Cloud Upgrade Automation

Project Report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

SUMANYU NANDWANI (151315)

Under the supervision of

Prem Mohan

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Waknaghat, Solan-173234, Himachal Pradesh
Candidate’s Declaration

I hereby declare that the work presented in this report entitled “Cloud Upgrade Automation” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2019 to May 2019 under the supervision of Shivani Sharma (Sr. Software Engineer (QA), Cloud Reliability Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sumanyu Nandwani (151315)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Name: Shivani Sharma
Designation: Sr. Software Engineer
Department Name: Cloud Reliability and Engineering
Dated: 21st May, 2019
Acknowledgement

I am highly grateful to Jaypee University of Information and Technology, Solan, for providing this opportunity to carry out six month industrial training at Zscaler Softech Pvt. India Ltd.

The constant guidance and encouragement received from Prof. Dr. Satya Prakash Ghrera, HoD Department of Computer Science and Engineering and Information Technology, JUIT has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I also take this opportunity to express a deep sense of gratitude to Mr. Prem Mohan, Sr. Manager, Cloud Reliability Engineering, Zscaler, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I take this opportunity to express my profound gratitude and deep regards to my mentor Ms. Shivani Sharma for her exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to staff members of Zscaler, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

I express gratitude to other faculty members of Computer Science and Engineering department of JUIT for their intellectual support throughout the course of this work.

Sumanyu Nandwani
# Table of Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>Chapter 1: INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>Chapter 2: INTRODUCTION TO PROJECT</td>
<td>9</td>
</tr>
<tr>
<td>Chapter 3: SYSTEM DESIGN</td>
<td>11</td>
</tr>
<tr>
<td>Chapter 4: RESULTS AND DISCUSSION</td>
<td>40</td>
</tr>
<tr>
<td>Chapter 5: CONCLUSION</td>
<td>43</td>
</tr>
<tr>
<td>Chapter 6: BIBLOGRAPHY &amp; REFERENCES</td>
<td>44</td>
</tr>
</tbody>
</table>
Abstract

The project ‘Cloud Upgrade Automation’ is basically made to reduce the human efforts and promote automation. In the era of automation, it is important for every organization to automate the available process. This automation will not only reduce manual work but also decrease the chance of human error. Therefore in Zscaler, where clouds are upgraded by NOC or operations team requires a lot of manual effort and time. Moreover it also requires continuously verification of the running process so as to ensure that all the clouds are upgraded in expected way. Therefore to reduce these manual efforts and enforce accuracy, the automation has to be implemented. My project ‘Cloud Upgrade Automation’ does the same. It upgrades the clouds based on certain information that is entered by user itself. Also validations are provided alongside on UI side so that no incorrect information is allowed to process.

The benefits of this project are:

1. Easy to use
2. Input validations
3. Shared visibility
4. Process orchestration/control - running multiple, interdependent upgrades
5. Location independence
6. Repeatable processes/templates
7. Usage control
8. Usage history
9. Dashboards/reporting
1.1: Introduction to Organization

Zscaler is a worldwide cloud-based data security organization that gives antivirus, weakness the executives and granular control of client movement in Web, email and versatile conditions. The company was founded by Jay Chaudhry, a security industry professional and K. Kailash, former chief architect of NetScaler. Prior to Zscaler, Jay Chaudhry founded and funded several successful companies, including CipherTrust, Air-defense, CoreHarbor, Air2Web, and SecureIT. The company competes with similar services offered by Blue Coat Systems, Cisco, MessageLabs, Webroot and Websense. Zscaler was founded and self-funded in 2008 by Jay Chaudhry and K. Kailash. Chaudhry is a serial security entrepreneur that previously founded AirDefense, CipherTrust, CoreHarbor and SecureIT. In 2012, Zscaler raised $38 million from Lightspeed Venture Partners. In 2013, Zscaler had a reported company valuation of $1 billion.

1.2: Vision of Organization

The universe of IT security has experienced huge change, started by the consumerization of the undertaking, the selection of distributed computing, the omnipresence of cell phones and the advancement of dangers, which are more genuine today than they have ever been previously. While customary security apparatuses were created in when none of this existed, Zscaler's cutting edge cloud security stage empowers organizations to grasp these advancements safely, while conveying a unrivaled client experience. Just as cloud applications transformed the market for CRM, Zscaler is transforming the market for Internet security.

