MAJOR-2 PROJECT

END TERM REPORT on

"Real-Time Log Visualization using DevOps"

Submitted by:

Name	Enrollment No.	Branch
Manik Khurana	R110216092	B.Tech CSE-CCVT
Babanjot Singh	R110216052	B.Tech CSE-CCVT
Ekanshu Dargan	R110216063	B.Tech CSE-CCVT
Chhavi Sharma	R110216055	B.Tech CSE-CCVT

Under the guidance of:

Mr. Harvinder Singh

Assistant Professor
Department of the Virtualization,
School of Computer Science,
University Of Petroleum and Energy Studies.
Dehradun- 248007

UPES

Approved By

(Mr. Harvinder Singh)   (Dr. Deepshikha Bhargava)

Project Guide   **Department Head**

1. Project Title

Real-time Log Visualization using DevOps.

2. Abstract

In this rapidly developing world, an ample amount of professionalism is required in every work environment. Therefore, software developers do not have enough time for everything that they are supposed to do. Earlier a simple task used to take weeks to be completed and now it happens in mere seconds. With this project, we aim to present a software solution with the assistance of the latest tools used in agile environments like DevOps and reduce the amount of time and effort to view the logs of any particular server. Crucial details of the server will be taken care of by the Elastic Stack collectively. The ELK stack will be up and running with the help of an ansible-playbook. This project will be done using only an ansible-playbook. This task, if not done this way, requires a lot of work to be done each and every time.

Keywords: Ansible, MetricBeats, Elasticsearch, Kibana, Git

3. Introduction

This project serves as an application - a future solution - to a time-costly problem of getting logs manually generated with the help of the outdated applications. Using our solution the developers will be able to save a lot of time and they can then focus on more important modules and grow at a greater rate. The solution is an ansible playbook on the remote host with ELK stack tools through which we will perform all the tasks. There are various DevOps tools through which we will create an interface from taking the logs and processing it, applying sorting on it, getting specific or desired outputs as per our requirements. Following are the tools that we will use

- **Ansible** — Ansible is an IT automation device. It can arrange systems, deploy software solutions, and coordinate further developed IT tasks, for example, consistent deployments or zero downtime moving updates.

  Ansible's principal objectives are straightforwardness and convenience. It likewise has a solid spotlight on security and dependability, including at least moving parts, use of OpenSSH for transport (with different transports and pull modes as options), and a language that is structured around audit ability by people even those curious about the program.

- **Metricbeat—** Metricbeat can be installed on the servers in our environment and used for monitoring their performance, as well as that of external services running on them. We can use Metricbeat to monitor and analyze system CPU usage, memory, and load. In Dockerized environments, Metricbeat can be installed on hosts for monitoring container performance metrics.

  Metricbeat is generally configured to ship the data directly to an Elasticsearch deployment.

- **Logstash** — Logstash is an open-source information collection device with constant pipelining abilities. Logstash can progressively bind together information from unique sources and standardize your preferred data into goals. Cleanse and democratize every one of your information for differing progressed downstream investigation and perception use cases.

  While Logstash initially drove development in log assortment, its capacities broaden well past that utilization case. Any kind of occasion can be advanced and changed with a broad array of input, channel, and output plugins, with numerous local codecs, further improving the ingestion procedure. Logstash quickens your bits of knowledge by bridging a more prominent volume and an assortment of data.

- **Elasticsearch** — Elasticsearch is the distributed search and analytics device at the core of the Elastic Stack. Logstash and Beats encourage gathering, totaling, and enhancing your data and putting it away in Elasticsearch. Kibana empowers you to intuitively investigate,

picture, and offer experiences into your information and oversee and screen the stack. Elasticsearch is the place the ordering, search, and analytics magic occurs.

Elasticsearch gives real-time search and analytics for a wide range of data. Regardless of whether you have structured or unstructured content, numerical information, or geospatial information, Elasticsearch can proficiently store and file it such that it supports quick hunts. You can go a long way past basic information recovery and total data to find patterns and examples in your data. Furthermore, as your data and query volume develops, the circulated idea of Elasticsearch empowers your deployment to become consistently directly alongside it.

- **Kibana** — Kibana is an open-source analytics and representation stage intended to work with Elasticsearch. You use Kibana to look, see, and collaborate with data put away in Elasticsearch indices. You can without much of a stretch perform propelled information investigation and imagine your information in an assortment of graphs, tables, and maps.

  Kibana makes it straightforward with huge volumes of information. It's basic, program based interface empowers you to rapidly make and offer powerful dashboards that show changes to Elasticsearch queries progressively.

4. Literature Review

F.M.A. Erich, C. Amrit & M. Daneva, University of Amsterdam [1], found main goals to achieve by implementing DevOps: Reduce lead-time, improve problem solving and improve feedback. Starting new projects took a very long time at the organization, as teams had problems obtaining development resources such as servers and software. During the project itself, it was hard to solve problems in which close collaboration between development and operation spersonnel was needed. By using DevOps, FinCom1 wanted to decrease the time required to solve these problems.

