

ELASTIC SEARCH IN CA PPM

Sriram Nandiraju (Ram)*

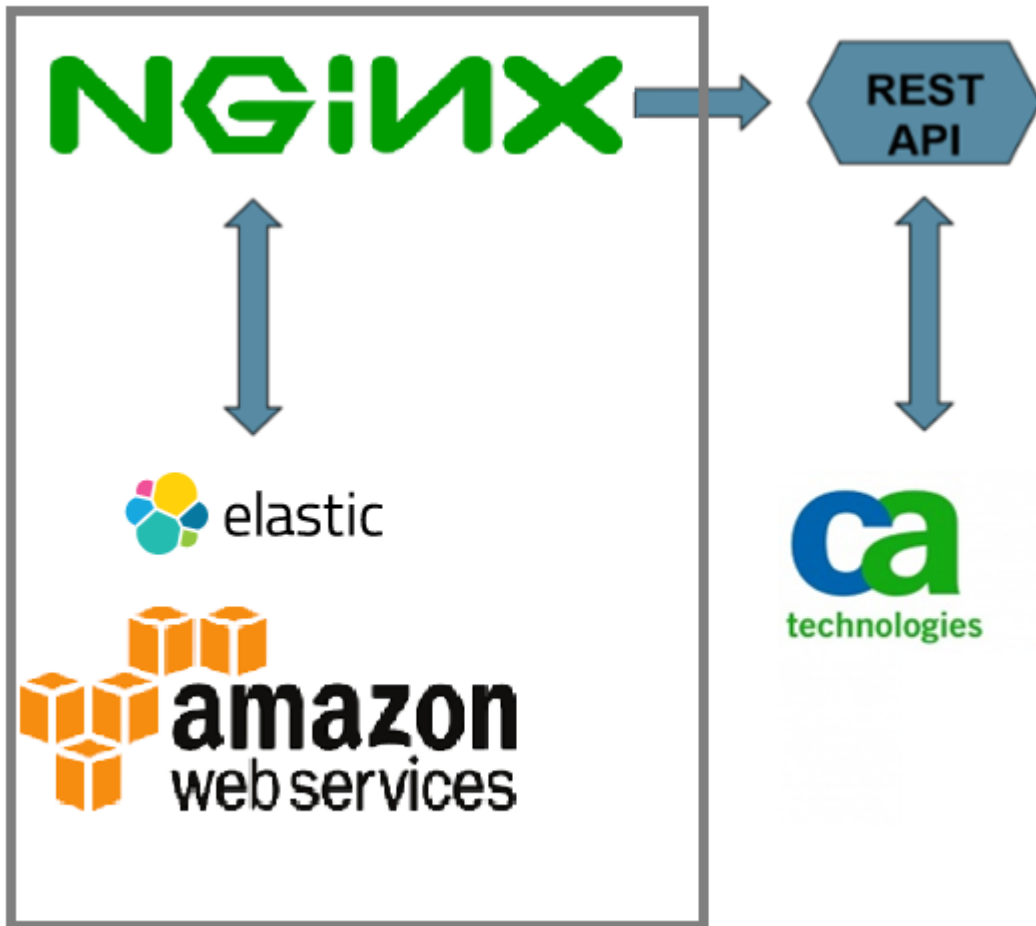
Introduction: Elasticsearch is a distributed, RESTful search and analytics engine based on Lucene capable of solving a growing number of use cases. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. Official clients are available in Java, .NET (C#), PHP, Python, Apache Groovy and many other languages. Elasticsearch is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.

CA PPM represents a single platform that enables you to manage your entire innovation lifecycle and make more informed strategic investments. CA PPM helps you track and prioritize market and customer requirements and make better decisions on how to invest limited resources, so you can optimize your enterprise, IT, service and product portfolio.

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis with a free-tier option available for 12 months. The technology allows subscribers to have at their disposal a full-fledged virtual cluster of computers, available all the time, through the internet. AWS's version of virtual computers have most of the attributes of a real computer including hardware (CPU(s) & GPU(s) for processing, local/RAM memory, hard-disk/SSD storage); a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, CRM, etc.