1.3: Technology

Zscaler provides cloud-based information security delivered through what is reportedly the world’s largest security cloud of more than 100 global data centers. Localized data centers store
security policies that can be pushed worldwide in seconds, following its users as they travel around the globe to enforce these policies without latency. Zscaler serves as a Web proxy, routing all traffic through its software to apply corporate and security policies, eliminating the time and money companies spend managing Web filtering and security on its own servers. Zscaler is designed to address the challenge of managing security in a world where cloud and mobility are eroding the network perimeter. Zscaler centralizes administration of user policies on a single Web interface with a simple visualization. Zscaler can provide comprehensive user reports in nearly real-time and is constantly gathering global threat data to protect its customers.

1.4: Additional Products

1.4.1: Zscaler for APT’s

Zscaler for APTs provides protection from zero-day attacks and advanced persistent threats by combining proactive protection against known threats, fill based behavior analysis and security analytic such as threat intelligence feeds. Zscaler for APTs provides a comprehensive solution that consolidates the commoditized features of existing security appliances to protect, detect and remediate advanced security threats.

1.4.2: Zscaler Mobile Security

Zscaler Mobile Security extends its real-time analysis and protection to mobile devices in BYOD environments by routing mobile traffic through its global cloud. Zscaler Mobile Security provides visibility into mobile application traffic, protection from web-based threats and rogue applications and policy enforcement on mobile devices.

1.5: Recognition

1.5.1: 2011

Information Week named Zscaler “Startup of the Week”.

Page 3
1.5.2: 2012
Zscaler was named a “Cool Vendors in Software as a Security, 2012” by Gartner.

1.5.3: 2013
Zscaler was named a “Cool Vendors in Cloud Security Services, 2013” by Gartner. Zscaler was named a “Leader” in the Gartner “Magic Quadrant for Secure Web Gateway.” Zscaler was named an “Emerging Vendor 2013: Security Vendors” by CRN. Zscaler was named a “Best Web Content Management Finalist” in the SC Magazine.

1.5.4: 2014
Zscaler was named a “Leader” in the Gartner “Magic Quadrant for Secure Web Gateways.” Zscaler was named an “Emerging Vendor 2014: Security Vendors” by CRN. Zscaler CEO Jay Chaudhry was named a “The Top 25 Disrupters Of 2014” by CRN.

1.5.5: 2015
Zscaler was named a “Leader” in the Gartner “Magic Quadrant for Secure Web Gateways.” Zscaler was named an “Emerging Vendor 2015: Security Vendors” by CRN. Zscaler was named a “Tech 10: Hot Antivirus Alternatives For 2015” by CRN. Zscaler was named a Red Herring “2015 Top 100 North America: Winners.”

1.5.6: 2016
Zscaler was named a “Leader” in the Gartner “Magic Quadrant for Secure Web Gateways.” Zscaler is voted as one of the top 50 best places to work at. Zscaler was named an “Emerging Vendor 2016: Security Vendors” by CRN. Zscaler was named a “Tech 10: Hot Antivirus Alternatives For 2016” by CRN. Zscaler was named a Red Herring “2016 Top 100 North America: Winners.”
1.5.7: 2017
On August 3, 2017, Zscaler announced a $100 million per-IPO fundraising led by TPG Capital Growth. On September 23, 2017, Zscaler announced that the $100 MM round had been oversubscribed and has been raised to $110 million including a $25 million investment from Google Capital.

1.5.8: 2018
On March 15, 2018, Zscaler, Inc. (NASDAQ:ZS) announced the closing of its initial public offering and the exercise in full of the underwriters’ option to purchase an additional 1,800,000 shares of its common stock. After the underwriters’ exercise of the option to purchase additional shares (often called “green shoe”), a total of 13,800,000 shares were sold at the price to the public of $16.00 per share for a total of $220.8 million in aggregate gross proceeds. The stock is now 68 percent above the offer price for a $3.2 billion valuation.

1.6: SSL Traffic Considerations

The Zscaler service operates in part by having all web traffic to be managed sent through Zscaler owned and operated devices. In order to monitor or inspect secure HTTPS connections, Zscaler implements what is known as a man-in-the-middle attack to decrypt SSL traffic for users going through the Zscaler service. When a user attempts to open an HTTPS website, Zscaler mimics the web browser, as the user accesses the server. In response to a CONNECT request by the web browser, the server will send Zscaler a server certificate. Zscaler will then check the validity of the cert and then create a new cert signed by Zscaler. The new cert will be sent to the Web Browser, and expecting that the client has pre introduced a Zscaler root cert, the program will check the legitimacy of the cert and after that acknowledge and introduce the cert and after that will keep on getting to the site. On the off chance that a root cert has not been introduced, at that point the client will get a mistake expressing that there is an issue with the site's security confirmation and client will have the choice to proceed or not. Since Zscaler can decode traffic,
they can check the substance for any noxious traffic that would have generally come over an encoded channel while applying strategy in light of the decoded traffic for the client. When they empower SSL unscrambling they likewise permit the client an alternative to indicate which URL classifications or custom areas ought not be decoded so as to guarantee protection. Zscaler additionally has the choice of blocking access to explicit URL classifications or client spaces, paying little respect to whether SSL decoding is empowered or not.

