



INSTITUTE *of*  
TECHNOLOGY  
CARLOW

Institiúid Teicneolaíochta Cheatharlach

April 30<sup>th</sup>, 2021

AUTOMATION OF  
NETWORK/SERVER SECURITY  
PROVISIONING USING DEVOPS  
TOOLS.

FINAL REPORT

STUDENT: KATIE BROPHY C00224531  
SUPERVISOR: JAMES EGAN

## *TABLE OF CONTENTS*

<i>Project overview.....</i>	<i>2</i>
<i>What you ACHIEVED. ....</i>	<i>3</i>
Test Environment:.....	3
Automations: .....	3
<i>What I dint get working .....</i>	<i>4</i>
The GUI.....	4
Ansible.....	5
Python .....	5
Virtual Machines/environment.....	5
<i>Problems encountered and how they were RESOLVED.....</i>	<i>5</i>
<i>What you would do differently if starting AGAIN. ....</i>	<i>6</i>
Planning the test environment: .....	6
Time management: .....	6
Do more research on django & Web Development .....	6
<i>Testing used to assess reliability of your software product. ....</i>	<i>6</i>
<i>Aknowledgements.....</i>	<i>7</i>
<i>BIBLIOGRAPHY .....</i>	<i>7</i>

# PROJECT OVERVIEW

The tool allows the user to run scripts that will automate the configuration, provisioning and security of networks and servers using DevOps principles and Ansible.

DevOps is a set of procedures that blends software development and IT operations. It aspires to reduce the systems development life cycle and provide continuous delivery with high software quality. [DevOps, 2020] Configuration management is the practice of sustaining and creating the performance of the products by maintaining its physical elements, functional elements, design, requirements, and operational information throughout its life cycle.

Manual creation, configuration and management of servers and network infrastructure is time consuming, repetitive, complex, and prone to errors. Using DevOps tools to automate the process reduces errors, speeds up provisioning and reduces costs.

The tool allows the user to create and control 3 key areas in their operations environment. First there is IT automation to automate the setup that would typically be done manually allowing for more efficiency. The second is configuration that is consistent; it will allow precise configuration that can be deployed to 100s of servers and devices that is guaranteed to replicate identically. Then there is deployment, for scalability in the servers so that new servers can be set up and the scripts will push configurations and provisioning to them automatically allowing for speed and efficiency in an operations team.

The tool achieves this by modelling push configuration rather than the traditional pull configuration method. There is no need for the remote servers to have client installed to try pull config from the main node, instead using Ansible the configurations are automatically pushed to the remote servers.

Ansible architecture employs a local 'control' node, that connects to the remote servers. This control node manages the modules and the inventory. The modules are the configuration playbooks containing instructions to be pushed to the servers and the inventory is what allows the local machine to connect to the remote hosts in the environment through SSH client.

The Ansible playbooks are code written in YAML, the code contains tasks or 'plays' that specify the configurations to be applied on the remote hosts.

The inventory specifies the node name, e.g., webserver and that name then points to the name/ IP address of the host. This node name is referenced in the 'hosts:' module in the Ansible playbook, this is what tells the script where to push the configurations and how to connect to the remote host.

```
*hosts x servers.yaml
1 [webservers]
2 192.168.1.10 ansible_ssh_user=kate ansible_ssh_pass=xxxxx
   ansible_sudo_pass=xxx|
```

```
1 ---
2 - hosts: webservers
3   become: yes
4   become_user: root
5   gather_facts: false
6   tasks:
7
8     - name: INSTALL PYTHON
9       apt: name=python3 state=present
10
11    - name: INSTALL pip
12      apt: name=python3-pip state=present
13
```

## WHAT YOU ACHIEVED.

### TEST ENVIRONMENT:

I created a virtualised test environment with 2 Ubuntu Virtual Machines and a Cisco Cloud services router Virtual Machines. I created an internal network for these machines and enabled NAT to allow internal and external communication.

#### Control Node:

I created an ubuntu machine for the ansible architecture to reside on. I installed and configured SSH and ansible on this machine and configured the Modules and Inventory.