Pavel Masek, Martin Stusek, Jan Krejci, Krystof Zeman, Jiri Pokorny, and Marek Kudlacek Department of Telecommunications, Brno University of Technology, Brno, Czech Republic [2] in this paper, considered building up a new layer for the utilized Ansible orchestration tool. In the realized scenario, more than 10 labora-tories at Brno University of Technology were utilized to test our developed framework in case of remote management.

Michael Jade Mitra, School of Computing and Information Technologies Asia Pacific College Makati, David Paulo Sy School of Computing and Information Technologies Asia Pacific College Makati, Philippines [3] presented the real-world usability and best practice compliance of ELK Stack and how ELK was able to convert big companies to use their stack. Uncover the potential growth of development for ELK given the conversion of companies and possible contribution.

Mrs.Radhika T V, Assistant Professor, Department of Information Science & Engineering, Dayananda Sagar College of Engineering, Bangalore, India Dr. S.Sathish Kumar, Associate

Professor, Department of Computer Science & Engineering, RNS Institute of Technology, Bangalore, India   Krushna Chandra Gouda, Scientist CSIR Centre for Mathematical Modeling and Computer Simulation.(C-MMACS) Wind Tunnel Road Bangalore, India [4] in this paper have discussed different concepts of VPC with scenarios along with the need of VPC and its advantages. Also how effectively we can use VPC is also discussed.

5. Problem Statement

Through log investigation, we can gather and examine the data that is created by the resources. As the log information develops the activities group begins confronting issues to solidify and deal with the huge amount of data - the log records. The issue with log data is that the logs are unstructured that are created from different layers and produce various kinds of logs. The test lies in merging the log information, preparing it to produce Business and specialized bits of knowledge.

6. Objective

The primary objective of the project is to analyze and visualize log information progressively. ELK accomplishes this objective. It will bring together logging that will be helpful when wanting to find issues with the servers or applications and resolve them. The whole process is automated and can be done with a single click thereby saving a lot of workload and downtime every time an error is encountered.
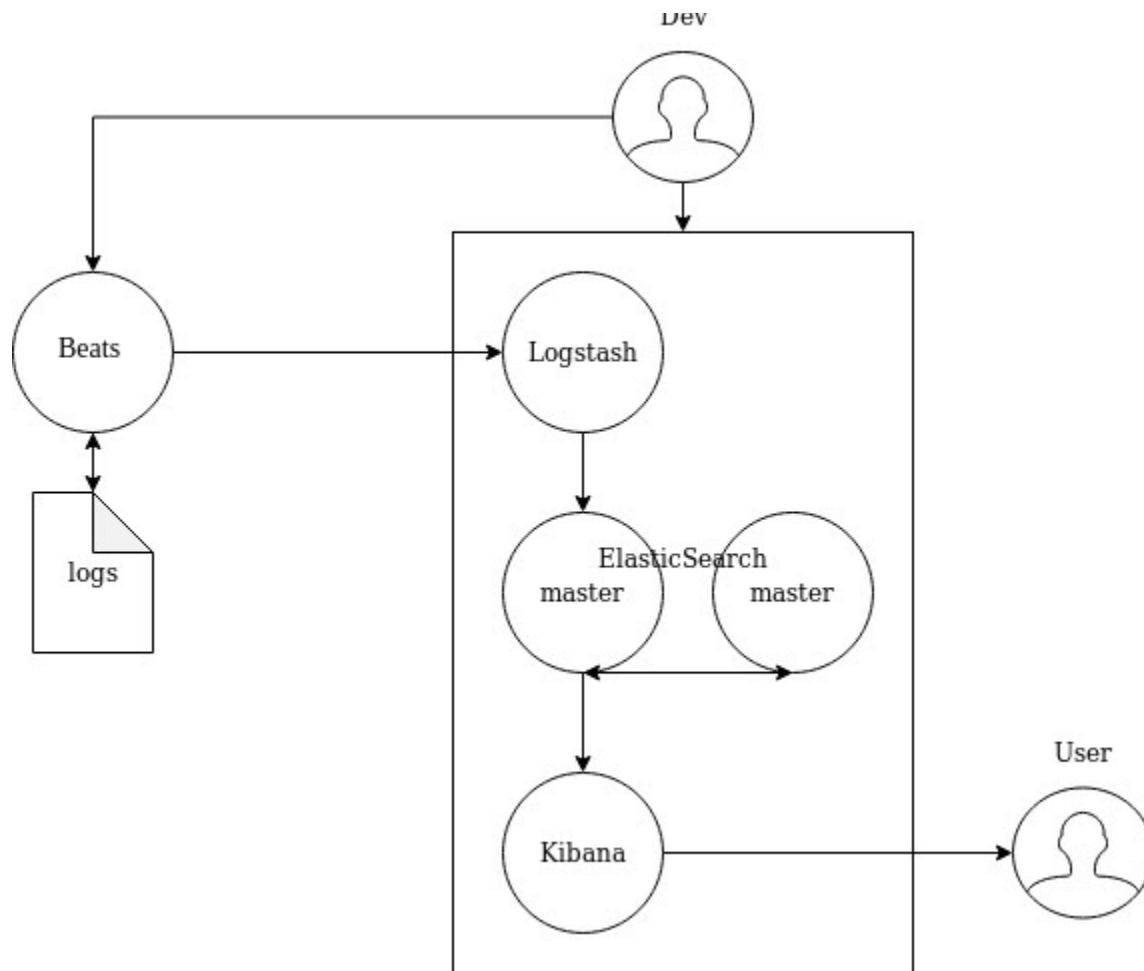
7. Methodology

