LinuxWorld Informatics PVT. LTD. Server-Side Industrial Automation System

An Industrial Internship Report

Submitted in partial fulfilment for the award of the degree of

B.TECH

In

Information Technology

by

P.Ameerul Ali Shah **17BIT0082**

VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

OCTOBER 2019

DECLARATION BY THE CANDIDATE

I here by declare that the project report entitled **"LINUXWORLD PVT. LTD. SERVER-SIDE INDUSTRIAL AUTOMATION SYSTEM"** submitted by me to School of Information Technology & Engineering, Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech (Information Technology)** is a record of bonafide **Industrial Internship – ITE3099** work carried out by me. I further declare that the work reported in this **Industrial Internship report** has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore   Signature of the Candidate

Date: Pattan Ameerul Ali Shah

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

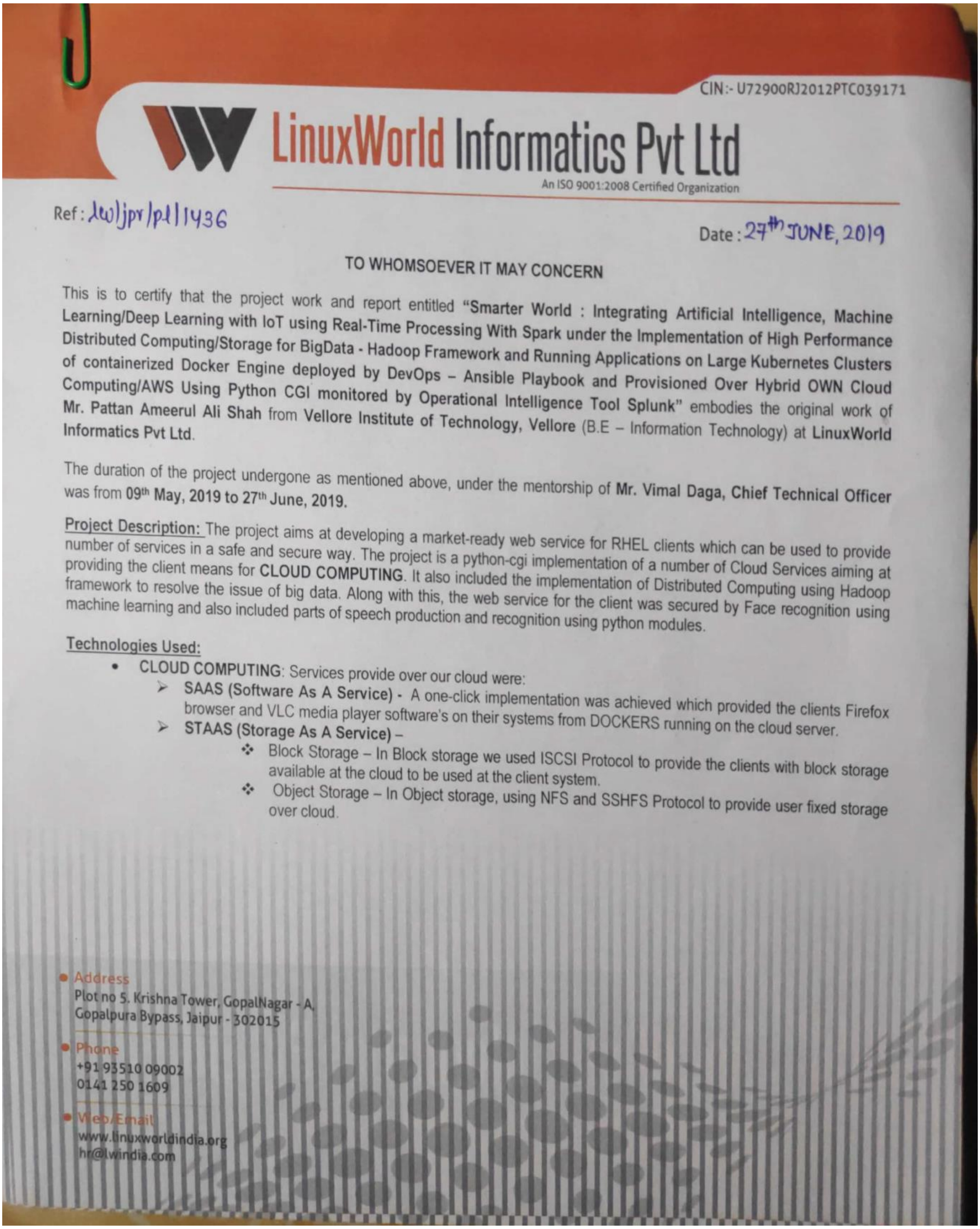
CERTIFICATE

This is to certify that the Industrial Internship report entitled **"LINUXWORLD PVT LTD SERVER-SIDE INDUSTRIAL AUTOMATION SYSTEM"** submitted by **Pattan Ameerul Ali Shah (17BIT0082)** to School of Information Technology & Engineering, Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech (Information Technology)** is a record of bonafide **Industrial Internship – ITE3099** work carried out by him in **LINUXWORLD INFORMATICS PVT LTD**. The **Industrial Internship** project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Mr. Vimal Daga
(Chief Technical Officer, LinuxWorld
Informatics Pvt Ltd, Plot No.5,
Krishna Tower, GopalNagar – A,
Gopalpura Bypass, Jaipur 302015)

Examiner – Panel In-Charge **Date**

(Name and Signature)



CIN:- U72900RJ2012PTC039171

LinuxWorld Informatics Pvt Ltd

An ISO 9001:2008 Certified Organization

Ref: lw/jpr/pl/1436

Date: 27th JUNE, 2019

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the project work and report entitled **"Smarter World : Integrating Artificial Intelligence, Machine Learning/Deep Learning with IoT using Real-Time Processing With Spark under the Implementation of High Performance Distributed Computing/Storage for BigData - Hadoop Framework and Running Applications on Large Kubernetes Clusters of containerized Docker Engine deployed by DevOps – Ansible Playbook and Provisioned Over Hybrid OWN Cloud Computing/AWS Using Python CGI monitored by Operational Intelligence Tool Splunk"** embodies the original work of **Mr. Pattan Ameerul Ali Shah** from **Vellore Institute of Technology, Vellore** (B.E – Information Technology) at **LinuxWorld Informatics Pvt Ltd**.