#### Remote Host:

I created Remote Linux Server and cloud services router to act as nodes to push the playbooks to.

### AUTOMATIONS:

#### *backup\_router.yaml*

This first playbook connects to a cloud services router through SSH. It then creates a ‘backups’ directory on the local machine. It checks the configurations of that router and saves them to this directory. This can provide 2 major benefits to the operations team. First it allows quick and efficient scalability – this configuration can easily be pushed to new routers as required, second it can allow for speed in incident response; when an event occurs that causes the router to go down/be compromised it can easily be isolated from the network and replaced with a new router that has identical configurations.

#### *config\_router.yaml*

This playbook connects to a router through SSH and applies configurations to it.

### *config\_server.yaml*

This playbook connects to a remote server through SSH and configures it. It escalates the privilege to root. It installs python and pip to the remote host. It also installs an Apache web server and configures it, it then installs MySQL server, MySQL client and PyMySQL and configures them. The playbook creates a secure user account for the MySQL server. It then creates databases and tables using these credentials. Then it deploys a secure web application to the web server.

### *backup\_server.yaml*

This playbook connects to the remote server through SSH and it escalates the privilege to root. It then logs in using the secure credentials and creates a backup of the databases and tables and stores them.

### *rebuild\_server.yaml*

This playbook connects to the remote server through SSH and it escalates the privilege to root. It then logs in using the secure credentials and then rebuilds the databases and tables using the backup files created in the script above.

## WHAT I DINT GET WORKING

### THE GUI

I did not manage to get the GUI implemented as planned to time constraint and the technical challenged faced with the virtual environment.

What you learned

## ANSIBLE

There was a lot of research and learning involved in learning Ansible.

## PYTHON

Prior to starting this project, I had never worked with Python programming language. While it was difficult to learn, it will benefit me greatly. Python is a very useful and comprehensive language.

## VIRTUAL MACHINES/ENVIRONMENT

The issues faced in this project lead to an in-depth learning of virtual machines and environments.

# PROBLEMS ENCOUNTERED AND HOW THEY WERE RESOLVED.

The main constraint faced implementing this project was the creation and configuration of a test environment for it to run on. I faced many issues with virtual machines. They would often crash without explanation requiring a complete reinstall and often brand new images. This was difficult to overcome and took quite a bit of time from the project work.

Once ansible was installed on the virtual machine the next challenge faced was to set up 'remote' hosts to push configurations to. In many examples this is achieved using Vagrant on the machine. I faced issues with this and got many errors. I attempted to resolve these errors by enabling VT-x in the BIOS, my laptop is college issued and I had to get a college administrator to edit the BIOS settings. VT-x was already enabled, so the error was confusing to all. After more research I discovered that VT-x cannot be nested with a 64-bit OS.

After This I began using cisco's DEVASC virtual machine and cloud services router and began running playbooks for routers. However, this scenario was very limited as this virtual machine can only interact with the router and could not be altered to interact with another virtual machine acting as a server.

I then began using two virtual machines, one as the control node and one as the remote host. This required enabling an internal network adapter, changing the access, and configuring IP addresses to each machine. Then I had to enable a NAT adapter to allow communication to the web. After doing this I successfully installed Ansible and began writing playbooks.

Aside from these technical errors with the virtual machines, they were incredibly unreliable in general. Throughout implementation of the project, the Virtual machines would crash with no explanation. I often had to set them up from scratch completely putting a strain on the timeframe. Also, when working with 2 identical virtual machines using the same ubuntu image, commands, settings, and scripts would sometimes only work on one machine.

The last big challenge I faced was late in the week of submission. There was an issue with SSH connection between the control node and the remote host. This issue had not been there a couple of hours prior, and no changes had been made in that timeframe. This took over 24 hours to troubleshoot and required extensive research. Any solution I found did not match my issue, I had the correct SSH configuration files, paths, and permissions. I re-installed the virtual machines and ansible many times also. On one of the finals Re installs the issue was solved, but

there were no other changes made to resolve the issue – just that reinstall worked unlike the previous attempts.

