

Alpha Presentation

Force Platform Ingestion Tool

The Capstone Experience

Team Rook

Roy Barnes

Matt Hammerly

Will McGee

Chiyu Song

Mark Velez

Department of Computer Science and Engineering
Michigan State University

Spring 2017



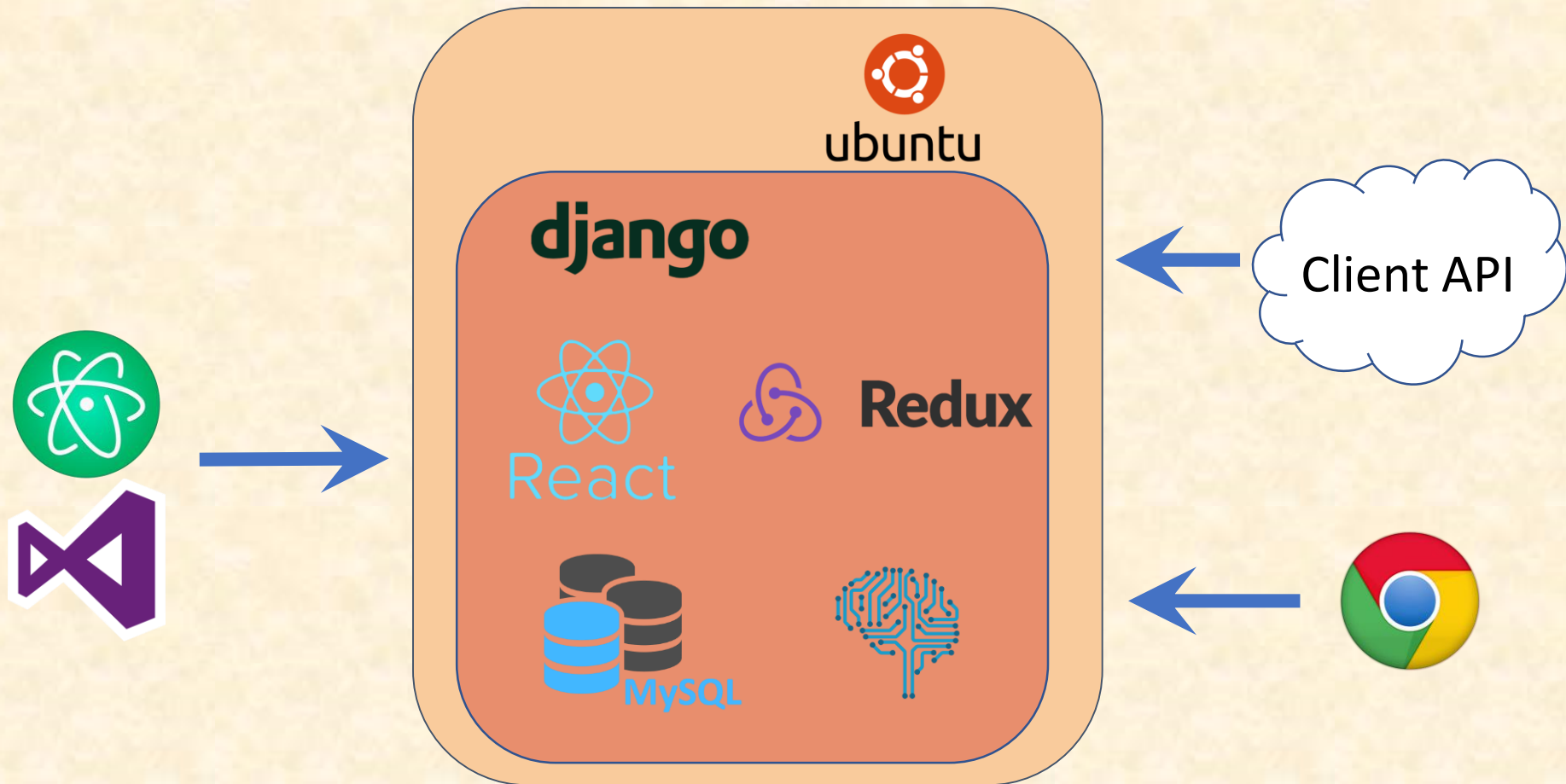
