

ULRENA – Simplifying Automation

A Project Report submitted in partial fulfilment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
with IBM specialization in INTERNET OF THINGS & SMART CITIES

Submitted by

JAI GUPTA (Enroll No. R164216024)
MAYANK GOEL (Enroll. No. R164216034)
NITISH SINGH (Enroll. No. R164216042)
SATVIK SHARMA (Enroll. No. R164216060)

Under the guidance of

Mrs. Amber Hayat
Sr. Assistant Professor
Department of Systemics, SoCSE



UNIVERSITY WITH A PURPOSE

SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM & ENERGY STUDIES
Bidholi Campus, Energy Acres, Dehradun – 248007.

March - 2020



I hereby declare that this submission is my own and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other Degree or Diploma of the University or other Institute of Higher learning, except where due acknowledgement has been made in the text.

JAI GUPTA (Enroll No. R164216024)

MAYANK GOEL (Enroll. No. R164216034)

NITISH SINGH (Enroll. No. R164216042)

SATVIK SHARMA (Enroll. No. R164216060)

CERTIFICATE

This is to certify that the project titled **ULRENA - Simplifying Automation** submitted by JAI GUPTA (Enroll No. R164216024), MAYANK GOEL (Enroll. No. R164216034), NITISH SINGH (Enroll. No. R164216042) & SATVIK SHARMA (Enroll. No. R164216060) to the University of Petroleum & Energy Studies, for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING is a bonafide record of project work carried out by them under my supervision and guidance. The content of the project, in full or parts have not been submitted to any other Institute or University for the award of any other degree or diploma.

Date: 20/04/2020

Mrs. Amber Hayat
(Sr. Assistant Professor)

Dr. Neelu Ahuja
(Dept. of Systemics)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide Mrs. **Amber Hayat**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected HoD of the Department, **Dr. Neelu Ahuja**, for his great support in doing our project in **Area (DevOps, Ansible, etc.)** at **SoCS**.

We are also grateful to **Dr. Manish Prateek, Director** SoCS, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	Jai Gupta	Mayank Goel	Nitish Singh	Satvik Sharma
Roll No.	R164216024	R164216034	R164216042	R164216060

ABSTRACT

Companies nowadays are working on different technologies due to which they must install various software for executing different tasks. They need to install them manually in different machines, which creates chaos for them and takes a lot of time. With the help of newer technologies, we find it easy for implementing such things inside a company. These objective working can help the company to save time and resources on such small tasks. We helped them out to make an automated user input by which we can carry out such tasks without wasting their time. Thus, by the help of our project they can carry out such tasks much more easily.

Keywords: *Automation, DevOps, Ansible. Ad-hoc Commands, Git, Tomcat, Apache httpd, SonarQube, Maven.*

TABLE OF CONTENTS

S.NO.	CONTENTS	PAGE NO.
1	Introduction	8
2	Problem Statement	9
3	Objective	9
4	Theory	10
	4.1 Overview	10
	4.2 Concept 4.2.1 DevOps	10
	4.3 Technologies	13
	4.3.1 Ansible	13
	4.3.2 K8s	14
	4.3.3 Git	15
	4.3.4 SonarQube	15
	4.3.5 Apache	16
	4.3.6 Tomcat	16
	4.3.7 Jenkins	17
	4.3.8 Maven	18
5	Literature Review	19
6	Methodology	21
7	System Requirements	22
8	Schedule	23
9	Design	24
10	Output Screen	27
11	Conclusion	29
12	References	30

LIST OF TABLES & FIGURES

S.NO.	CONTENTS	PAGE NO.
1	Pert Chart	23
2	Use Case Diagram	24
3	Sequence Diagram	25
4	Flow Chart Diagram	26
5	Object Diagram - Ansible	26
6	Ulrena – Main Page	27
7	Website Technologies – I	27
8	Website Technologies – II	28
9	Jenkins Successful Installation	28

INTRODUCTION

In this project we aim to build an automation utility tool to deploy various technologies in various machines present at remote locations. In this project we are making an automated working of the different technologies in various machines by creating a tool, which will work on the basis of input given by the user, our tool will go to that place and find the appropriate output for the query asked. From our project we make it easy for the user to carry out different tasks without asking the operating system, our tool will automatically complete the task. These eases of tools will help out the companies in every aspect. We use them as a part of any process to be followed by the user and help the companies by making this project. We use various things to automate in our project, they are like: *Git*, *Tomcat*, *Apache httpd*, *SonarQube*, *Maven*, etc.

PROBLEM STATEMENT

- In office and work places, people find it difficult to install a service or software's.
- They find it difficult as they have to complete the entire process manually due to which the process becomes hectic and time taking.
- All this process also requires lots of things to be configured and many changes to be made manually which is a bigger task.
- It also costs a lot for a company for time as well as maintenance and services.

OBJECTIVE

- To create a tool which solves the problem for the IT companies to complete the tasks as per requirements and install the service and software's.
- With our project a user can completely deploy any service across 'n' number of computers with different configuration at once with a single cli-command or web-based working.
- Our project aims to solve these problems. It's a utility tool for automated deployment of infrastructure software services.
- Ulrena can help work by its own and complete the task of automation without the use of any manpower.

THEORY

OVERVIEW

Technology, an ever increasing thing, which is making changes in everything and automating things for the ease of life, which leads to the best practice to grow and learn. The world is ever increasing thing, which is going up as for the betterment of humanity and providing various helping features for the people around. These technical things are all controlled by humans which help the general people to understand the future technology and adopt it in our life and which will help them for living quality and providing with a different exposure to learn and work. We all want the futuristic design and automation of daily working to be done. Thus, our project deals with various technologies which helps them to carry out the tasks and help to install the various software's for many computers. This project which leads for the betterment of the people and helps to increase the overall productivity.

