***Elasticsearch - Kibana for Kubernetes Integration***

***Prerequisites:***

1. Ubuntu Machine with a minimum configuration of 2 vCPUs and 8 GB RAM.
2. A Kubernetes Cluster

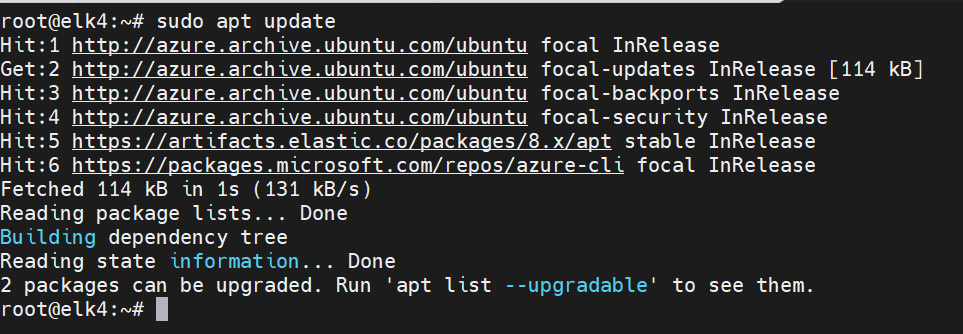
***Observations:***

1. Installed ELK 7.x and integrated with Jaeger for all necessary traces from the DevOps tenant.
2. In 7.x, Kubernetes couldn't be configured with ELK because of Security enhancements and license restrictions.
3. These enhancements were included in the 8.x version, and we were able to integrate Kubernetes with ELK using version 8.8.
4. Jaeger couldn't be integrated with the 8.x version due to security and SSL certificate requirements.
5. We were able to create Alerting rules but most of the connectors there require a gold license to configure.