Figure 1: Elastic Stack

Module 1:Setting up AWS Environment

Setting up an AWS environment for Ansible involves creating a custom VPC and further launching our AWS EC2 instances (Ansible Nodes) in it.
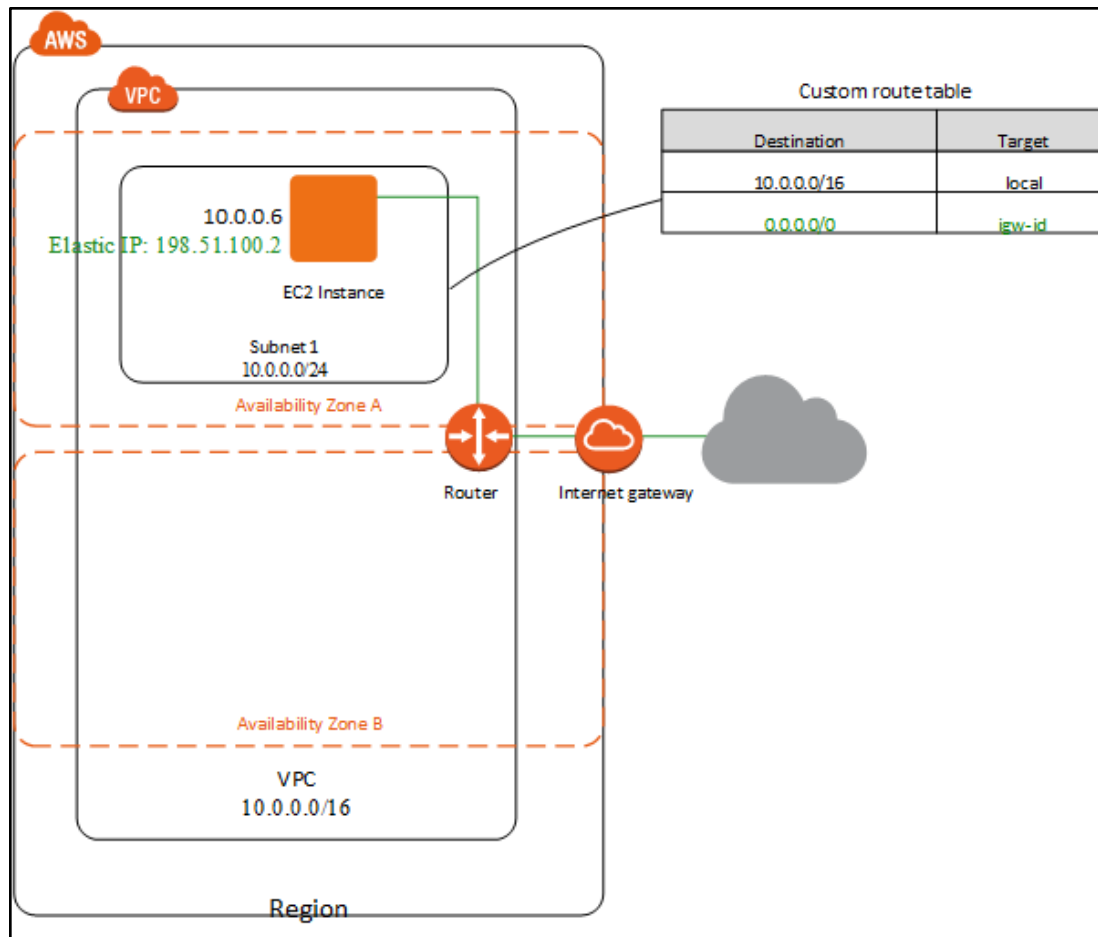
Figure 2: EC2 - AWS - VPC

The following are the steps involved-

1. Creation   of   VPC
   Amazon Virtual Private Cloud lets us arrange a sensibly secluded system or segment of the
   AWS Cloud. Here we have unlimited oversight over the virtual systems administration
   condition. This incorporates choosing our own IP address ranges, Creating subnets, and
   designing course tables and system doors. We can utilize IPv4 or IPv6 for secure and
   simple access to the resources.

