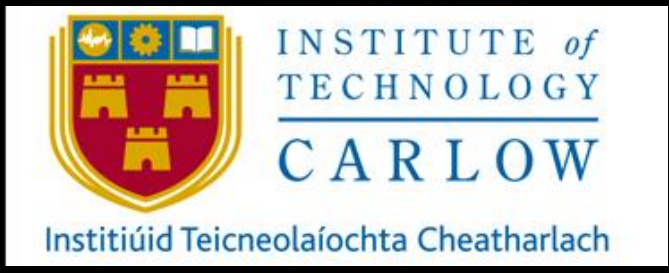



April 30th, 2021

# AUTOMATION OF NETWORK/SERVER SECURITY PROVISIONING USING DEVOPS TOOLS.

## FUNCTIONAL SPEC

STUDENT: KATIE BROPHY C00224531
SUPERVISOR: JAMES EGAN

*TABLE OF CONTENTS*

# DEFINE THE PROJECT.

## WHAT IS THE TOOL?

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 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 servers.yaml

```
[webservers]
192.168.1.10 ansible_ssh_user=kate ansible_ssh_pass=xxxxx
ansible_sudo_pass=xxx
```

```yaml
---
- hosts: webservers
  become: yes
  become_user: root
  gather_facts: false
  tasks:

    - name: INSTALL PYTHON
      apt: name=python3 state=present

    - name: INSTALL pip
      apt: name=python3-pip state=present

```

# WHAT DOES THE TOOL DO?

## AUTOMATING CONFIGURATION MANAGEMENT

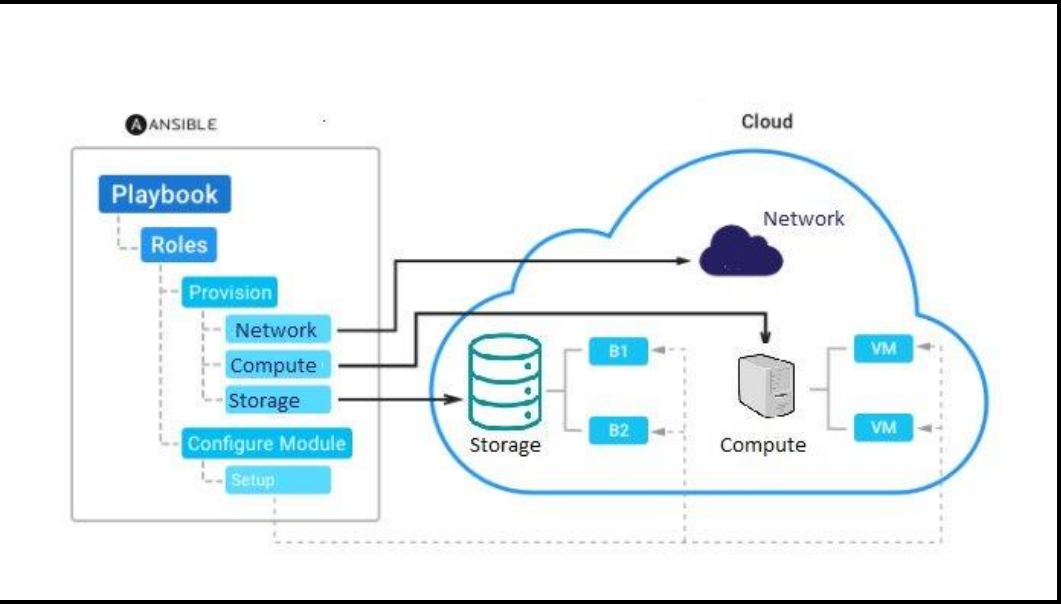
Most system's configuration management is either handled manually with a combination of numerous scripts and impromptu practices carefully selected by the administrators, or automatically with complex high maintenance automation frameworks. This poses a challenge that is growing alongside the virtualization of systems and the complex cross-platform management that requires. [Ansible, 2020]

The application will allow the user to solve this using Ansible playbooks that are accessible, consistent and secure. The configurations are human readable and machine-parsable, requiring minimal training for administrators, IT managers and developers. Only a SSH key or password are needed to use the automation playbooks and they do not require installation of any other software meaning very little management is needed. The simplicity allows for quicker turnaround of the management cycles allowing for more availability to focus on more crucial and tactical work. [Ansible, 2020]