***ELK Installation:***

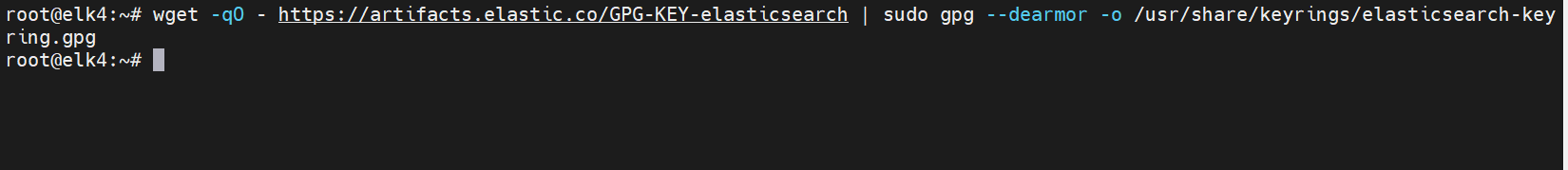
1. **Update System Packages:**

sudo apt update



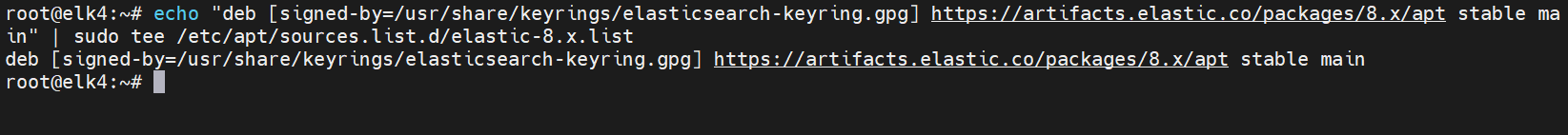
1. **Import Elastic GPG key:**

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg



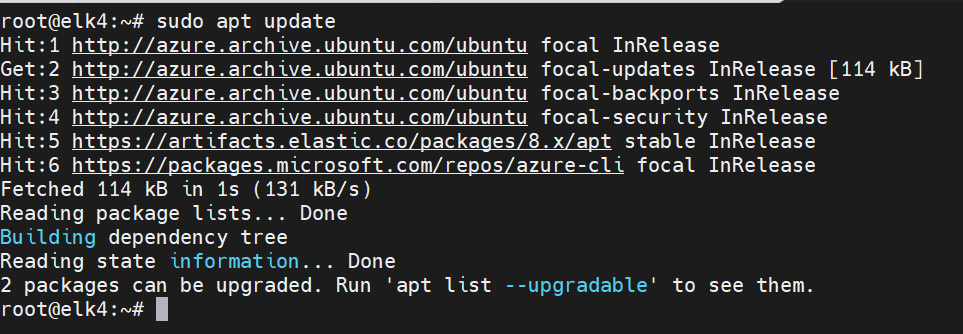
1. **Add Elastic APT repository:**

echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list



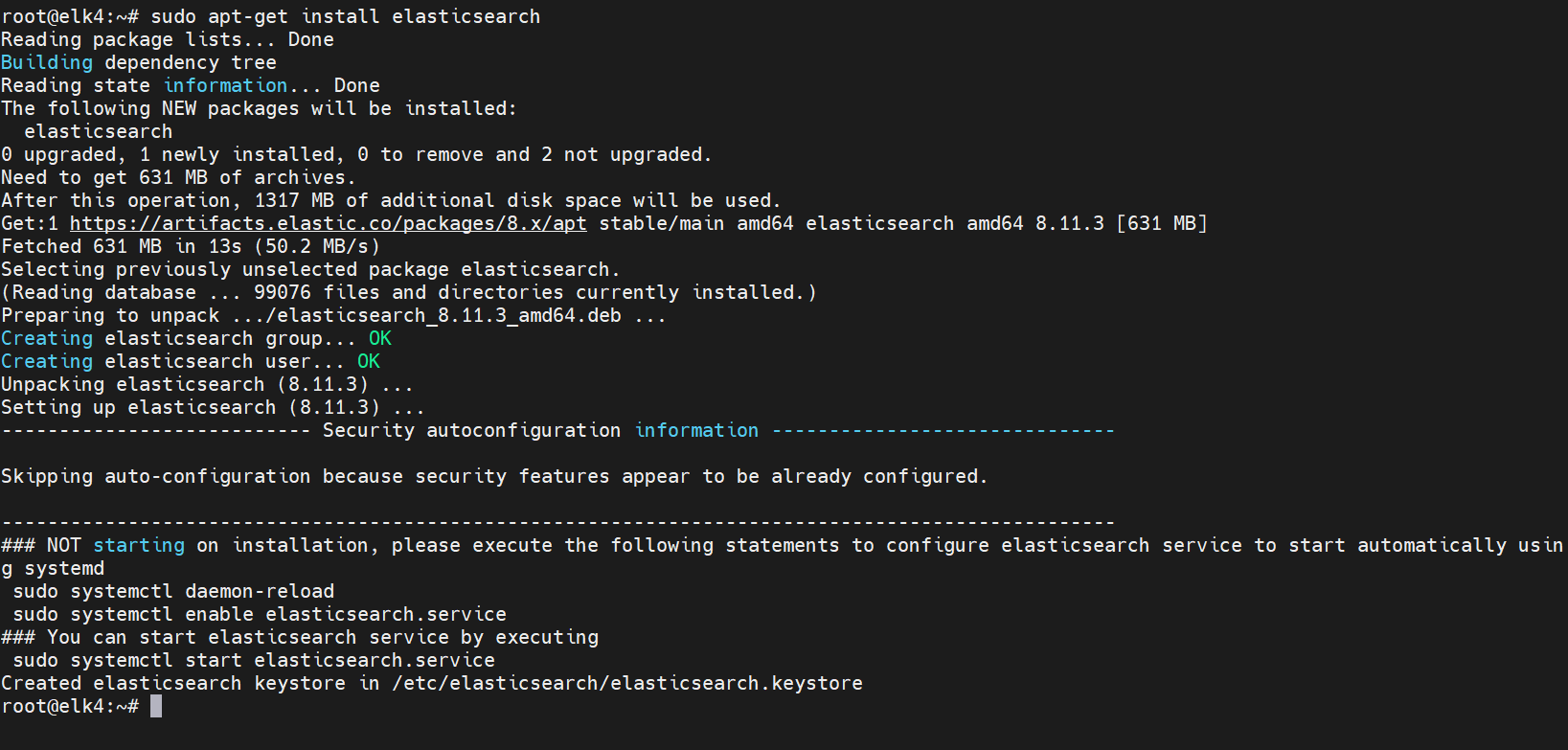
1. **Update APT repositories:**

sudo apt-get update



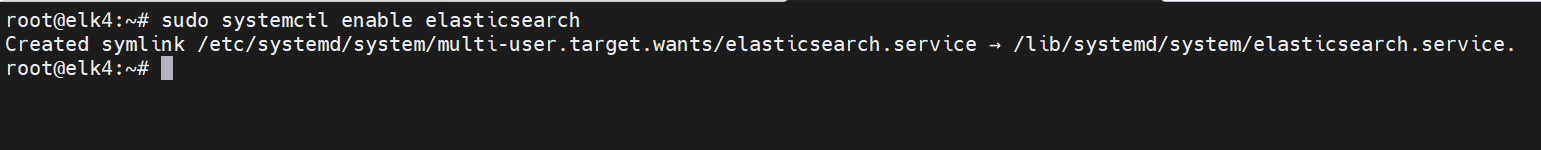
1. **Install Elasticsearch:**

sudo apt-get install elasticsearch



1. **Enable Elasticsearch service:**

sudo systemctl enable elasticsearch



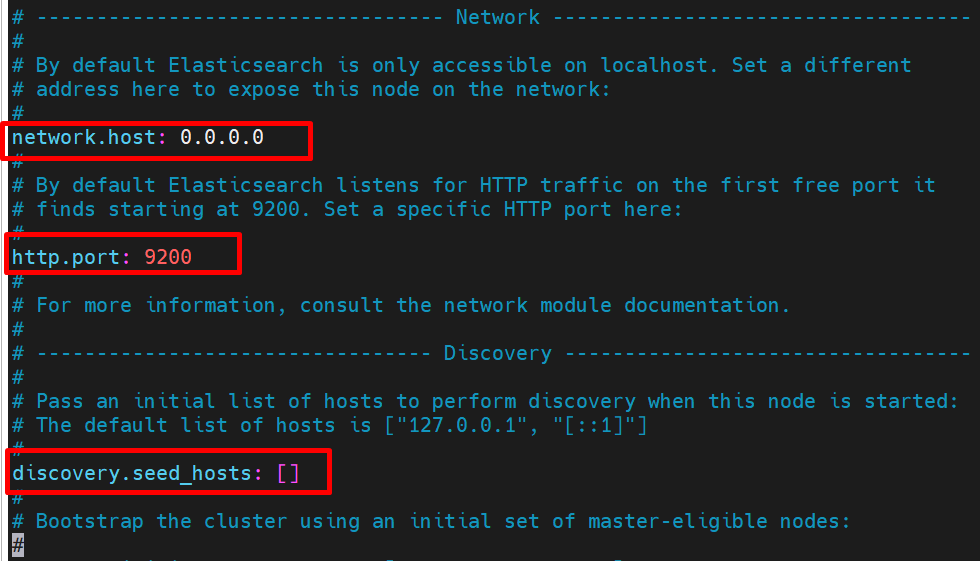
1. **Edit Elasticsearch configuration:**

vi /etc/elasticsearch/elasticsearch.yml

network.host: 0.0.0.0

http.port: 9200

discovery.seed\_hosts: [ ]



***network.host: 0.0.0.0***

This setting in Elasticsearch's configuration file determines the network interface on which Elasticsearch is accessible. Setting it to 0.0.0.0 means that Elasticsearch will bind and listen for incoming requests on all network interfaces available on the machine. It allows Elasticsearch to be accessible from any network interface available on the host. This is a broad setting and should be used carefully in production environments due to potential security implications.

***http.port: 9200***

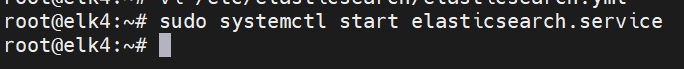
This sets the port on which Elasticsearch's HTTP service listens for incoming requests. By default, Elasticsearch uses port 9200 for handling RESTful API requests. This configuration parameter sets the HTTP port to 9200, allowing communication with Elasticsearch via HTTP.

***discovery.seed\_hosts: [ ]***

This setting is related to the initial discovery of other nodes in a cluster. In a clustered environment where multiple Elasticsearch nodes exist, this parameter specifies a list of hostnames or IP addresses that the node should contact to join the cluster or discover other nodes. The empty brackets '[]' indicate that no initial seed hosts are specified for discovery. In a production cluster, you would typically specify the addresses of other nodes in the cluster.