1.7: The Zscaler Difference

With major global brands grappling with fallout from high profile security breaches, security has been elevated to a board level discussion and a new approach is clearly needed. Many vendors with roots in the security appliance or software world claim to offer cloud based solutions. Only Zscaler was architected from the ground up as a multi-tenant, distributed cloud security platform. Zscaler has effectively moved security into the Internet backbone, operating in more than 100 data centers around the world and enabling organizations to unlock the promise of cloud and mobile computing by providing users with the shortest and safest path to the Internet. As the main genuinely incorporated cloud security stage, Zscaler conveys bearer grade Internet security, progressed constant danger (APT) insurance, information misfortune anticipation, SSL decoding, traffic molding, approach the executives and danger insight – all without the requirement for on-premise equipment, apparatuses or programming. In contrast with conventional apparatus based ways to deal with Internet security, Zscaler's cutting edge cloud based stage conveys better security, better head and end client efficiency, and better financial matters.
Fig 1.1: Active and Upcoming Data Centers of Zscaler
1.8: Zscaler Cloud Architecture

Fig 1.2: Zscaler Cloud Architecture


Chapter 2 : Introduction to Project

2.1: Objective

In the era of automation, it is important for every organization to automate the available process. This automation will not only reduce manual work but also decrease the chance of human error. Therefore in Zscaler, where clouds are upgraded by NOC or operations team requires a lot of manual effort and time. Moreover it also requires continuously verification of the running process so as to ensure that all the clouds are upgraded in expected way.

Therefore to reduce these manual efforts and enforce accuracy, the automation has to be implemented. My project ‘Cloud Upgrade Automation’ does the same. It upgrades the clouds based on certain information that is entered by user itself. Also validations are provided alongside on UI side so that no incorrect information is allowed to process.

2.2: Introduction to assigned job

During the training period, various techniques were taught which are as follows:

2.2.1: Introduction to Cloud

Goal: To understand the Zscaler Cloud Architecture and become familiar with concepts of SMCA, SME, SMSM and SMUI.

2.2.2: Networking Fundamentals

Goal: To utilize prior knowledge of Networking Fundamentals and become practically familiar with concepts such as telnet, raw udp sessions, Trace-route, Performing ssl connections, applying forward proxy, creating self signed ssl cert and etc.
2.2.3: Introduction to Linux Commands
Goal: To get familiar with basic Linux commands and text editors.

2.2.4: Ansible
Goal: To understand and implement this open-source back-end web application in Ansible of ‘Cloud Upgrade Automation’ Project.

2.2.5: Python
Goal: To implement Python for creating high-level, structured, open-source programming language to support ‘Cloud Upgrade Automation’ Project.

2.2.6: PostGres SQL
Goal: To understand and simplify the SQL commands used in the company sanctioned SQL and use it for Database purposes and to make it interact with

2.2.7: Flask
Goal: To understand creation of API (REST) using python library.

2.2.8: uWSGI Emperor
Goal: To serve a flask application on a network. API’s needed for CDA tool.

2.2.9: NGINX
Goal: To serve a flask with uWSGI in a network to act as a server. It keeps the API secure.
3.1: Introduction to Cloud

3.1.1: Introduction to Zscaler

Zscaler is a global cloud-based information security company that provides antivirus, vulnerability management, and granular control of user activity in Web, email, and mobile environments. Jay Chaudhry, a security industry professional and K. Kailash, former chief architect of NetScaler, founded the company. Prior to Zscaler, Jay Chaudhry founded and funded several successful companies, including CipherTrust, AirDefense, CoreHarbor, Air2Web, and SecureIT. The company competes with similar services offered by Blue Coat Systems, Cisco, MessageLabs, Webroot, and Websense.