* Principal Software Developer in Sabre Corporation. He has 14 years of experience in IT and has worked in multiple technologies.

This paper successfully demonstrates the functionality of Elasticsearch installed on an AWS AMI and its real-time integration with CA PPM via REST based web services and JavaScript.



This document is divided into 6 sections.

1. Install Elasticsearch on AWS (click [here](#))
2. Install & Enable SSL through NGINX (click [here](#))
3. Load data into ElasticSearch (click [here](#))
4. Query data through CA PPM Interface (click [here](#))
5. Demo (click [here](#))
6. References (click [here](#))

Install Elasticsearch on AWS

Below is the EC2 Ubuntu Amazon Machine Image which I have provisioned in my personal AWS account.

| Description | Status Checks | Monitoring | Tags |
|-----------------------|--|------------|------|
| Instance ID | i-██████████2 | | |
| Instance state | running | | |
| Instance type | t2.micro | | |
| Elastic IPs | | | |
| Availability zone | us-east-1b | | |
| Security groups | launch-wizard-18. view inbound rules | | |
| Scheduled events | No scheduled events | | |
| AMI ID | ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20170721 (ami-cd0f5cb6) | | |
| Platform | - | | |
| IAM role | RamEc2Dec272016 | | |
| Key pair name | Dec282016 | | |
| Owner | ██████████ | | |
| Launch time | September 4, 2017 at 3:47:02 AM UTC-4 (1249 hours) | | |
| Public DNS (IPv4) | ec2-██████████-1.amazonaws.com | | |
| IPv4 Public IP | ██████████ | | |
| IPv6 IPs | - | | |
| Private DNS | ip-██████████ec2.internal | | |
| Private IPs | ██████████ | | |
| Secondary private IPs | | | |
| VPC ID | ██████████3 | | |
| Subnet ID | ██████████B | | |
| Network interfaces | eth0 | | |
| Source/dest. check | True | | |
| EBS-optimized | False | | |
| Root device type | ebs | | |

Once the EC2 instance is provisioned, follow the below simple installation instructions.

Install Java8

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

Install Elasticsearch

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
sudo apt-get install apt-transport-https
echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-5.x.list
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.5.2.deb
sha1sum elasticsearch-5.5.2.deb
sudo dpkg -i elasticsearch-5.5.2.deb
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

```
chown -R elasticsearch:elasticsearch /var/lib/elasticsearch/  
Add START_DAEMON=true in /var/lib/elasticsearch/
```

Install & Enable SSL through NGINX

Install NGINX

```
sudo apt-get update
sudo apt-get install nginx
sudo service nginx start
```

Enable SSL

```
sudo mkdir /etc/elasticsearch/ssl
cd /etc/elasticsearch/ssl
sudo openssl genrsa -des3 -out es_domain.key 1024
```

```
sudo openssl req -new -key es_domain.key -out es_domain.csr
sudo cp es_domain.key es_domain.key.bk
sudo openssl rsa -in es_domain.key.bk -out es_domain.key
sudo openssl x509 -req -days 3650 -in es_domain.csr -signkey es_domain.key -out
es_domain.crt
sudo mkdir -p /var/log/nginx/elasticsearch/
sudo chown www-data:www-data /var/log/nginx/elasticsearch/
```

```
nano /etc/nginx/sites-available/elasticsearch
server {
    listen 443;
    ssl on;
    ssl_certificate /etc/elasticsearch/ssl/es_domain.crt;
    ssl_certificate_key /etc/elasticsearch/ssl/es_domain.key;
    access_log /var/log/nginx/elasticsearch/access.log;
    error_log /var/log/nginx/elasticsearch/error.log debug;

    location / {
        if ($request_method ~* "(GET|POST)") {
            add_header "Access-Control-Allow-Origin" *;
        }

        rewrite ^/(.*) /$1 break;
```

```
proxy_ignore_client_abort on;
proxy_pass http://localhost:9200;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $http_host;
}
}
server {
    listen 80;
    return 301 https://$host$request_uri;
}
```

```
sudo ln /etc/nginx/sites-available/elasticsearch /etc/nginx/sites-enabled/
sudo service nginx reload
```

```
sudo scp ec2-user@ec2-99-999-999-999.compute-
1.amazonaws.com:/etc/elasticsearch/ssl/es_domain.crt /local/path/to/store/cert
```

