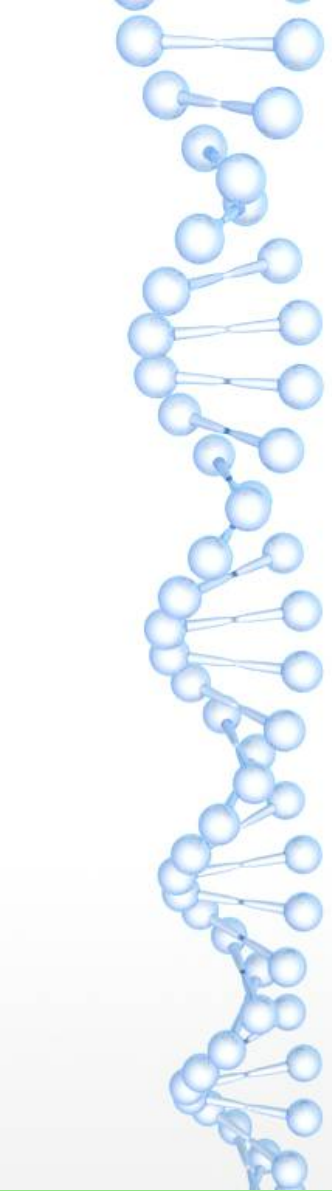


CSI-ZG628T
DISSERTATION PRESENTATION

Study of Ansible as an automation tool for Site Reliability

by
BASKAR BALASUBRAMANIAN
BITS ID - 2019HT66015

MENTOR INFORMATION

ID No.	**: 2019HT66015**
NAME OF THE STUDENT	**:** BASKAR BALASUBRAMANIAN
EMAIL ADDRESS	**:** 2019HT66015@wilp.bits-pilani.ac.in
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	**:** IBM India
SUPERVISOR	**:** Swetha J, SAP Technical Architect, IBM
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	**:** IBM India, Chennai
SUPERVISOR'S EMAIL	**: swetha.iyer@in.ibm.com**
ADDITIONAL EXAMINER	**:** Raghu Srinivasan, Senior Technical Staff Member - Technology Architect Lead Client Transformation SRE
ADDITIONAL EXAMINER'S EMAIL	**: rsriniv@us.ibm.com**
BITS FACULTY	**:** Rajesh Kumar, Assistant Professor, Dept of Computer Science
BITS FACULTY EMAIL	**: rajesh.k@pilani.bits-pilani.ac.in**
DISSERTATION TITLE	**:** Study of Ansible as an automation tool for Site Reliability

Website from the End User Point of View..

how flexible is the system
when there is a need to address
an increase in the number of users ?

Is the application available
every time accessed by the user?

how quickly
can the system recover from a failure?

how effectively
application changes can be incorporated?

What is the level of
confidentiality and integrity
that the system provides to user's data
within the systems and the network?

how robust does the system responds
to sudden surge or drop in the processing load?

Is there any economic value?

Reliability of a Site

A **Site** can be defined as any useful application or software available for use over computer networks which is accessible over the Internet or private interconnected networks.

Reliability of a Site

- Reliability is directly associated with user experience.
- From the perspective of the service provider, they would have to satisfy the expectations of the end users and facilitate the best experience.

A reliable site is one where the users of the sites would fall back for the services of the service provider.

Sites are considered reliable from user experience perspective, which becomes the primary goal for anyone providing information services.

When reliable services are realized with the hosted websites, then the sites are meant to have an added quality called as **Site Reliability.**

The art of practicing the principles to meet the expectations from reliability perspective can be named as **Site Reliability Engineering.**

About the project - PROJECT OBJECTIVE

Address the problem of site reliability through:

- **Automation** of tasks that are manual, repetitive, tactical, devoid of enduring value, and that scales linearly as a service grows. In IT industry terms this could be called as **eliminating the toil** or backlogs.

- **Measurement and Interpretation** of the system data which is essential in a system that automatically adjusts its resources and configurations, there by meet the demands of the end users. This could be termed as the **Observability** principle in Site Reliability.

