Integrate Ezeelogin SSH Jump host with ELK for SIEM

358 admin July 26, 2024 Technical 5240

Integrating Ezeelogin SSH gateway with ELK stack for SIEM

Overview: This article outlines the process of integrating Ezeelogin SSH gateway with the ELK stack for SIEM. It includes installing and configuring ElasticSearch and Kibana on a monitoring server and Logstash on the Ezeelogin Jump Host server. Detailed instructions are provided for sending MySQL table data from the Ezeelogin server to ElasticSearch using Logstash, as well as setting up Kibana for data visualization. The article also covers verifying and visualizing data in Kibana once it is indexed in ElasticSearch.

In the example below, we would be installing ElasticSearch and Kibana on the monitoring server [monitor.eznoc.com (Centos 7 OS)] and the Logstash daemon on the Ezeelogin Jumphost server.

Step 1. Install Java 8

```
[root@monitor-eznoc ~]# yum install java-1.8.0-openjdk.x86_64

[root@monitor-eznoc ~]# java -version

openjdk version "1.8.0_272"

OpenJDK Runtime Environment (build 1.8.0_272-b10)

OpenJDK 64-Bit Server VM (build 25.272-b10, mixed mode)
```

Step 2. Download the ElasticSearch

```
[root@monitor-eznoc ~]# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.9.2-x86_64.rpm
[root@monitor-eznoc ~]# rpm -ivh elasticsearch-7.9.2-x86_64.rpm
[root@monitor-eznoc ~]# systemctl enable elasticsearch.service
[root@monitor-eznoc ~]# service elasticsearch restart
```

Step 3. Edit the ElasticSearch Configuration file and set the following variables.

```
[root@monitor-eznoc ~]# vi /etc/elasticsearch/elasticsearch.yml
cluster.name: elk-test
node.name: node-elk
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 0.0.0.0
discovery.type: single-node
[root@monitor-eznoc ~]# service elasticsearch restart
```

Step 4. Test the ElasticSearch Response

```
[root@monitor-eznoc ~]# curl -X GET 'http://localhost:9200'

{



 "name" : "elk-test",


 "cluster_name" : "elasticsearch",
```

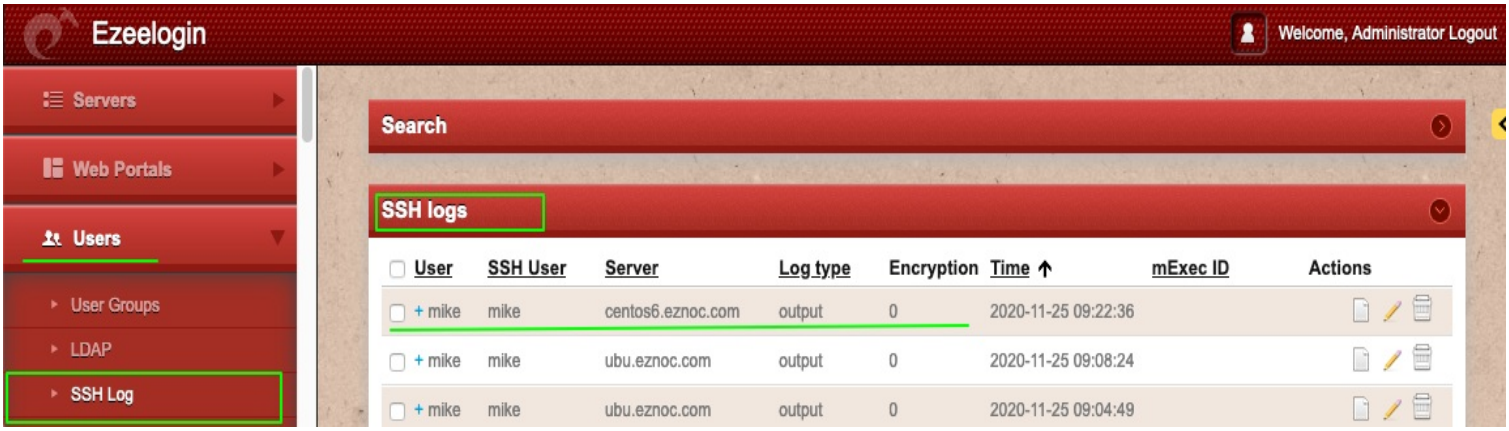
Step 5. Installing and Configuring Logstash on the Ezeelogin Jump Host server to send the Ezeelogin mysql table data (fqn_sshlogs, fqn_gwactivity_logs, fqn_serveractivity_logs,fqn_webactivity_logs) to Elastic Search.

```
[root@jump-host ~]#rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

[root@jump-host ~] vi /etc/yum.repos.d/logstash.repo

[logstash-7.x]

name=Elastic repository for 7.x packages

baseurl=https://artifacts.elastic.co/packages/7.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

[root@jump-server ~] yum install logstash
```