   We can likewise modify the system setup of the VPC. We can make an open confronting
   subnet for the web servers that approach the web. What's more, simultaneously, we can
   likewise put the backend frameworks, for example, the databases or the application servers,
   in a private-confronting subnet with no web get to. We can utilize various layers of security,
   including security gatherings and systems get to control records (ACLs), to assist control
   with getting to Amazon EC2 occasions in each subnet.

   To create a new VPC :

   1.1.   Open   the   Amazon   VPC   console.

1.2. In the left navigation pane, choose Your VPCs and then click on Create VPC.

1.3. Here you must specify an IPv4 address run for your VPC. Determine the IPv4 address run as a Classless Inter-Domain Routing (CIDR) block; for instance, 10.0.0.0/16. You can't determine an IPv4 CIDR block bigger than/16. You can alternatively relate an IPv6 CIDR block with the VPC.

2. Creation   of   a   Subnet
A subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. AWS provides two types of subnetting one is Public which allows the internet to access the machine and another is private which is hidden from the internet. They are containers within your VPC that segment of a slice of the CIDR block you define in your VPC. Subnets allow you to give different access rules and place resources in different containers where those rules should apply.

To   create   a   new   subnet   in   the   VPC   created   :

2.1.   In   the   VPC   console,   on   the   left   navigation   pane   select   subnets.

2.2.   Next,   click   on   Create   Subnet.

2.3.   Here, give a name tag to your subnet and choose the VPC that you created.

2.4. Also, specify the Availability can be implemented by using the local database of the device   or   by   using   an   external   ACS   server.y   Zone   for   your   subnet.

2.5. Finally, specify your subnet's IP address block in CIDR format. For example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask and can be the   same   size   as   your   VPC.

Now, the subnet created is private by default. To make it public, we need to attach an Internet   Gateway.

3. Creation   of   an   Internet   Gateway
An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows the communication between our instances in VPC and the internet. It is basically a virtual router that connects a VPC to the internet. An internet gateway (IGW)   serves   two   purposes:
To provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.   It   supports   both   IPv4   and   IPv6   traffic.
To   create   an   Internet   Gateway:

3.1.   In   the   VPC   console,   on   the   left   navigation   pane   select   Internet   Gateway.

3.2.   Next,   click   on   Create   Internet   Gateway.

3.3.   Here,   specify   the   name   for   the   gateway   and   click   on   create.

Once the Internet Gateway is created you need to attach it to your VPC.

3.4. Now, select your Internet Gateway from the list and click on the Actions button on top.

3.5.   Now   click   on   the   Attach   VPC   option.

3.6 Next, you'll be asked to enter the VPC. Choose your VPC from the list of available options.

4. Creation   of   a   Custom   Route   Table
When you create a subnet, it is automatically associated with the main route table for the VPC. By default, the main route table doesn't contain a route to an internet gateway. The following procedure creates a custom route table with a route that sends traffic destined outside the VPC to the internet gateway and then associates it with your subnet.

4.1.   Open   the   Amazon   VPC   console

4.2. In the navigation pane, choose Route Tables, and then choose Create Route Table.

4.3. In the Create Route Table dialog box, optionally name your route table, then select your   VPC,   and   then   choose   Yes,   Create.

4.4. Select the custom route table that you just created. The details pane displays tabs for working   with   its   routes,   associations,   and   route   propagation.

4.5. On the Routes tab, choose Edit, Add another route, and add the following routes as necessary. Choose Save when you're done.

For IPv4 traffic, specify 0.0.0.0/0 in the Destination box, and select the internet gateway ID in the Target list.

For IPv6 traffic, specify::/0 in the Destination box, and select the internet gateway ID in the Target list.

4.6. On the Subnet Associations tab, click on Edit, select the Associate check box for the subnet, and then click on Save.

5. Creation   of   a   Security   Group

By default, a VPC security group allows all outbound traffic. You can create a new security group and add rules that allow inbound traffic from the internet. You can then associate the security   group   with   instances   in   the   public   subnet.

To create a security group:

5.1. Open the Amazon VPC console

5.2. In the navigation pane, choose Security Groups, and then choose to Create Security Group.

5.3. In the Create Security Group dialog box, specify a name for the security group and a description. Select the ID of your VPC from the VPC list, and then choose Yes, Create.

5.4. On the Inbound Rules tab, choose Edit. Choose Add Rule, and complete the required information. For example, select HTTP or HTTPS from the Type list, and enter the Source as 0.0.0.0/0 for IPv4 traffic. Here, you need to select SSH from the Type list (for Ansible). Choose Save when you're done.

Security Group is created. Now while creating an EC2 instance attach the security group to it.