Zscaler provides cloud-based information security delivered through what is reportedly the world’s largest security cloud of more than 100 global data centers. Localized data centers store security policies that can be pushed worldwide in seconds, following its users as they travel around the globe to enforce these policies without latency. Zscaler serves as a Web proxy, routing all traffic through its software to apply corporate and security policies, eliminating the time and money companies spend managing Web filtering and security on their own servers. Zscaler is designed to address the challenge of managing security in a world where cloud and mobility are eroding the network perimeter. Zscaler centralizes administration of user policies on a single Web interface with a simple visualization. Zscaler can provide comprehensive user reports in nearly real time and is constantly gathering global threat data to protect its customers.

3.1.2: Zscaler Cloud Architecture

SMCA: Safe March Central Authority is the brain of the entire system. It is the point in the cloud where the policies are defined by the admin. The CA is connected to DB
which stores all the policies that are defined by the admin. Whenever a user connects to SME (ZEN), he is redirected to CA for authentication and the CA pushes the policies to SME for policy enforcement.

**SME: Safe March Enforcement (ZEN: Zscaler Enforcement Node):** These nodes are the one which are actually connected to the end users and acts as proxy through which traffic passes. The proxy can be configured in the client machine via three ways: browser proxy, PAC files and GRE tunneling. It is the ZEN which provides the first login page to the user. The session is initiated here by forwarding the traffic to the CA to check which policies are applicable.

**SMSM (Nanolog):** Safe March Statistical Manager is responsible for recording all the transactions that pass through the SME's (ZEN's) that are connected to it. These records are used for reporting and extracting the statistics from the data. The SMSM is further connected to the CA from where the information is processed and presented.

**SMUI:** The Safe March User Interface is basically implemented for the organization’s admin and the Z-admin. The organization admin is the one who makes the policies for different users in his organization. The admin has various tabs like secure, comply, reporting, manage, administration which provide him different options for enforcement of policies and reporting and the management of users. The Z-admin interface is provided for management of all the companies where he can add, delete, configure, enable or disable a company’s subscription.
**Zscaler Admin UI:** It is the interface through which admin can have broad view of traffic from different users, configure policies to be applied on the users, manage user accounts and view logs of all transactions on SME for analytics.

**Dashboard:** It presents the broad idea of cloud app traffic and URL traffic from different users to the admin in the form of pie chart.
3.2: Policy

Various policies can be applied on either individual users or group of users or even department.

Fig 3.3: Policies of Z-Admin UI

3.2.1: Web Policies
Provide security by using malware protection, advanced threat protection, behavioral analysis and browser control.

3.2.2: Access Control
Access control can be applied on URL, file type by allowing or blocking specific url’s and file type.

3.2.3: Bandwidth control
Bandwidth Policy is applied to location based users only and not on road warriors. It can be of two types either session based or defining minimum and maximum bandwidth limit.
To apply bandwidth policy, following steps are followed:

I. The static IP of company is added through ZA.
II. Location is added in SMUI under administration tab, ssl scanning is enabled and download and upload limit is specified.
III. Bandwidth control can be applied from policy tab on particular location.

![Fig 3.4: URL and Cloud App Control](image-url)
3.3: Administration

3.3.1: Account Management
Admin can manage his own account.

Fig 3.5: Administration Panel of Z-Admin UI

Fig 3.6: Setting up new profile
3.3.2: Role Based Access Control

Admin can manage roles of different users as super admin, admin and provide access permissions.

Fig 3.7: Administrator Management

Fig 3.8: Add administrator role
3.4: Analytics

SMSM has two ways to access reports. One is based on counter which are stored in RAM. Admin can get interactive reports and Insights from here (faster). While logs are fetched from secondary memory (slower than interactive reports and insights)

3.4.1: Insights

Admin can view web insights, mobile insights, Email insights, Firewall insights, DNS insights. These can be viewed in the form of bar graph, pie chart, line graph or table. Admin can choose time frame and apply filters to view these insights.
3.5: Networking Fundamentals

3.5.1: Http Live Header

Helps in debugging web application. See which kind of web server the remote site is using.

3.5.2: Wireshark

Wireshark is the world's foremost network protocol analyzer. It lets you see what's happening on your network at a microscopic level. It is the defector standard across many industries and educational institutions. It does deep inspection of hundreds of protocols, live capture and offline analysis.