*From Students...
...to Professionals*

Project Overview

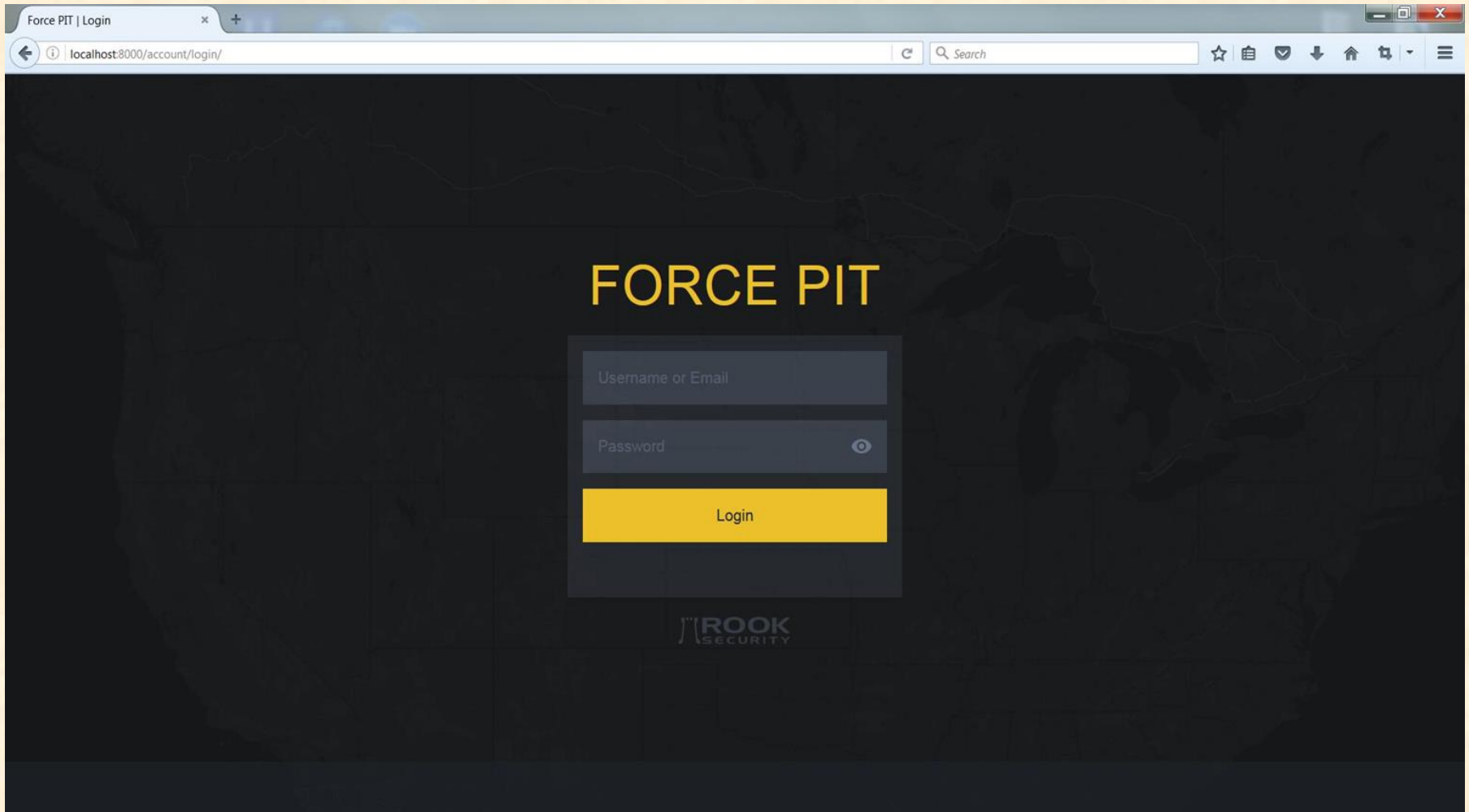
- Integration into Rook's new Force platform
- Enhance analyst efficiency in daily work
- Provide an easy way to integrate new clients
- Machine learning to improve alert correlation



