Building reproducible system setups with FreeBSD and ansible

Albert Dengg
<albert@fsfe.org>

Free Software Foundation Europe

2023-05-20

What do I mean by "reproducible" in this context

IN SCOPE

- the ability to a new system to the desired functional state
- the ability to bring a current system (that is in an at least semi defined state) to the new functional state
- to do so in an automated or at least mostly hands off way

OUT OF SCOPE

- having all timestamps etc identical
- really controlling all files on the sytstem
- for that one would more likely have to go to things like Reproducible builds instead of simple configuration managment

So i use configuration managment software to configure the system and I'm done, right?

Ansible

Definition

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.[pc23]

Pro

- automation helps to repeatbly apply the steps to systems in the same order and depending on how many systems one has makes things much faster
- having the steps that get applied in (version controlled) code is at least some documentation what steps where taken to get a system to a specific state

Ansible

Con

- it does not directly define a target step but how you got there
- if you are not carefull your steps might depend on external resoruces that can change
- if you develop your playbooks over time and apply them in chunks your might run into circular dependencies you only realize once you have to rebuild a system from scratch

Start early with automation in the process that defines your systems

- for physical systems, consider automatic things like BIOS/UEFI and IPMI settings
- for VMs/Jails/... consider automating the creation/modification of these

Start early with automation in the process that defines your systems

Examples

- dellemc.openmanage.idrac_server_config_profile module
- community.general.proxmox_kvm modulecommunity.general.proxmox_kvm module
- . . . or script it in another way (you can also script rest api's using ansible . . .)

External Dependencies

Possible Problems

- Packages / Ports
- files downloaded from external sources
- iternal resources that are not directly tracked with the playbook
  - internal gitlab
  - internal mirrors
  - files/folders on the deployment machine not within the playbook structure
  - ...
- ssh server keys of servers accessed by the playbook
- ...

External Dependencies

Possible Solutions

- Packages
  - quaterly vs latest
  - fixing package vs relying on the latest version availible
  - using your own mirror / poudriere server to control the availible versions
- files downloaded from external sources
  - use internal mirrors if sensible
  - use checksums to at least fail in a defined way if the file is not what you expected
- iternal resources that are not directly tracked with the playbook
  - is it nessecary to be doing it that way?
  - defined processes to ensure integrity
  - checksums, see above
- ...

Editing Files

- specialiced modules
- line/blockinfile
- templating
- unpacking archives
- running commands to generate content
- editing files vs adding files to include directories
- using filters to transform variable content into file content
- copying
- ...

Editing Files

copying

- from local machine
- inline, static or with templating

Editing Files

templating

- uses Jinja2
- not shure if it is actually turing complete or not, but it feels close
- can be used for file templating (template module)
- at least most of it can also be used inline in strings in a lot of places

Editing Files

running commands to generate content

- if there is no module availible for what you want to do, you can always run command via
  - command
  - raw
  - shell
  - expect
- however, portability becomes a bit more of an issue
- writing your playbook in a way where it is obvious if a certain step changed anything on the machine or not becomes harder

Editing Files

editing files vs adding files to include directories

- splitting the configuration in multiple files can be great
  - simpler templates
  - less problems if for one reason or another parts of the configuration come from diffrent roles
  - ...
- ...but how to deal with removals to prevent leftover pieces lying around that might or might not affect something?

Filters

Can be used to

- filter/select data
- sort data
- transform data
- format data
- are easy to write yourself in python if no suitable one exists

Lookup Plugins

looking up external data

- file listings
- file contents
- content from urls
- enviroment variables
- ...

complex lookups

- command output
- randomization
- more complex dict/list lookups
- ...

Connections

Connection Plugins

By default, Ansible uses ssh using OpenSSH, however there are other options availible:

- running command on the local machine directly
- paramiko
- some windows specific stuff
- custom plugins like ansible-sshjail for managing jails on remote hosts without sshd in the jails

Connections

Delegations

You might want/have to run commands on other hosts then the target you are currently configuring, for example

- calling api's from localhost that might not be reachable from the target host
- preparing the backupserver to recieve backups from the new server
- ...

Inventories

Static vs. Dynamic

Ansible has a few different formats to describe the inventory statically, why would you want to do a dynamic inventory?

Inventories

Static vs. Dynamic

Ansible has a few different formats to describe the inventory statically, why would you want to do a dynamic inventory?

Inventories

Dynamic

- if the specified inventory is executable it will be executed
- it simply has to spit out the data formated as json
- usefull for example if you want to get the inventory and host/group variables out of a single-source-of-truth system
- . . . or want to apply updates to all your jails/vms (for a lot of such usecases there are existing plugins)

Handling events after stuff was changed

- registering states of commands into variables and then use that as a precondition to do stuff
  - output and/or return state can be used as a filter
  - can be used in combination with loops
  - can be used to immediatly run actions based on the results
- handlers
  - will be run bunched up
  - might save you from having to restart services multiple times
  - if you have multiple handlers the ordering is not really guaranteed

Playbook Development

Circular Dependencies

Be awaere with iterative devlopment of playbook and possible circular dependencies you might generate you only notice once you have to rebuild the machine...

Playbook Development

Enviroment

- ansible setups
  - system packages
  - venvs
  - AWX
- syncronizing playbooks

Sources I

[pc23]


Ansible project contributors.
Ansible documentation, 3 2023.
https://docs.ansible.com/ansible/latest/index.html.

Questions?

THANK YOU