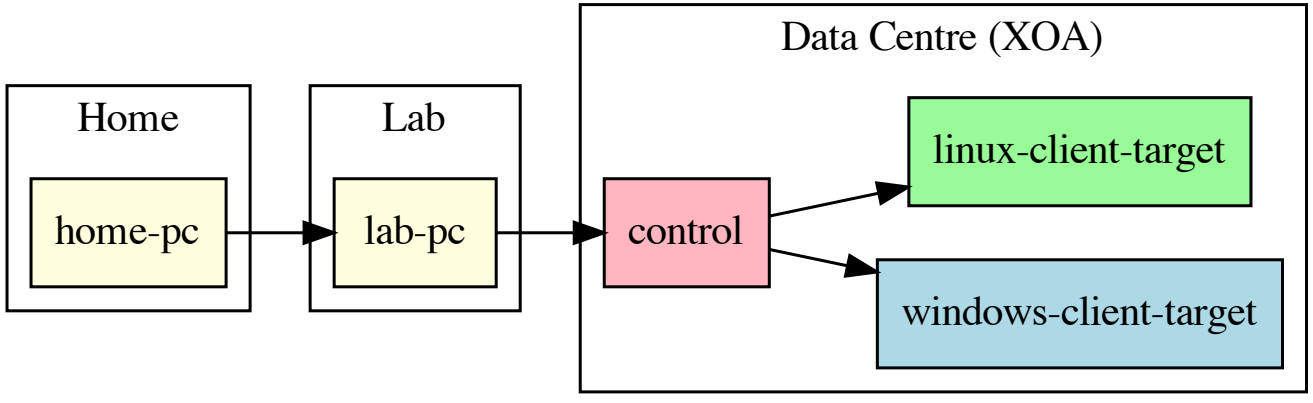
Practices

Dr Peadar Grant

November 14, 2024

Contents

1 Scenario


Home
home-pc
Lab
lab-pc
Data Centre (XOA)
control
linux-client-target
windows-client-target

2 Infrastructural practices

2.1 Addressing

Your target nodes are inventoried as either IP addresses or hostnames.

Sometimes a playbook will reboot a host:

- Best to ensure that hosts always receive the same IP address.
- DHCP reservation is better than a static address.
- Dynamic DNS registration also a possibility (not for here!)

Ensure hosts receive same IP address

- Hosts should have a DHCP reservation.
- Consider generating inventory file dynamically from DNS / DHCP files.

2.2 Uniformity

While ansible itself is cross-platform, its modules are not.

For example, in terms of package management:

- Debian and Ubuntu use `apt`
- RedHat uses `yum`
- Windows internally uses MSI files

As we have seen so far, the same operations on Windows and Linux are handled by different ansible modules.

Standardise where possible

- When you have the opportunity to standardise your environment, take it!

2.3 Hostnames

We often use hostnames instead of IP addresses.

Although the aim of automation is to reduce manual intervention, you can often find yourself logged in to multiple similar hosts:

- The only way to distingush them is by means of their displayed hostname.
- This will only work if the hostname has been set.
- Can happen during OS installation
- Or can be set by DHCP client if received from DHCP server.

Set the hostname!

- Manually using `hostnamectl`
- Or by `hostname` module in Ansible playbook

3 Coding practices

3.1 Version control your code

Your playbook(s) are living documents:

- You need to be able to experiment with changes.
- Bugs / regressions can creep in when changes are made.
- Requirements change over time.
- Syntax and other mistakes can be amde.

You need to be able to see changes between versions and roll back to previous versions.

Use version control

- Your playbooks should be kept in a version control system.
- Highly recommend `git` as used throughout industry.

3.2 Editor

Playbooks are written in YAML / YML syntax.

Don't even try to edit this in Word!

Use a suitable editor

- At a minimum, use an editor that can syntax highlight YML.
- Better still if it can automate some YML formatting as well.

I use `emacs` with `yaml-mode` on both linux and windows.

3.3 Check syntax

Ansible has a `--check` option.

This depends on the implementation of the dry run / check operation in the modules used in the playbook.

Less useful if there are conditional / dependent operations in the playbook.

4 Playbook practices

4.1 Handling reboots

Reboots are sometimes required or recommended following certain system updates.

Ansible can initiate and wait for a reboot on a target node.

Smart reboots

- Reboots should normally be conditional (if possible)
- Reboots won’t work when running a playbook against the localhost

4.2 Avoid shell commands

On both Windows and Linux you can use Ansible to execute a shell command as if in a Bash / PowerShell script.

Use ansible native tasks

- Replace any shell commands by ansible native tasks, where possible