Ansible includes a state-driven source mode that defines the preferred state of computer systems and services, instead of the paths needed to achieve this state. Regardless of the current state of a system, ansible will understand how to configure the preferred state described. This provides a consistent, replicable solution and reduces failures seen in script-based configuration. Ansible is provides secure configurations. It uses Open SSH for its default transport layer and does not require any 3rd party software or root login privileges meaning it has a low attack surface and is easily deployed [Ansible, 2020]

AUTOMATION OF PROVISIONING

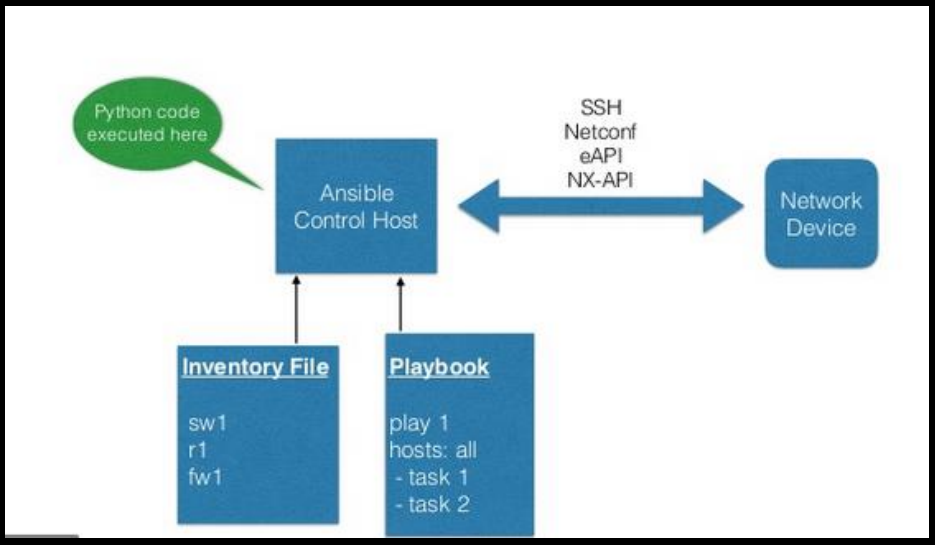
When trying to automate the operational life cycle of your applications the first step needs to be automating the provisioning of any infrastructure. Ansible can provision the latest cloud platforms, virtualized hosts and hypervisors, network devices and bare-metal servers. Ansible provisioning playbooks exists for bare metal infrastructure to both call and perform the provisioning steps required. They also exist for virtualized infrastructure. [Ansible, 2020]




[Kolappan, 2020]

AUTOMATING NETWORKS

The application will use the Ansible playbooks to provide automation of configuring the network stack from system to access to core services, testing and validating the current network state and continuous compliance to check for network configuration drift




[Network Automation with Ansible, 2020]

**AUTOMATING SECURITY**

The application will use Ansible to automate the firewalls regulating what traffic can pass between networks, shielding applications that are internet or intranet facing. It will employ policies and log configuration, allowing a quicker turnaround for investigations and remediation. Intrusion detection & prevention systems [IDPS] monitor network traffic for suspect activity and broadcast alerts and prevent attacks when a recognised attack pattern is detected. The application will use Ansible to simplify rule and log administration, resulting in more proficient security operations. Security information and event management [SIEM] systems gather and examine security events aid in detection and response to incidents. Ansible provides security analysts with wide range of data sources to help better assess incidents. Privileged Access Management [PAM] tools monitor and oversee privileged accounts and access, Ansible simplifies the alternation and managing of privileged authorisations to automatically prevent and remediate high-risk activities. [Ansible, 2020]

Automating Security Operations allow for:

- Configuration a sequence of jobs that share inventory, playbooks, or permissions to fully automate investigations.
- Centralising job access and execution and role provisioning/building by pairing user directory with infrastructure
- classify patterns, analyse infrastructure events, monitor glitches, and compare unrelated events, by integrating with 3rd party external log cluster services [Ansible, 2020]

THE GUI

The main project will also include developing a GUI using Django and python, this front end will have the ability to be connected to Ansible play books and run them to provide automation for network and server security provisioning.