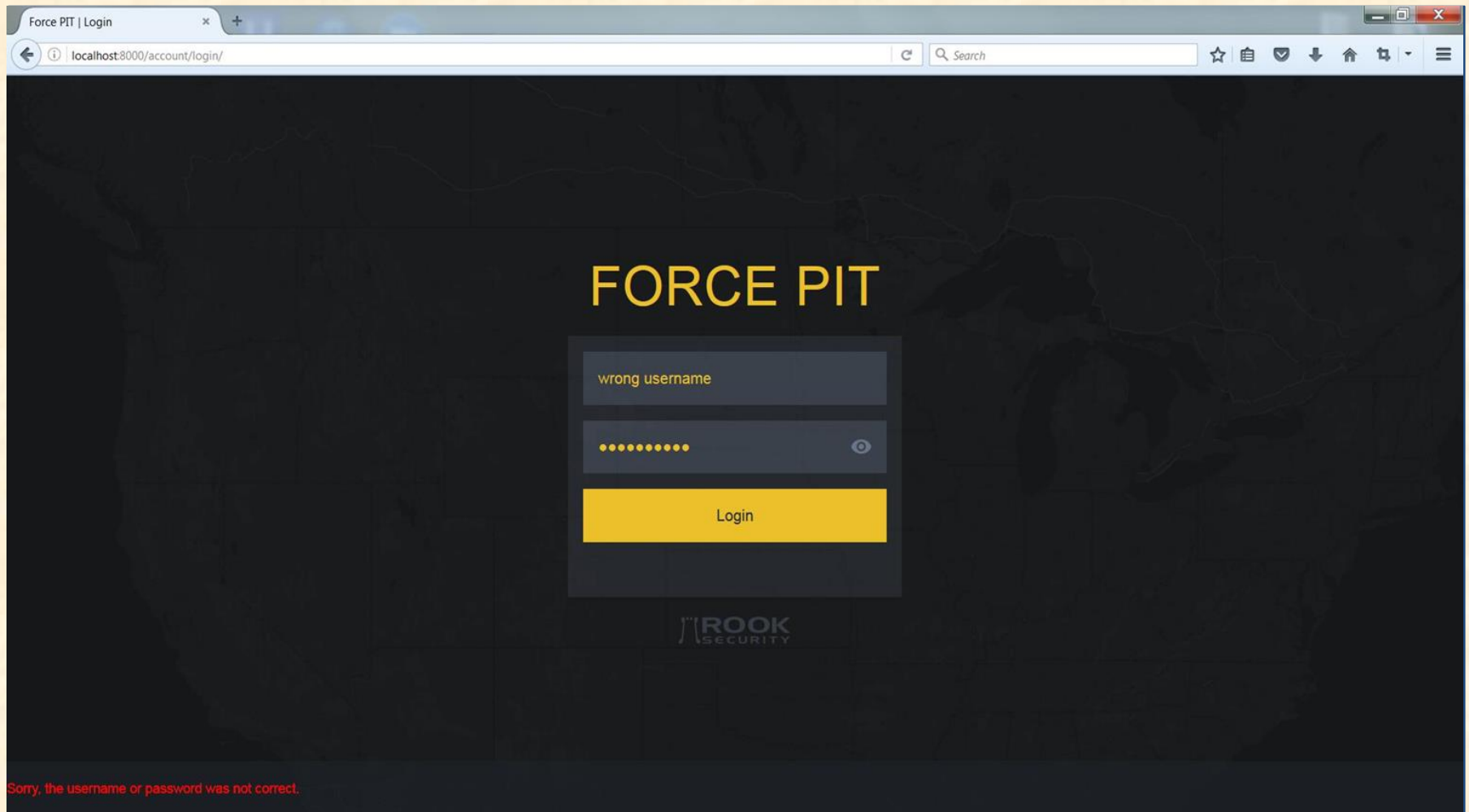
System Architecture



Login Page



Login Page - Error Message



Alerts Page

The screenshot shows a web browser window displaying the 'Force PIT | Alerts' page. The browser's address bar shows 'localhost:3000'. The page header features the 'ROOK SECURITY' logo on the left, the title 'FORCE PIT' in large yellow letters in the center, and a user greeting 'Hello, User!' on the right. Below the header, there are three yellow buttons: 'Edit Cases', 'Make Case', and 'Add to case'. A search bar with the placeholder text 'Search...' is located on the right side of the header area. The main content is a table of alerts, each row representing a different company. Each row includes a status indicator (five blue squares), the company name, an IP address, a timestamp, and two action icons (a tag with a plus sign and a vertical ellipsis).

Status	Company	IP Address	Timestamp	Actions
■■■■■	Company 1	62.14.233.98	2016-08-15 11:14	[Tag+] [More]
■■■■■	Company 2	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 3	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 4	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 5	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 6	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 7	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 8	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 9	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 10	test ip	TEST TIME	[Tag+] [More]
■■■■■	Company 11	test ip	TEST TIME	[Tag+] [More]



Alerts Page - Selected Alert

The screenshot shows a web browser window displaying the Force PIT Alerts page. The browser's address bar shows 'localhost:3000'. The page header includes the Rook Security logo, the title 'FORCE PIT', and a user greeting 'Hello, User!'. Below the header, there are three buttons: 'Edit Cases', 'Make Case', and 'Add to case'. A search bar is also present. The main content area displays a table of alerts. The first alert is selected and expanded to show detailed information.