6. Creation of an EC2 Instance

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that gives a secure, resizable compute capacity in the cloud. It is intended to make web-scale distributed computing simpler for designers. Amazon EC2's straightforward web administration interface permits us to acquire and design limits with insignificant contact. It gives us full oversight of our processing assets and lets us run on Amazon's demonstrated registering condition.

To launch an EC2 Instance :

6.1. Open the Amazon EC2 console after logging in

6.2. From the console dashboard, click on Launch Instance.

6.3. The "Choose an Amazon Machine Image (AMI)" page displays a list of basic configurations, called "Amazon Machine Images (AMIs)", that serve as layouts for your instance.

6.4. On the Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default.

6.5. Configure Instance Details. In this step choose the VPC and Subnet you just created. Check the Auto Assign Public IPs option.

6.6. Add Storage. The instance has been automatically provisioned a General Purpose SSD root volume of 8GB. You can add more storage if you want.

6.7. Tag Instance. You can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are a lot of instances.

6.8. Configure Security Groups. Here select the security group you've just created.

6.9. Next, Review Instances and launch. Now you'll be prompted to create a key pair to log in to your instance. A key pair is a set of public-private keys. Create a key-pair and download the private key.

Your instance is now successfully created.

Module 2: Configuring Elasticsearch

For this, we need to create an ansible role for Elasticsearch. The Elastic search role needs to be configured such that it sets up the Elasticsearch apt repo, installs Elasticsearch, and some other 5601configuration tasks. The tasks included in the YAML file are :

1. Add Elasticsearch apt-key
   Import the GPG key for elasticsearch packages.

2. Add the Elasticsearch apt repo
   "The Elasticsearch official team provides an apt repository to install Elasticsearch on the Ubuntu Linux system. Then configure the apt repository on your system."

3. Installing Elasticsearch
   After adding the repository to your system install elasticsearch.

4. Update Elasticsearch config file to allow access (to secure Elasticsearch, bind to 'localhost').
   The Elasticsearch is now installed on your PC. You can customize this by editing the "Elasticsearch configuration file", elasticsearch.yml. Use the lineinfile module for this. In the file set the "network host" to "0.0.0.0 to listen on all interfaces and make it publically available." We can also use our "LAN address for LAN access only."

5. Update Elasticsearch port in the config file
   Also in the elasticsearch.yml file change the HTTP port to 9200. Use the lineinfile module for this.

6. Start Elasticsearch
   Next, start elasticsearch service using the service module.

Module 3: Configuring Kibana

For this, we need to create an ansible role for Kibana. The Kibana role installs and configures Kibana. The tasks included in the YAML file are :

1. Install   Kibana
   Install the Kibana package with apt.

2. Updating the config file to allow outside access
   The configuration file here is the kibana.yml file in the /etc/kibana directory. Here 'server.host' is set to '0.0.0.0'.

3. Define   server   port
   Again in the same configuration file (/etc/kibana/kibana.yml) set the 'server.port' to '5601'.

4. Defining   Elasticsearch   URL
   In the same configuration file (/etc/kibana/kibana.yml) specify Elasticsearch server to connect to the elasticsearch URL :
   elasticsearch.url: "http://localhost:9200"

5. Starting   Kibana
   Now start kibana service using the service module.

Module 4: Setting up Metricbeat

Now we need to create a role for metricbeat. The metricbeat role will install and start it with the default settings. The tasks included in the yaml file are :

1. Installing   Metricbeat
   Install Metricbeat package with apt.

2. Starting   Metricbeat.
   Now start Metricbeat service using the service module.

Module 5: ANSIBLE Environment and ssh keys

Ansible lets us automate the advent, configuration, and management of machines. Instead of manually retaining servers up to date, making configurations, transferring documents, and so on., you could use Ansible to automate this for groups of servers from one manipulating system.

For development and testing purposes, but, you might locate yourself by putting it in the stack repeatedly. While the installation method is straightforward and should take you no extra than 5 mins, a one-liner solution for installing and configuring the numerous components might be even higher.

It will allow us to configure by allowing users to write the scripts in YAML files with fixed or unchanged execution. The script will perform tasks and set up servers with direct use of any ssh service with each other.

- **Playbooks** — Execution entry point of ansible scripts

  Setting up Ansible s might be even higher.

  It will allow us to configure by allowing users to write the scripts in YAML files with fixed or unchanged execution. The script will perform tasks and set up servers with direct use of any ssh service with each other.

- **How to install Ansible into your system**

  sudo apt update

  sudo apt install software-properties-common

  sudo apt-add-repository --yes --update ppa:ansible/ansible

  sudo apt install ansible

Ensure our SSH keys are set up

Try to ssh into our target server using ssh **<user>@<host_ip>** to check if ssh-agent has got our keys first. This is because ansible will try to SSH into the target servers specified in the ./hosts file with the keys ssh-agent has.