CONCEPT

DEVOPS:

- DevOps has become an increasingly common approach to software delivery that development and operations teams use to build, test, deploy, and monitor applications with speed, quality, and control.
- DevOps is a culture which promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way.
- The word 'DevOps' is a combination of two words 'development' and 'operations.'
- DevOps helps to increase an organization's speed to deliver applications and services.
- It allows organizations to serve their customers better and compete more strongly in the market.

- DevOps can be defined as an alignment of development and IT operations with better communication and collaboration.
- Development teams need to design, develop, deliver and run the software as quickly and reliably as possible.
- Operations teams need to identify and resolve problems as soon as possible by monitoring, predicting failure, managing the environment and fixing issues.

Benefits of DevOps

- Faster code delivery
- Faster time to market
- Higher-quality software
- Improved collaboration between developers and operations
- Decreased time to resolution for fixing bugs and vulnerabilities
- A culture that brings business, development, and operations together for improved responsiveness to market demands

DevOps tools

- **Define and plan**, which focuses on planning DevOps workflows for iterations, release management, and issue tracking. Notable tools or tool vendors in this space include Atlassian, CA Technologies, IBM, iRise, and Jama Software.
- **Code, build, and configure**, which focuses on code development and review, source code management, and code merging. Notable tools/tool vendors include BitBucket, Electric Cloud, GitLab, GitHub, and IBM.
- **Test**, which verifies that the quality of the software release and code are maintained throughout the development process and that the highest quality deploys to production. Notable tools/tool vendors include Delphix, FlawCheck, HP, IBM, Microsoft, Parasoft, SonarSource, Skytap, and ThoughtWorks.
- **Packaging and preproduction**, which refers to the activities involved once the release is ready for deployment; it's also called staging or

preproduction. Notable tools/tool vendors include IBM, Inedo's ProGet, Jfrog's Artifactory, Sonatype Nexus repository.

- **Release, deploy, and orchestration**, which is the process of actually releasing software and usually involves change management, release approvals, release automation, schedule orchestration, provisioning, and deploying into production. Tools/tool vendors in this space include Automatic, Clarive, BMC, IBM, Flexagon, Vmware, and XebiaLabs.
 - **Continuous management and configuration** includes continuous configuration automation, configuration management, and infrastructure as code. Notable tools/tool vendors include Ansible, Chef, IBM, Puppet Labs, Otter, and Salt.
 - **Monitoring** reports application performance and helps identify issues impacting the user experience. Tools/tool vendors include Big Panda, IBM, New Relic, Plumb, and Wireshark.
- DevOps grew out of Agile. Agile is a way of producing software in short iterations on a continuous delivery schedule of new features and bug fixes in rapid cycles from two to four weeks.

DevOps methodologies include the following:

- **Continuous integration**, which is where coding, building, integrating, and testing take place.
- **Continuous delivery**, which includes continuous integration, but mainly focuses on product releases.
- **Continuous deployment**, which focuses on automating releases of projects as soon as possible.
- **Operate** for conducting the development operations of configuration management and continuous monitoring.

DevOps strategy you can do the following:

- Accelerate the delivery of reliable software

- Balance speed, cost, quality, and risk with increased capacity to innovate
- Reduce time to customer feedback with improved customer experience

TECHNOLOGIES

(1) ANSIBLE

- Ansible is a software tool that provides simple but powerful automation for cross-platform computer support.
- It is primarily intended for IT professionals, who use it for application deployment, updates on workstations and servers, cloud provisioning, configuration management, intra-service orchestration, and nearly anything a systems administrator does on a weekly or daily basis.
- Ansible doesn't depend on agent software and has no additional security infrastructure, so it's easy to deploy.
- Because Ansible is all about automation, it requires instructions to accomplish each job. With everything written down in simple script form, it's easy to do version control.
- The practical result of this is a major contribution to the "infrastructure as code" movement in IT: the idea that the maintenance of server and client infrastructure can and should be treated the same as software development, with repositories of self-documenting, proven, and executable solutions capable of running an organization regardless of staff changes.
- While Ansible may be at the forefront of automation, systems administration, and DevOps, it's also useful to everyday users.
- Ansible allows you to configure not just one computer, but potentially a whole network of computers at once, and using it requires no programming skills.
- Instructions written for Ansible are human-readable. Whether you're entirely new to computers or an expert, Ansible files are easy to understand.

- A playbook is a configuration file written in YAML that provides instructions for what needs to be done in order to bring a managed node into the desired state.
- Playbooks are meant to be simple, human-readable, and self-documenting.
- If a playbook is run on a system that's already properly configured and in its desired state, then that system should still be properly configured after a playbook runs.

(2) K8s

- **Kubernetes** is an open-source Container Management tool which automates container deployment, container scaling, and descaling and container load balancing (also called as container orchestration tool).
- It is written in Golang and has a huge community because it was first developed by Google and later donated to CNCF (Cloud Native Computing Foundation).
- Kubernetes can group 'n' number of containers into one logical unit for managing and deploying them easily.
- It works brilliantly with all cloud vendors i.e. public, hybrid and on-premises. Kubernetes is an open-source platform that manages Docker containers in the form of a cluster.

Features of Kubernetes –

1. **Automated Scheduling**– Kubernetes provides advanced scheduler to launch container on cluster nodes. It performs resource optimization.
2. **Self-Healing Capabilities**– It provides rescheduling, replacing and restarting the containers which are died.
3. **Automated Rollouts and Rollbacks**– It supports rollouts and rollbacks for the desired state of the containerized application.
4. **Horizontal Scaling and Load Balancing**– Kubernetes can scale up and scale down the application as per the requirements.