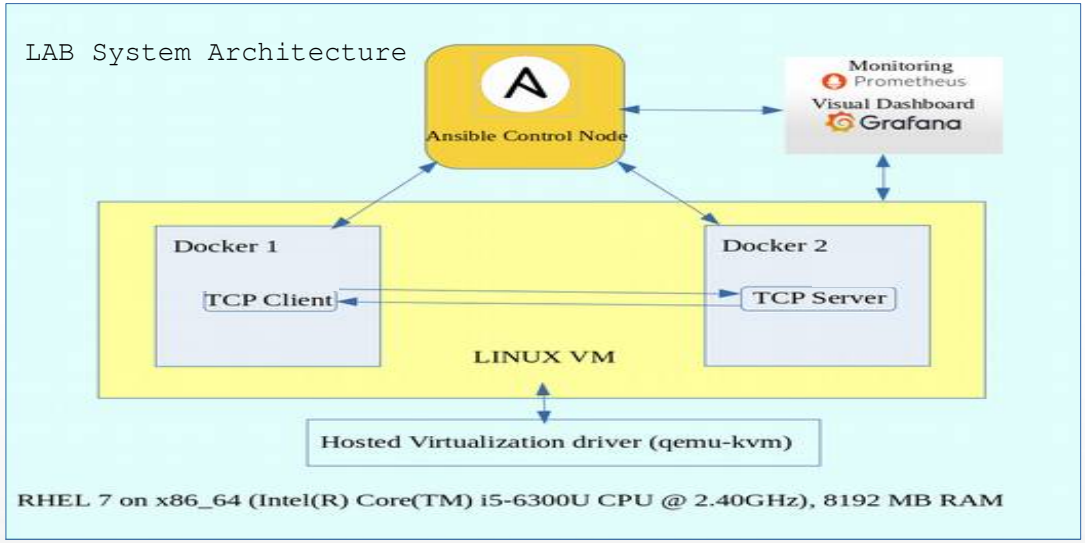
- **Alerting** the support personnel and experts and effective communication among them, about the system malfunctioning and take corrective actions for speedy recovery. This functionality is normally categorized as **Event Management** or **Incident Management** based on the severity of the issue.