Step 5.a. Sending the mysql tables fqn_sshlogs from Jump Host server to Remote ElasticSearch server using Logstash.

```
root@jump-host:/etc/logstash/conf.d# vi ez_ssh_log.conf
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
 jdbc {
   jdbc_driver_library => "/usr/share/java/mysql-connector-java-8.0.22.jar"
   jdbc_driver_class => "com.mysql.jdbc.Driver"
   jdbc_connection_string => "jdbc:mysql://localhost:3306/ezlogin_eqb"
   jdbc_user => "logstas"
   jdbc_password => "Baiweigh6Voh6m"
   #parameters => { "favorite_artist" => "Beethoven" }
   schedule => "* * * * *"
   statement => "SELECT * from fqn_sshlogs"
   tracking_column => "created"
   use_column_value => "true"
 }
}

output {
 elasticsearch {
   hosts => ["http://monitor.eznoc.com:9200"]
   #index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{ YYYY.MM.dd}"
   index =>"gw_ssh"
   document_id => "%{id}"
   #user => "elastic"
   #password => "changeme"
 }
}
```

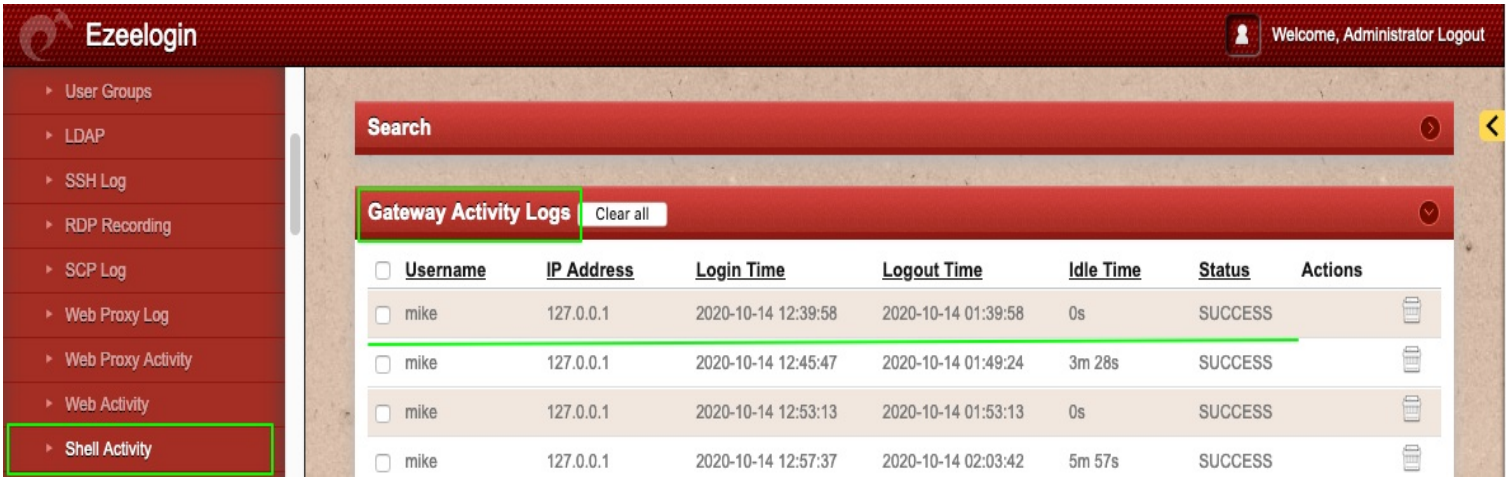
The mysql table fqn_sshlogs displays the following information in the GUI



Step 5.b. Sending the mysql table fqn_gwactivity_logs from Jump Host server to Remote ElasticSearch server using Logstash.

```
root@jump-host:/etc/logstash/conf.d# vi gateway_activity_log.conf
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
 jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-java-8.0.22.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/ezlogin_eqb"
  jdbc_user => "logstas"
  jdbc_password => "Baiweigh6Voh6m"
  #parameters => { "favorite_artist" => "Beethoven" }
  schedule => "* * * * *"
  statement => "SELECT `fqn_gwactivity_logs`.`id` as id, `remote_ip` , `login_time`, `logout_time`, `idle_time`,
`fqn_gwactivity_logs`.`status` as status, `username` FROM `fqn_gwactivity_logs` JOIN `fqn_users` ON
`fqn_users`.`id` = `fqn_gwactivity_logs`.`user_id`"
  tracking_column => "id"
  use_column_value => "true"
 }
}

output {
 elasticsearch {
  hosts => ["http://monitor.eznoc.com:9200"]
  #index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{ YYYY.MM.dd}"
  index =>"gw_activity"
  document_id => "%{id}"
  #user => "elastic"
  #password => "changeme"
 }
```

The mysql table fqn_gwactivity_logs displays the following information in the GUI

