matrix
        2. Pretix
    2. PHP
        1. Nextcloud
        2. Blogs
        3. Wiki
        4. roundcube (webmail)
        5. paste(bin)
    3. NodeJS
        1. etherpad
    4. JS-only
        1. riot-web
2. Mail
    1. Postfächer
    2. Weiterleitungen
    3. SPAM-Filter
    4. Mailinglisten
3. Datenbanken
    1. mysql (Webdienste, Mail)
    2. postgresql (Matrix)
4. Sonstige Dienste
    1. ByteBot (Python)
    2. Freifunk-API (Python)
    3. Status MS/BS (Python)
5. User-Space (aktuell nicht vorhanden)

Testumgebung

- VirtualBox mit Debain 10 NetInstall ohne graphisches Frontend, Auswahl bei der Installation nur „SSH Server“.
- Portweiterleitung in VirtualBox von localhost:8822 auf <interneIP>:22 für den SSH Zugang

Ansible auf dem Host System

Zur Dokumentation und ggfs. Wiederholbarkeit soll die Konfiguration mittels Ansible-Skripten erfolgen. Dies benötigt keine Installation auf dem Server/Testsystem, sondern lediglich auf dem Host-System.

Windows (Cygwin)

- Mittels Setup folgende Pakete installieren: ansible binutils curl gcc-core gmp libffi-devel libgmp-devel make python27 python27-crypto python27-openssl python27-setuptools python27-devel git nano openssh openssl
- Per ssh-geygen ein Schlüsselpaar erzeugen

Inventory

Ansible benötigt ein Inventory mit den zu steuernden Server.

| inventory.ini

```
[all]
```

```
localhost:8822 ansible_ssh_user=chaos
```

In dieser Datei wird auch angegeben, dass der ssh-Nutzer nicht dem Windows/Cygwin-Nutzer entspricht.

Configuration

| anisble.cfg

```
[defaults]
# path to key for encrypted parameters
vault_password_file = ./vault_pass.txt
# enable timing information for tasks
callback_whitelist = profile_tasks

[privilege_escalation]
# ask for the sudo pass on start
become_ask_pass = true

[ssh_connection]
# ControlMaster=no was suggested for use with cygwin
# ConnectTimeout=0 fixes long waits for timeouts on my system
ssh_args = -o ControlMaster=no -o ConnectTimeout=0
```

vault password file

Ansible kann verschlüsselte Variablen verwenden, um sensible Daten wie Passwörter oder access-tokens in öffentlich zugänglichen Playbooks zu schützen. Das Passwort kann auf der Kommandozeile angegeben oder in einer (nicht zu veröffentlichenden) Datei gespeichert werden. Letzteres erlaubt die Angabe des Pfades in der .cfg und wird in den Beispielen benutzt.

Die Datei enthält eine beliebe Zeichenfolge in einer Zeile.

Startup

```
ansible -i inventory.txt -l all -v playbook.yml

#for generating encrypted variables:
ansible-vault encrypt-string --vault-password-file vault_password.txt
<put_string_to_encrypt_here>
```

Vorbereitung auf dem Server

- einen Nutzer anlegen (üblicherweise durch die Installation oder via adduser)

```
su # login als root
apt-get install sudo
/sbin/adduser <nutzername> sudo # fügt nutzer zur sudo-gruppe hinzu
exit # zurück zum User
mkdir ~/.ssh
nano ~/.ssh/authorized_keys #public key aus der cygwin-session des hosts hier her kopieren
exit
```

Idealerweise sollte dies die einzigen direkt ausgeführten Befehle auf dem Server bleiben.

Playbooks

Set time

- während der Debian10-Installation kann man Sprache und Zeitzone nicht unabhängig voneinander wählen
- ich hatte danach als Zeitzone PST und die Hardware Clock war auf Systemzeit statt auf UTC gestellt
- dies wird hier repariert, indem erst grob die Zeit anhand der Host-Zeit gestellt und dann ntp installiert wird

| play_timezone.yml

```
- hosts: all
  become: yes
  tasks:
    - name: check if ntp is installed
      apt:
        pkg:
          - ntp
        state: present
        update_cache: no
        force_apt_get: True
      register: ntp
    - name: set hw clock to UTC from host
      command: hwclock --set --date "{{ lookup('pipe','date -u') }}"
      when: ntp.failed
    - name: Set timezone to Europe/Berlin
      timezone:
        name: Europe/Berlin
        hwclock: UTC
      register: tz
    - name: Unconditionally reboot the machine to apply tz change
      reboot:
      when: tz.changed
    - name: install ntp
      apt:
        pkg:
          - ntp
        state: present
        update_cache: yes
        force_apt_get: True
      when: ntp.failed
    - name: Make sure NTP is started up
      service: name=ntp state=started enabled=yes
```