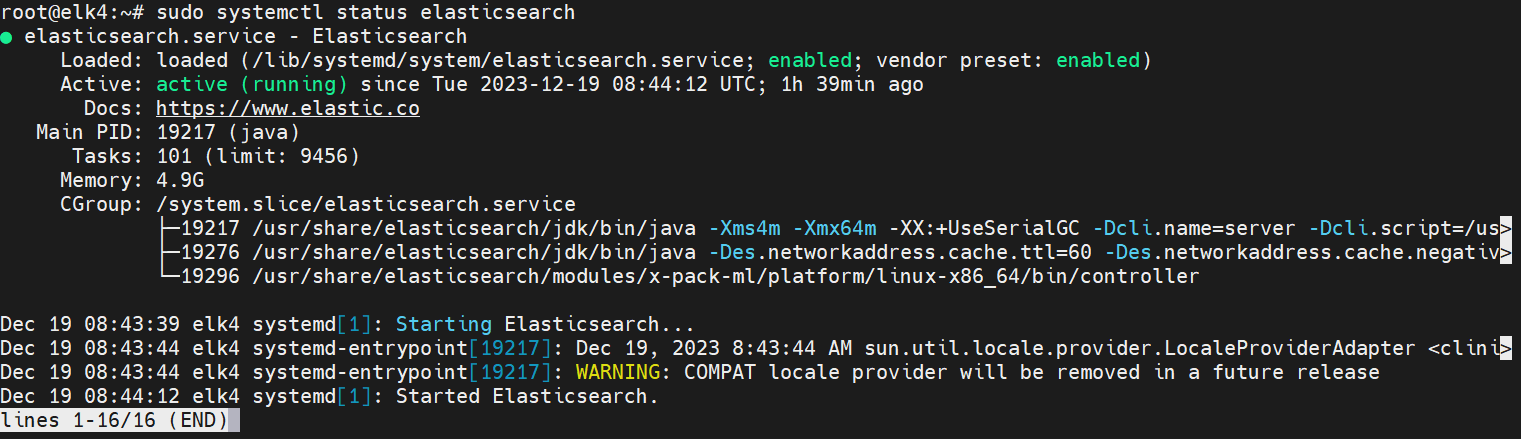
1. **Start Elasticsearch service:**

sudo systemctl start elasticsearch



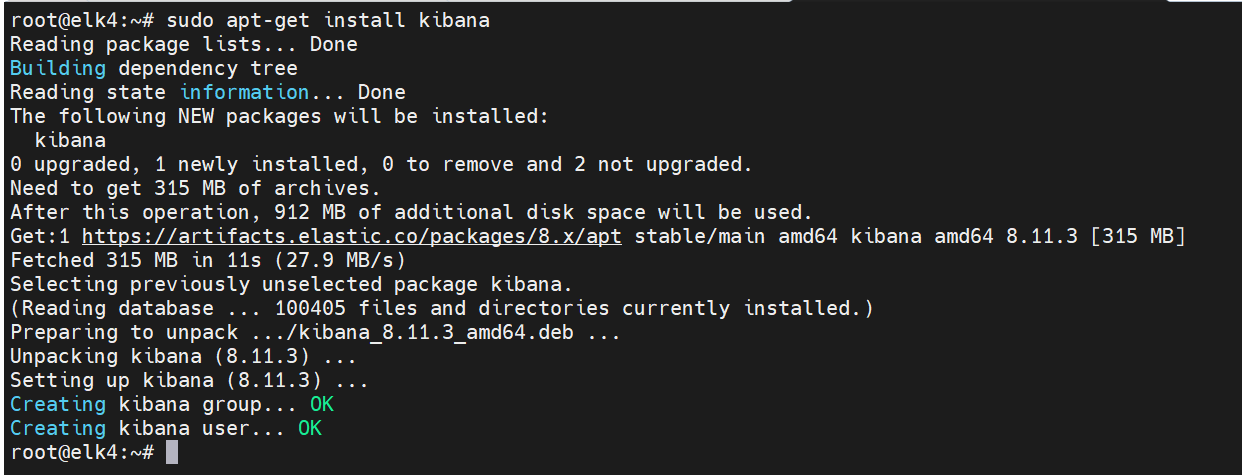
1. **Check Elasticsearch service status:**

sudo systemctl status elasticsearch



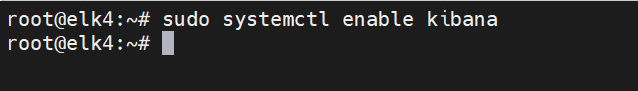
1. **Install Kibana:**

sudo apt-get install kibana



1. **Enable Kibana service:**

sudo systemctl enable kibana

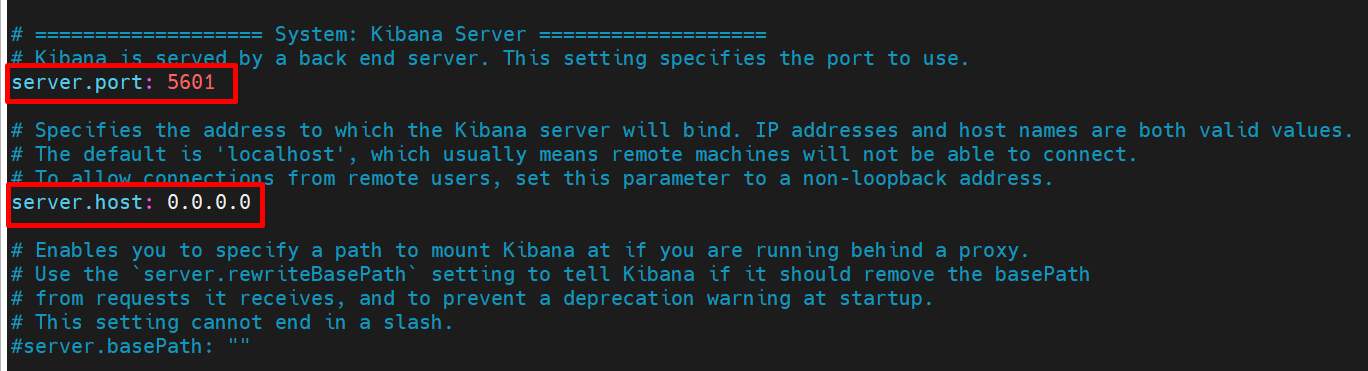


1. **Edit Kibana configuration:**

vi /etc/kibana/kibana.yml

server.port: 5601

server.host: 0.0.0.0



***server.port: 5601***

This setting in Kibana's configuration file defines the port on which the Kibana server will listen for incoming connections. By default, Kibana uses port 5601 for its web interface. This configuration parameter sets the server's port to 5601, allowing users to access Kibana's web interface through a web browser via this port.