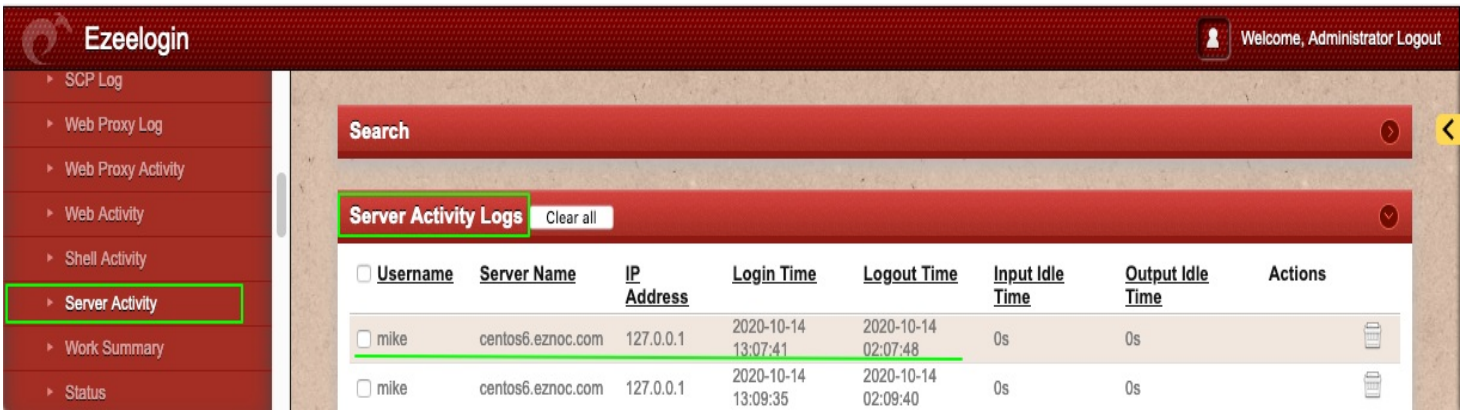


Step 5.c. Sending the mysql table fqn_serveractivity_logs from Ezeelogin Jump Host server to the monitor.eznoc.com server using Logstash.

```
root@jump-host:/etc/logstash/conf.d# vi serveractivity_logs.conf
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
 jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-java-8.0.22.jar"
```

```
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/ezlogin_eqb"
  jdbc_user => "logstas"
  jdbc_password => "Baiweigh6Voh6m"
  #parameters => { "favorite_artist" => "Beethoven" }
  schedule => "* * * * *"
  statement => "SELECT fqn_serveractivity_logs.id, fqn_users.username,  fqn_servers.name AS servername, fqn_gwactivity_logs.remote_ip, .fqn_serveractivity_logs.login_time, fqn_serveractivity_logs.logout_time, fqn_serveractivity_logs.input_idle_time, fqn_serveractivity_logs.output_idle_time, fqn_serveractivity_logs.status FROM fqn_serveractivity_logs JOIN fqn_users ON  fqn_users.id = fqn_serveractivity_logs.user_id JOIN fqn_servers ON fqn_servers.id = fqn_serveractivity_logs.server_id JOIN fqn_gwactivity_logs ON fqn_gwactivity_logs.id = fqn_serveractivity_logs.gwactivity_id"
  tracking_column => "id"
  use_column_value => "true"
 }
}

output {
 elasticsearch {
  hosts => ["http://monitor.eznoc.com:9200"]
  #index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{ YYYY.MM.dd}"
  index =>"server_activity"
  document_id =>"%{id}"
  #user =>"elastic"
  #password => "changeme"
 }
}
```

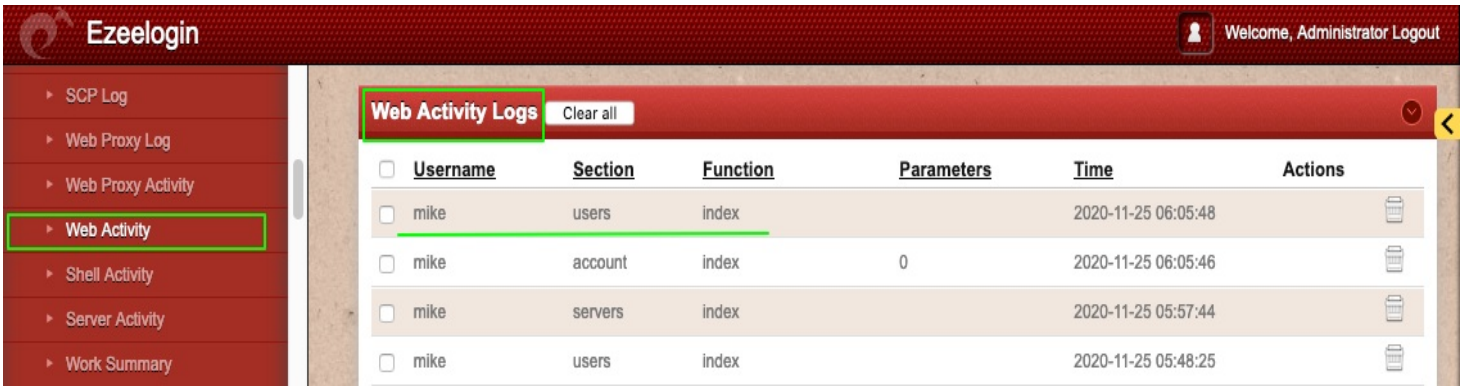
The mysql table fqn_serveractivity_logs displays the following information in the GUI