The duration of the project undergone as mentioned above, under the mentorship of **Mr. Vimal Daga, Chief Technical Officer** was from 09th **May, 2019 to 27th June, 2019.**

Project Description: The project aims at developing a market-ready web service for RHEL clients which can be used to provide number of services in a safe and secure way. The project is a python-cgi implementation of a number of Cloud Services aiming at providing the client means for **CLOUD COMPUTING**. It also included the implementation of Distributed Computing using Hadoop framework to resolve the issue of big data. Along with this, the web service for the client was secured by Face recognition using machine learning and also included parts of speech production and recognition using python modules.

Technologies Used:

- CLOUD COMPUTING: Services provide over our cloud were:
  - ➢ SAAS (Software As A Service) - A one-click implementation was achieved which provided the clients Firefox browser and VLC media player software's on their systems from DOCKERS running on the cloud server.
  - ➢ STAAS (Storage As A Service) –
    - ❖ Block Storage – In Block storage we used ISCSI Protocol to provide the clients with block storage available at the cloud to be used at the client system.
    - ❖ Object Storage – In Object storage, using NFS and SSHFS Protocol to provide user fixed storage over cloud.

Ref: lw/jpr/pl/1436

Date: 27th JUNE, 2019

- CAAS (Containers As A Service) - A one-click launch, starting, stopping and removing and providing the terminal of Docker containers was achieved.
- PAAS (Platform As A Service) - A one-click launch of a python platform was provided for clients where they can code as they wish.

- **Hadoop:** A CGI based web portal was developed for a one click set up of hdfs and map reduce cluster depending on the requirements of the client per se, the size of the cluster required, and the number of processors required for the big data analysis.
- **Machine Learning:** The web service for the client was secured by Face recognition using machine learning and also included parts of speech production and recognition using python modules. IMAGE PROCESSING constitutes a large portion of the security aspect of the service.
- **Ansible:** The entire configuration requirement of the project was met by creation of ANSIBLE playbooks for both client and the cloud.
- **Dockers:** For the safe and secure single client based service production, most of the services were provided through dockers only.

Note: The backend includes Python scripting in version 3.6 integrated with HTML to provide a cgi portal.

We wish him all the success for his future

Regards

LinuxWorld Informatics Pvt Ltd

LinuxWorld Informatics Pvt Ltd
Jaipur
Authorised

Mr. Vimal Daga – Chief Technical Officer

ACKNOWLEDGEMENT

First, I would like thank **Mr. Vimal Daga, Chief Technical Officer of LinuxWorld Informatics PVT. LTD.** for giving me an opportunity to do an internship within the organization. For me it was a unique experience to work on a project for a company. It helped me to have new plans for my future career.

I also would like to thank all the people that worked in the office of **LinuxWorld Informatics PVT. LTD.** for always being patient and lending me a helping hand.

Furthermore, I want to thank all the fellow interns, with whom I worked on the projects and learnt how to work as a team.

Place : Vellore

Date : Pattan Ameerul Ali Shah

TABLE OF CONTENTS

TABLE OF FIGURES

1. SYNOPSIS OF THE REPORT

This report is based on the tasks done in Industry 4.0 internship pursued in the ISO 9001:2008 Certified Organisation named LinuxWorld Informatics Pvt. Ltd from 09/05/2019 to 2/06/2019 accounting for 50 days. During my internship I worked on Industry several 4.0 technologies like Artificial Intelligence, Machine Learning, Deep Learning, IoT, ML/DL libraries Tensorflow, Keras, DevOps – Ansible, Cloud Computing, AWS Cloud, Operational Intelligence Tool Splunk ,Cloud Computing Unit, IaaS, PaaS, SaaS, Storage As-a service, CaaS, Virtualization-KVM, Linux development, Security, Python with Web Technology CGI and Socket Programming. I also worked on Big Data technologies like Hadoop, Docker Engine Kubernetes. Big Data is used in helping companies to make more informative business decisions by analysing large volumes of data that simplify workload and reduce the time. During my internship period I worked on a menu-based automation project with other team members and gained deep insight knowledge of these technologies. In the duration of this internship, I experienced first-hand the environment in a corporate setting, the subdivision and the distribution of tasks as well as the flow of work between various teams. I also learnt how each group contributes in an important way to achieve the final result. In this report I have provided all the details of the work I did and the I experiences gathered during this internship period.

2. ABOUT THE COMPANY – LinuxWorld Informatics PVT. LTD.

2.1 HISTORY

LinuxWorld Informatics Pvt. Ltd is a very fast-growing ISO 9001:2008 Certified Organisation/Company primarily based in Jaipur, India. For around a decade now, the company has successfully built a niche of its own by offering an integrated portfolio of high-end Linux, Open Source internship and support services. This adventurous journey was started by my Mentor Vimal Daga known as an Technologist and also a Technology Motivational Speaker, Sr. IT Consultant and Corporate Trainer having Expertise in Latest and High-End Technologies like Machine Learning, Artificial Intelligence, Deep Learning, IoT, NLP, Splunk, PingFederate, Delphix, AppDynamics, Docker, DevOps, AWS, Cloud Computing, Big Data Analytics Dollar

Universe whose main aim was Making India Future Ready and he strived to do something explosive and innovative in this field. Since its inception in the year 2005, LW(LinuxWorld) have achieved the status of centre of excellence wherein there is latest technology, innovative developing methodology, state of the art infrastructure and individual needs of employees are identified and executed professionally, efficiently and ethically.