Install Packages

- nötige Pakete via apt installieren
- diese Liste wird bei Bedarf ergänzt

| play_packages.yml

```
- hosts: all
  become: yes
  tasks:
    - name: install apt packages
      apt:
        pkg:
          - git
          - python3
          - metastore
        state: present
        update_cache: yes
```

```
        force_apt_get: True
```

Backup /etc as git

- Um Konfigurationsänderungen zu dokumentieren wird das /etc Verzeichnis sowie bei bedarf weitere Pfade zu einem GIT repo hinzugefügt
- Da GIT die Nutzerrechte nicht mit speichert, wird dies extern getan
- Scripte stellen sicher, dass Paketinstallationen Commits triggern oder ohne Commit nicht möglich sind.

| play_git.yml

```
- hosts: all
  become: yes
  vars:
      access_token_decrypt:  !vault |
          $ANSIBLE_VAULT;1.1;AES256
35383539633666323861613033623039646561613965353464366536623833623361623764303531
33336133303939393033386161616533396362643136623 20a656464396239663335393539336434
34386666303363646639363831363134333733386262623462633536326235613335333633653830
34636634623237313 10a663137346332626565656230356138303437353836363133363063373333
          6133333036353532663165393532656665303237333130626233386432643636656 6
      email_decrypt: !vault |
          $ANSIBLE_VAULT;1.1;AES256
31376661383735323533663362363861353337376230363636613462363734643634636634363430
65333933613534616132316338333365663336383461613 60a313361633636373136646530323235
36393136396262323334643333353839623532373938316539363737376263666162326633356364
38383138336134386 10a313761323737363332386262316332666330366437393136613364303631
          3962313965323465323166316330626166386235663935323266633933336131316 5
      gitlab_pass_decrypt: !vault |
          $ANSIBLE_VAULT;1.1;AES256
39376239396166313232393035393864346232343131623563373363636365613038613064333939
30653335336236363762643431363232616337643263636 60a366264326562366539363934636534
31643365343864633335343135383939646238373430376239313034356338386636636634313435
36663823365663064 0a663935633065646638363864613636366335316465663035393232333866
          6230

  tasks:
    - name: Creates directory for git hooks
      file:
        path: /etc/etc-git
        state: directory
    - name: Copy git hooks
      copy:
        src: ./conf/aptpostinstall.bash
        dest: /etc/etc-git/
        owner: root
        group: root
        mode: '0744'
    - name: Copy git hooks2
      copy:
        src: ./conf/aptpreinstall.bash
        dest: /etc/etc-git/
        owner: root
        group: root
        mode: '0744'

    - name: register pre-install hook to dpgk
      lineinfile:
        path: /etc/apt/apt.conf.d/90etc-git
        create: yes
```

```
      regexp: 'aptpreinstall'
      line: DPkg::Pre-Invoke { '/etc/etc-git/aptpreinstall.bash' };

  - name: register post-install hook to dpgk
    lineinfile:
      path: /etc/apt/apt.conf.d/90etc-git
      create: yes
      regexp: 'aptpostinstall'
      line: DPkg::Post-Invoke { '/etc/etc-git/aptpostinstall.bash' };

  - name: Creates .ssh folder
    file:
      path: ~/.ssh
      state: directory

  - name: create ssh key
    openssh_keypair:
      path: ~/.ssh/id_ssh_rsa
    register: sshkey

  - name: "Adding ssh key to gitlab"
    gitlab_user:
      #api_url: https://gitlab-heimatsender.ddns.net/
      api_token: "{{ access_token_decrypt }}"
      server_url: https://gitlab-heimatsender.ddns.net/
      username: Chaos_99
      name: Chaos_99
      password: "{{ gitlab_pass_decrypt }}"
      email: "{{email_decrypt}}"
      sshkey_name: Ansible
      sshkey_file: "{{ sshkey.public_key }}"
      state: present

  - name: Creates temp directory for git clone
    file:
      path: ~/tempgit
      state: directory

  - name: clone homedir git repo in ~
    git:
      repo: 'git@gitlab.heimatsender:Chaos_99/bytecluster.git'
      dest: ~/tempgit
      update: no
      accept_hostkey: true
    register: gitclone

  - name: install homedir
    synchronize:
      src: ~/tempgit
      dest: /
      recursive: yes
    when: gitclone.changed
```

Dauerhafter Link zu diesem Dokument:
https://wiki.technikkultur-erfurt.de/dienste:bytecluster0002-alpha

Dokument zuletzt bearbeitet am: **25.06.2020 14:11**

Verein zur Förderung von Technikkultur in Erfurt e.V
https://wiki.technikkultur-erfurt.de/