Our ssh-user should have sudo privileges to allow creations of user and group ids that we will be using in the images we are going to build.

Module 6: Deploying the ELK Stack and Metricbeat using Ansible

Our final playbook will look something like this :

Now to deploy the Stack, execute the playbook on the machine hosting Ansible using the following command :

sudo ansible-playbook major.yml
The major.yml playbook includes the invocation of all the roles that we have created till now.

Now ansible established a connection to the target host, and begins to execute the various roles and tasks.

Module 7: Check that all services are up and running

In this module we will check if Elasticsearch and Kibana are up and running. For this we will curl their URLs and see if we get the desired pages as the result.

For Elasticsearch :

curl http://<<ip>>:portnumber
For Kibana:

curl http://<<ip>>:portnumber

Module 8 : Setting up our index patterns and browsing logs

Lets test run our log shipper deployed in our Step 6 previously.

Now hit our kibana URL with <kibana_ip>:<port>.

1. You should see the "Kibana Welcome page". Now click the "Explore my Own" button.

2. After this, you should be taken to the "Kibana Home Page."

3. Now in order to visualize and explore data in kibana, we need to create an index pattern to retrieve data from Elasticsearch. Now click the "Discover" tab on the left side of the page. Choose "Create an index pattern".

4. Then value the index pattern as "metricbeat-*".

5. Click next and choose '@timestamp' and click "Create index pattern". The "Time Filter" will use this field to filter the data by time.

6. Index pattern should get created.

7. Now navigate to Management/Kibana/Index patterns. Here you can see that the "page lists every field" in the metricbeat-* index and the "field associated core type as recorded by elasticsearch."

8. Click the "Discover" menu to check the server logs. Here you can see the time and the source of the logs created.

9. The logs will be displayed as per the time stamp. Click on the timestamp of any log to expand it and check the log file contents and all its details.