No matter how long the path is, but it starts with a single small step, and henceforth, we began our journey. Because, we believe that dreams should not be followed but should be led by, LinuxWorld is a dream lead by a group of entrepreneurs, developers and researchers.

2.2 OVERVIEW

LinuxWorld is Red Hat Partner; which is the world's most trusted provider of Linux, open source technology and the most recognized Linux brand in the world. Red Hat serves global enterprises through technology and services made possible by the open source model, include Red Hat Enterprise Linux operating platforms and features RHCE, the global standard Linux certification.

LINUXWORLD provides support for all major Open Source applications. The main objective of LW R&D's team is to work dedicate on the loopholes present in the existing Linux Operating System, to ensure that the same are removed and plug-in the new development which does not exist in Linux OS at present. In other words, it follows Push-Pull Strategy i.e. Pull out the loopholes and Push in the new developments.

It also provide summer internship for the students of about 45 days in Industry 4.0 technologies like technologies like Artificial Intelligence, Machine Learning, Deep Learning, IoT, BigData, ML/DL libraries Tensorflow, Keras, DevOps – Ansible, Cloud Computing, AWS Cloud, Operational Intelligence Tool Splunk etc Along with that they also help the interns understand the value of sharing knowledge and all over Personality Development because we know that if you are not able to present in front of others than your knowledge has no meaning and they provide simple, secure, portable, fully responsive, dynamic websites, software solutions, and cost-efficient with best maintenance services.

2.3 AIM

The use of technology is to get the things done in a fast and easy way. Their aim is to provide the best quality of industry 4.0 technology services at an affordable cost. They just want to make a world where the things can be get done at just a single click. The cost of technology should be as so that it can reach to every required person without any hassle and our company has already started work upon this. They provide the best quality of software services across the globe. According to my Mentor "No technology is challenging or difficult or complex as the world says to us or approaches to us since it is MAN-MADE. You just need the right path or approach to understand the technology and take it further as per one's requirement". Right thinking or positive thinking that "Yes I Can Do It" can change one's life drastically.

3. SKILLSET BEFORE INTERNSHIP (Gained through curriculum)

3.1 HTML - HTML is a software solution stack that defines the properties and behaviours of web page content by implementing a mark-up based pattern to it. HTML 5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the mark-up available for documents, and introduces mark-up and application programming interfaces (APIs) for complex web applications. Many new syntactic features are included in HTML5 as compared to HTML. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav>, and <figure> are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite>, and <menu> have been changed, redefined, or standardized.

3.2 Cascading Style Sheets - CSS is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

3.3 SQL - Structured Query Language or SQL is a standard Database language which is used to create, maintain and retrieve the data from relational databases like MySQL, Oracle, SQL Server, Postgre, etc. The recent ISO standard version of SQL is SQL ,2019. As the name suggests, it is used when we have structured data (in the form of tables). SQL is required to perform different operations on the database and on various tables of the database. SQL is required to create database, create tables, add constraints on table field, altering tables, deleting tables, inserting values and updating values in tables etc. Knowledge of SQL is required for backend development as we deal with a number of databases when working on backend.

3.4 Data Structure and Algorithm - Data Structures and Algorithms are the identity of a good Software Developer. The main reason behind this is Data Structures and Algorithms improves the problem-solving ability of a candidate to a great extent. In programming, algorithm is a set of well-defined instructions in sequence to solve the problem. A data structure is a specialized format for organizing, processing, retrieving and storing data. While there are several basic and advanced structure types, any data structure is designed to arrange data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. Arrays, linked list, stacks, queues, heaps, hash tables are the common data structures.

3.5 Programming Languages (C, C++, Java, Python) - Coding in basic programming languages and all OOPs (Object oriented Programming) concepts were taught in the curriculum. Concepts of inheritance, polymorphism, abstraction, encapsulation and modularity are main OOPs concept. Coding in such languages improved my problem-solving skills.

4. KNOWLEDGE ACQUIRED FROM IN-PLANT INTERNSHIP

4.1 Ansible

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intraservice orchestration and provisioning. Automation is crucial these days, with IT environments that are too complex and often need to scale too quickly for system administrators and developers to keep up if they had to do everything manually. Automation simplifies complex tasks, not just making developers' jobs more manageable but allowing them to focus attention on other tasks that add value to an organization. In other words, it frees up time and increases efficiency. And Ansible, as noted above, is rapidly rising to the top in the world of automation tools.

4.2 Docker

Docker is a containerization platform that packages your application and all its dependencies together in the form of a docker container to ensure that your application works seamlessly in any environment. Docker Container is a standardized unit which can be created on the fly to deploy a particular application or environment. It could be an Ubuntu container, CentOs container, etc. to full-fill the requirement from an operating system point of view. Docker has become an extremely popular way to configure, save, and share server environments using containers. Because of this, installing an application or even a large stack can often be as simple as running docker pull or docker run. Separating application functions into different containers also offers advantages in security and dependency management.

4.3 Hadoop

Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analytics, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouses provide.

4.4 Python CGI

It is a set of standards that define a standard way of passing information or web-user request to an application program & to get data back to forward it to users. This is the exchange of information between web-server and a custom script. When the users requested the web-page, the server sends the requested web-page. The web server usually passes the information to all application programs that process data and sends back an acknowledged message; this technique of passing data back-and-forth between server and application is the Common Gateway Interface.

4.5 Computer Vision

Computer vision is a field of computer science that works on enabling computers to see, identify and process images in the same way that human vision does, and then provide appropriate output. It is like imparting human intelligence and instincts to a computer. In reality though, it is a difficult task to enable computers to recognize images of different objects. Computer vision is closely linked with artificial intelligence, as the computer must interpret what it sees, and then perform appropriate analysis or act accordingly. The CV tool used here is OpenCV.