![Wireshark Screenshot](Figure3.11: Wireshark)
3.5.3: Telnet Commands

telnet <ip><port>

3.5.4: nc command for creating raw UDP Session
3.5.5: Applying forward proxy using Curl

```
source /usr/lib/lamp/ssl.sh
set -e
crib -O -s -k /etc/squid/squid-ssl.conf
rebuild config;
```

Fig 3.14: Forward proxy using curl

3.5.6: Creating self-signed SSL Certificate

```bash
#!/usr/bin/env python
from OpenSSL import crypto
import base64

# Create a self-signed certificate

# Generate a private key
key = crypto.PKey()
key.generate_key(crypto.PK�示密算法('RSA'), 2048)

# Create a certificate request
certreq = crypto.X509Req()
certreq.get_subject().CN = 'localhost'

# Sign the certificate request with the private key
certreq.set_pubkey(key)
certreq.sign(key, 'sha256')

# Create a certificate
cert = crypto.X509()
cert.set_subject(certreq.get_subject())
cert.set_pubkey(key)
cert.set_serial_number(1)
cert.gmtime_adj_notBefore(0)
cert.gmtime_adj_notAfter(31536000)
cert.sign(key, 'sha256')

# Encode the certificate as a DER binary data
data = crypto.dump_certificate(crypto.FILETYPE_PEM, cert)
```

Fig 3.15: Self-Signed SSL Certificate
3.6: Introduction to LINUX Commands

Problem: Start multiple perl processes and kill first perl process.

![Screenshot of multiple perl processes]

Fig 3.16: Multiple perl processes

Problem: One line command to display that others are logged in to the system and print their IP addresses, result should contain only unique values.

```bash
prabhleen@zarms ZARGS-DevTest]$ who | awk '{print $1 $NF}' | cut -d ' ' -f1 | uniq

root(10.37.144.65):
Niranjan(10.66.105.4):
gunjan(10.66.63.21):
shivan(10.66.105.4):
prabhleen(10.37.144.200):
root(10.37.144.65):
kelees(10.66.63.21):
Niranjan(10.66.105.4):
shivan(10.66.105.4):
root(zarms.corp.zscaler.com):
prabhleen@zarms.corp.zscaler.com):
root(zarms.corp.zscaler.com):
prabhleen@zarms.corp.zscaler.com):
root(zarms.corp.zscaler.com):
prabhleen@zarms.corp.zscaler.com):
root(zarms.corp.zscaler.com):
root(zarms.corp.zscaler.com):
prabhleen@zarms ZARGS-DevTest]$ prabhleen@zarms ZARGS-DevTest]$
```

Fig 3.17: Get IP address of other users
Problem: Write VI editor commands for following:

I. Search a keyword and go back and forth in search result.(/pattern)

II. Go to specific line.(:lineno:)

III. Cut/copy and paste a word and whole line(dd,yy,p)(y w)(v for selecting then y/d)

IV. Replace specific word in vim. s/pattern1/pattern2/(single word),/g for whole line ,%s for whole paragraph

V. Paste with comment (when auto and smart in dent is enabled) :set paste then paste

Fig 3.18: VI editor commands
Problem: Write a command to replace a word without opening a file

```bash
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~$ cat scripts/script_dir.sh
#!/bin/bash
h='du -h / | sort -rh | head -10'
date > /home/saurabh/Documents/out.log
echo $h > /home/saurabh/Documents/out.log
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~$ sudo ./scripts/script_dir.sh
du: cannot access `/run/user/1000/gvfs': Permission denied
du: cannot access `/proc/14353/task/14353/fd/4': No such file or directory
du: cannot access `/proc/14353/task/14353/fdinfo/4': No such file or directory
du: cannot access `/proc/14353/fdinfo/4': No such file or directory
saurabh@saurabh-hp:~$ crontab -l
*/5 * * * * /bin/execute/thls/scripts/script_dir.sh
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
```

Fig 3.19: Replacing a word

Problem: Creating a bash script to check the disk usage of all directories and list top 10 dirs and running it automatically every 5 minutes.(crontab)

```bash
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~$ cat scripts/script_dir.sh
#!/bin/bash
h='du -h / | sort -rh | head -10'
date > /home/saurabh/Documents/out.log
echo $h > /home/saurabh/Documents/out.log
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~$ sudo ./scripts/script_dir.sh
du: cannot access `/run/user/1000/gvfs': Permission denied
du: cannot access `/proc/14353/task/14353/fd/4': No such file or directory
du: cannot access `/proc/14353/task/14353/fdinfo/4': No such file or directory
du: cannot access `/proc/14353/fdinfo/4': No such file or directory
saurabh@saurabh-hp:~$ crontab -l
*/5 * * * * /bin/execute/thls/scripts/script_dir.sh
saurabh@saurabh-hp:~
saurabh@saurabh-hp:~
```