Step 5.d. Sending mysql table fqn_web_activity log from Jump Host server to Remote ElasticSearch server using Logstash.

```
root@jump-host:/etc/logstash/conf.d# vi web_activity_logs.conf
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
 jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-java-8.0.22.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/ezlogin_eqb"
  jdbc_user => "logstas"
  jdbc_password => "Baiweigh6Voh6m"
  #parameters => { "favorite_artist" => "Beethoven" }
  schedule => "* * * * *"
  statement => "SELECT `fqn_webactivity_logs`.`id`, `controller`, `function`, `objective`, `fqn_webactivity_logs`.`created`, `username` FROM `fqn_webactivity_logs` JOIN `fqn_users` ON `fqn_users`.`id` = `fqn_webactivity_logs`.`user_id`"
  tracking_column => "id"
  use_column_value => "true"
 }
}

output {
 elasticsearch {
  hosts => ["http://monitor.eznoc.com:9200"]
  #index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{ YYYY.MM.dd}"
  index =>"web_activity"
  document_id => "%{id}"
  #user =>"elastic"
  #password => "changeme"
 }
}
```

The mysql table fqn_webactivity_logs displays the following information in the GUI



Restart the logstash daemon

```
[root@jump-host ~] systemctl restart logstash
```

To find the sql queries to be used in logstash configuration file, simply enable general log in mysql daemon and run the search query in GUI. You can generate any number of queries to tailor the data according to your requirement.

```
mysql > "SET GLOBAL general_log = 'ON';”
mysql [(none)]> show variables like "%general%";
+------------------+---------+
| Variable_name    | Value   |
+------------------+---------+
| general_log      | ON      |
| general_log_file | gate.log |

+------------------+————+

tail -f /var/lib/mysql/gate.log
 855 Query SELECT `servers`.`id` AS `id`, `servers`.`name` AS `servername`, `ips`.`ip` AS `ip`, `ssh_user`, `ssh_port`, `servers`.`description` AS `description`, `enable_ssh`, `controlpanel_id`, `datacenter_id`, `servergroup_id`, `servergroups`.`name` AS `groupname`, `ipmi_enable`, `ipmi_request_uri`, `servers`.`switch_sudo`, `servers`.`switch_user`, `keep_password`, `ssh_key`, `rdp`, `windows_domain` FROM `nfo_servers` AS `servers` JOIN `nfo_servergroups` AS `servergroups` ON `servers`.`servergroup_id` = `servergroups`.`id` JOIN `nfo_ips` AS `ips` ON `ips`.`server_id` = `servers`.`id` WHERE `ips`.`main` = 1 AND (`servers`.`id` IN (SELECT `id` FROM `nfo_servers` WHERE `servergroup_id` IN (SELECT `servergroup_id` FROM `nfo_usergroup_servergroup_acls` WHERE `usergroup_id` = '1' ) OR `servergroup_id` IN (SELECT `servergroup_id` FROM `nfo_user_servergroup_acls` WHERE access='Y' and `user_id`= '1') ) AND `servers`.`id` NOT IN (SELECT `server_id` FROM `nfo_user_server_acls` WHERE `user_id`= '1'  and access='N' ) AND `servers`.`id` NOT IN (SELECT `id` FROM `nfo_servers` WHERE `servergroup_id` IN (SELECT `servergroup_id` FROM `nfo_user_servergroup_acls` WHERE `user_id` = '1' AND `access`='N')) OR `servers`.`id` IN (SELECT `server_id` FROM `nfo_user_server_acls` WHERE `user_id`= '1' AND `access`='Y')) ORDER BY `servername` ASC

 855 Query SELECT `nfo_users`.`username` AS `user`, `nfo_servers`.`name` AS `server`, `nfo_sshlogs`.*
```