About the project - SCOPE OF WORK

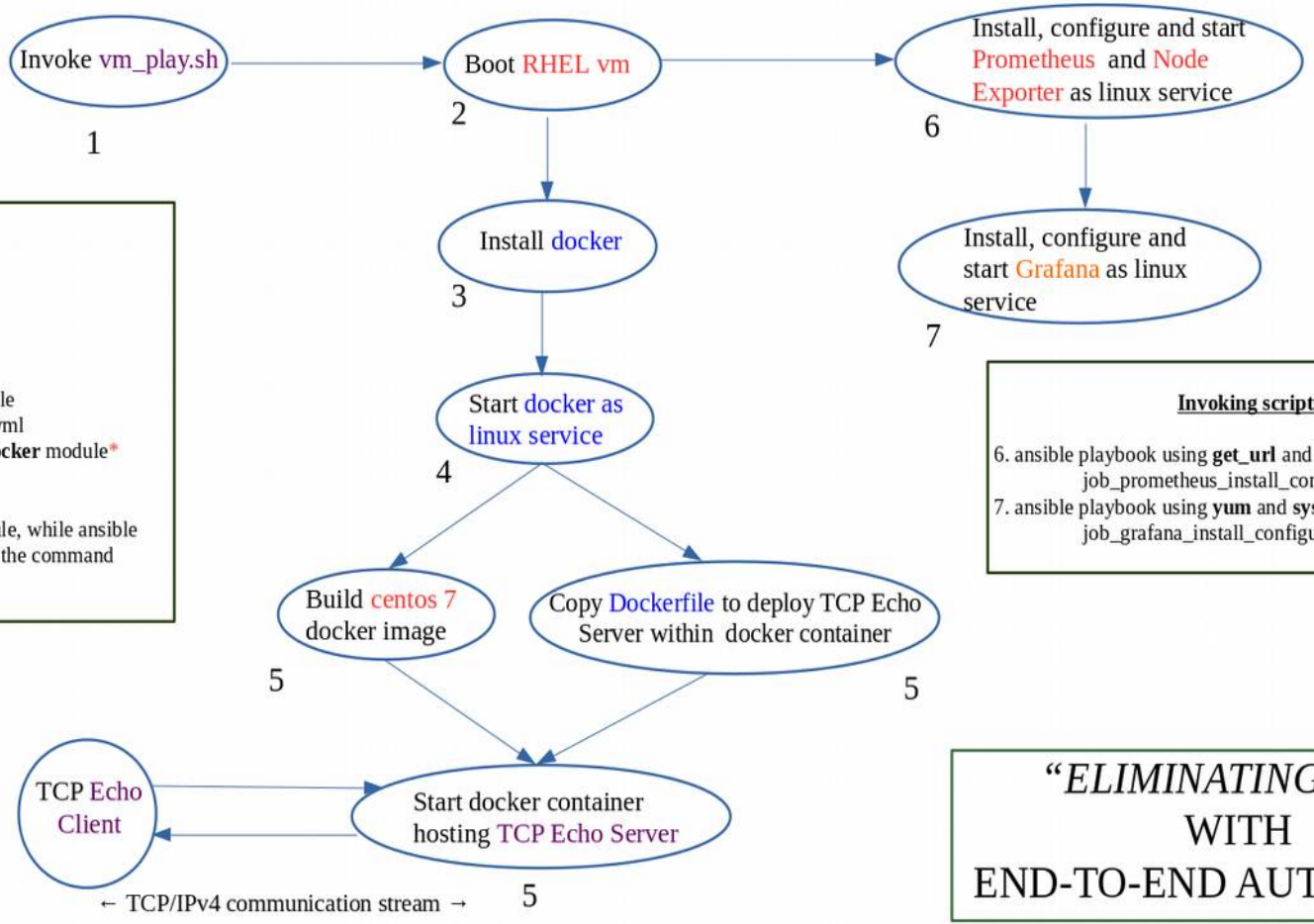
- Study, practice and document how **Ansible** would help **to automate system functions** and reduce toil using the rich set of ansible modules.
- **System Monitoring and Visualization** of the metrics using **Prometheus** and **Grafana** respectively.
- **Alerting** when there is anamoly found in the system functionality **by sending an email** to the systems support.

Use case to eliminate toil with ansible automation

- Boot a Linux virtual machine from a pre-built image
- Install Docker on top of the VM
- Spin up a Docker container with TCP Echo Server hosted in it.
- Demonstrate TCP/IP Client-Server communication between TCP Echo Client and Echo Server



END TO END AUTOMATION FLOW



Invoking scripts

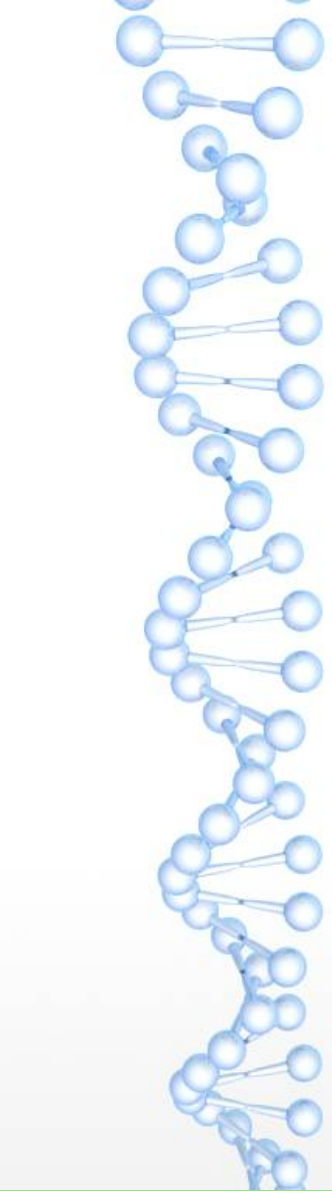
1. **shell** script
   vm_play.sh
2. ansible command using **virt** module
3. ansible playbook using **yum** module
4. ansible playbook using **systemd** module
   job_system_services_control.yml
5. ansible playbook using **command / docker** module*
   job_docker_operations.yml