5. APPLICATION OF THE GAINED KNOWLEDGE IN/DURING THE INTERNSHIP

5.1 OVERVIEW OF PROPOSED SYSTEM

The proposed system was implemented under RHEL 7.5 under master slave architecture. There were 3 slaves which can be easily scaled, the RHEL 7.5 installation of nodes was standardized using Kickstart and the setup task was automated. The client services were accessible through the Main Webserver on the Master. The rest deployment was carried on the client. No reverse proxy exists so the client was redirected to the appropriate node IP and port for accessing the service. The following are the main components: -

Table 1: Programming Components and Frameworks

COMPONENT	CATEGORY	USE CASE
Python CGI	Programming Language	Dynamic Webpage and Ansible persistence
HTML	Markup Language	To make basic webpages
YAML	Markup Language	Perform tasks using Ansible
OpenCV	Framework	For Image Processing and Facial Identification
Python3	Programming Language	Text to Speech and viceversa, file handling and video streaming.
Containerization	Technology	The core cloud services were implemented under containerization

The client was provided services like – SaaS, PaaS, CaaS, STaaS through containers. They were provided with isolated environments. For proving STaaS simple NFS with LVM was used. CaaS (Container as a service) was provided through Shellinabox which provided the shell of the container in the web page itself. The SaaS (Software as a Service) can be directly streamed to the webpage without any overhead to the client to install a software to make use of the services.

Another main component of the project is Big Data. There is a separate internal cluster where the network packet dumps are stored. The dumps are then analyzed through Map Reduce. Then the results can be used to blacklist the DoS attacker’s IP Address in the Firewall.

Computer Vision paired with Speech recognition was also one of the main features. The administrator is required to run a container remotely which will capture his face through the webcam and the face is cropped (through Haar Cascade) and streamed to the main server where the identification is done.

After identification, the admin gets access to the voice activated menu for setting up the new nodes after the Kickstart. The nodes are then added permanently to the cluster. The Admin also can provision any node for provisioning any type of services like StaaS or other cloud services or Hadoop cluster.

Table 2: Software and Frameworks

SOFTWARE	CATEGORY	USE CASE
Docker	Containerization	To deploy all core cloud services and remote facial authentication.
Ansible	Automation	To automate the containers and dynamic node setup
Apache Hadoop	Big Data	To store and process data with high volume, velocity and variety.
Apache HTTP	Web Services	To provide web interface to the client for performing tasks.
Shellinabox	Interface	Provides remote shell access in the web interface.
FTP	Server	To setup YUM repositories and kickstart boot files.
DHCP	Server	For dynamically configuring host when connected to the network.
NFS	Server	To provide Object based Storage

Docker Registry	Registry	Storage of docker images, and helps to utilize layered structure of docker images.

5.2 DESIGN OF THE SYSTEM

5.2.1 PYTHON CGI

Now it's time for CGI. As the name suggests, CGI means "Common" gateway interface for everything. CGI is one of the essential parts of HTTP (Hyper-Text Transfer Protocol).

It is a set of standards that define a standard way of passing information or web-user request to an application program & to get data back to forward it to users. This is the exchange of information between web-server and a custom script. When the users requested the web-page, the server sends the requested web-page. The web server usually passes the information to all application programs that process data and sends back an acknowledged message; this technique of passing data back-and-forth between server and application is the Common Gateway Interface.

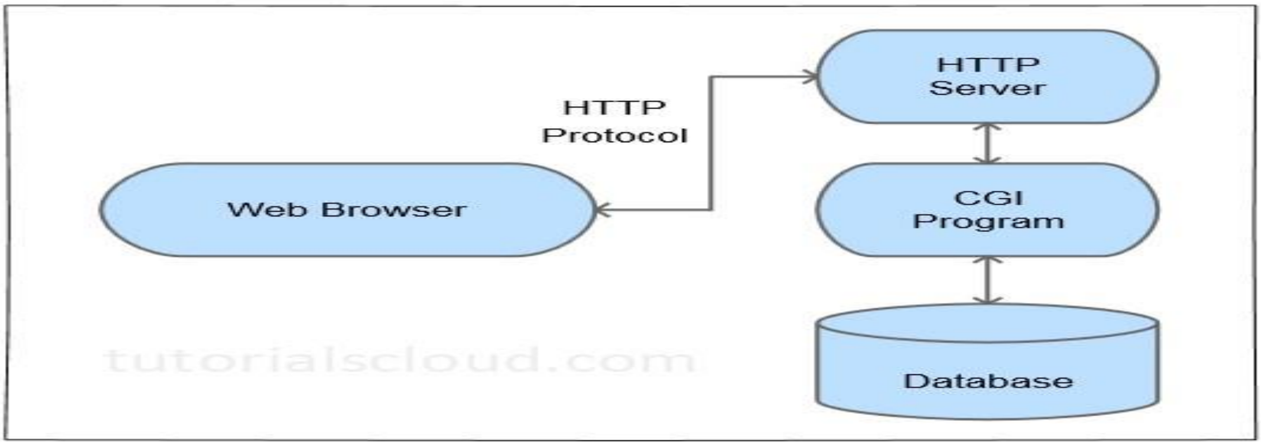


Figure 1: Architecture

5.2.2 ANSIBLE:

Ansible is an open-source software provisioning, configuration management, and application deployment tool. It runs on many Unix-like systems, and can configure both Unix-like systems as well as Microsoft Windows. It includes its own declarative language to describe system configuration.

Architecture

Unlike most configuration-management software, Ansible does not require a single controlling machine where orchestration begins. Ansible works against multiple systems in your infrastructure by selecting portions of Ansible's inventory, stored as edit-able, version-able ASCII text files. Not only is this inventory configurable, but you can also use multiple inventory files at the same time and pull inventory from dynamic or cloud sources or different formats (YAML, INI, etc). Any machine with Ansible utilities installed can leverage a set of files/directories to orchestrate other nodes, the absence of a central-server requirement greatly simplifies disaster-recovery planning. Nodes are managed by this controlling machine - typically over SSH. The controlling machine describes the location of nodes through its inventory. Sensitive data can be stored in encrypted files using Ansible Vault.