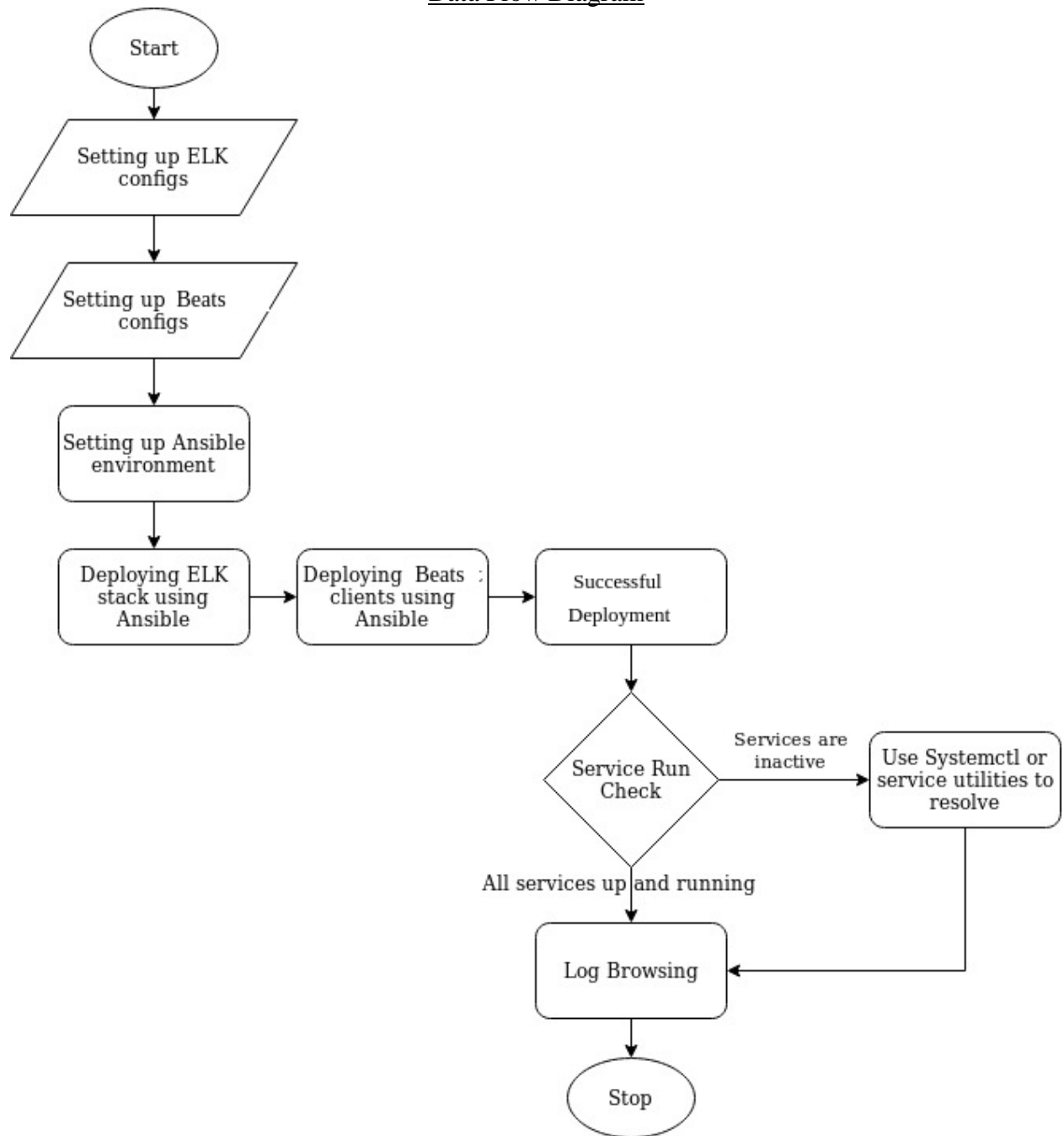# 8. Design and Flowchart

Data Flow Diagram

Figure 3: Dataflow diagram

9. System Requirements (Software/Hardware)

- **Hardware:**
  - Intel(R) Core(TM) i3-6200 CPU @ 1.5 GHz
  - Minimum 2 GB RAM
  - The system type is a 32/64-bit Operating System
- **Software:**
  - Linux based operating system - Ubuntu
  - Ansible

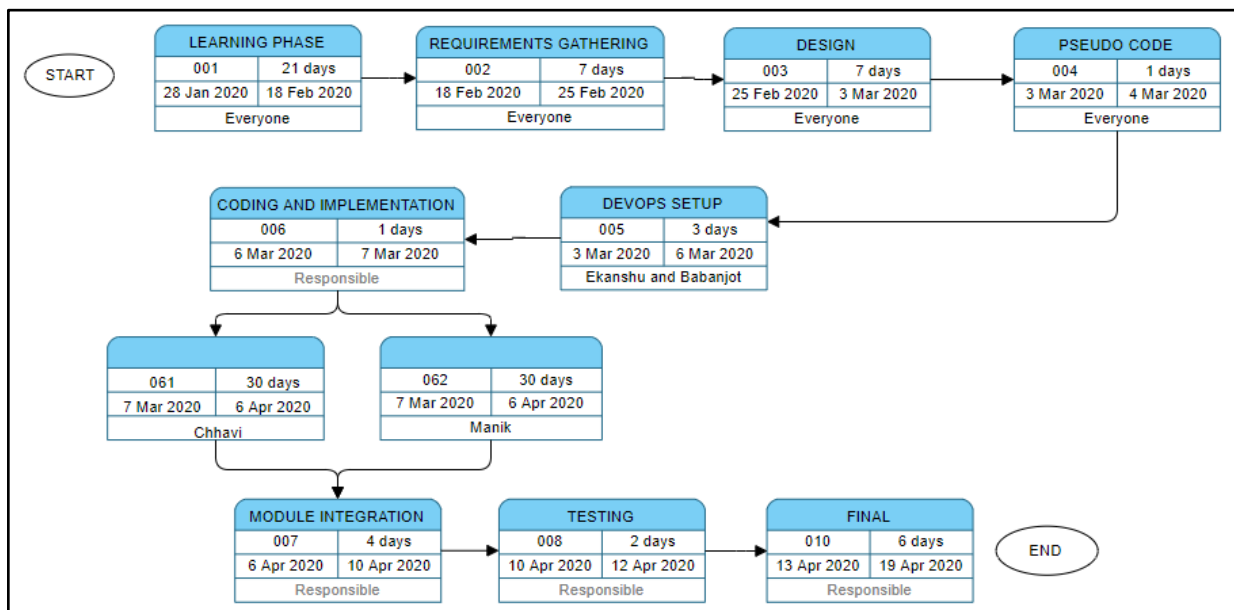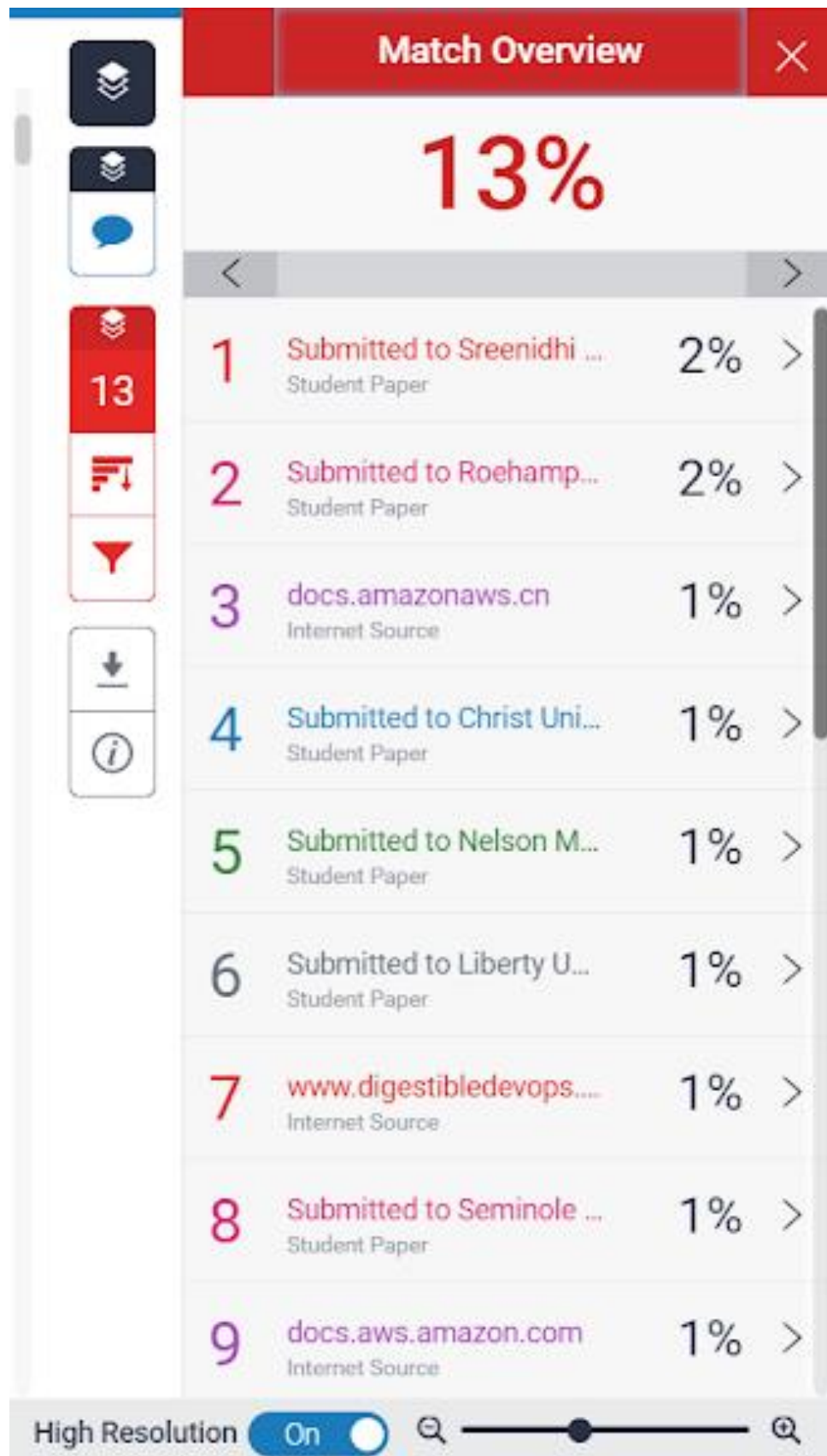## 10. Schedule (Project Evaluation and Review Technique Chart)

