Automated setups for devops and data science

Piet Stam

2025-06-01

Table of contents

1 Introduction to YAML Scripts

This is a Quarto document that describes the YAML scripts that you can find in the subdirectory scripts of this repo. There is a chapter for each script, and the chapter contains a description of the script, its purpose, and how to use it.

1.1 Cloud-config scripts

The scripts in this repository are written in YAML format and can be used as user data when launching cloud instances on various cloud providers, such as Hetzner or DigitalOcean. The contents of the user data field are passed to the instance as a file, which is then processed by the cloud-init service. This allows you to automate the setup of your instance, such as installing software, configuring services, and running scripts. Cloud init is the standard for customising cloud instances.

1.2 Requirements

Check these things manually first: - Make sure the R apt source matches the system (see https://cran.rstudio.com/bin/linux/ubuntu/fullREADME.html for names) - If running on smaller droplet uncomment the section that enlarges swapfile - Optional: add SSH public key for user account

1.3 Scripts

You can find the scripts in the `scripts` directory, or click here:

- `analysis.yaml`: Installs R (with the hadleyverse and close friends) + Python 3 + numpy + pandas + Docker
- `openvpn.yaml`: Installs and configures an OpenVPN server and generates a ready-to-use client certificate
- `shiny-server.yaml`: Installs R + Nginx + Shiny
- `pptx2md.yaml`: Installs pptx2md

1.4 Documentation

You can find the explanations of the scripts in the documentation that is created with the help of Quarto and GitHub Pages. See the link in the About section of this repository for the documentation site.

1.5 Sources

The original code was adapted from various sources, including the scripts in this respository by Andrew Heiss and this tutorial by the Digital Ocean community. Also, I benefited from the results of penetration tests that have been performed for acquiring ISO and NEN certifications for my consulting firm Equalis.

2 analysis.yml

2.1 The script

- Adds CRAN and Docker APT sources
- Installs packages for:
  - Python (pip, libraries)
- R (base, dev tools, packages)
- Installs Docker and Docker Compose
- Add security (fail2ban, ufw)
- Creates a user piet with sudo and Docker access
- Configures SSH on port 4444
- Saves a log as /var/log/cloud-init-output.log
- Sets up fail2ban and a firewall
- Installs Python and R packages
- Optionally adds swap space for R
- Installs a large set of R packages
- Installs Node.js for JavaScript-based services
- Installs npm for JavaScript package management
- Installs Express.js generator
- Creates a simple marker file to verify cloud-init completion

2.2 References

Help on cloud-init syntax from:

- https://cloudinit.readthedocs.io/en/latest/topics/examples.html
- https://www.digitalocean.com/community/tutorials/how-to-use-cloud-config-for-your-initial-server-setup
  - ERROR: THE `ssh-authorized-keys` (AND `sudo`) SYNTAX IS WRONG **IN ALL DIGITAL OCEAN EXAMPLES**, SEE NEXT REFERENCE

- https://superuser.com/questions/1725127/invalid-config-for-cloud-init-but-apparently-still-works-how-do-i-remove-the

Help on R and Docker installation syntax from:

- http://deanattali.com/2015/05/09/setup-rstudio-shiny-server-digital-ocean/
- https://www.digitalocean.com/community/tutorials/how-to-set-up-r-on-ubuntu-14-04
- https://gist.github.com/mdlincoln/1f40f4e88ce32c55b5f3
- https://gist.github.com/mdlincoln/1f40f4e88ce32c55b5f3#gistcomment-3009981
- https://cran.rstudio.com/bin/linux/ubuntu/fullREADME.html#using-apt-key
- https://github.com/rocker-org/hadleyverse/blob/master/Dockerfile
- https://stackoverflow.com/questions/24418815/how-do-i-install-docker-using-cloud-init

Detailed explanations of lines of code for Node.js setup:

- `git`: Essential for version control and cloning repositories.
- `nodejs` and `npm`: Required for running and managing JavaScript packages.
- `npm install -g npm@latest`: Ensures you're using the latest npm version, which can fix bugs and improve compatibility.
- `express-generator`: A helpful tool for quickly scaffolding Express.js applications.