In contrast with other popular configuration-management software — such as Chef, Puppet, and CFEngine — Ansible uses an agentless architecture, with Ansible software not normally running or even installed on the controlled node. Instead, Ansible orchestrates a node by installing and running modules on the node temporarily via SSH. For the duration of an orchestration task, a process running the module communicates with the controlling machine with a JSON-based protocol via its standard input and output. When Ansible is not managing a node, it does not consume resources on the node because no daemons are executing of software installed.

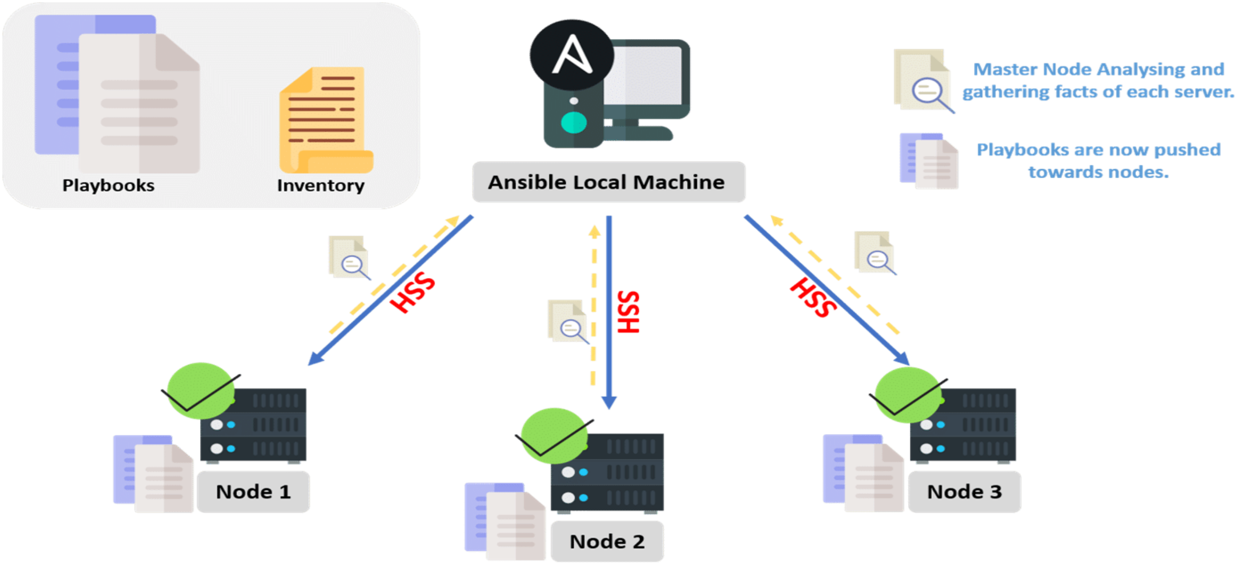

Playbooks
Inventory
Ansible Local Machine
Master Node Analysing and gathering facts of each server.
Playbooks are now pushed towards nodes.
SSH
SSH
SSH
Node 1
Node 2
Node 3

Figure 2: Ansible Architecture

Design goals

The design goals of Ansible include:

- Minimal in nature. Management systems should not impose additional dependencies on the environment.
- Consistent. With Ansible one should be able to create consistent environments.
- Secure. Ansible does not deploy agents to nodes. Only OpenSSH and Python are required on the managed nodes.
- Highly reliable. When carefully written, an Ansible playbook can be idempotent, to prevent unexpected side-effects on the managed systems. It is entirely possible to have a poorly written playbook that is not idempotent.
- Minimal learning required. Playbooks use an easy and descriptive language based on YAML and Jinja templates.

Modules

Modules are mostly standalone and can be written in a standard scripting language (such as Python, Perl, Ruby, Bash, etc.). One of the guiding properties of modules is idempotency, which means that even if an operation is repeated multiple times (e.g., upon recovery from an outage), it will always place the system into the same state.

Inventory configuration

The Inventory is a description of the nodes that can be accessed by Ansible. By default, the Inventory is described by a configuration file, in INI or YAML format, whose default location is in /etc/ansible/hosts. The configuration file lists either the IP address or hostname of each node that is accessible by Ansible. In addition, nodes can be assigned to groups.

An example inventory:

```
192.168.6.1
```

This configuration file specifies three nodes: the first node is specified by an IP address and the latter two nodes are specified by hostnames. Additionally, the latter two nodes are grouped under the webservers group.

Ansible can also use a custom Dynamic Inventory script, which can dynamically pull data from a different system.

Playbooks

Playbooks are YAML files that express configurations, deployment, and orchestration in Ansible, and allow Ansible to perform operations on managed nodes. Each Playbook maps a group of hosts to a set of roles. Each role is represented by calls to Ansible tasks.