However, when the SSH connection error was resolved, there were errors thrown in the ansible scripts that had been previously working. There was a lot of troubleshooting involved in this too again putting a strain on the project schedule.

These issues and the time strain they caused had a negative impact on fully implementing the GUI.

## WHAT YOU WOULD DO DIFFERENTLY IF STARTING AGAIN.

If I had the chance to start working on this project from the start, there are a few things I would change to achieve more implementation.

### PLANNING THE TEST ENVIRONMENT:

During November and December 2020, I spent a lot of time researching ansible and planning scenarios I could implement. This learning and researching were significant, and the information found was very consistent. Most scenarios and examples I discovered were identical and I underestimated the challenges I would face just to set up a test environment.

As outlined in the issues above, the environment used was very unreliable and inconsistent and it was often hard to decide if the issue was in the virtual machine itself or in the code. The time spent researching solutions and fixing the issues caused significant delays. If starting again I would have sought guidance from my supervisor sooner and researched more into the test environment before beginning

### TIME MANAGEMENT:

One challenge I faced in this project was time. Unfortunately, some of it due to remote learning, which involved more assignments with high weightings. However I feel I could've managed time better in the beginning of the year by being more consistent in my work and focusing on a single module at a time. This did manage better as the year went on, and achieving this balance earlier would've helped.

### DO MORE RESEARCH ON DJANGO & WEB DEVELOPMENT:

Web development is one of my weakest areas of IT, and I could have spent more time researching into these areas to have produced a better implementation. My plan had been to implement the automation first and then connect it to the web application, however I underestimated the time it would take to achieve this automation due to the challenges faced above.

## TESTING USED TO ASSESS RELIABILITY OF YOUR SOFTWARE PRODUCT.

The project was tested by running the playbooks and confirming the carried out the instructions

## ACKNOWLEDGEMENTS

Throughout this project, and my time at IT Carlow I have received a great deal of support and assistance.

I would first like to thank my supervisor, Dr James Egan, your knowledge, support, and advice was invaluable in completing this project. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would like to thank my lecturers, for your support and guidance since first year especially Richard Butler and Keara Barrett, you both went above and beyond to support my work and learning, it is very much appreciated.

To my colleagues and best friends Hazel and Síne, I am incredibly grateful for all your support, advice and guidance, in this project and in every project, exam and challenge we faced. Without you both I could not have made it through the last 4 years.

Finally I would like to thank Dad, Mam, Nana, my aunt and uncles, Mike and Puggins for all your support.

## BIBLIOGRAPHY

[DevOps,2020]	En.wikipedia.org. 2020. <i>Devops</i> . [online] Available at: < <a href="https://en.wikipedia.org/wiki/DevOps#cite_note-loukides-2012-2">https://en.wikipedia.org/wiki/DevOps#cite_note-loukides-2012-2</a> > [Accessed 13 November 2020].
[Chef Tutorial: Components and Configuration Management, and More Explained, 2020]	Simplilearn.com. 2020. <i>Chef Tutorial: Components And Configuration Management, And More Explained</i> . [online] Available at: < <a href="https://www.simplilearn.com/chef-tutorial-article">https://www.simplilearn.com/chef-tutorial-article</a> > [Accessed 12 November 2020].
[Puri, 2020]	Puri, M., 2020. <i>A Complete Beginner'S Guide To Chef And Infrastructure As Code</i> . [online] freeCodeCamp.org. Available at: < <a href="https://www.freecodecamp.org/news/an-introduction-to-chef-and-infrastructure-as-code-7d8ad2689b8/">https://www.freecodecamp.org/news/an-introduction-to-chef-and-infrastructure-as-code-7d8ad2689b8/</a> > [Accessed 12 November 2020].
[What is Chef   Automate Infrastructure Configuration, 2020]	Apachebooster Blog: <i>What Is Chef   Automate Infrastructure Configuration</i> . [online] Available at: < <a href="https://apachebooster.com/blog/automate-infrastructure-configuration/">https://apachebooster.com/blog/automate-infrastructure-configuration/</a> > [Accessed 12 November 2020]. Showcasing the tech blogs written by our writers. 2020.
[What is a Puppet in DevOps?   Why Puppet Software for DevOps?, 2020]	Staragile.com. 2020. <i>What Is A Puppet In Devops?   Why Puppet Software For Devops?</i> . [online] Available at: < <a href="https://staragile.com/blog/puppet-in-devops">https://staragile.com/blog/puppet-in-devops</a> > [Accessed 12 November 2020].
[Sheet and Sheet, 2020]	Sheet, P. and Sheet, P., 2020. <i>Puppet Cheat Sheet - Download In PDF &amp; JPG Format - Intellipaat</i> . [online] Intellipaat Blog. Available at:



	< <a href="https://intellipaat.com/blog/tutorial/devops-tutorial/puppet-cheat-sheet/">https://intellipaat.com/blog/tutorial/devops-tutorial/puppet-cheat-sheet/</a> > [Accessed 12 November 2020].
[Anon, 2020]	Puppet.com. 2020. [online] Available at: < <a href="https://puppet.com/docs/continuous-delivery/3.x/cd_architecture.html">https://puppet.com/docs/continuous-delivery/3.x/cd_architecture.html</a> > [Accessed 12 November 2020].
[Ansible, 2020]	Ansible, R., 2020. <i>Ansible For Provisioning</i> . [online] Ansible.com. Available at: < <a href="https://www.ansible.com/use-cases/provisioning">https://www.ansible.com/use-cases/provisioning</a> > [Accessed 12 November 2020].
[Kolappan, 2020]	Kolappan, K., 2020. <i>Automating Infrastructure Deployments [IAAS] In AWS Cloud With Ansible</i> . [online] Oneglobesystems.com. Available at: < <a href="https://www.oneglobesystems.com/blog/automating-infrastructure-deployments-iaas-in-aws-cloud-with-ansible">https://www.oneglobesystems.com/blog/automating-infrastructure-deployments-iaas-in-aws-cloud-with-ansible</a> > [Accessed 12 November 2020].
Ansible, 2020	Ansible, R., 2020. <i>Red Hat Ansible Network Automation - Red Hat Ansible</i> . [online] Ansible.com. Available at: < <a href="https://www.ansible.com/use-cases/network-automation">https://www.ansible.com/use-cases/network-automation</a> > [Accessed 12 November 2020].
[Network Automation with Ansible, 2020]	Slideshare.net. 2020. <i>Network Automation With Ansible</i> . [online] Available at: < <a href="https://www.slideshare.net/atarsha/network-automation-with-ansible-banog-meetup">https://www.slideshare.net/atarsha/network-automation-with-ansible-banog-meetup</a> > [Accessed 12 November 2020].
[Ansible, 2020]	Ansible, R., 2020. <i>Ansible For Configuration Management</i> . [online] Ansible.com. Available at: < <a href="https://www.ansible.com/use-cases/configuration-management">https://www.ansible.com/use-cases/configuration-management</a> > [Accessed 12 November 2020].
Ansible, 2020]	Ansible, R., 2020. <i>Red Hat Ansible   Security Automation</i> . [online] Ansible.com. Available at: < <a href="https://www.ansible.com/use-cases/security-automation">https://www.ansible.com/use-cases/security-automation</a> > [Accessed 12 November 2020].
[Johari, 2020]	Johari, A., 2020. <i>Chef Vs Puppet Vs Ansible Vs Saltstack: Which One To Choose   Edureka</i> . [online] Edureka. Available at: < <a href="https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/">https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/</a> > [Accessed 12 November 2020].
[“Security in Django   Django Documentation   Django”]	Security in Django   Django Documentation   Django.” Docs.Djangoproject.com, docs.djangoproject.com/en/3.1/topics/security/. Accessed 12 Nov. 2020.
[“Flask vs Django in 2020: Which Framework to Choose?”]	“Flask vs Django in 2020: Which Framework to Choose?” Hackr.io, hackr.io/blog/flask-vs-django. Accessed 13 Nov. 2020.