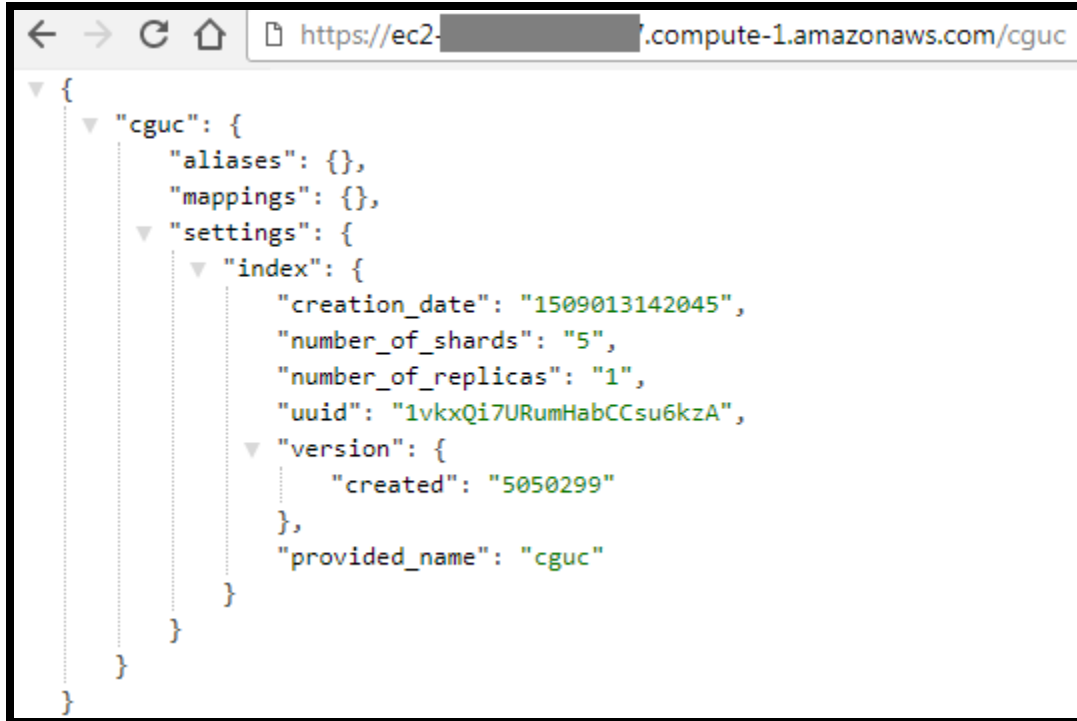
```
curl --cacert es_domain.crt --tlsv1 https://ec2-99-999-999-999.compute-
1.amazonaws.com
```

```
Add below in nano /etc/elasticsearch/elasticsearch.yml
http.cors.enabled: true
http.cors.allow-origin: "*"

```

Load data into Elasticsearch

We can create the data model by making the http POST request as shown below. The data gets loaded into the cguc header



```

{
  "cguc": {
    "aliases": {},
    "mappings": {},
    "settings": {
      "index": {
        "creation_date": "1509013142045",
        "number_of_shards": "5",
        "number_of_replicas": "1",
        "uuid": "1vkxQi7URumHabCCsu6kzA",
        "version": {
          "created": "5050299"
        },
        "provided_name": "cguc"
      }
    }
  }
}

```

Load the data of CAPPm projects in the below CURL format from command line.

```

curl -X POST 'http://localhost:9200/cguc/project/PRJ000112' -d '{ "prj_id":
"PRJ000112", "prj_name": "Idea test", "is_active": "Yes", "prj_url":
"https://cppm1109.ondemand.ca.com/niku/nu#action:projmgr.projectProperties&id=5
020009" ,"prj_mgr": "Geraldine, Judith" }'

```

Elasticsearch gives the below confirmation

```

{
  "_index": "cguc",
  "_type": "project",
  "_id": "PRJ000112",
  "_version": 1,
  "result": "created",
  "_shards": {

```



```
"total":2,  
"successful":1,  
"failed":0  
},  
"created":true  
}
```

As much data as possible can be loaded via the command line.