5.3 IMPLEMENTATION OF THE SYSTEM

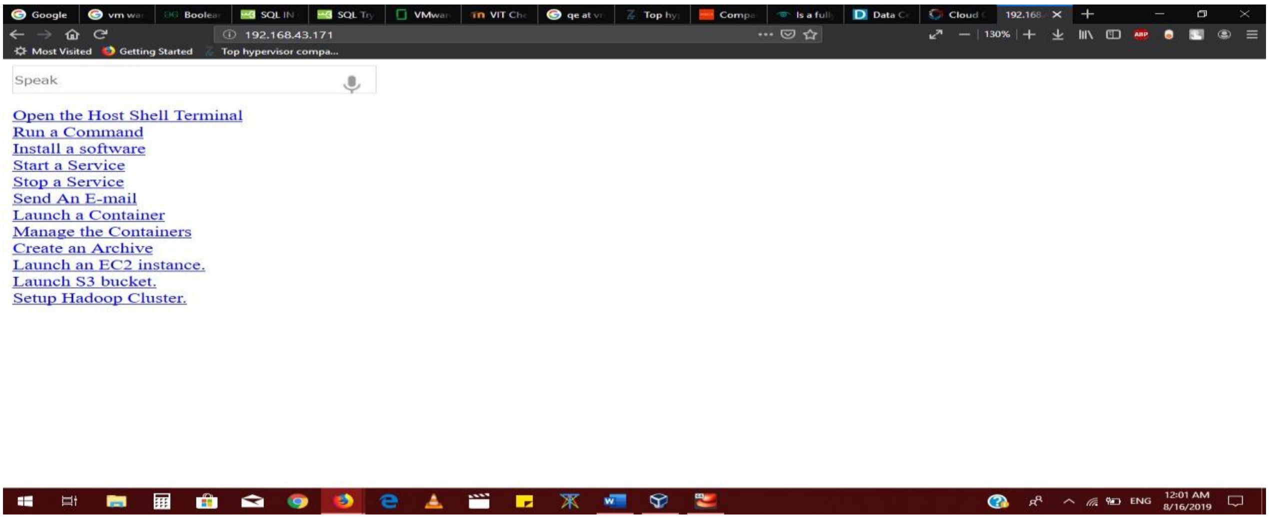


Figure 3 ; Home page of Server-Side Industrial Automation System Web Portal

This is a site in which we have encapsulated the project's main idea. This is a html front face with python CGI running in backend. We have also given some CSS styling to make it a bit attractive. The site uses Ansible notebooks when called, to execute the given task. For e.g.: if we want to run a docker/container we just need to click the button run a docker and as we have put the speech recognition feature too, you can even speak it out "Run a docker" to execute the command. Obviously, it will ask for certain interactive questions like

1. Do you want it on your local system or on a remote system?
2. If remote desktop then provides it with the necessary IP Address details
3. Also provide which image of a docker would you like to choose.

The site in backend tracks the docker ansible playbook and then it will run on the local/remote host as per the given instructions.

6. COMPARISON OF COMPETENCY LEVELS BEFORE AND AFTER THE INTERNSHIP AND SELF EVALUATION

Despite the fact that it was a course of only one and half months, the experience was altogether extraordinary. I saw an enormous change in myself as far as self-contemplation. I pushed my cut-off points and attempted to discover some new information and it turned out to be beneficial. Additionally, I felt that the educational program covers every one of the things yet then I understood that there were things up to a level and further we need to pursue ourselves.

The significant part that influenced my perspective and working is that we need to work everything under a cut-off time and a pre-characterized technique. Likewise, presently I am sure about myself that any work given to me should be possible by me. It can require some investment, however without a doubt it will be done, as I have confronted a few difficulties during the temporary position. Likewise, the thing which made me increasingly more centred around the work is the way that now I realize that other's work is additionally subject to my work, so following a timetable carefully is must during occupation. Likewise, before I was not ready to work in group as I was a little loner yet in the wake of getting into the organization and getting in discussion routinely with the colleagues, I understood that the vast majority of the assistance you get in an organization is from your partners and associates working with you. It makes progressively helpful for us to get an answer as opposed to looking on the web, on the grounds that there are a ton of subject specialists in any industry. This improved a great deal significant part inside me.

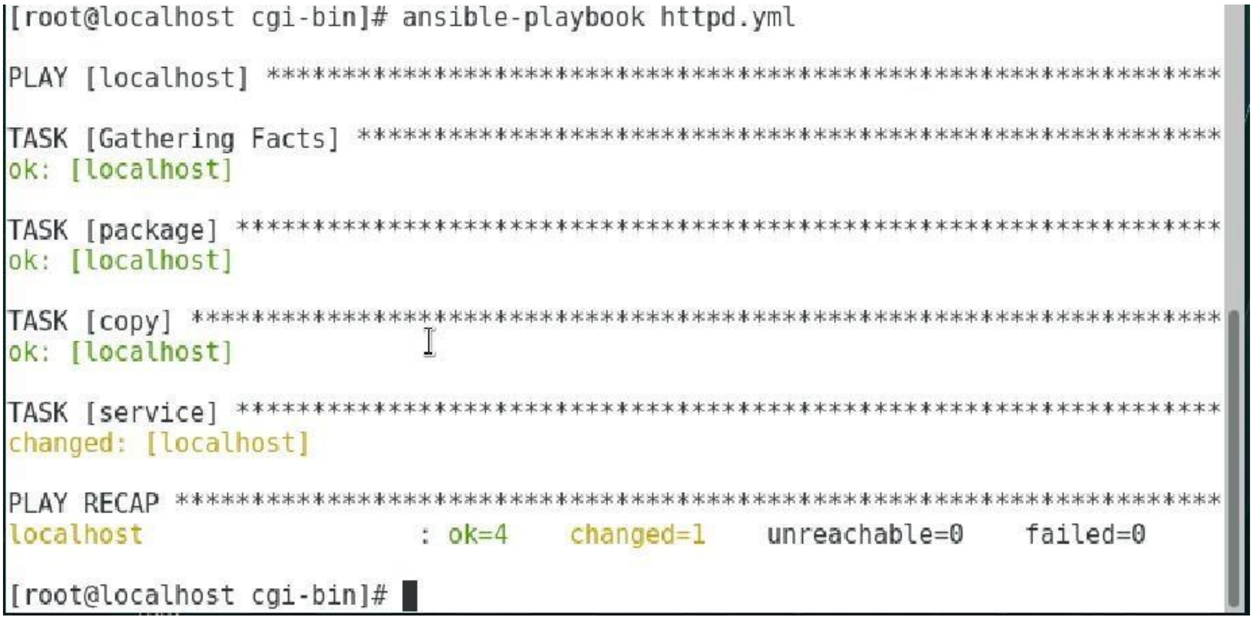
During this undertaking I adapted some corporate culture, group the board, great basic leadership, demonstrable skill, venture arranging, cohesiveness and asset the executives. In fact, I became acquainted with about different research areas in Computer Science and their uses and learnt essential application advancement which improved and honed my coding abilities too.

