LS Lab Assignment: Infrastructure as Code

Paul Danielse Maurice Mouw Koen Greuell Shashikant Ilager

Feedback deadline:
December 11, 2024 23:59 CET

Abstract

This lab assignment will let you practice with the concept of Infrastructure as Code. We use the hypervisor setup created in the first week of lab assignments to provide the "hardware". You can choose which configuration management software you want to use. For beginners we recommend Ansible. For something more exotic than Puppet or Ansible, please consult with the lab teacher first. We refer to your tool of choice as CMT in the rest of this document. You can also use e.g. scripting, as long as the whole process stays automated and stored in your repository. You should work alone on this assignment. For this lab, it is more important that you demonstrate your understand the IaC concepts and tools, than to have perfectly working server configurations.

1 Version Control

With the Infrastructure as Code concept, the configuration of your infrastructure (servers, switches, routers) is stored in a source code repository that supports versioning. For this exercise we will use the Git repositories provided by `gitlab.os3.nl`.

If you are unfamiliar with Git or GitLab, read up on these topics first.

Questions

1. Create a new repository on our GitLab server to store your configuration files.
   (a) Checkout this repository on your desktop.
   (b) Commit a README file to the repository, and push it to the server. Provide a screenshot of the GitLab GUI to show this worked.
   (c) Make sure the lab teachers have read access to this repository, and publish its URL on your Wiki.

2 Basic Infrastructure

You are starting a new Master education, called Closed Source Rules, or CSR. You have registered the DNS domain `csr4.nl`, bought a physical server and installed a KVM hypervisor with a Linux Dom0 on it. Now you want to create some infrastructure on this server to run your education. To be precise you want to have:

- An authorative DNS server for `csr4.nl`
- A Web server for your domain that shows a basic page about your new education.

- A mail server for your domain.

2. Create 2 virtual machines on your hypervisor.
3. Write a configuration script for your chosen CMT that creates the above infrastructure. The DNS and mail server should go on one VM, the Web server on the other. Each service should get a DNS entry in the authoritative DNS server. (You do not have to create the VMs with your CMT, that is a bonus question later.)
4. Store this configuration in your Git repository.
5. Deploy the infra using your CMT, and show that **one** of the services works. Do everything from your desktop, do not make any manual changes on the VMs.

3 Students

What is an education without students? To support actual students you will need 3 things

1. A home page for each student on your Web server
2. A mail account for each student on your mail server
3. A desktop system for each student

Questions

6. Extend the configuration file for your CMT to also create the home page and mail account for each student. Assume there are 25 students, each with a unique user name. You are not required to set up a central authentication service like LDAP for this, but you may do so, of course.
7. Show the resulting configuration files that your CMT installed on the VM.
8. BONUS: Show the home page of a student and that the mail server accepts mail for a student.

For the desktops, you order 25 machines from Dell. Dell send you a list of service tags (needed to report issues to Dell), and the MAC address of the Ethernet interface, as a `.csv` file. For example:

```
#ServiceTag,MainMACAddress
3Y6VSL5,16:ab:ee:00:00:01
3Z2UHD2,16:ab:ee:00:00:02
...
```

As you cannot afford real IPv4 addresses, you decide to use the 10.0.0.0/8 range as the IP space for your desktops.

9. Extend your infrastructure with a DHCP server for the desktops, via Git and your CMT. Each desktop should get a fixed IP address. Think about how you can autogenerate such an IP assignment from the `.csv` provided by Dell. Also think about the fact that you may want to renumber, or extend the IP assignment with IPv6 addresses in the future. The DHCP server should run on the DNS VM. *Hint:* `https://wiki.debian.org/DHCP_Server`
10. BONUS: Add an caching name server to your DNS VM for the desktops to use. Make sure the DHCP reply they receive points to this caching server.

11. Deploy the DHCP server on the VM and show the resulting configuration files that your CMT installed on the VM.

12. Extend your DNS configuration with Git and your CMT to assign a DNS name to each of the desktops. Again think about future changes. At a minimum generate a zone file for your DNS server. *Hint:* `https://en.wikipedia.org/wiki/Zone_file`

13. Show the resulting DNS zone files on the VM.

4 Student Servers

But wait, students also need servers to run their Closed Source Operating Systems. So you order 25 Dell servers, and Dell sends you another list of service tags, main Ethernet interfaces, and the MAC address of the DRAC remote management module you had them install, as another `.csv` file. For example:

```
#ServiceTag,MainMACAddress,DRACMACAddress
5Y6VSL5,16:ab:ee:02:00:01,17:ab:ff:02:00:01
5Z2UHD2,16:ab:ee:02:00:02,17:ab:ff:02:00:02
...
```

14. Rewrite your IaC configuration such that all servers
    (a) get a fixed IP address from the 10.0.127.0/24 range for their main Ethernet interface.
    (b) get a fixed IP address from the 10.0.42.0/24 range for their DRAC interface.
    (c) a unique DNS name (you may choose the name space (wisely!))
    (d) a `drac.<servername>` DNS name for the DRAC's IP address.

    Make sure this configuration can be easily extended when more student servers are added!

15. Deploy these configuration changes with your CMT, and show the resulting DHCP and DNS configuration files on the VM.

5 Disaster Strikes!

Someone accidentally deletes your two VMs.

16. Restore your two VMs from the configuration stored in your Git repository.

17. BONUS: Recreate your VMs using a tool like Terraform, or KVM integration for your CMT.