Query data through CA PPM Interface

A HTML portlet written with Javascript makes a real-time REST webservice call to Elasticsearch which is installed on AWS and renders the data in less than 10 milliseconds.

```
<html>
<head>

<style>
  body {
    font-family: Arial, Helvetica, Verdana, sans-serif;
  }
  table{
    width: 100%;
    background: #e8ebf2;
    font-size: 12px;
  }

  .thead1:hover{
    background-color:#00ffff;
  }
  .tbody1:hover{
    background-color:#f7f4b4;
  }
  .featuresTable{
    outline: 1px solid; /* use instead of border */
    margin-top: 1px;
    margin-left: 1px;
  }
  .noPadding{
    padding: 0;
  }
  .vTop{
    vertical-align: top;
    border-right: 1px solid black;
  }
  .thead1{
    font-weight: bold;
    border-bottom: 1px solid black;
    background: #98acbf;
  }
  .featuresTable tr td{
    padding: 1px;
  }
  /* Button styles */
  .ppm_button {
    border-top: 1px solid #c4d1e1;
    border-bottom: 1px solid #889ab5;
    border-left: 1px solid #bccee4;
    border-right: 1px solid #889ab5;
    background: -webkit-gradient(linear, center top, center bottom, color-stop(0, #ebf4fc), color-stop(.5, #d9e9fc), color-stop(.52, #c6ddf6), color-stop(1, #c3d3eb));
    background-image: -moz-linear-gradient(center top, #ebf4fc 0%, #d9e9fc 50%, #c6ddf6 52%, #c3d3eb 100%);
  }

```

```

background: linear-gradient(center top, #ebf4fc 0%, #d9e9fc 50%, #c6ddf6 52%, #c3d3eb 100%);
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr=#ebf4fc,
endColorstr=#c3d3eb)";
}

/* Button styles when hovered */
.ppm_button:hover {
background: -webkit-gradient(linear, center top, center bottom, color-stop(0, #f8fbff), color-
stop(.5, #e4effb), color-stop(.52, #d1e4f8), color-stop(1, #c5d5eb));
background-image: -moz-linear-gradient(center top, #f8fbff 0%, #e4effb 50%, #d1e4f8 52%,
#c5d5eb 100%);
background: linear-gradient(center top, #f8fbff 0%, #e4effb 50%, #d1e4f8 52%, #c5d5eb 100%);
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr=#f8fbff,
endColorstr=#c5d5eb)";
}
.label {
font-family: Verdana, Arial, sans-serif;
font-size: 10px;
font-weight: bold;
}

</style>

</head>
<body>
<br>
<font class="label">Project ID </font><input type="text" size="12" name="srch_prj_id"> <font class="label">Project
Name </font><input type="text" size="50" name="srch_prj_name">
<br><br>
<input type="button" value="Filter" class="ppm_button" onclick="loadXMLDoc('hi',undefined)" /> <input type="button"
value="Show All" class="ppm_button" onclick="loadXMLDoc('showall',1)" />
<br>
<div id="listingTable"></div>
<br>
<input type="btn_prev" size="5" value="Prev 20" class="ppm_button" onclick="prevPage()" /> <input type="btn_next" size="5"
value="Next 20" class="ppm_button" onclick="nextPage()" /> <b>Page: </b><span id="page"></span>

<br>
<br>
<table>
<tr>
  |
```



```

var page_span = document.getElementById("page");

// Validate page
if (page < 1) page = 1;
if (page > numPages()) page = numPages();

listing_table.innerHTML = "";

if ( (page-1) < (page * records_per_page) ) {
    var page_size_from;
    page_size_from = (page * records_per_page);
    loadXMLDoc(endpoint_url_parm,page_size_from);
}
page_span.innerHTML = page + " of " + numPages() + " pages";

/*
if (page == 1) {
    btn_prev.style.visibility = "hidden";
} else {
    btn_prev.style.visibility = "visible";
}

if (page == numPages()) {
    btn_next.style.visibility = "hidden";
} else {
    btn_next.style.visibility = "visible";
}
*/
}

function loadXMLDoc(endpoint_url_parm,param_page_size_from)
{
    console.log("endpoint_url_parm: " + endpoint_url_parm);
    console.log("parm_page_size_from: " + parm_page_size_from);

    document.getElementById('prj_name').innerHTML=document.getElementById('prj_id').innerHTML = "";
    document.getElementById('prj_name').innerHTML=document.getElementById('prj_name').innerHTML = "";
    document.getElementById('prj_name').innerHTML=document.getElementById('prj_mgr').innerHTML = "";
    document.getElementById('prj_name').innerHTML=document.getElementById('timeTaken').innerHTML = "";

    if (endpoint_url_parm == undefined)
    {
        return;
    }

    if (parm_page_size_from == undefined)
    {
        console.log("parm_page_size_from is undefined");
        parm_page_size_from = 1;
        console.log("changed parm_page_size_from: " + parm_page_size_from);
    }

    if (endpoint_url_parm == 'showall')
    {
        str="";
        document.getElementsByName('srch_prj_id')[0].value = "";
        document.getElementsByName('srch_prj_name')[0].value = "";
        var endpoint = aws_url + "/cguc/project/_search?pretty=true&q=*:*&size=" + records_per_page +
"&from="+parm_page_size_from;
    }
}