FUNCTIONALITY IN THIS PROJECT
Playbooks:

***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.

# METRICS

| **Criteria** | **Description** |
|---|---|
| **Security** | **All connections to the remote hosts are secure** |
| **Security** | **The playbooks all allow for efficient and successful Incident response Handling** |
| **Security** | **The web application deployed is secured** |
| **Security** | **The database stores passwords salted and hashed** |
| **Security** | **The webserver has security configurations to access it** |
| **Automation, security, networking** | **The playbooks successfully create and configure networks/servers** |

# BIBLIOGRAPHY


| | |
|---|---|
| [DevOps,2020] | En.wikipedia.org. 2020. *Devops*. [online] Available at: <https://en.wikipedia.org/wiki/DevOps#cite_note-loukides-2012-2> [Accessed 13 November 2020]. |
| [Chef Tutorial: Components and Configuration Management, and More Explained, 2020] | Simplilearn.com. 2020. *Chef Tutorial: Components And Configuration Management, And More Explained*. [online] Available at: <https://www.simplilearn.com/chef-tutorial-article> [Accessed 12 November 2020]. |
| [Puri, 2020] | Puri, M., 2020. *A Complete Beginner'S Guide To Chef And Infrastructure As Code*. [online] freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/an-introduction-to-chef-and-infrastructure-as-code-7d8ad2689b8/> [Accessed 12 November 2020]. |
| [What is Chef \| Automate Infrastructure Configuration, 2020] | Apachebooster Blog: *What Is Chef \| Automate Infrastructure Configuration*. [online] Available at: <https://apachebooster.com/blog/automate-infrastructure-configuration/> [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. *What Is A Puppet In Devops? \| Why Puppet Software For Devops?*. [online] Available at: <https://staragile.com/blog/puppet-in-devops> [Accessed 12 November 2020]. |
| [Sheet and Sheet, 2020] | Sheet, P. and Sheet, P., 2020. *Puppet Cheat Sheet - Download In PDF & JPG Format - Intellipaat*. [online] Intellipaat Blog. Available at: <https://intellipaat.com/blog/tutorial/devops-tutorial/puppet-cheat-sheet/> [Accessed 12 November 2020]. |
| [Anon, 2020] | Puppet.com. 2020. [online] Available at: <https://puppet.com/docs/continuous-delivery/3.x/cd_architecture.html> [Accessed 12 November 2020]. |
| [Ansible, 2020] | Ansible, R., 2020. *Ansible For Provisioning*. [online] Ansible.com. Available at: <https://www.ansible.com/use-cases/provisioning> [Accessed 12 November 2020]. |
| [Kolappan, 2020] | Kolappan, K., 2020. *Automating Infrastructure Deployments [IAAS] In AWS Cloud With Ansible*. [online] Oneglobesystems.com. Available at: <https://www.oneglobesystems.com/blog/automating-infrastructure-deployments-iaas-in-aws-cloud-with-ansible> [Accessed 12 November 2020]. |
| Ansible, 2020 | Ansible, R., 2020. *Red Hat Ansible Network Automation - Red Hat Ansible*. [online] Ansible.com. Available at: <https://www.ansible.com/use-cases/network-automation> [Accessed 12 November 2020]. |
| [Network Automation with Ansible, 2020] | Slideshare.net. 2020. *Network Automation With Ansible*. [online] Available at: <https://www.slideshare.net/atarsha/network-automation-with-ansible-banog-meetup> [Accessed 12 November 2020]. |

| | |
|---|---|
| [Ansible, 2020] | Ansible, R., 2020. *Ansible For Configuration Management*. [online] Ansible.com. Available at: <https://www.ansible.com/use-cases/configuration-management> [Accessed 12 November 2020]. |
| Ansible, 2020] | Ansible, R., 2020. *Red Hat Ansible | Security Automation*. [online] Ansible.com. Available at: <https://www.ansible.com/use-cases/security-automation> [Accessed 12 November 2020]. |
| [Johari, 2020] | Johari, A., 2020. *Chef Vs Puppet Vs Ansible Vs Saltstack: Which One To Choose | Edureka*. [online] Edureka. Available at: <https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/> [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. |
| | |
| | |