

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

ansible.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 beliebige Zeichenfolge in einer Zeile.

Startup

Vorbereitung auf dem Server

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

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

Install Packages

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

play_packages.yml

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

Dauerhafter Link zu diesem Dokument:

<https://wiki.technikkultur-erfurt.de/dienste:bytecluster0002-alpha?rev=1582827393>

Dokument zuletzt bearbeitet am: **27.02.2020 19:16**

Verein zur Förderung von Technikkultur in Erfurt e.V

<https://wiki.technikkultur-erfurt.de/>