* The project work uses command module, while ansible study reveals docker module can replace the command module

Invoking scripts

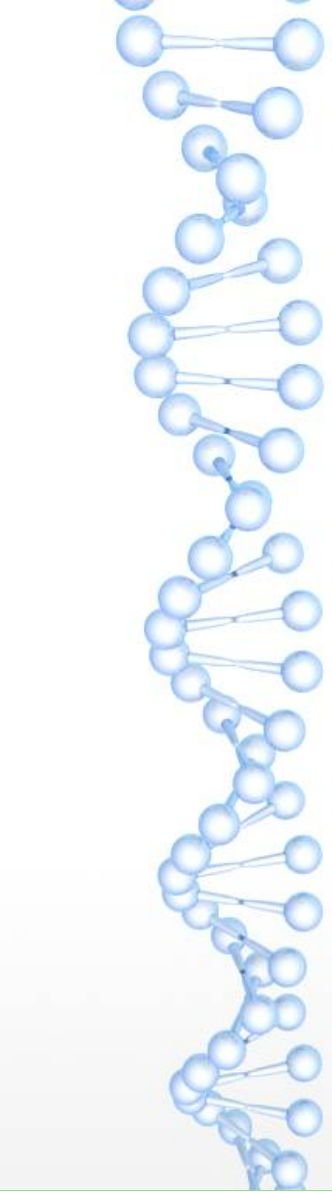
6. ansible playbook using **get_url** and and **systemd** modules
   job_prometheus_install_configure_start.yml
7. ansible playbook using **yum** and **systemd** modules
   job_grafana_install_configure_start.yml

"ELIMINATING TOIL"
WITH
END-TO-END AUTOMATION



ASSUMPTIONS IN THE AUTOMATION

- RHEL VM is started from a **pre-built OS image.** Building an image from the scratch is not within the scope in this project.
- **Network interfaces** are **well defined and configured** on both host and virtual operating systems., so that the development work in the project involves only at the application layer and TCP layers of the Network protocol stack.
- Linux VM would be readily **connecting to the tool repositories**, which includes RHEL repositories, Prometheus and Grafana download locations, etc.
- **Network firewall rules are enabled** so that access to the TCP ports used for prometheus (9090), node_exporter (9100) and grafana (3000) is available to the host Linux machine.
- **SMTP configuration is setup** on the Linux VM in order to send the alert messages from Grafana to the support personnel.



Use case for Monitoring and Alerting

Project designed to check the following two basic information on the server.

- TCP Echo Server Running Status – up (1) or down (0)
- TCP Echo Server Socket State – Listening (1) or Not listening (0)

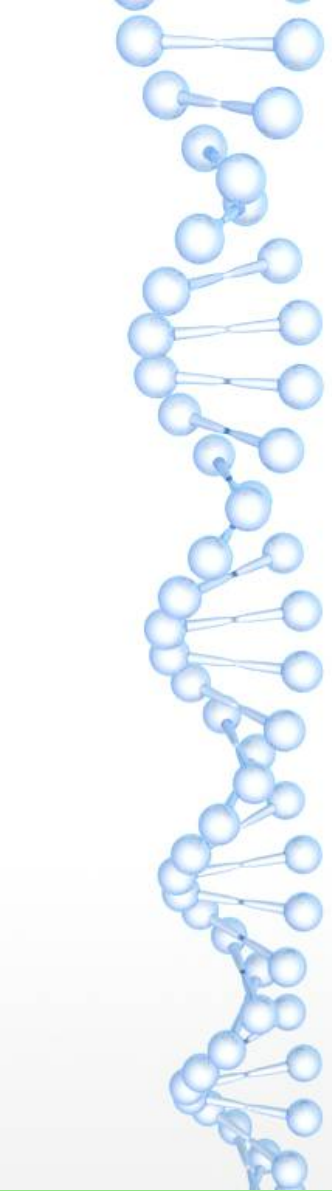
Using Node Exporter Standard Dashboard the following metrics are visualized.