APPENDIX 1: ANSIBLE

Playbook to configure httpd service

```
- hosts: all   tasks:        - package:
                 name:       "httpd"
state: present

- copy:
                dest:              "/var/www/html"
src: "/var/www/html/index.html"

- service:
                name:
"httpd"             state:
started
enabled: yes
```

Output:



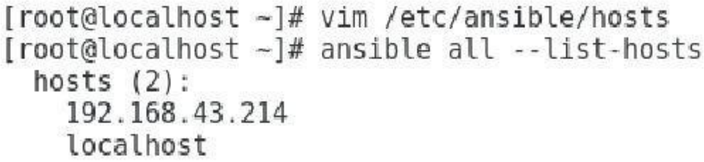
```
[root@localhost cgi-bin]# ansible-playbook httpd.yml

PLAY [localhost] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [package] *****************************************************************
ok: [localhost]

TASK [copy] ********************************************************************
ok: [localhost]

TASK [service] *****************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=0

[root@localhost cgi-bin]# 
```

Figure 4 : Output of what runs in backend when httpd playbook is executed

This is the playbook that would run in backend if “configuration of http service” is chosen as one of the many offered tasks. This output is not shown it can be seen only if we run it manually. Hence it is just for understanding the backend part

Note: All the yml files(i.e the playbooks) are stored under the /var/www/cgi folder in the given OS (i,e RHEL)

Listing the hosts present in ansible inventory file:

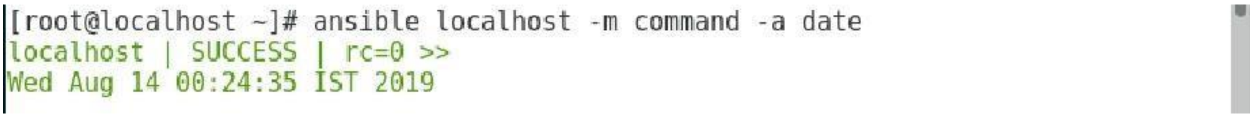


```
[root@localhost ~]# vim /etc/ansible/hosts
[root@localhost ~]# ansible all --list-hosts
  hosts (2):
    192.168.43.214
    localhost
```

Figure 5 : Inventory host file

The above command lists out all the nodes that needs to be configured. Here we can see tha IP 192.168.43.214 and localhost are the 2 nodes. Here we have taken an example that’s why we took 2 nodes but in real life scenario as there are many nodes in an actual company.

Basic Ansible Command



```
[root@localhost ~]# ansible localhost -m command -a date
localhost | SUCCESS | rc=0 >>
Wed Aug 14 00:24:35 IST 2019
```

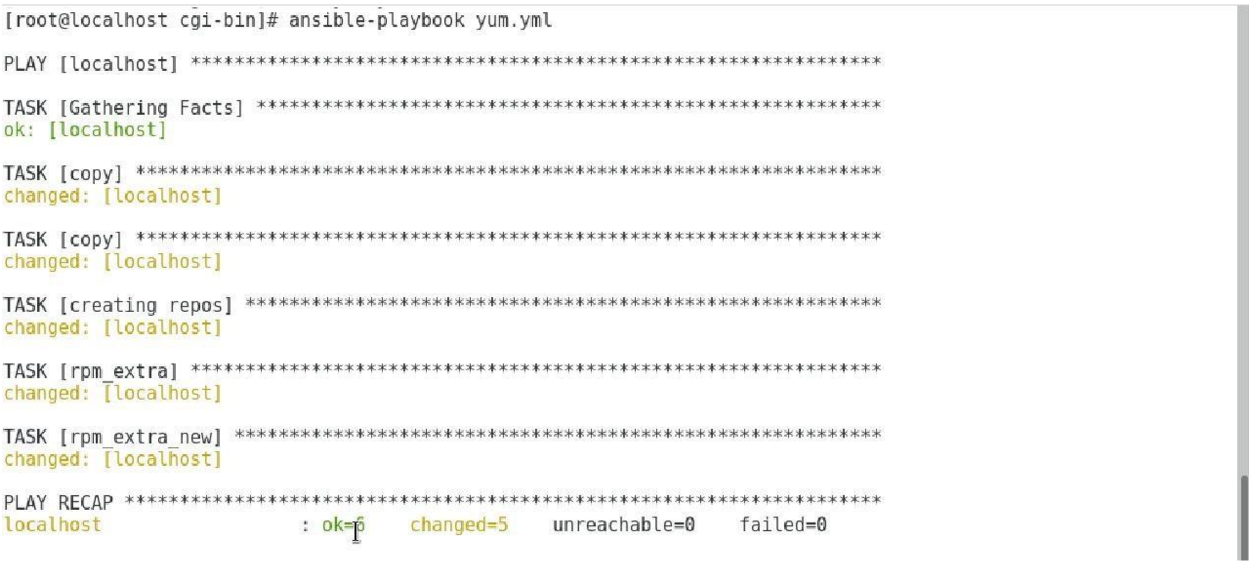
Figure 6 : Simple example of an ansible adhoc command

The above command is a basic command that shows the date of localhost. If we want we can also write IP of any of the node that is present in the ansible inventory file

Playbook to configure yum

```
-          hosts: all   tasks:     - copy:
        src:
/root/rhel7_5_rpm_extras
dest: /root     - copy:
        src:      /root/rhel7_extra_new_rpm
dest: /root
-          name: creating repos
yum_repository:
        name: dvd        description: dvd yum repo        file:
dvd        baseurl: file:///run/media/root/RHEL-7.5\
Server.x86_64        gpgcheck: no     - name: rpm_extra
yum_repository:        name: rpm_extra        file:
rpm_extra        description: rpm1        baseurl:
file:///root/rhel7_5_rpm_extras
        gpgcheck: no
-          name: rpm_extra_new
yum_repository:        name: rpm_new
description: rpm2        file: rpm_new
baseurl:
file:///root/rhel7_extra_new_rpm
gpgcheck: no
```