(3) GIT

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Git is easy to learn and has a tiny footprint with lightning fast performance.
- It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.
- Git is designed to manage source code, which in most languages consists of lines of text.
- Git can manage other formats of files, but it treats them as blobs.
- It is a great way to manage versions of your files and it is a powerful tool.

(4) SONARQUBE

- SonarQube (formerly Sonar) is an open-source platform developed by Sonar Source for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages.
- SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities.
- SonarQube can record metrics history and provides evolution graphs.
- SonarQube provides fully automated analysis and integration with Maven, Ant, Gradle, MSBuild and continuous integration tools

(5) APACHE HTTPD

- Apache HTTPD is one of the most used web servers on the Internet.
- Apache HTTP Server is a free software/open source web server for Unix-like systems and other operating systems.
- A web server is a daemon that speaks the http(s) protocol, a text-based protocol for sending and receiving objects over a network connection.
- The http protocol is sent over the wire in clear text, using port 80/TCP by default (though other ports can be used).
- There is also a TLS/SSL encrypted version of the protocol called https that uses port 443/TCP by default.
- A basic http exchange has the client connection to the server.
- Then it requests a resource using the GET command. Other commands like HEAD and POST exist, allowing clients to request just metadata for a resource, or send the server more information.
- Apache features configurable error messages, DBMS-based authentication databases, and content negotiation.
- It is also supported by several graphical user interfaces (GUIs) which permit easier, more intuitive configuration of the server.
- The length of this content must match the length indicated in the Content-Length header.

(6) TOMCAT

- Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding.
- Described as a “reference implementation” of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions.
- Tomcat can be used as either a standalone product with its own internal Web server or together with other Web servers, including Apache, Netscape

Enterprise Server, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server.

- Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later.

(7) JENKINS

- **Jenkins** is an open source Continuous Integration server capable of orchestrating a chain of actions that help to achieve the Continuous Integration process (and not only) in an automated fashion.
- Jenkins is free and is entirely written in Java.
- Jenkins is a widely used application around the world that has around 300k installations and growing day by day.
- It is a server-based application and requires a web server like Apache Tomcat.
- The reason Jenkins became so popular is that of its monitoring of repeated tasks which arise during the development of a project.
- Jenkins will continuously test your project builds and show you the errors in early stages of your development.
- Software companies can accelerate their software development process, as Jenkins can automate build and test at a rapid rate.
- Jenkins supports the complete development lifecycle of software from building, testing, documenting the software, deploying and other stages of a software development lifecycle.
- Jenkins also supports cloud-based architecture so that you can deploy Jenkins in cloud-based platforms.
- One of the reasons why many people don't implement Jenkins is due to its difficulty in installing and configuring Jenkins.
- Jenkins is an open source Continuous Integration server capable of orchestrating a chain of actions
- The biggest pros of Jenkins is that it is managed by the community which holds public meetings and take inputs from the public for the development of Jenkins projects

(8) MAVEN

- Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework.
- Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.
- In case of multiple development team's environment, Maven can set-up the way to work as per standards in a very short time.
- As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups.
- Maven simplifies and standardizes the project build process.
- It handles compilation, distribution, documentation, team collaboration and other tasks seamlessly.
- Maven increases reusability and takes care of most of the build related tasks.
- Maven provides developers ways to manage the following –
 - Builds
 - Documentation
 - Reporting
 - Dependencies
 - SCMs
 - Releases
 - Distribution
 - Mailing List

LITERATURE REVIEW

- Nishant Kumar Singh, Sanjeev Thakur, Himanshu Chaurasiya and Himanshu Nagdev in their paper [1] describes how cloud has become a predominant innovation in the current state of data innovation and there is a need to satisfy the growing interest of customers. The weight of data innovation partnerships continues to grow to provide the client application via a private cloud. This change has come about since countless customers have legitimately started contacting cloud merchants for help. In addition, DevOps groups are now much more concentrated as they are responsible for mechanization and supplying the entire condition with the client application. This document focuses on the computerization of client applications directly from the provisioning of conditions to the provision of the application.
- AiZhen Chen, Qiniu Cloud in their paper [2] describes Instructions for using Ansible to assemble a sparse reserve structure, to complete the activity and maintenance stage for a large number of poles of the information assortment and dispersion of tasks and other huge scope planning tasks. By subdividing the interface layer, the standby layer, the pilot layer, is passed on to recognize the variety of data and embark on circulation for the basic work environment, storage and organization. In addition, by modifying the Ansible source code to probe companies as they approach the judgment of parameters.
- Hylson V. Netto, Lau Cheuk Lung, Miguel Correia, Aldelir Fernando Luiz, and Luciana Moreira Sá de Souza in their paper [3] provides information about how PC virtualization has brought rapid asset provisioning to server farms and the organization of pay-per-use cost models. This document proposes to combine the advantages of coordination in a compartment of the card framework called Kubernetes (k8s), while trying to limit the size of the titles and offer a planned state replication. A convention was created by which the applications shared the accessible memory on Kubernetes and an evaluation was carried out to show the practicality of the proposal.