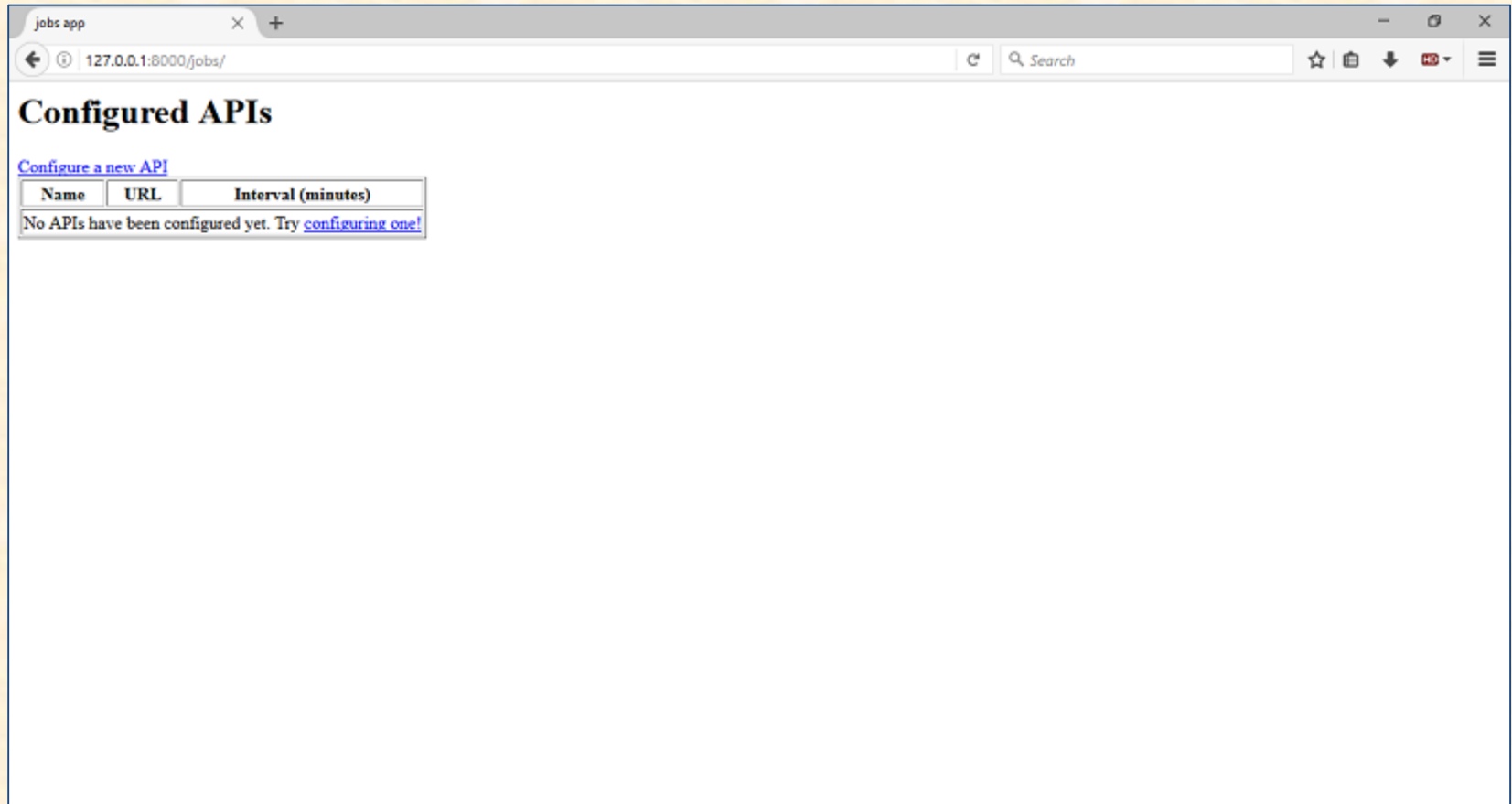
Company	IP Address	Timestamp	Actions
Company 1	62.14.233.98	2016-08-15 11:14	[Tag] [More]
Company 2	test ip	TEST TIME	[Tag] [More]
Company 3	test ip	TEST TIME	[Tag] [More]
Company 4	test ip	TEST TIME	[Tag] [More]
Company 5	test ip	TEST TIME	[Tag] [More]
Company 6	test ip	TEST TIME	[Tag] [More]

Selected Alert Details:

Attacker IP Address	Victim IP Address	Host Criticality	Payload
62.14.233.98	10.90.5.27	2	No Payload
Geolocation	Host Name	Asset Type	
Madrid, Madrid	web-srv-01	Financial	
Reputation	Host Information		
Unknown [3/5]	Windows 2008R2		