```

```

else
{
    if (document.getElementsByName('srch_prj_name')[0].value == "")
    {
        var endpoint = aws_url + "/cguc/project/" + document.getElementsByName('srch_prj_id')[0].value;
    }
    if (document.getElementsByName('srch_prj_id')[0].value == "")
    {
        var endpoint = aws_url + "/cguc/project/_search?pretty=true&size=" + records_per_page +
"&from=" + parm_page_size_from + "&q=prj_name:" + document.getElementsByName('srch_prj_name')[0].value;
    }
    if (
        document.getElementsByName('srch_prj_id')[0].value == ""
        &&
        document.getElementsByName('srch_prj_name')[0].value == ""
    )
    {
        return;
    }
}

var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function()
{
    //alert("After Ready state change");
    //alert(xmlhttp.readyState);
    //alert("hi");
    //alert(xmlhttp.status);
    //alert("xmlhttp.responseText: " + xmlhttp.responseText);
    var theXML = xmlhttp.responseText;
    //alert(theXML);

    if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
    {
        //alert("Step-4");
        //var theXML = xmlhttp.responseText;
        //alert(theXML);
        myObj = JSON.parse(theXML);

        if (endpoint_url_parm == 'hi')
        {
            if (document.getElementsByName('srch_prj_id')[0].value != "")
            {
                var str_prj_mgr="";

                document.getElementById('timeTaken').innerHTML="";
                str = "<tr class=tCell><td><a href="+myObj._source.prj_url+"
target=_blank>"+myObj._source.prj_id + "</a></td></tr>";

                document.getElementById('prj_id').innerHTML=document.getElementById('prj_id').innerHTML+str;
                str = "<tr class=tCell><td>"+myObj._source.prj_name + "</td></tr>";

                document.getElementById('prj_name').innerHTML=document.getElementById('prj_name').innerHTML+str;

                if (myObj._source.prj_mgr == "")
                {
                    str_prj_mgr = '--';
                }
            }
        }
    }
}