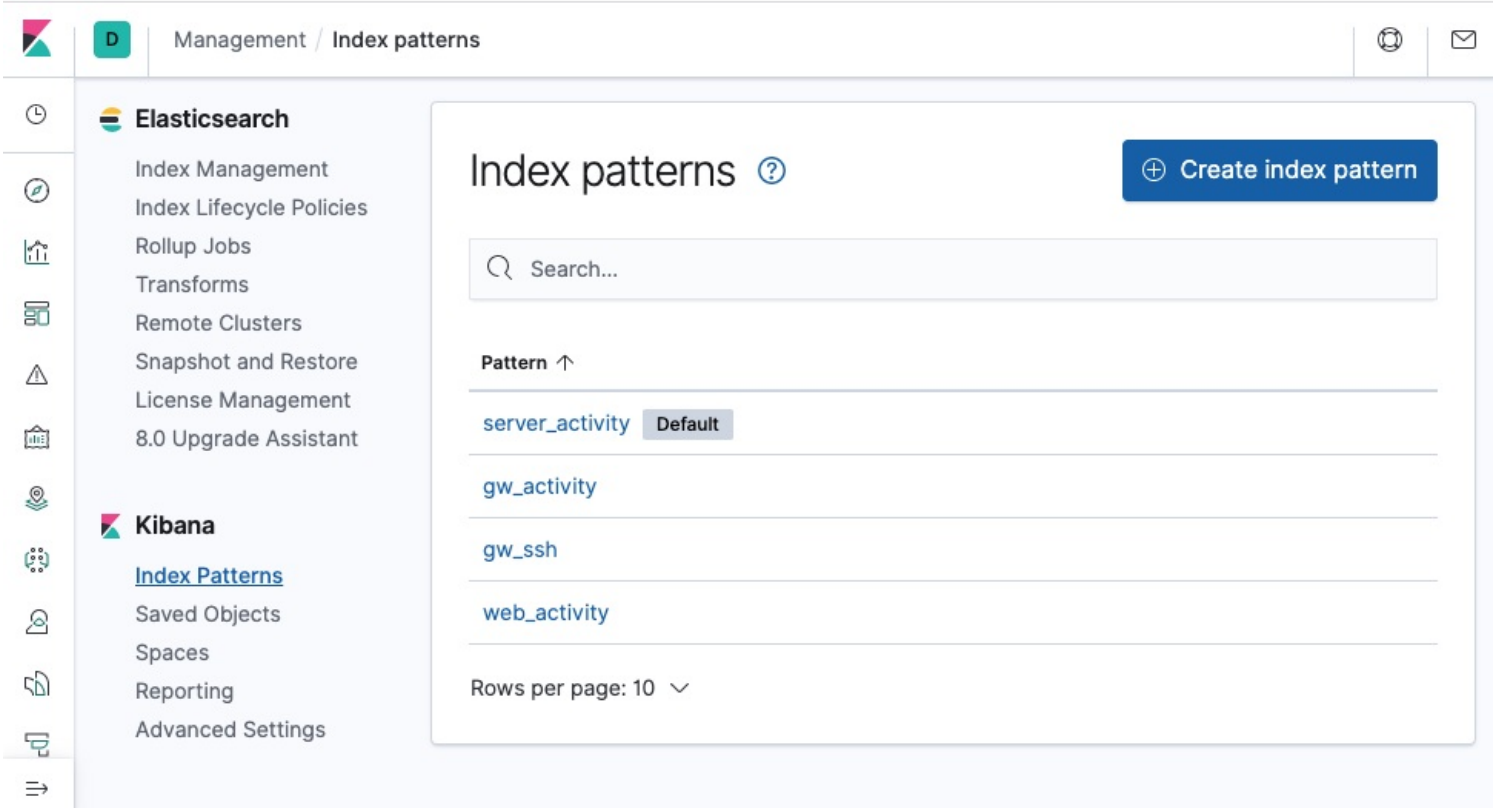
Step 6. Installing and Configuring the Kibana Dashboard on the monitoring server. Refer the article

```
[root@jump-host ~]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

[root@jump-host ~]#vi /etc/yum.repos.d/kibana.repo

 [kibana-7.x]

name=Kibana repository for 7.x packages
```