Configuring an API



jobs app

127.0.0.1:8000/jobs/

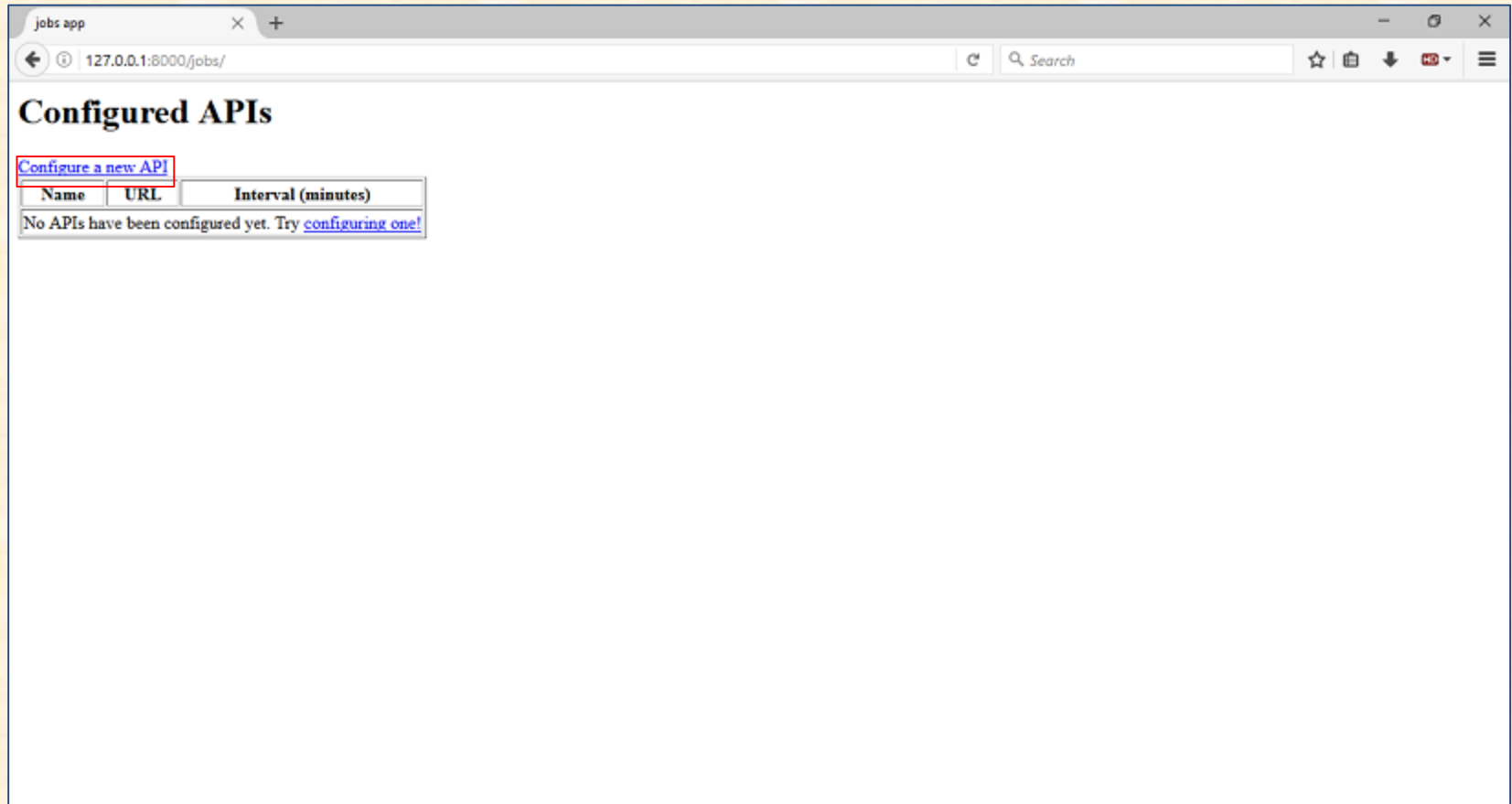
Configured APIs

[Configure a new API](#)

Name	URL	Interval (minutes)
No APIs have been configured yet. Try configuring one!		