***server.host: 0.0.0.0***

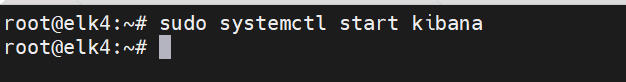
Similar to the `network.host` setting in Elasticsearch, this setting in Kibana's configuration determines the network interface on which Kibana is accessible. Setting it to 0.0.0.0 means that Kibana will bind and listen for incoming requests on all available network interfaces on the machine. This means Kibana will be accessible from any network interface available on the host. As with Elasticsearch, this is a broad setting and should be used carefully in production environments due to potential security implications.

***elasticsearch.hosts: ['https://<ElasticsearchServerIP:9200']***

This setting specifies the Elasticsearch instance that Kibana should connect to. It defines the endpoint or address of the Elasticsearch cluster that Kibana will use for data retrieval and visualization. In this case, it's configured to connect to an Elasticsearch node with the address `172.16.0.4` and using the HTTPS protocol on port 9200. This setting allows Kibana to communicate with the specified Elasticsearch instance.

1. **Start Kibana service:**

sudo systemctl start kibana

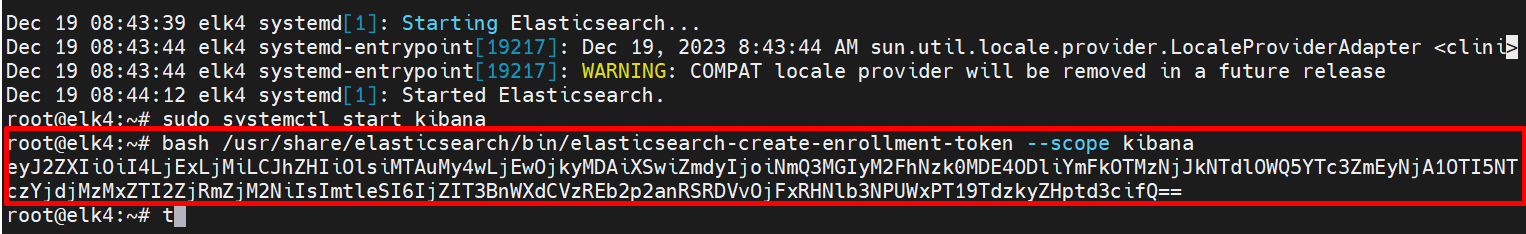


***Elastic User Configuration:***

1. **Generate Elasticsearch Enrolment Token for Kibana:**

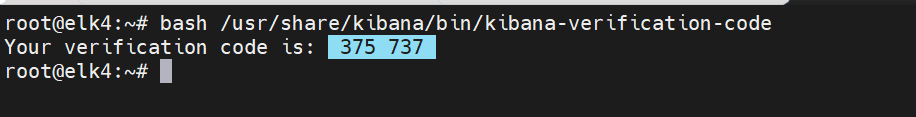
bash /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token --scope kibana

This command creates an enrolment token specifically for Kibana to enrol with Elasticsearch securely. It allows Kibana to authenticate itself to Elasticsearch and gain the necessary permissions or access rights.



1. **Generate Kibana Verification Code:**

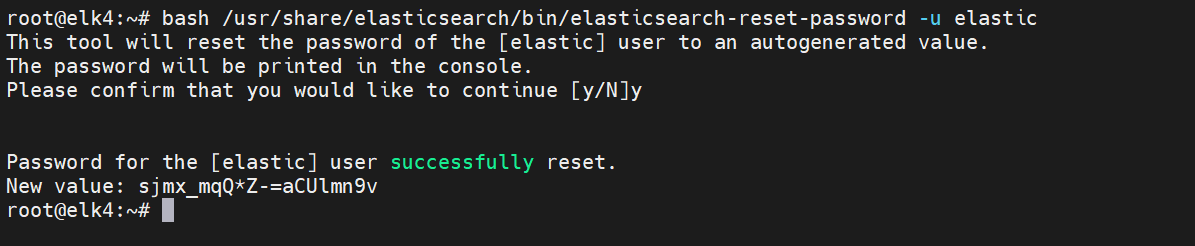
bash /usr/share/kibana/bin/kibana-verification-code



This command is used to generate a verification code required during the setup or authentication process. The verification code is likely used for the initial setup or verification of Kibana with certain settings, possibly during the authentication process with Elasticsearch or other components.

1. **Reset Elasticsearch Password for 'elastic' user:**

bash /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic



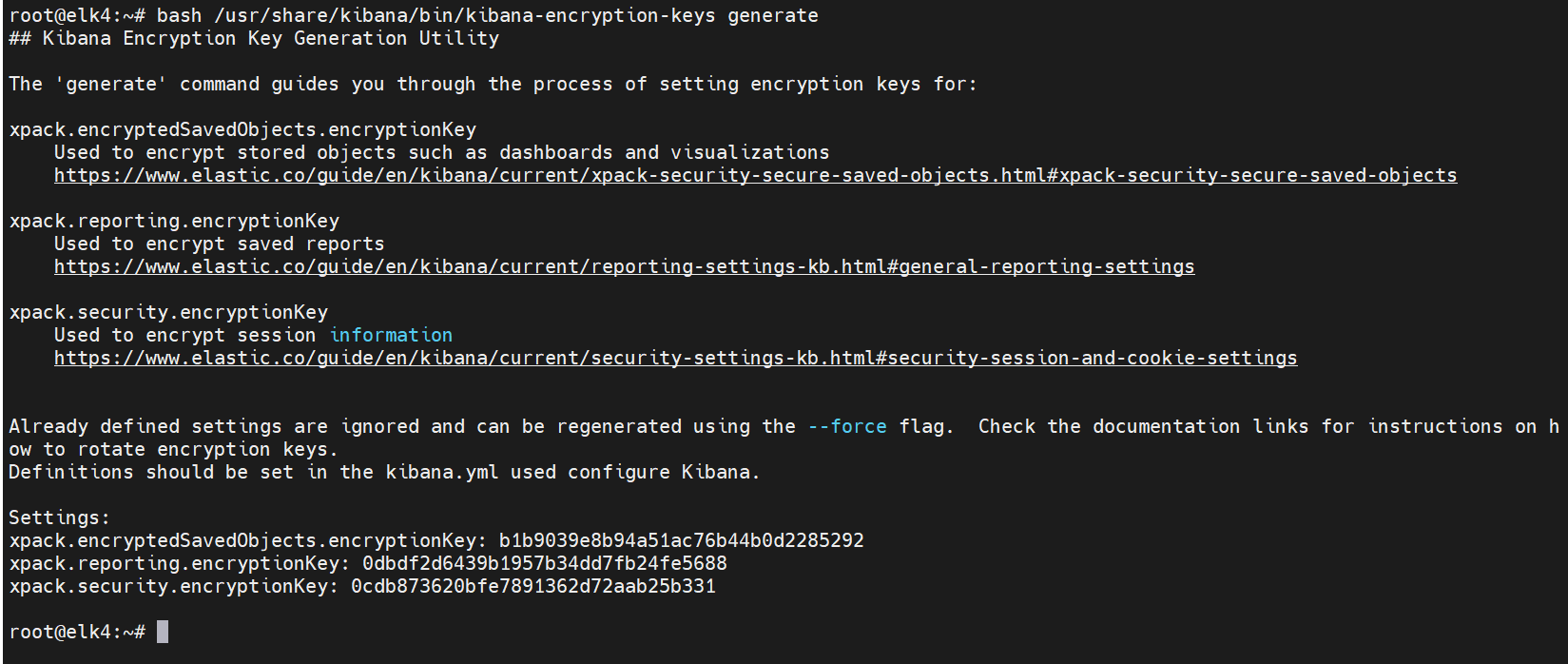
The command is used to reset the password for the built-in user 'elastic' in Elasticsearch. This user is usually the default superuser/administrator for the Elasticsearch instance. After executing this command, a new password will be set for the 'elastic' user.