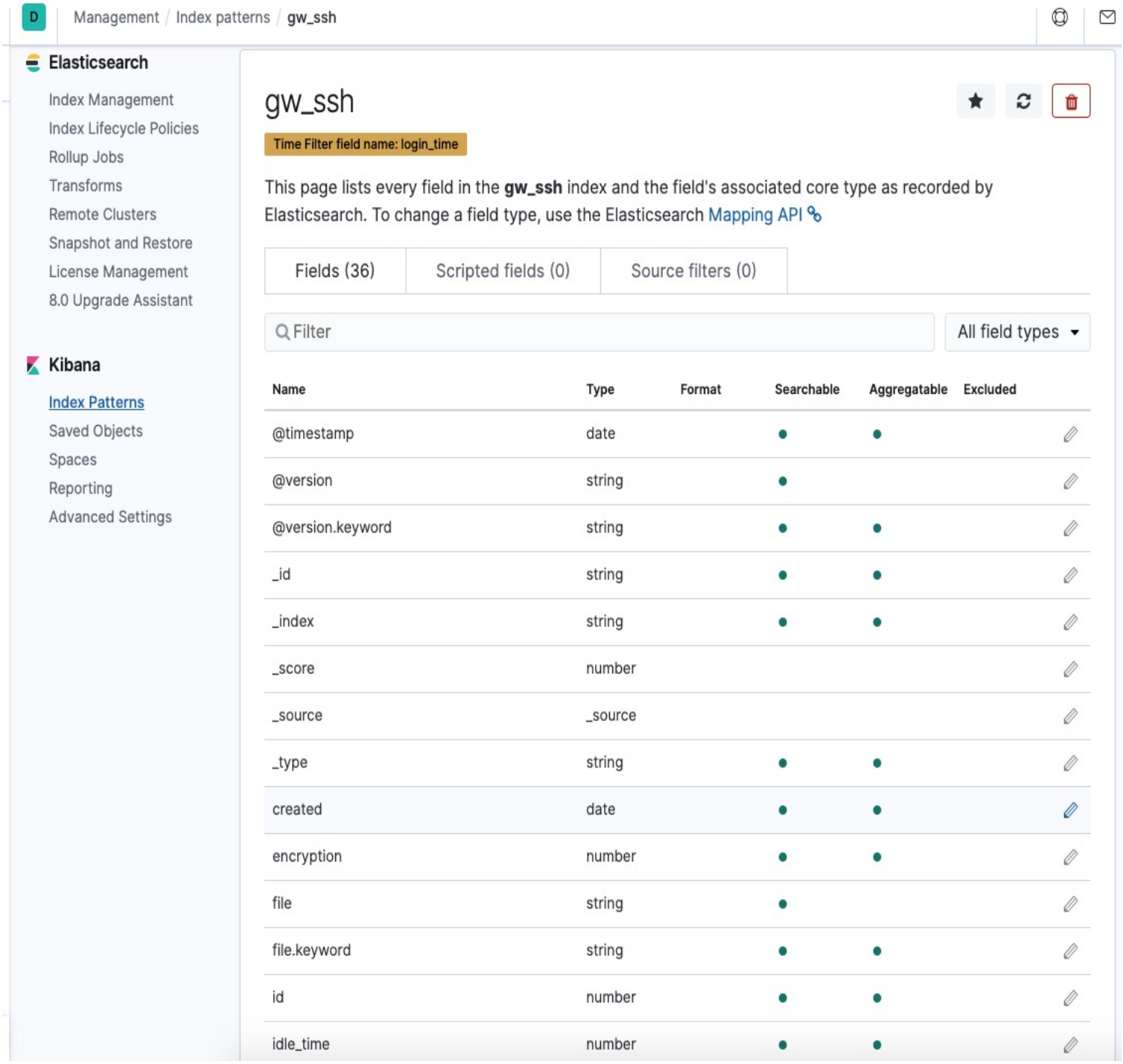
```
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

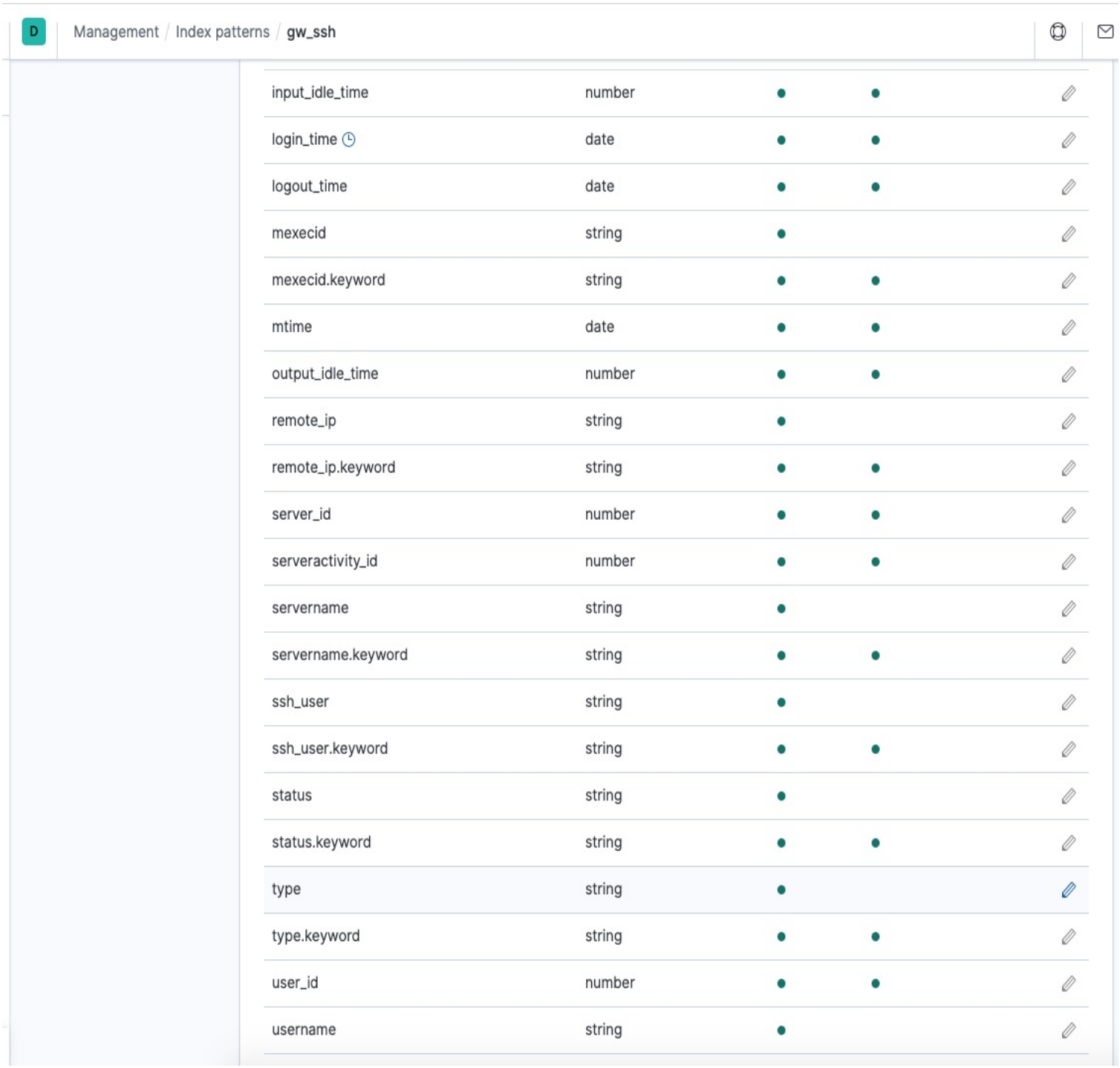
```
[root@jump-host ~]# yum install kibana
[root@jump-host ~]# systemctl enable kibana.service;systemctl start kibana.service
```

Access the Kibana Dashboard http://monitor.eznoc.com/app/kibana#/management/kibana/index_patterns?_g=()