- Patrick Vogel, Thijs Klooster, Vasilios Andrikopoulos, Mircea Lungu in their paper [4] explains how Tens of thousands of web applications are written in Flask, a Python-based web framework. Despite a rich ecosystem of extensions, none allow the developer to better understand the evolution of the performance of their service. In this paper, they presented a Flask dashboard, a library that solves this problem. They presented the ease with which the library can be integrated into an existing web application, discussed some of the viewing perspectives the library offers, and highlighted some future challenges for similar libraries.
- Teemu Laukkanen, Kati Kuusinen, Tommi Mikkonen in their paper [5] expects to bring closer Devops and regulated software development. For starters, they had to make it easier for engineers to create controlled programming with the tools and practices they know. They were also intended to allow administrative specialists to gain confidence in the provisions given by manufacturers by characterizing a mapping between DevOps and improving administrative programming. To create closer coordination between improvement teams, executive prerequisites, deployment control, and the organization's portfolio will simplify the production of management evidence of sustainable progress.
- Rémi Pieplu in his paper [6] presents a strategy to simplify the organization of a ground control room which depends on a few isolated machines at certain stages of approval. Naturally, he had to start these steps from a typical and combined design. For this reason, he described how to configure, assemble and design frameworks using infrastructure as a code approach. We learnt to discover the advantages of this creative technique.

METHODOLOGY

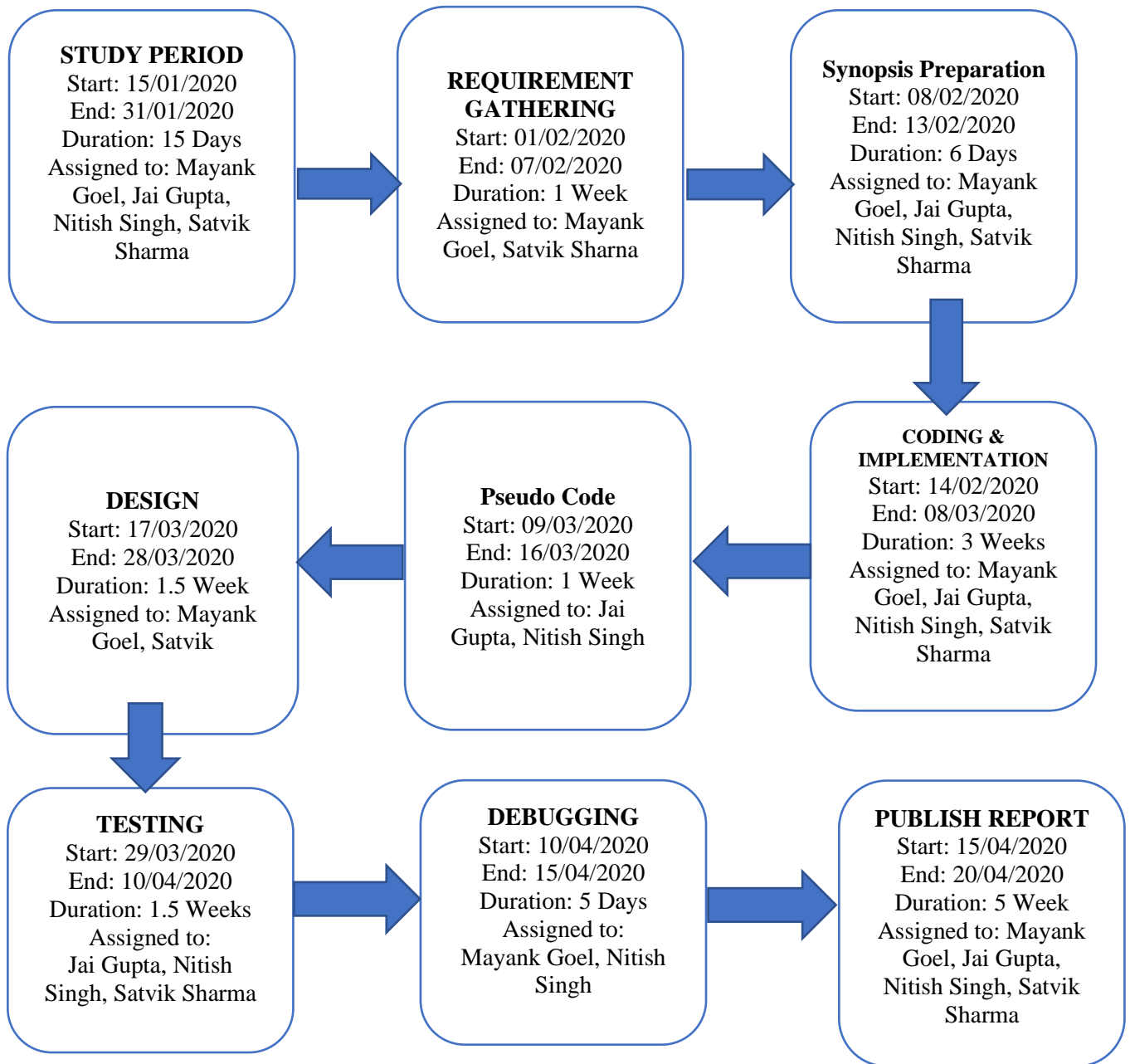
- Ulrena aims to provide an easy, intuitive and interactive way to assist the user in managing and deploying enterprise application services.
- Ulrena provides the user with a web dashboard UI, which is accessible through anywhere in the world.
- This dashboard provides a list of services which the user can select to deploy on either of local or remote workstations.
- The user selects the services through the dashboard which they wish to deploy and provides the hostname or ip address of the remote workstations.
- The selected services are then configured and deployed to the given workstations.
- The primary advantage which Ulrena provides to the user is that irrespective of the operating system of the remote workstation and its underlying hardware architecture the selected application services will be deployed.
- Ulrena provides a conceptual abstraction of the underlying architecture and thus even in cases of different operating systems on different systems the application services are deployed without concerning the user with hassles like using different versions, package manager etc. for the installation, configuration and deployment. Ulrena can install, configure and deploy the application services across multiple local and remote workstations.