1. **Copy the password.**

***Alerting Configuration:***

1. **Generate Kibana Encryption Keys:**

bash /usr/share/kibana/bin/kibana-encryption-keys generate



The command generates the necessary encryption keys used for securing sensitive data within Kibana, such as saved objects, reporting, and security-related data. These keys are crucial for encryption and decryption purposes to protect data.

1. **Edit Kibana configuration to add encryption keys:**

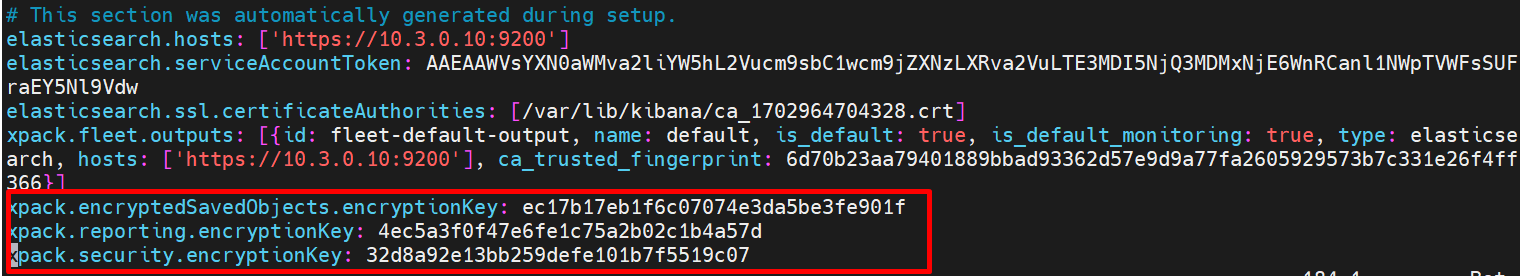
vi /etc/kibana/kibana.yml

xpack.encryptedSavedObjects.encryptionKey: your\_encryption\_key

xpack.reporting.encryptionKey: your\_encryption\_key

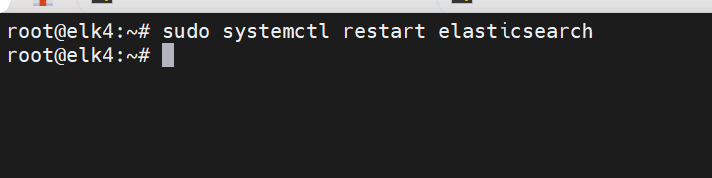
xpack.security.encryptionKey: your\_encryption\_key

These settings are configured to specify the encryption keys generated in the previous step. These keys are crucial for securing different aspects of Kibana's functionalities.