Figure 4: PERT Chart

11. References

[1] F.M.A. Erich, C. Amrit & M. Daneva, "A Qualitative Study of DevOps Usage in Practice" *University of Amsterdam, 2017.*

[2] Pavel Masek, Martin Stusek, Jan Krejci, Krystof Zeman, Jiri Pokorny, and Marek Kudlacek "Unleashing Full Potential of Ansible Framework", *1Department of Telecommunications, Brno University of Technology, Brno, Czech Republic, 2018.*

[3] Michael Jade Mitra" The Rise of Elastic Stack " *David Paulo Sy School of Computing and Information Technologies Asia Pacific College Makati, Philippines 2015.*

[4]Mrs.Radhika T V, Dr. S.Sathish Kumar, Krushna Chandra Gouda, "A study on the different aspects of the Virtual Private Cloud" *Bangalore 2015.*

[5].https://docs.ansible.com/

[6].https://www.elastic.co/guide

[7].https://docs.aws.amazon.com

[8]. https://guides.github.com

Report: Major 2 Inbox ×

MANIK KHURANA
Dear Sir,Greetings for the day! Kindly find the Major 2 End Term Report attached below. Please ap|

MANIK KHURANA
Dear Sir, The revised report is attached below.

Harvinder Singh
to me, Babanjot, Ekanshu, Chhavi ▾

Ok..Approved..

--
With Regards

Harvinder Singh

Assistant Professor (S.S) | Department of Virtualization | School of Computer Science
University of Petroleum & Energy Studies (UPES)
Energy Acres Building, PO Bidholi, via Prem-Nagar
Dehradun- 248007 Uttarakhand, India
E-mail: hsingh@upes.ac.in
E-mail: harvinderjabbal@gmail.com
LinkedIn Profile: https://www.linkedin.com/in/harvinder-jabbal-38104816/
Mobile: (+91) 9896980070

UPES