```

```

else
{
    str_prj_mgr = myObj._source.prj_mgr;
}
strs= "<tr class=tCell><td>" + str_prj_mgr + "</td></tr>";

document.getElementById('prj_mgr').innerHTML=document.getElementById('prj_mgr').innerHTML+strs;
}
}

if (
    (endpoint_url_parm == 'hi' &&
document.getElementsByName('srch_prj_name')[0].value != "")
    ||
    (endpoint_url_parm == 'showall')
)
{
    var totalCount=records_per_page; //myObj._shards.total;
    //alert('totalCount' +totalCount);
    //var strs="";
    for( var c=0; c<totalCount; c++)
    {
        //alert(c);
        //alert(myObj.hits.[c]);

        var str_prj_mgr="";

        if (c==0)
        {
            strs= "<tr><td><b><font color=red>" + myObj.took
+ "</font></b></td></tr>";
            document.getElementById('timeTaken').innerHTML="Time taken
(in <b><i>Milliseconds</i></b></i>) to query the data from AWS ElasticSearch: " + document.getElementById('timeTaken').innerHTML+ " " + strs;
        }
        strs= "<tr class=tCell><td><a href="+myObj.hits.hits[c]._source.prj_url+"
target=_blank">"+myObj.hits.hits[c]._source.prj_id + "</a></td></tr>";

document.getElementById('prj_id').innerHTML=document.getElementById('prj_id').innerHTML+strs;
        strs= "<tr class=tCell><td>" + myObj.hits.hits[c]._source.prj_name
+ "</td></tr>";

document.getElementById('prj_name').innerHTML=document.getElementById('prj_name').innerHTML+strs;

        if (myObj.hits.hits[c]._source.prj_mgr == "")
        {
            str_prj_mgr = '--';
        }
        else
        {
            str_prj_mgr = myObj.hits.hits[c]._source.prj_mgr;
        }

        strs= "<tr class=tCell><td>" + str_prj_mgr + "</td></tr>";

document.getElementById('prj_mgr').innerHTML=document.getElementById('prj_mgr').innerHTML+strs;
    } //for
} //else
} //if
//alert("Step-5");
} //fun

```

```
//alert("Step-3");

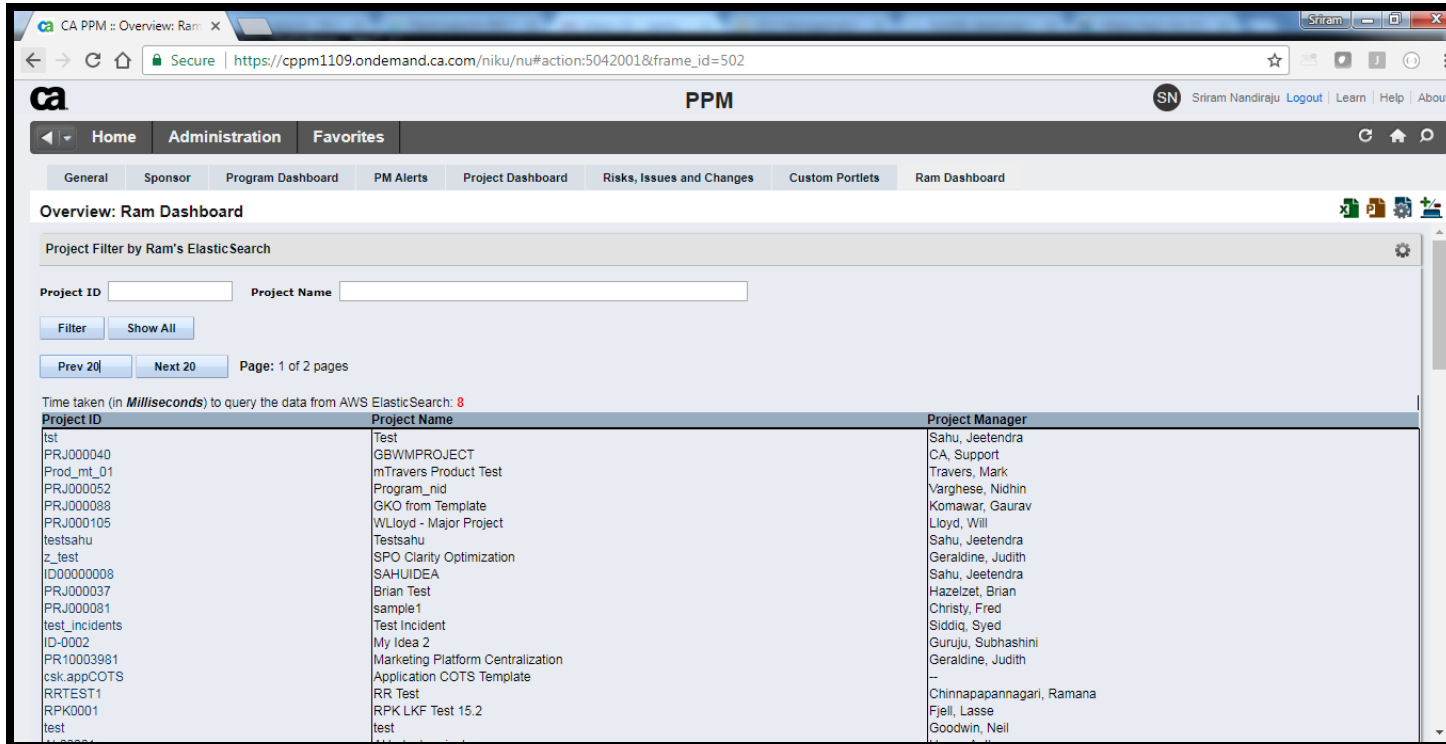
xmlhttp.open("GET", endpoint, false );
//xmlhttp.setRequestHeader("Access-Control-Allow-Headers", "*");
//xmlhttp.setRequestHeader("Access-Control-Allow-Origin", "*");
xmlhttp.send();
}
//alert("begin");

loadXMLDoc("");

</script> </body>
</html>
```