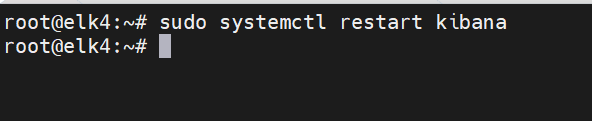
1. **Restart Elasticsearch service:**

sudo systemctl restart elasticsearch



1. **Restart Kibana service:**

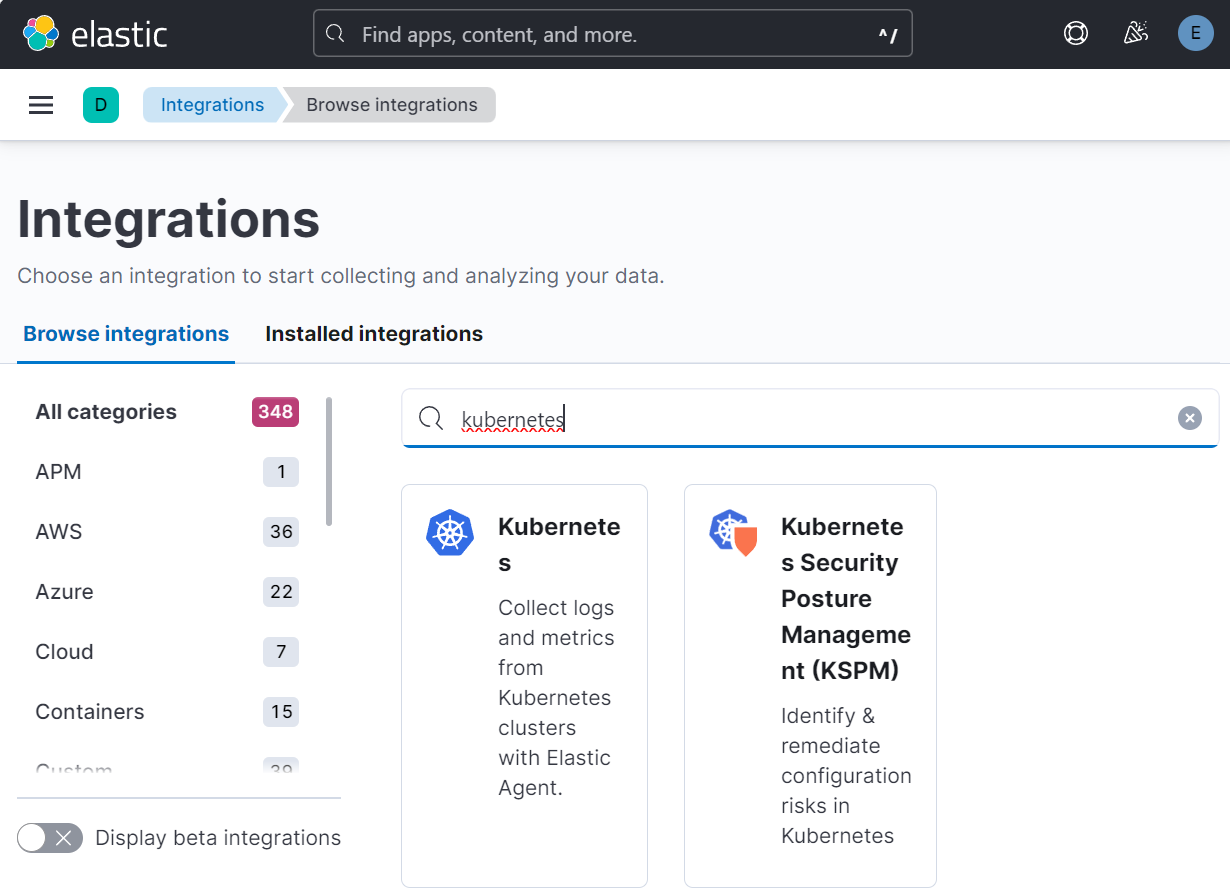
sudo systemctl restart kibana



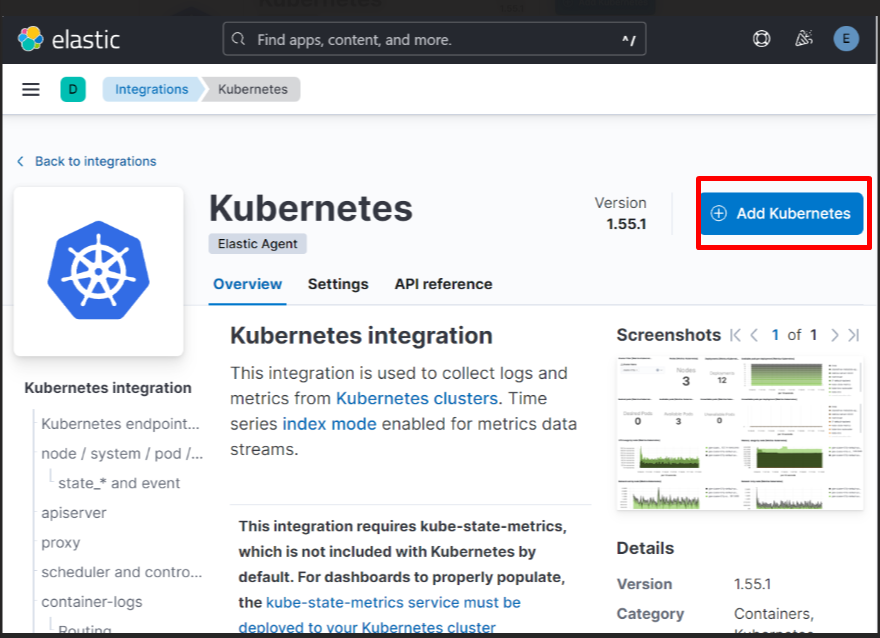
1. **Access Kibana URL:** http://<VMPublicIP>:5601
2. **Access Elasticsearch URL:** http://<VMPublicIP>:9200

***Kubernetes Agent Installation:***

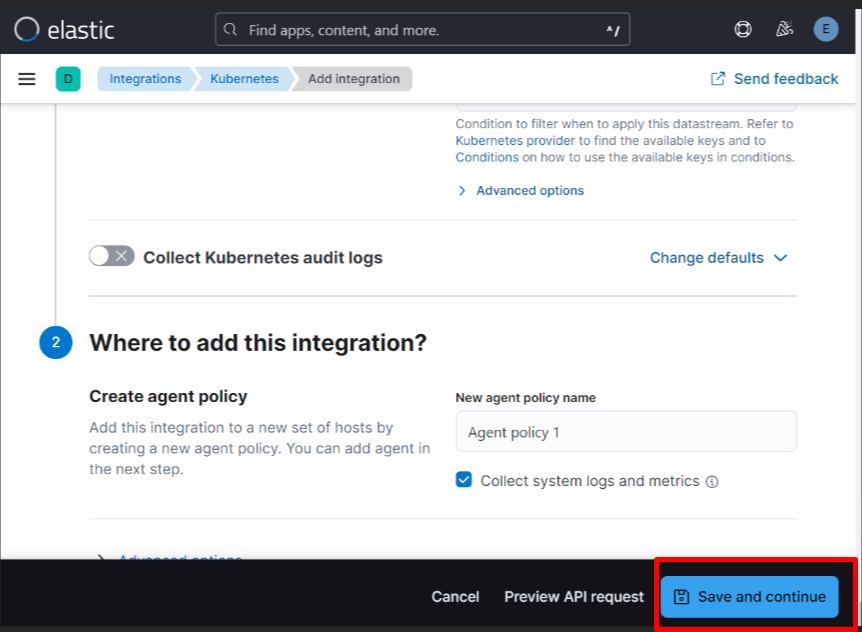
1. **Go to the Integrations page in Kibana.**
2. **Search for Kubernetes and select it.**