SYSTEM REQUIREMENTS

- Hardware:
 - Processor: Core i3 and above
 - RAM: 4GB and above
 - Proper input and output devices

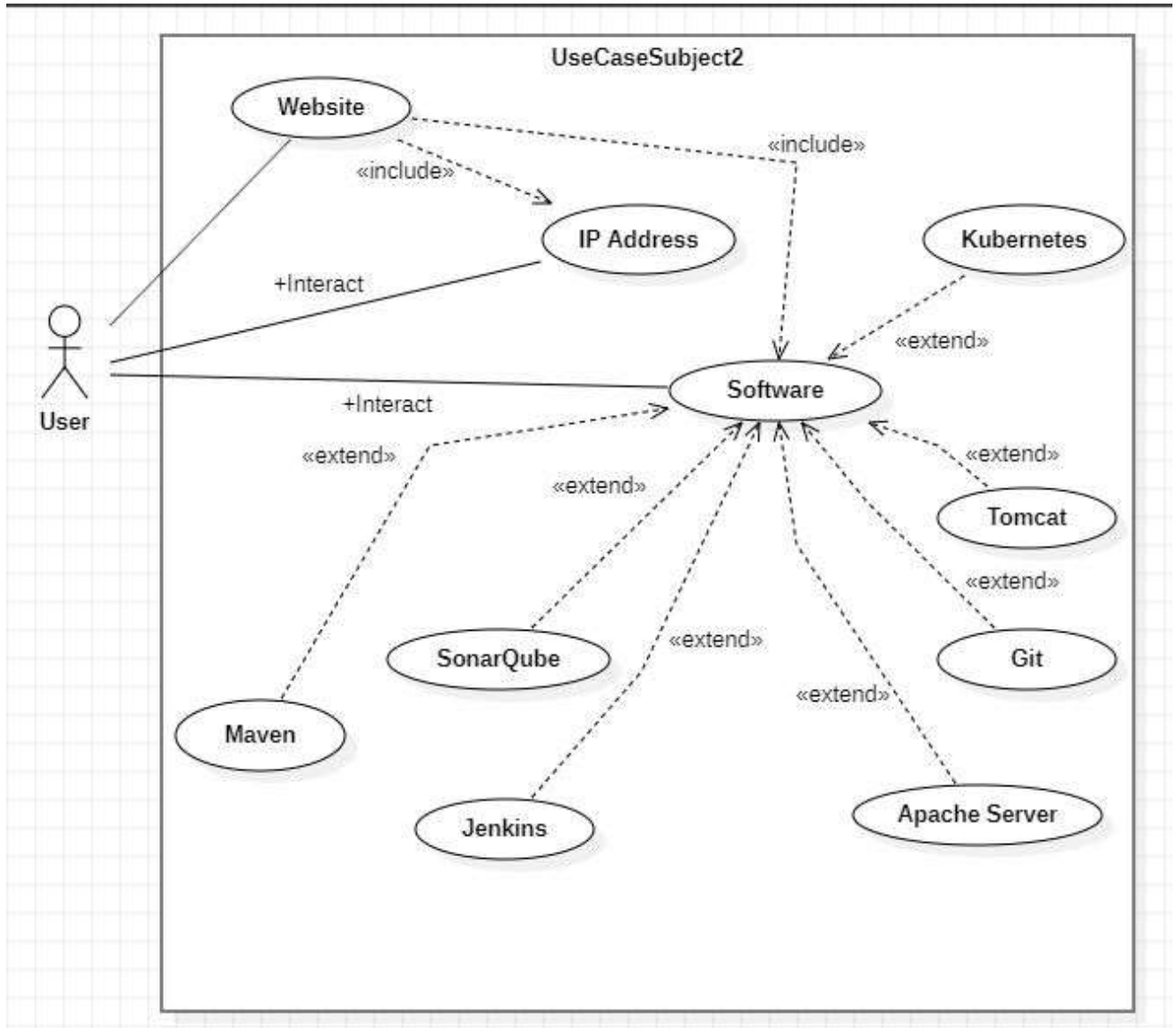
- Software:
 - Operating System: Windows 7 or above and RHEL (Version 7 and above)
 - Virtualization Tool: Oracle VirtualBox
 - Cloud platform: AWS
 - Languages/Modules: Python3, VirtualENV, Flask, YAML