Demo

A simple search ALL fetches all the projects in 6 milliseconds.



Overview: Ram Dashboard

Project Filter by Ram's ElasticSearch

Project ID Project Name

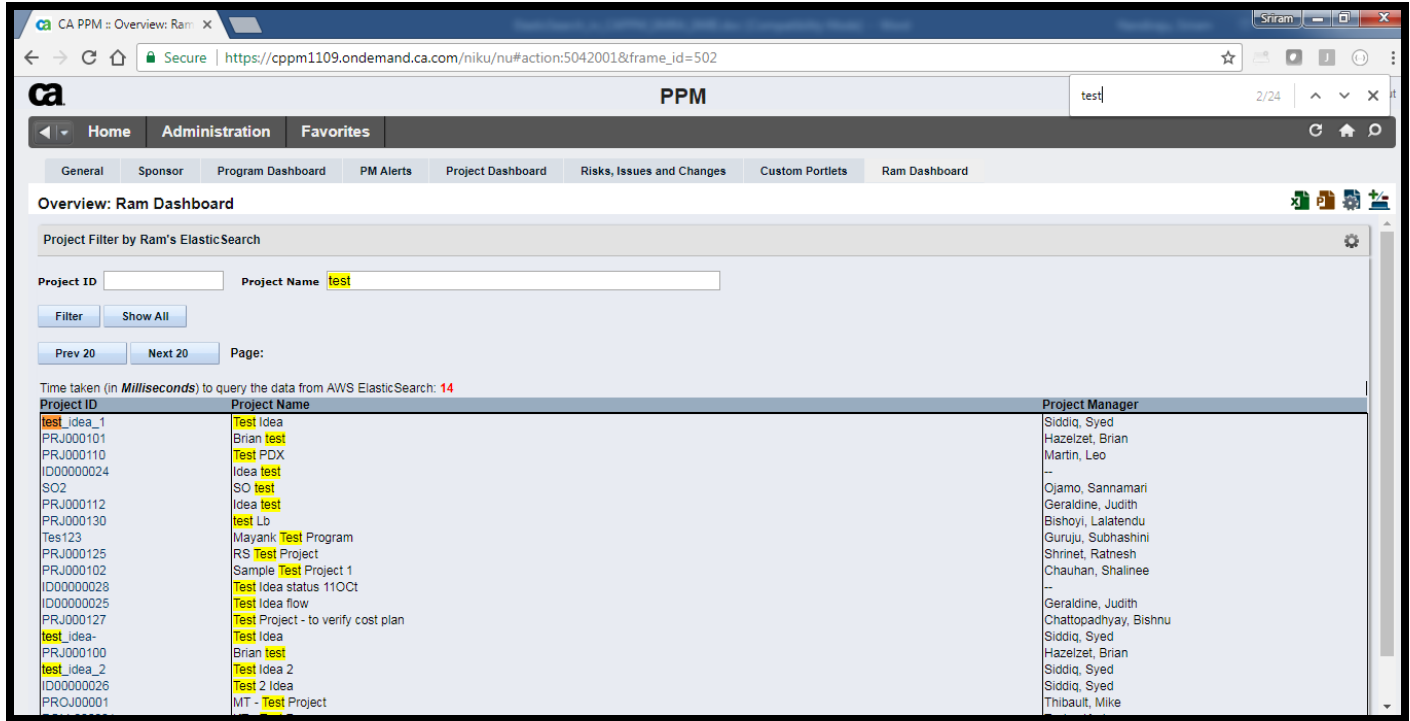
Filter Show All

Prev 20 Next 20 Page: 1 of 2 pages

Time taken (in **Milliseconds**) to query the data from AWS ElasticSearch: **8**

| Project ID | Project Name | Project Manager |
|----------------|-----------------------------------|-------------------------|
| tst | Test | Sahu, Jeetendra |
| PRJ000040 | GBWMPROJECT | CA, Support |
| Prod_mt_01 | mTravers Product Test | Travers, Mark |
| PRJ000052 | Program_nid | Varghese, Nidhin |
| PRJ000088 | GKO from Template | Komawar, Gaurav |
| PRJ000105 | WLloyd - Major Project | Lloyd, Will |
| testsahu | Testsahu | Sahu, Jeetendra |
| z_test | SPO Clarity Optimization | Geraldine, Judith |
| ID0000008 | SAHUJDEA | Sahu, Jeetendra |
| PRJ000037 | Brian Test | Hazelzet, Brian |
| PRJ000081 | sample1 | Christy, Fred |
| test_incidents | Test Incident | Siddiq, Syed |
| ID-0002 | My Idea 2 | Guruju, Subhashini |
| PR10003981 | Marketing Platform Centralization | Geraldine, Judith |
| csk.appCOTS | Application COTS Template | - |
| RRTEST1 | RR Test | Chinnapannagari, Ramana |
| RPK0001 | RPK LKF Test 15.2 | Fjell, Lasse |
| test | test | Goodwin, Neil |

Elasticsearch can even search substrings as well. See below.