Fig 3.20: Creating a bash script
Problem: Download a File from command line "http://10.65.1.220/ba_testfiles/Exe.zip"

Fig 3.21: Downloading a file
3.7. Ansible

Ansible is programming that robotizes programming provisioning, setup the executives, and application arrangement. Similarly as with most setup the executives programming, Ansible has two kinds of servers: controlling machines and hubs. To start with, there is a solitary controlling machine which is the place coordination starts. Hubs are overseen by a controlling machine over SSH. The controlling machine portrays the area of hubs through its stock

![Fig 3.22: Downloading Ansible](image)

**3.7.1: Architecture Overview**

To coordinate hubs, Ansible sends modules to hubs over SSH. Modules are incidentally put away in the hubs and speak with the controlling machine through a JSON convention over the standard yield. At the point when Ansible isn't overseeing hubs, it doesn't expend assets in light of the fact that no daemons or projects are executing for Ansible out of sight.

Conversely with mainstream arrangement the board programming —, for example, Chef, Puppet, and CFEngine—Ansible utilisations an operator less engineering. With an operator based design, hubs must have a privately introduced daemon that speaks with a controlling machine. With an operator less engineering, hubs are not required to introduce and run foundation daemons to interface with a controlling machine. This
kind of design decreases the overhead on the system by keeping the hubs from surveying the controlling machine.

3.7.2: Ansible Components

I. Playbooks

II. Inventory Configuration

III. Callback Plugins

IV. Ansible Engine

V. Configuration Files

3.7.3: Playbooks

Ansible playbooks are an approach to send directions to remote PCs in a scripted manner. Rather than utilizing Ansible directions independently to remotely design PCs from the order line, you can arrange whole complex conditions by passing a content to at least one frameworks. Ansible playbooks are written in the YAML information serialization position. It is an approach to interpret an automatic information structure (records, exhibits, word references, and so on) into a configuration that can be effectively put away to circle. The document would then be able to be utilized to reproduce the structure at a later point. JSON is another prominent information serialization position, yet YAML is a lot simpler to peruse.

Executing an Ansible playbook constitutes of installing the Ansible and them using the command.

Command: Ansible-playbook playbookname.yml
Fig 3.23: Example of how playbooks look

As playbook are in YAML format, it’s easy to add to the existing commons in a KEY-VALUE pair format so that later on its easy to insert and delete or skip commands on the convenience of the user.

Fig 3.24: Example of how playbooks are executed

This setup record indicates three hubs: the primary hub is determined by an IP address and the last two hubs are indicated by host names. Moreover, the last two hubs are gathered under the web servers gathering.

Ansible can likewise utilize a custom Dynamic Inventory content, which can progressively pull information from an alternate framework.

Common host file parameters-
I. ansible_connection: Connection type to the host. This can be the name of any of ansible's connection plugins. SSH protocol types are smart, ssh or paramiko. The default is smart. Non-SSH based types are described in the next section.

II. ansible_host: The name of the host to connect to, if different from the alias you wish to give to it.

III. ansible_port : The ssh port number, if not 22

IV. ansible_user: The default ssh user name to use.
3.7.4: Inventory configuration

The Inventory is a depiction of the hubs that can be gotten to by Ansible. As a matter of course, the Inventory is portrayed by a design record, in INI or YAML position, whose default area is in/and so on/ansible/has. The design document records either the IP address or host name of every hub that is available by Ansible. Likewise, hubs can be appointed to gatherings. An example inventory:

![Example inventory](image)

This setup record indicates three hubs: the primary hub is determined by an IP address and the last two hubs are indicated by host names. Moreover, the last two hubs are gathered under the web servers gathering.

Ansible can likewise utilize a custom Dynamic Inventory content, which can progressively pull information from an alternate framework.

Common host file parameters-

V. ansible_connection: Connection type to the host. This can be the name of any of ansible’s connection plugins. SSH protocol types are `smart`, `ssh` or `paramiko`. The default is smart. Non-SSH based types are described in the next section.
VI. **ansible_host**: The name of the host to connect to, if different from the alias you wish to give to it.