SCHEDULE

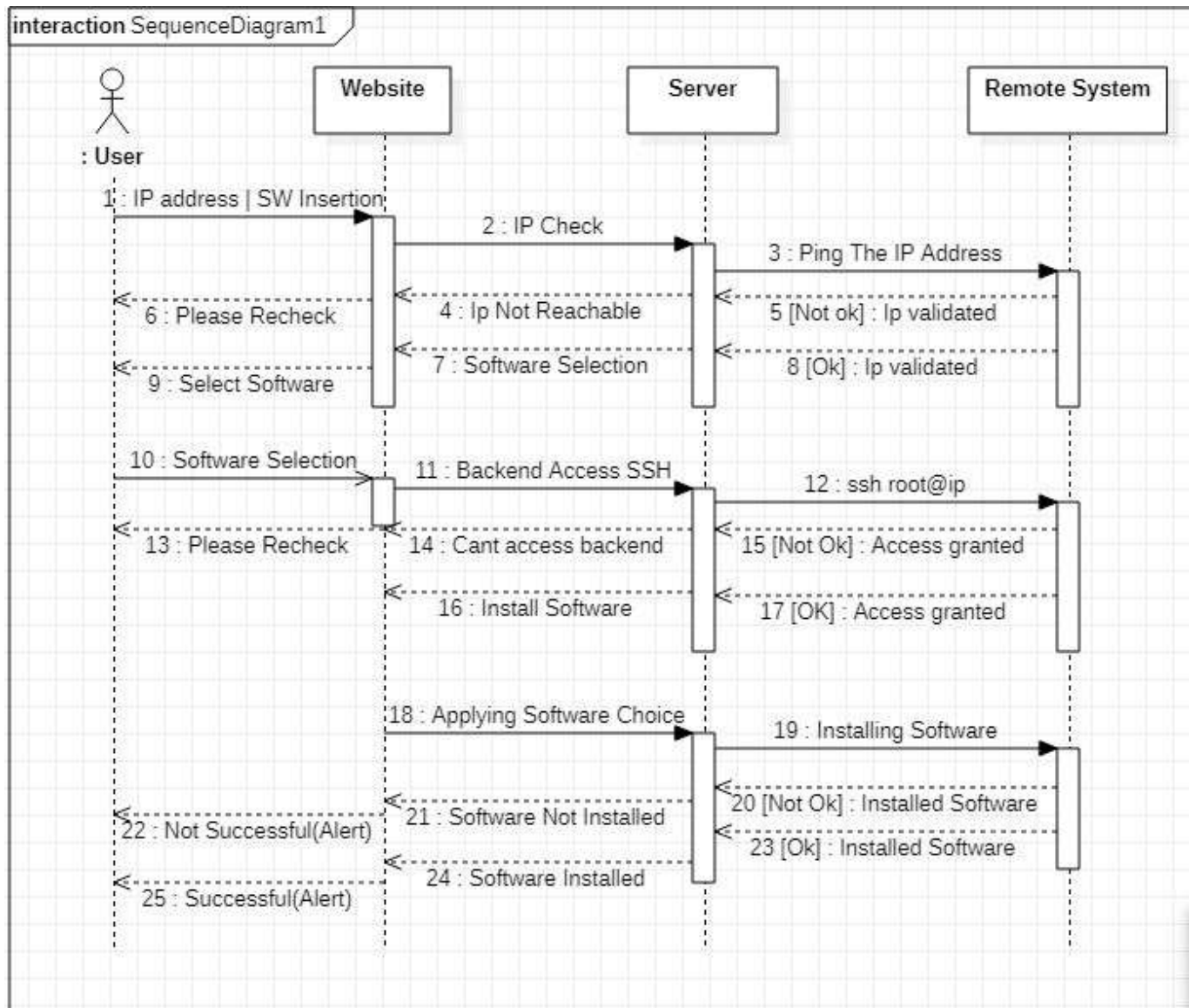


DESIGN

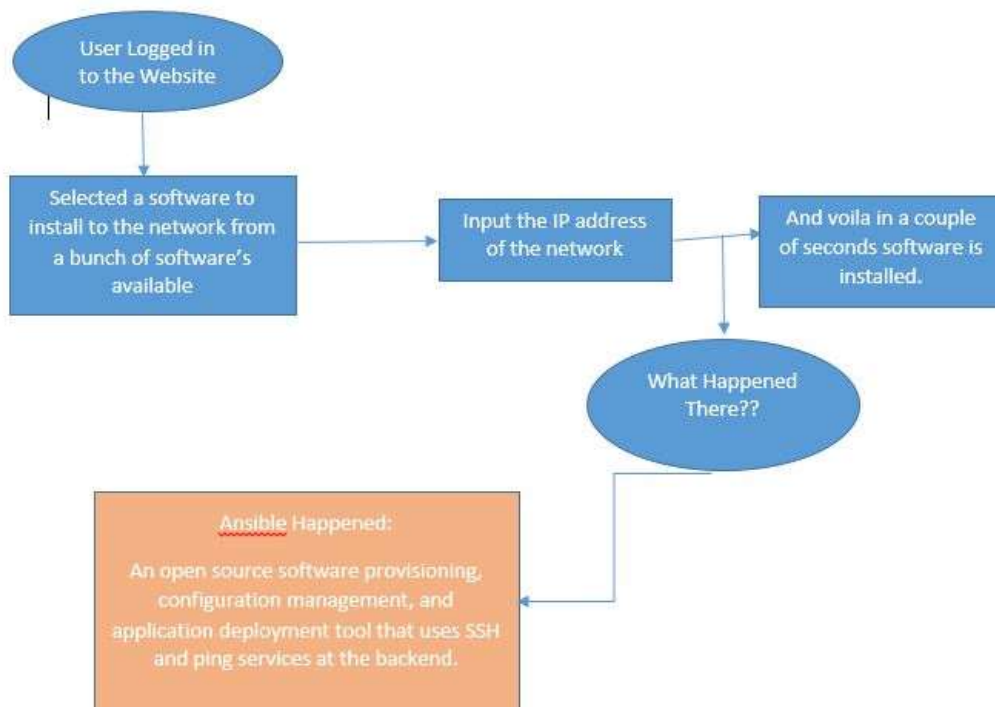
(1) Use Case Diagram



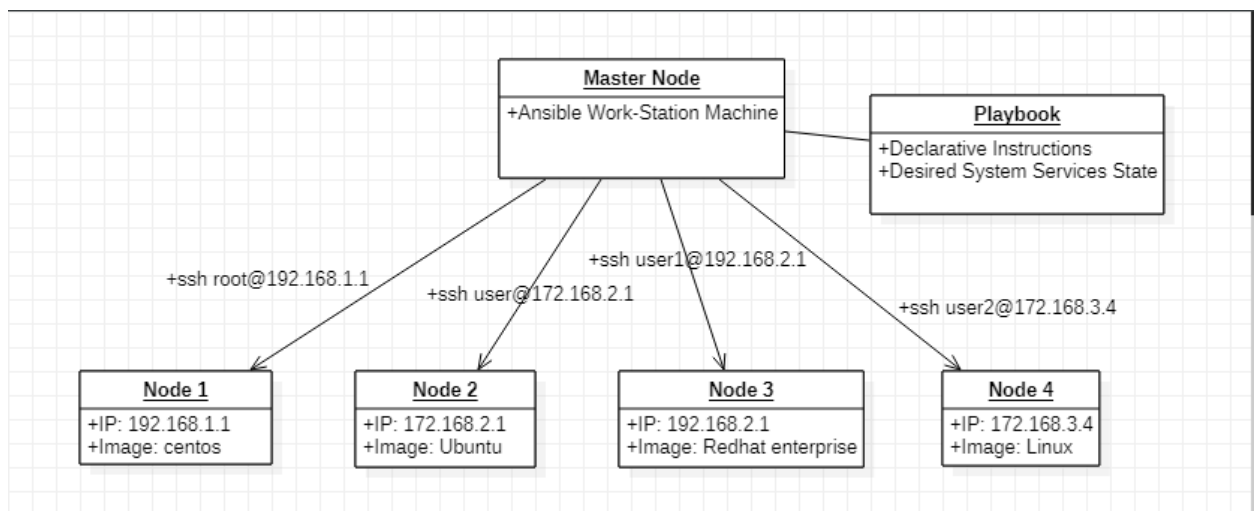
(2) Sequence Diagram



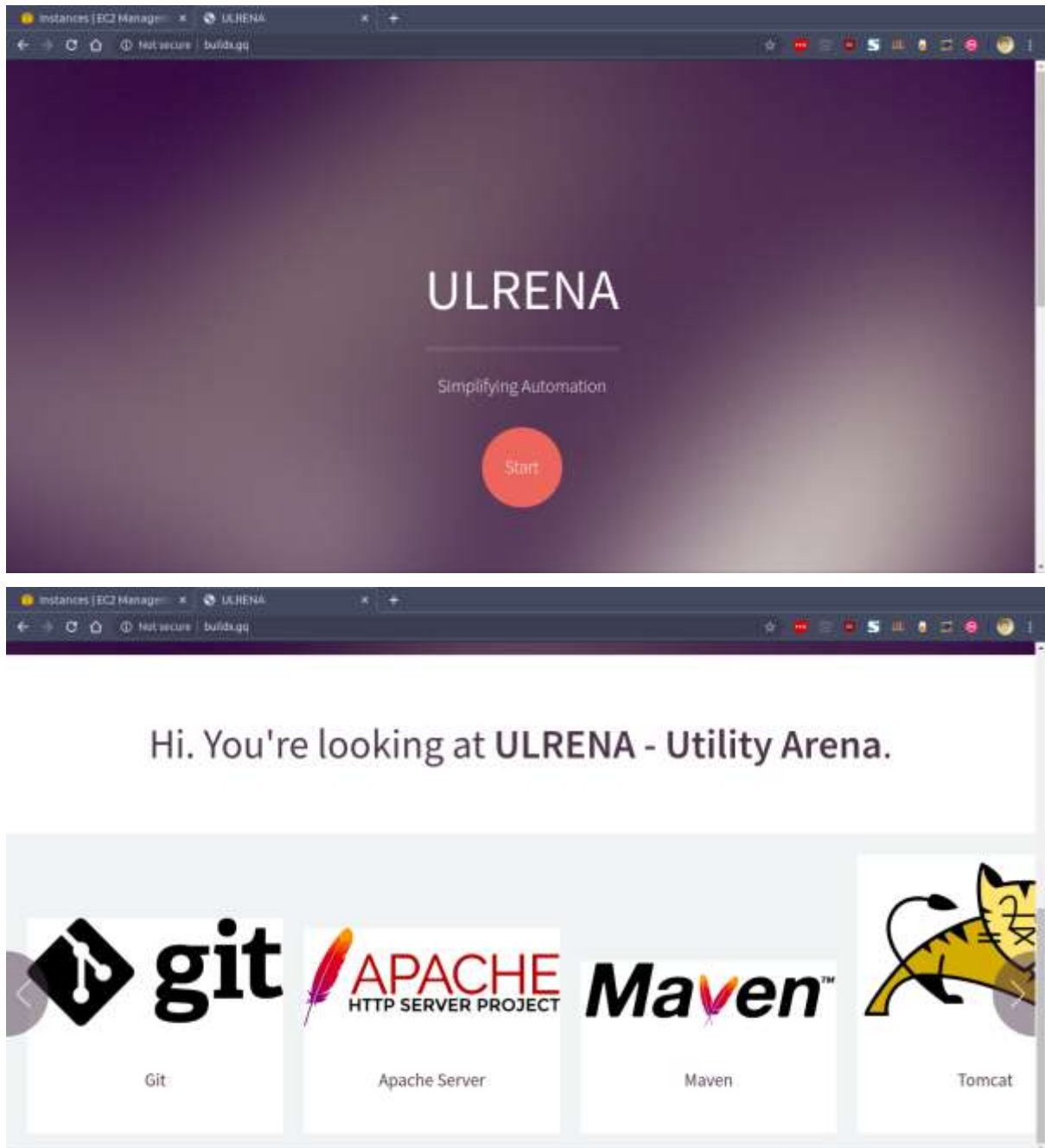
(3) Flow Chart

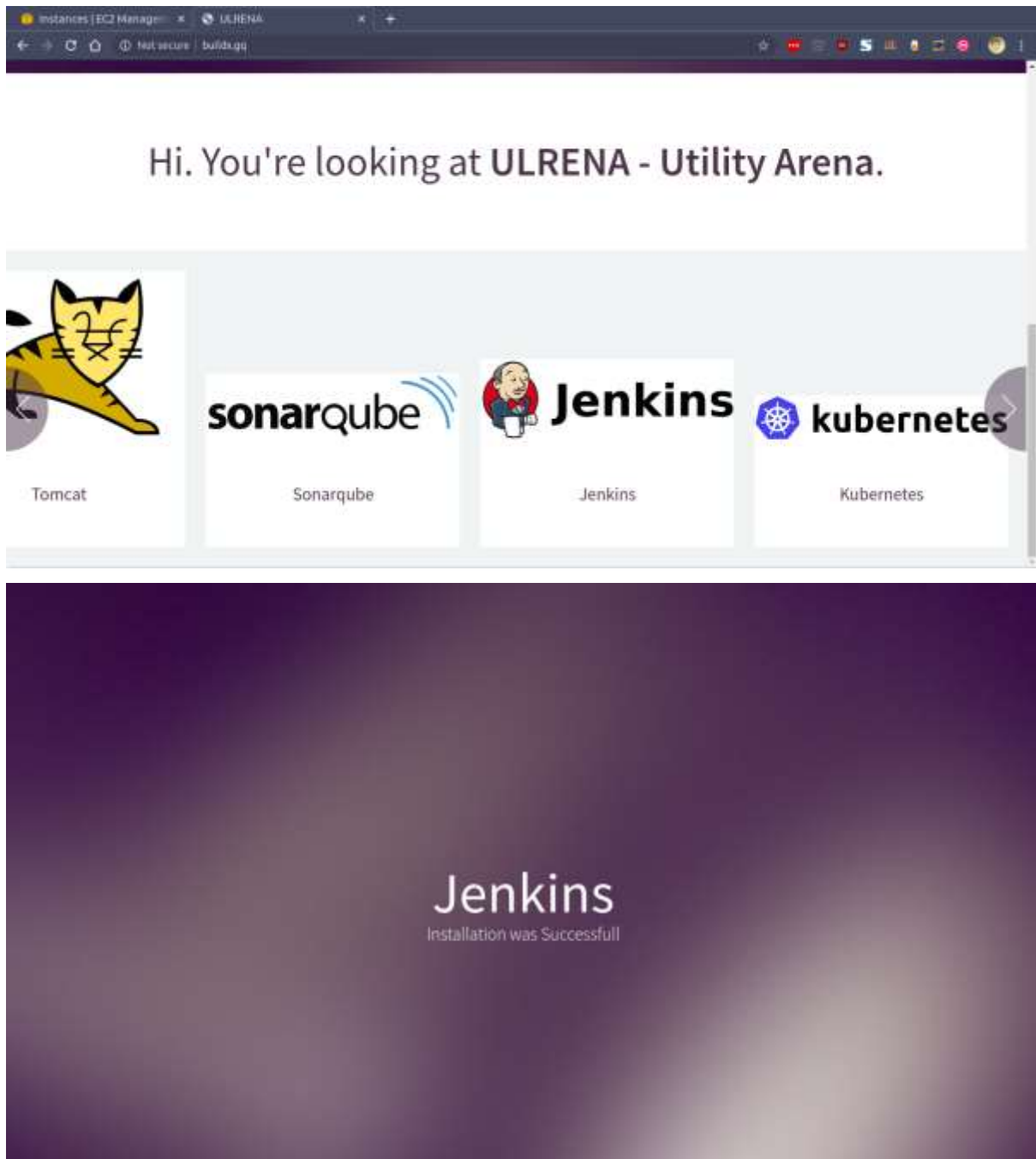


(4) Object Diagram (Ansible)



OUTPUT SCREENS





CONCLUSION

The people are working hard to bring some new and real systems to come up and understand the demands of people and helps them to adapt the working of the system of the new tech, This all leads for better advancements of systems and understand the best out of works by the help of new tech. The DevOps tools will help to execute the tasks very easily and adaptive working. It just overwrites the process at each level of working. This process deals with the installation of services and software's to n number of computer nodes and can help to work on all type of systems irrespective of operating system. We took the IP from the user and then