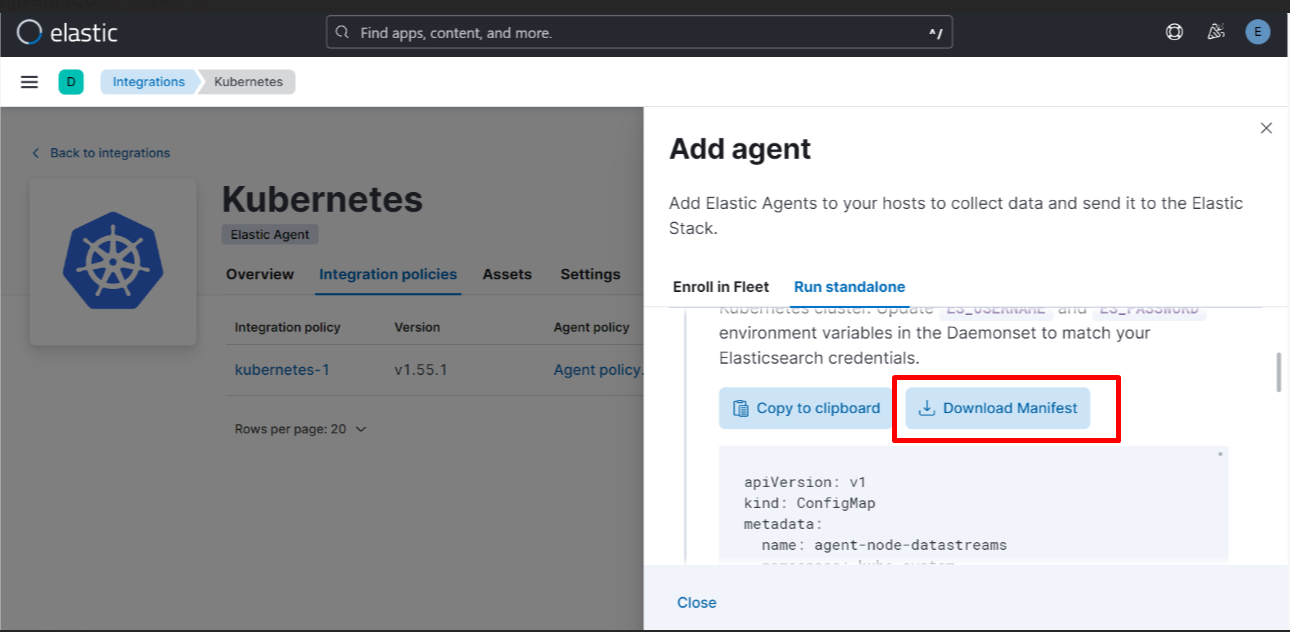
1. **Click on Add Integration.**

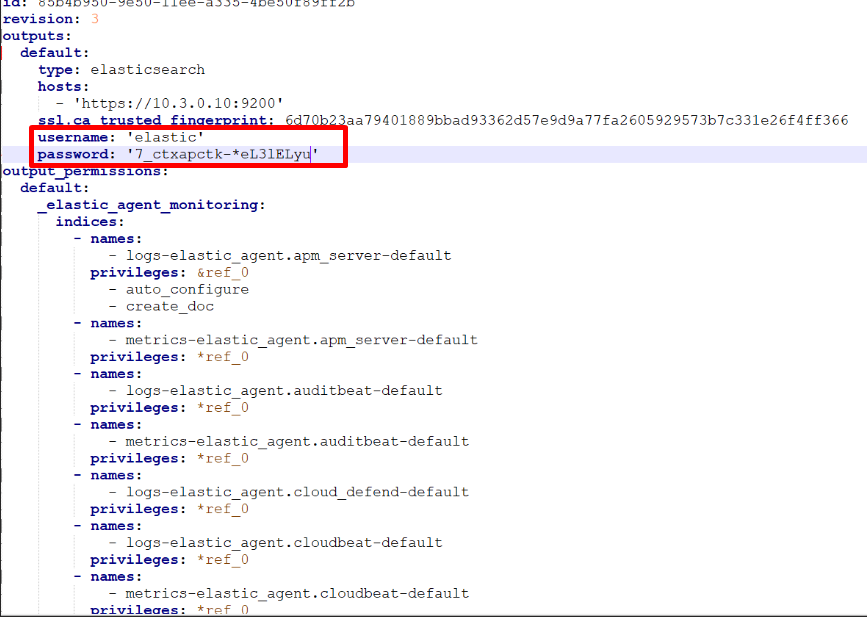


1. **Save and continue.**



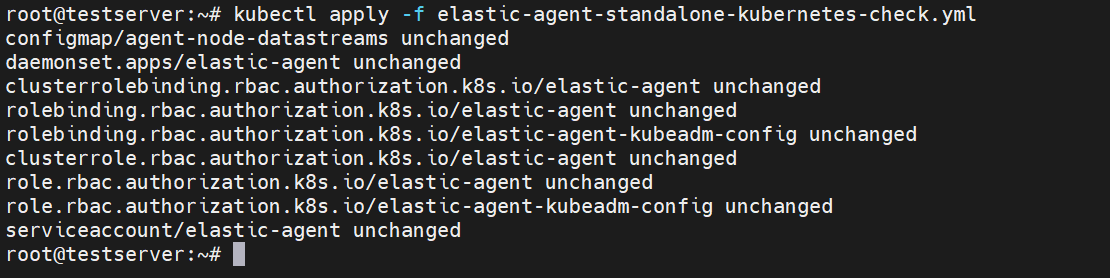
1. **Download the YAML file and replace the username and password for the elastic user.**





1. **Apply the YAML on the cluster.**

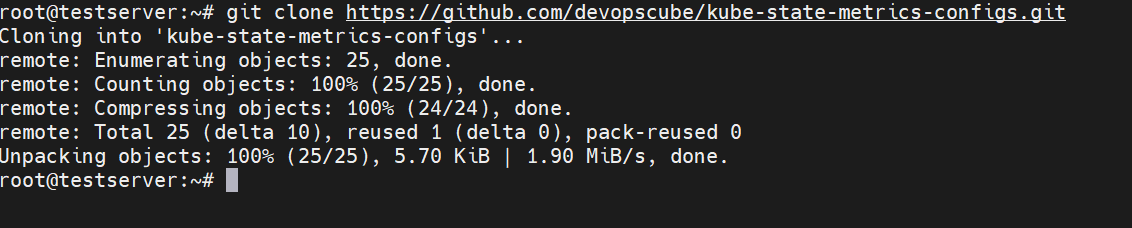
kubectl apply –f <yml filename>



***Kube State Metrics Installation (To be installed on Kubernetes Cluster):***

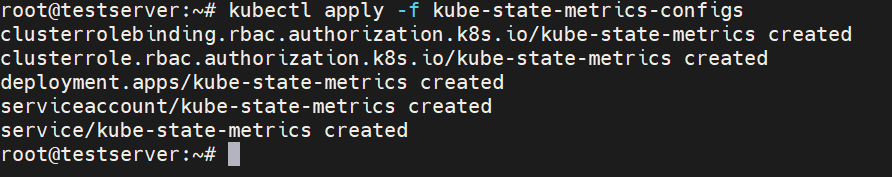
1. **Clone Kube State Metrics Configs:**