Output:



```
[root@localhost cgi-bin]# ansible-playbook yum.yml

PLAY [localhost] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [copy] ********************************************************************
changed: [localhost]

TASK [copy] ********************************************************************
changed: [localhost]

TASK [creating repos] **********************************************************
changed: [localhost]

TASK [rpm_extra] ***************************************************************
changed: [localhost]

TASK [rpm_extra_new] ***********************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=6    changed=5    unreachable=0    failed=0
```

Figure 7 : Output after running the yum configuration playbook

Note : The green color shows that no change has been made to the system state or whichever state it suggests . And the yellow color shows change in the prevous settings. For eg : It is clearly visible that [Gathering facts] has no change in it , hence it is in green color. While the [copy] has yellow color . This shows that the copy task has been executed successfully and it was done for the first time(i.e no other file of same name has been copied to the same stination ever before.)

APPENDIX 2: HADOOP

Playbook to configure Hadoop cluster

```
-        hosts: all   tasks:
-        name: Check the hadoop
software     command: "rpm -q
hadoop"     register: hq

-        name: Check the java
version     command: "java -
version"     register: jq

 # - name: Sending the JDK Software
#   copy:
 #        src: /media/sf_share/Software/jdk-8u171-linux-x64.rpm
 #        dest: /root/Desktop/
 #    when: "'Java HotSpot' not in jq.stdout"

#  - name: Sending the Hadoop Software
#    copy:
#        src: /media/sf_share/Software/hadoop-1.2.1-1.x86_64.rpm
#        dest: /root/Desktop/
#    when: "'not installed' in hq.stdout"

#  - name: "Installing JDK "
#    package:
#        name: /root/Desktop/jdk-8u171-linux-x64.rpm
#        state: "installed"
#    register: x
#    ignore_errors: true
#    when: "'Java HotSpot' not in jq.stdout"

#  - get_url:
#        dest: /root/jdk.rpm
#        url: #Write the url here
#    when: "'No package' in x.results[0] "
#    ignore_errors: True

#  - name: "Installing Hadoop "
#    command: "rpm -ivh /root/Desktop/hadoop-1.2.1-1.x86_64.rpm  --force"
#    when: "'not installed' not in hq.stdout"

#  - name: " Setting the path of Oracle JAVA"
#    command: "{{ item }}"
#    with_items:
#        - "export JAVA_HOME=/usr/java/jdk1.8.0_171-amd64"
#        - "export PATH=/usr/java/jdk1.8.0_171-amd64/bin/:$PATH"
#    ignore_errors: True
#    when: "'Java HotSpot' not in jq.stdout"

#  - name: "Setting the path of JAVA permanent"
#    shell: "{{ item }}"
#    with_items:
#        - "echo 'export JAVA_HOME=/usr/java/jdk1.8.0_171-amd64' >> /root/.bashrc"
#        - "echo 'PATH=/usr/java/jdk1.8.0_171-amd64/bin/:$PATH' >> /root/.bashrc"
#    when: "'Java HotSpot' not in jq.stdout"
```

```
-          hosts: name   tasks:
-          name:              "Making
Directory"    file:
      path:      "/share"
state: "directory"
-          name:    "Copying    the
HDFS     configuration     file"
copy:
      src: "/code2/automate/hdfs-site-
name.xml"       dest: "/etc/hadoop/hdfs-
site.xml"   - name: "Copying the Core
configuration file"    copy:
      src: "/code2/automate/core-
site.xml"       dest: "/etc/hadoop/core-
site.xml"   - name: "Configuring the
HDFS file"    lineinfile:
      path: /etc/hadoop/hdfs-site.xml
regexp: '^<value>dir</value>'
line: "<value>/share</value>"   -
name: "Configuring the CORE file"
lineinfile:
      path: /etc/hadoop/core-site.xml       regexp:
'^<value>ip</value>'       line: "<value>hdfs://{{
groups['name'][0] }}:9001</value>"

-          name:    "Reformat    the    File
System"    command: "echo Y | hadoop
namenode  -format"    -  name:  "  Start
Hadoop   Services"            command:
"hadoop-daemon.sh start namenode"

-          name:   "Starting   the   Firewall
Services"    service:
    name:    "firewalld"
state: "restarted"

-     name:     "Firewall     Rule"
firewalld:
    port: "9001-
9002/tcp"
immediate: true
permanent: true
state: "enabled"


-     hosts: data   tasks:
-     name: "Making Directory"   file:
    path:    "/data"
state: "directory"
-     name:   "Copying   the   HDFS
configuration file"   copy:
    src: "/code2/automate/hdfs-site-
data.xml"     dest: "/etc/hadoop/hdfs-
site.xml"   - name: "Copying the Core
configuration file"   copy:
    src: "/code2/automate/core-
site.xml"     dest: "/etc/hadoop/core-
site.xml"   - name: "Configuring the
HDFS file"   lineinfile:
    path: /etc/hadoop/hdfs-site.xml
    regexp: '^<value>dir</value>'
line: "<value>/data</value>"   -
name: "Configuring the CORE file"
lineinfile:
    path: /etc/hadoop/core-site.xml     regexp:
'^<value>ip</value>'     line: "<value>hdfs://{{
groups['name'][0] }}:9001</value>"
```

```
-        name:   "   Start   Hadoop   Services"
command: "hadoop-daemon.sh start datanode"

-        name: "Starting the Firewall Services"
service:
     name:     "firewalld"
state: "restarted"

-        name: "Firewall Rule"    firewalld:
     port: "9001-
9002/tcp"
immediate: true
permanent: true
state: "enabled"
```