The screenshot shows the CA PPM interface. At the top, there's a navigation bar with 'Home', 'Administration', and 'Favorites'. Below that, a secondary navigation bar includes 'General', 'Sponsor', 'Program Dashboard', 'PM Alerts', 'Project Dashboard', 'Risks, Issues and Changes', 'Custom Portlets', and 'Ram Dashboard'. The main content area is titled 'Overview: Ram Dashboard' and features a search filter for 'test'. Below the filter, there are buttons for 'Filter', 'Show All', 'Prev 20', and 'Next 20'. A message indicates the query took 14 milliseconds. The main data is presented in a table with three columns: Project ID, Project Name, and Project Manager.

| Project ID | Project Name | Project Manager |
|-------------|------------------------------------|-----------------------|
| test_idea_1 | Test Idea | Siddiq, Syed |
| PRJ000101 | Brian test | Hazelzet, Brian |
| PRJ000110 | Test PDX | Martin, Leo |
| ID0000024 | Idea test | -- |
| SO2 | SO test | Ojamo, Sannamari |
| PRJ000112 | Idea test | Geraldine, Judith |
| PRJ000130 | test Lb | Bishoyi, Lalatendu |
| Tes123 | Mayank Test Program | Guruji, Subhashini |
| PRJ000125 | RS Test Project | Shrinet, Ratnesh |
| PRJ000102 | Sample Test Project 1 | Chauhan, Shalinee |
| ID0000028 | Test Idea status 11OCT | -- |
| ID0000025 | Test Idea flow | Geraldine, Judith |
| PRJ000127 | Test Project - to verify cost plan | Chattopadhyay, Bishnu |
| test_idea- | Test Idea | Siddiq, Syed |
| PRJ000100 | Brian test | Hazelzet, Brian |
| test_idea_2 | Test Idea 2 | Siddiq, Syed |
| ID0000026 | Test 2 Idea | Siddiq, Syed |
| PROJ00001 | MT - Test Project | Thibault, Mike |

References

1. <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-16-04>
2. <https://discuss.elastic.co/>
3. <https://aws.amazon.com/ec2/faqs/>
4. <https://communities.ca.com/>

THE END