transfers it from a website to the installation nodes and leads the process to carry out the tasks as desired. This all is very good in executing the tasks and finding the appropriate results as asked by the company. Our project "ULRENA", simplifies the working of various executions and ease the steps as done manually by a person. This also reduces the manpower and helps to complete the tasks easily. By concluding we all some up to say that adapting the new technology can be risky at various moments but by understanding the concept fully and then work from basics to high up according to their needs and which will help them to ease the tasks. The DevOps Automation is the future which needs to be adopted for better adoption of technology and ease for the users well as for companies.

REFERENCES

1. Singh, N.K., Thakur, S., Chaurasiya, H. and Nagdev, H., 2015, September. Automated provisioning of application in IAAS cloud using Ansible configuration management. In 2015 1st International Conference on Next Generation Computing Technologies (NGCT) (pp. 81-85). IEEE.
2. Chen, A., 2017. A Scheduling Framework For Large-Scale Based on Ansible.
3. Netto, H.V., Lung, L.C., Correia, M., Luiz, A.F. and de Souza, L.M.S., 2017. State machine replication in containers managed by Kubernetes. *Journal of Systems Architecture*, 73, pp.53-59.s
4. Vogel, P., Klooster, T., Andrikopoulos, V. and Lungu, M., 2017, September. A low-effort analytics platform for visualizing evolving Flask-based Python web services. In 2017 IEEE Working Conference on Software Visualization (VISSOFT) (pp. 109-113). IEEE.
5. Laukkarinen, T., Kuusinen, K. and Mikkonen, T., 2018. Regulated software meets DevOps. *Information and Software Technology*, 97, pp.176-178.
6. Pieplu, R., 2018. Ground Control Segment automated deployment and configuration with ANSIBLE and GIT. In *2018 SpaceOps Conference* (p. 2337).

Synopsis Draft verified by

Mrs. Amber Hayat
(Sr. Assistant Professor)

Dr. Neelu Ahuja
(Dept. of Systemics)