- CPU Usage
- Load Average
- Memory Usage
- Disk I/O
- Disk Usage
- Network Received
- Network Transmitted

Monitoring tools used are prometheus and node_exporter

Visual dashboards are created using Grafana

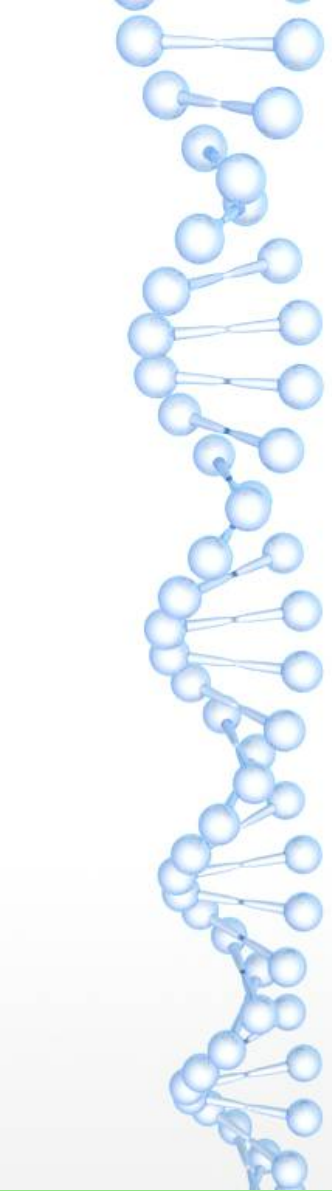
E-mail sent to support personnel when



DEMONSTRATION VIDEO

https://www.youtube.com/watch?v=J3yoHYbJeJs

- Automation using Ansible
- Monitoring with Prometheus
- Visualization and Alerting with Grafana



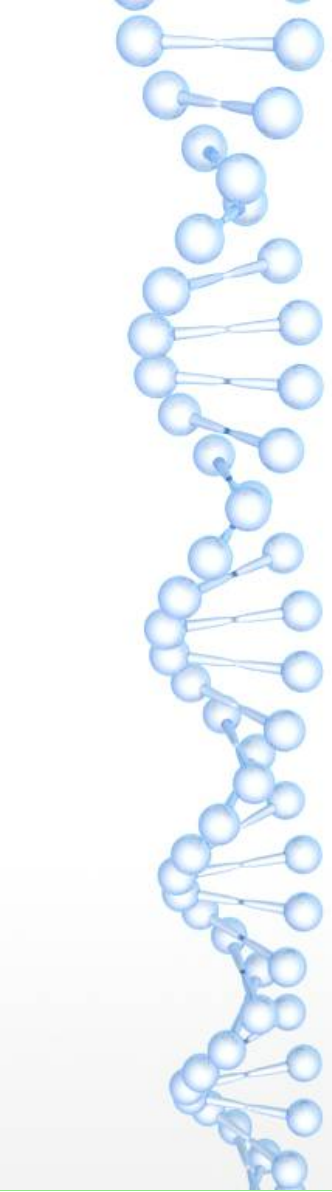
• SCOPE FOR FUTURE WORK

- The study can help students in the future to make extensive study about the fundamental nature of TCP communication between client and server.

- Create additional custom metrics for analysing the server behaviour and performance. For example response times to the client requests, number of clients connected to the server can be added to the future scope of the project.

- The study can replace TCP Echo Server with any other server like webservers, app servers or db servers by changing a single DockerFile configuration to address the problem of Site Reliability on those servers.

- RHEL pre-built OS image can be replaced with free Linux OS like centos or ubuntu, so that entire project can be made available in a public repository.



THANK YOU.