3 openvpn.yml

3.1 The script

- Creates a new user named "vpn" with password-less sudo capabilities
- Makes SSH available on port 4444

3.2 Client configuration

OpenVPN is configured automatically and the client certificate is available at `/home/vpn/client.ovpn` at your newly installed OpenVPN server. Then follow these steps:

1. Transfer this ovpn file to your client (Windows: `C:\Users<USERNAME>\OpenVPN\config`) using scp like so: `scp -P 4444 vpn@IPADDRESS:/home/vpn/client.ovpn client.ovpn`
2. Install the OpenVPN client.
3. Run the OpenVPN client (see the system tray in Windows).
4. Click yes if asked to add yourself to the "OpenVPN Administrators" group (administrator password required)

3.3 References

- https://github.com/Nyr/openvpn-install

4 shiny-server.yml

4.1 Set up a Shiny server on Ubuntu 20.04

- without SSL certificate
- see https://cloud.r-project.org/bin/linux/ubuntu/ for the choice of the Ubuntu version
- also check the shiny server version number at https://www.rstudio.com/products/shiny/download-server/ubuntu/

4.2 Checks

Some checks to do after the cloud-init script has finished:

- (SSH) Is SSH key added? `sudo nano ~/.ssh/authorized_keys`
- (SSH) Has port changed from 22 to 4444? `sudo nano /etc/ssh/sshd_config`
- (Nginx) Is the content of the server block written to the file that is included in the Nginx config file `/etc/nginx/conf.d`? `sudo nano /etc/nginx/sites-enabled/default`
- Are .deb files deleted? `sudo ls -l /var/cache/apt/archives`
  - If not, apply sudo apt-get clean (source: https://www.jverdeyen.be/ubuntu/digital-ocean-ubuntu-free-up-disk-space/)
- (sed) Check sed syntax at https://sed.js.org/index.html, this tool was used to solve the issue with port 4444 (see commit w/ hash `fcb426dd536952ce929329a865d737660d12f112`)

4.3 References

Help on cloud-init syntax from:

- https://cloudinit.readthedocs.io/en/latest/topics/examples.html
- https://www.digitalocean.com/community/tutorials/how-to-use-cloud-config-for-your-initial-server-setup
  - (ERROR: THE `ssh-authorized-keys` (AND sudo) SYNTAX IS WRONG **IN ALL DIGITAL OCEAN EXAMPLES**, SEE NEXT REFERENCE)

- https://superuser.com/questions/1725127/invalid-config-for-cloud-init-but-apparently-still-works-how-do-i-remove-the
- https://github.com/andrewheiss/cloud-config-files

Help on Shiny server installation syntax from:

- http://deanattali.com/2015/05/09/setup-rstudio-shiny-server-digital-ocean/
- https://www.digitalocean.com/community/tutorials/how-to-set-up-shiny-server-on-ubuntu-20-04
- https://gist.github.com/mdlincoln/1f40f4e88ce32c55b5f3
- https://gist.github.com/mdlincoln/1f40f4e88ce32c55b5f3#gistcomment-3009981
- https://cran.rstudio.com/bin/linux/ubuntu/fullREADME.html#using-apt-key

5 pptx2md.yml

5.1 Prerequisites

- Use python 3.8 or 3.9 (not 3.10, see issue https://github.com/scanny/python-pptx/issues/770)
- Execute pptx2md as a root (see issue https://github.com/ssine/pptx2md/issues/25)

5.2 The script

- Creates an Ubuntu server
- Installs pptx2md
- Converts a Powerpoint pptx file into markdown

5.3 Manual steps

- After the install, create a subdirectory called tmp
- chdir tmp
- Upload the pptx file
- Excute the following cmd: `/usr/bin/python3.8 /usr/local/bin/pptx2md filename.pptx`

5.4 References

- The tool to convert a Powerpoint pptx file into markdown: https://github.com/ssine/pptx2md
- How to Install and Switch Python Versions on Ubuntu 20.04: https://www.rosehosting.com/blog/how-to-install-and-switch-python-versions-on-ubuntu-20-04/