VII. **ansible_port**: The ssh port number, if not 22.

VIII. **ansible_user**: The default ssh user name to use.

Examples from an Ansible-INI host file:

<table>
<thead>
<tr>
<th>Host Type</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>some_host</td>
<td>ansible_port=2222   ansible_user=manager</td>
</tr>
<tr>
<td>aws_host</td>
<td>ansible_ssh_private_key_file=/home/example/.ssh/aws.pem</td>
</tr>
<tr>
<td>freebsd_host</td>
<td>ansible_python_interpreter=/usr/local/bin/python</td>
</tr>
<tr>
<td>ruby_module_host</td>
<td>ansible_ruby_interpreter=/usr/bin/ruby.1.9.3</td>
</tr>
</tbody>
</table>

Fig 3.26: Example of hostfile Hierarchy

### 3.7.5: Callback Plugins

Callback modules empower adding new practices to Ansible when reacting to occasions. As a matter of course, callback modules control the vast majority of the yield you see when running the direction line programs, yet can likewise be utilized to include extra yield, incorporate with different devices and marshall the occasions to a capacity backend. You can initiate a custom callback by either dropping it into a callback_plugins directory neighboring your play, inside a job, or by placing it in one of the callback index sources arranged in ansible.cfg. Modules are stacked in alphanumeric request. For instance, a module executed in a document named 1_first.py would keep running before a module record named 2_second.py. Most callbacks transported with Ansible are impaired as a matter of course and should be white recorded in your ansible.cfg file so as to work.
For example:

3.7.6: Ansible Engine

Ansible is a drastically basic IT computerization motor that robotizes cloud provisioning, configuration the board, application arrangement, intro-administration coordination, and numerous other IT needs.

Intended for multi-level organizations since the very beginning, Ansible models your IT foundation by portraying how the majority of your frameworks between relate, as opposed to simply overseeing one framework at any given moment. With Ansible, you can without much of a stretch arrangement, pulverize, take stock, and deal with your virtual surroundings. Furthermore, Ansible's broad help enables you to then send and arrange over the majority of your virtual surroundings once you've provisioned them.
3.7.7: Configuration file

Ansible supports a few ways of providing configuration variables, mainly through environment variables, command line switches and an ini file named ansible.cfg.

Starting at Ansible 2.4 the ansible-config utility allows users to see the entire configuration settings available, their defaults, how to set them and where their current value comes from. Changes can be made and used in a configuration file which will be searched for in the following order:

I. ANSIBLE_CONFIG (environment variable if set)

II. ansible.cfg (in the current directory)

III. ~/.ansible.cfg (in the home directory)

IV. /etc/ansible/ansible.cfg
Ansible will process the above list and use the first file found, all others are ignored.
3.8 : Python

Python is an abnormal state, organized, open-source programming language that can be utilized for a wide assortment of programming assignments. Python was made by Guido Van Rossum in the mid 1990s; its following has developed relentlessly and intrigue has expanded extraordinarily over the most recent couple of years or something like that. It is named after Monty Python's Flying Circus parody program.

![Python Command](image)

**Fig 3.30 : python command**

Python is utilized widely for framework organization (numerous essential segments of Linux dispersion are written in it); additionally, it is an incredible language to instruct programming to amateurs. NASA has utilized Python for its product frameworks and has embraced it as the standard scripting language for its Integrated Planning System. Python is likewise broadly utilized by Google to execute numerous parts of its Web Crawler and Search Engine and Yahoo! for dealing with its exchange gatherings. Python inside itself is a deciphered programming
language that is naturally accumulated into byte code before execution (the byte code is then typically spared to plate, similarly as consequently, with the goal that gathering need not occur again until and except if the source gets changed). It is additionally a progressively composed language that incorporates (yet does not expect one to utilize) object-arranged highlights and builds. The most abnormal part of Python is that blank area is critical; rather than square delimiters (supports → "{}" in the C group of dialects), space is utilized to demonstrate where squares start and end. Another incredible element of Python is its accessibility for all stages. Python can keep running on Microsoft Windows, Macintosh and all Linux circulations effortlessly. This makes the projects entirely compact, as any program composed for one stage can without much of a stretch be utilized on another.

![Python script]

Python gives a ground-breaking combination of implicit kinds (e.g., records, lexicons and strings), various inherent capacities, and a couple of builds, for the most part proclamations. For instance, circle develops that can repeat over things in a gathering as opposed to being constrained to a straightforward scope of number qualities. Python additionally accompanies a ground-breaking standard library, which incorporates several modules to give schedules to a wide assortment of administrations including customary articulations and TCP/IP sessions.
Python is utilized and upheld by a huge Python Community that exists on the Internet. The mailing records and newsgroups like the mentor list effectively backing and help new python software engineers. While they demoralize doing homework for you, they are very useful and are populated by the writers of a significant number of the Python course readings as of now accessible available.