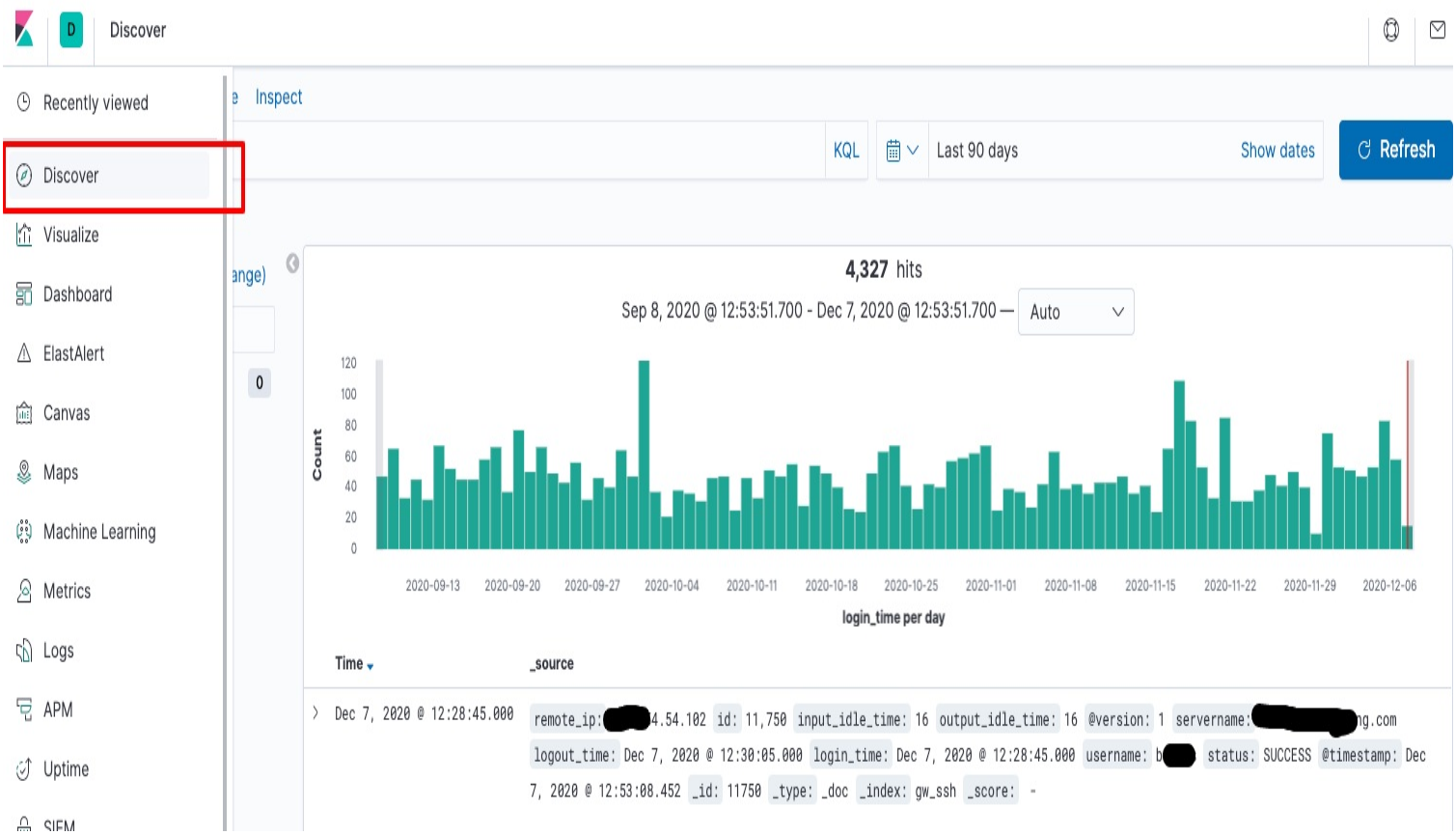


Click on the gw_ssh index pattern and you will be able to see the data from mysql table fqn_sshlogs belonging to Ezeelogin Jump Server database.

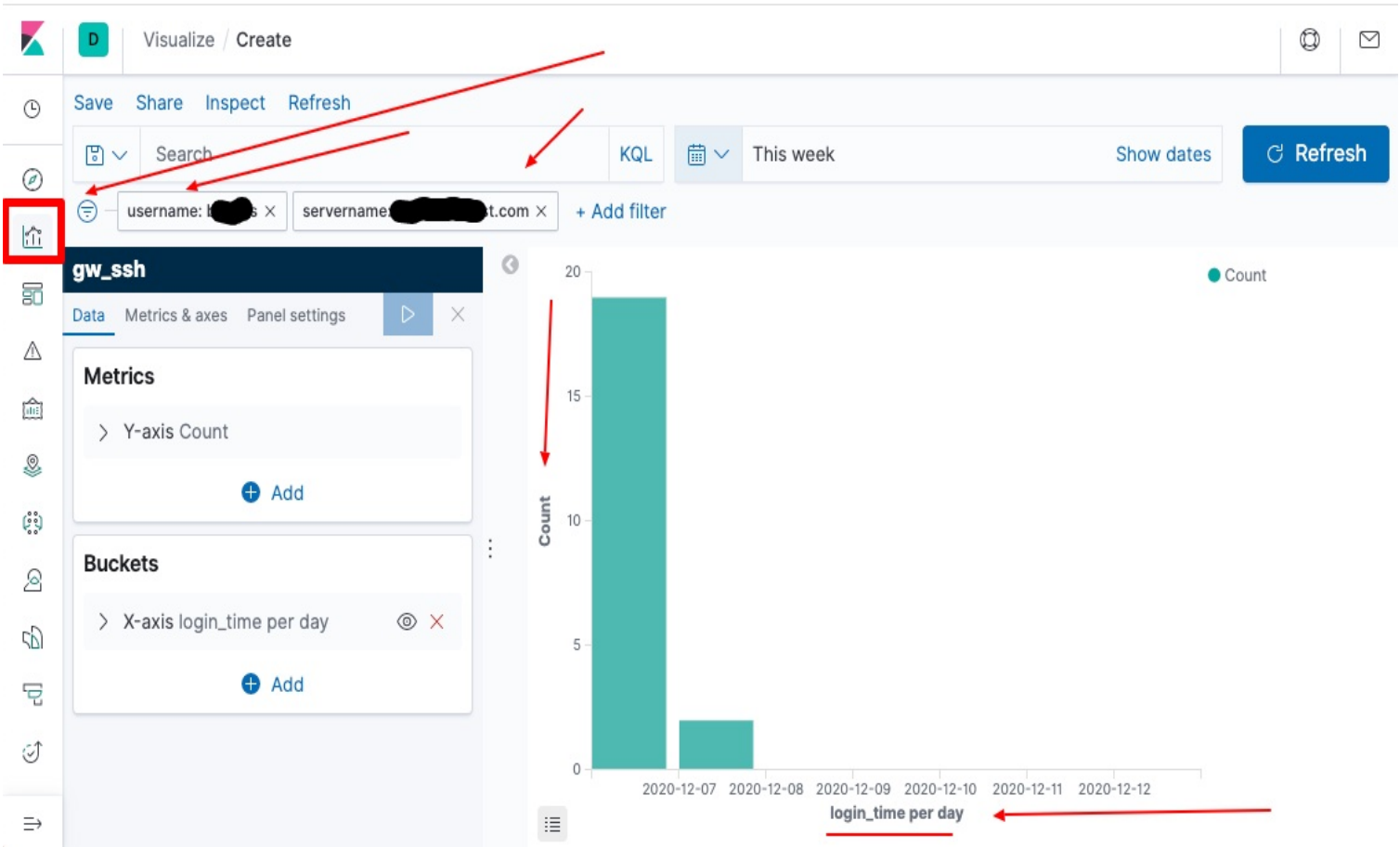




Once you have the data in ElasticSearch, you can verify the mysql data fetched under "Discovery".



You can create different type of Visualizations from data imported using the "Visualize" feature.



You can also create a dashboard for in Kibana to view the various graphical visualization that was created.

Related Articles

Integrate SSH Jump server with splunk for SIEM

Integrate SSH Jump Server with syslog