git clone <https://github.com/devopscube/kube-state-metrics-configs.git>



1. **Apply configurations to the Kubernetes cluster:**

kubectl apply -f kube-state-metrics-configs/

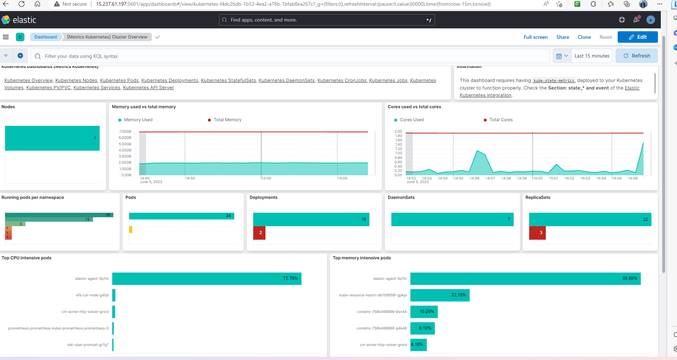
.

***Visualizing Kubernetes metrics on Elastic Observability****.*

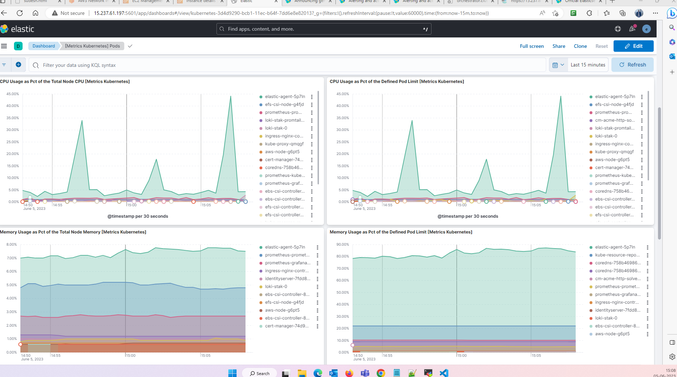
Navigate to **Dashboards** located under **Analytics** on the left and search **Kubernetes**.

If we look at the **Kubernetes Overview**, we can see metrics for the entire Kubernetes Cluster. All the nodes, pods, and CPU and memory usage.

Kubernetes Integration:



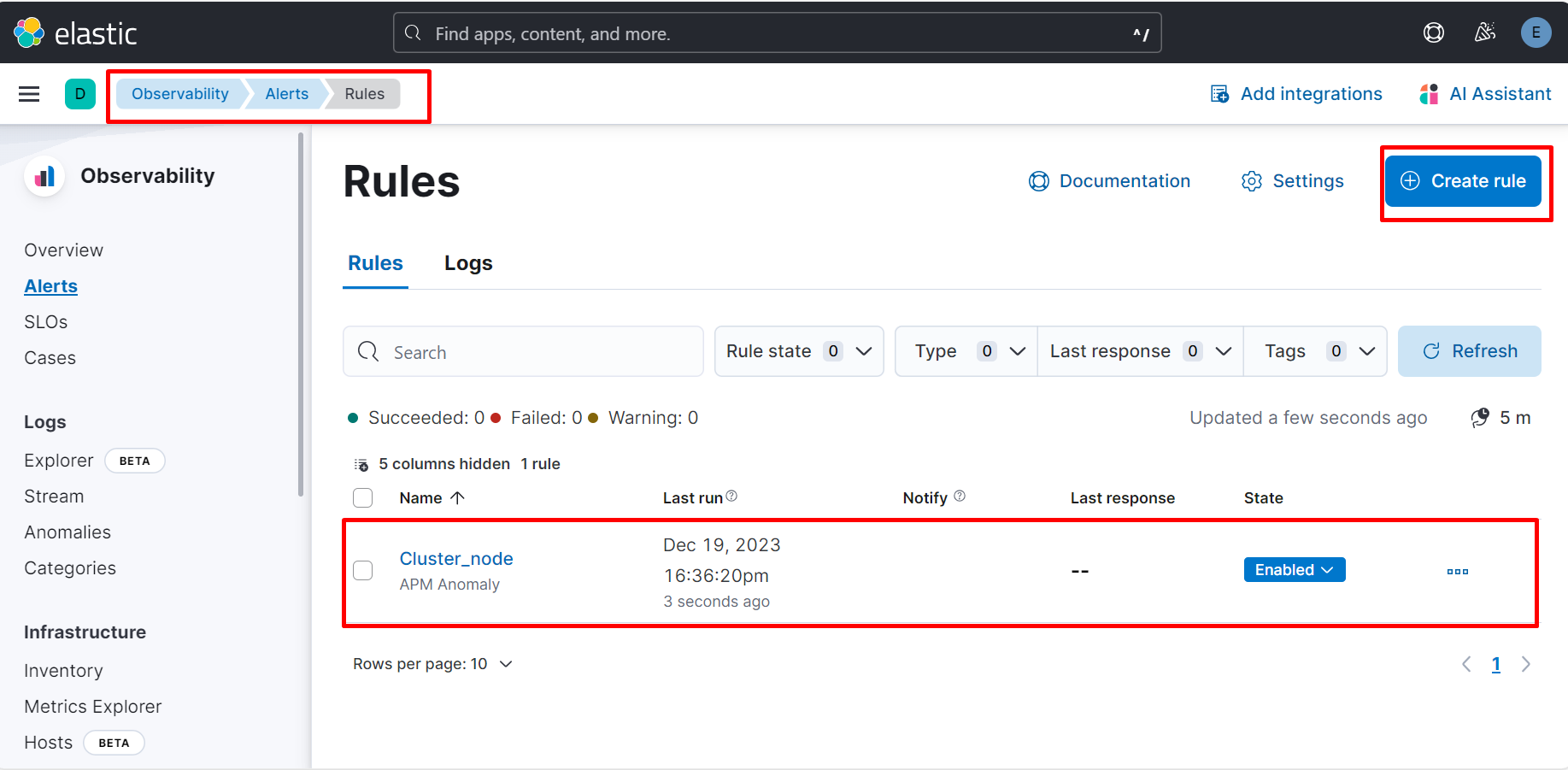
Pod details:



***Set up alerts.***

On the left navigation under **Observability**, next you'll want to select **Alerts** and **Create rules**. This will allow you to get notifications when various events happen via email, Jira, Slack, and more.

***Note:*** License is required



***license cost:***

image

[Official Elasticsearch Pricing: Elastic Cloud, Managed Elasticsearch](https://www.elastic.co/pricing/)