3.9 : PostgreSQL

PostgreSQL, frequently just Postgres, is an item social database (ORDBMS) – for example a RDBMS, with extra (discretionary use) "object" highlights – with an accentuation on extensibility and gauges consistence. As a database server, it’s essential capacities are to store information safely and return that information in light of solicitations from other programming applications.

Fig 3.32 : PostgreSQL Commands
It can deal with remaining tasks at hand extending from little single-machine applications to huge Internet-confronting applications (or for information warehousing) with numerous simultaneous clients; on macOS Server, PostgreSQL is the default database; and it is likewise accessible for Microsoft Windows and Linux (provided in many disseminations). PostgreSQL is ACID-agreeable and value-based. PostgreSQL has updatable perspectives and appeared sees, triggers, remote keys; underpins works and put away methodology, and other expandability. PostgreSQL is created by the PostgreSQL Global Development Group, a different gathering of numerous organizations and individual benefactors. It is free and open-source programming, discharged under the terms of the PostgreSQL License, a lenient free-programming permit.

Multiversion simultaneousness control (MVCC) PostgreSQL oversees simultaneousness through a framework known as multiversion simultaneousness control (MVCC), which gives every exchange a "preview" of the database, enabling changes to be made without being unmistakable to different exchanges until the progressions are submitted. This to a great extent kills the requirement for read bolts, and guarantees the database keeps up the ACID (atomicity, consistency, segregation, sturdiness) standards in a proficient way.
PostgreSQL offers three dimensions of exchange detachment: Read Committed, Repeatable Read and Serializable. Since PostgreSQL is invulnerable to filthy peruses, mentioning a Read Uncommitted exchange disconnection level gives read submitted. PostgreSQL bolsters full serializability by means of the serializable depiction detachment (SSI) method.

![Fig 3.33: PostgreSQL Table](image-url)
Chapter 4: Results and Discussion

User Interface Design and Snapshots of Project

4.1: Login Page
User needs to login into the page to access the project.

![Login Page](image1)

Fig 4.1 Login Page

4.2: Dashboard Page

![Dashboard Page](image2)

Fig 4.2 Dashboard Page
4.3: Create new pipeline Stage

Fig 4.3 Create new pipeline page

4.4: View Ansible Command

Fig 4.4 View Ansible Commands
4.5: View Playbooks

Fig 4.5 View Playbooks

4.6: Reporting Page

Fig 4.6 Reporting Page
Chapter 5: Conclusion

As an undergraduate of Jaypee University of Information Technology, I would like to say that this co-op training program is an excellent opportunity for us to get to the ground level and experience the things that we would have never gained through going straight into a job. I am grateful to JUIT and Zcaler Softech India Private Limited for giving me this wonderful opportunity.

The main objective of the internship is to provide an opportunity to undergraduates to identify, observe and practice how engineering is applicable in the real industry. It is not only to get experience on technical practices but also to observe management practices and to interact with fellow workers.

It is easy to work with sophisticated machines, but not with people. The only chance that an undergraduate has to have this experience is the internship period. I feel I got the maximum out of that experience. Also I learnt the way of work in an organization, the importance of being punctual, the importance of maximum commitment, and the importance of team spirit.

In my opinion, I have gained lots of knowledge and experience needed to be successful in a great engineering challenge, as in my opinion; Engineering is after all a Challenge, and not a Job.

Zcaler Softech India Private Limited

The co-op training that I received at Zcaler was great experience for me not only on technical terms but also in terms of interaction with other employees. I learnt a great deal on applying the knowledge I have gained during my B. Tech course. Learning something from books and lectures is nothing like having firsthand experience. I got to apply my Programming knowledge.

The company gives the full freedom for the trainee to get a sufficient training. The trainees are allowed to ask and do anything relevant to the training. Engineers are always very enthusiastic on giving us the best training Page experience. The employees do not think of their designation but only on the work to be done. Teamwork has been excellent. Everyone was friendly from top to bottom. Although employees were busy with their work but they always help us to improve our knowledge.
Chapter 6 Bibliography And References

I. www.ansible.com

II. www.python.org

III. Tango with Django Book

IV. www.w3schools.com

V. Stackoverflow.com

VI. Python Docs, Ansible Docs, Django Docs

VII. Learn Python The Hard Way

VIII. Python Cookbook

IX. www.hackerrank.com

X. www.hackerearth.com