Configuring an API



jobs app

127.0.0.1:8000/jobs/

Configured APIs

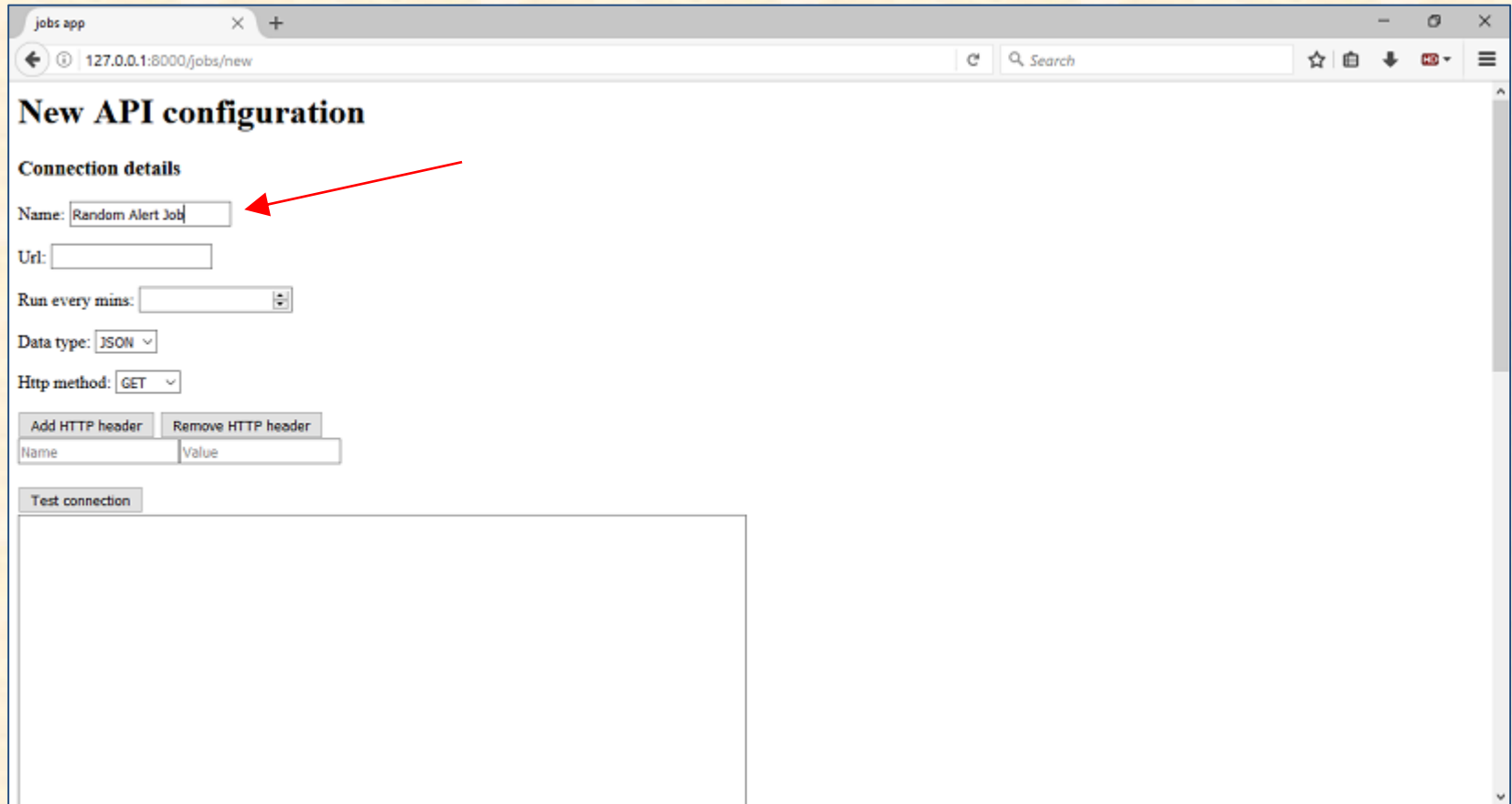
[Configure a new API](#)

Name	URL	Interval (minutes)
------	-----	--------------------

No APIs have been configured yet. Try [configuring one!](#)



Configuring an API



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/jobs/new". The page title is "New API configuration". The form is titled "Connection details" and contains the following fields and controls:

- Name:** A text input field containing "Random Alert Job", with a red arrow pointing to it.
- Url:** An empty text input field.
- Run every mins:** A dropdown menu.
- Data type:** A dropdown menu set to "JSON".
- Http method:** A dropdown menu set to "GET".
- HTTP Headers:** Two buttons, "Add HTTP header" and "Remove HTTP header", are positioned above a table with two columns: "Name" and "Value".
- Test connection:** A button above a large empty rectangular area.



Configuring an API

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/jobs/new'. The page title is 'New API configuration'. The form contains the following fields and controls:

- Name:** Text input field containing 'Random Alert Job'.
- Url:** Text input field containing '00/dotoparser/getmerak'. Two red arrows point to this field.
- Run every mins:** Spin box containing '1'. A red arrow points to this field.
- Data type:** Dropdown menu set to 'JSON'.
- Http method:** Dropdown menu set to 'GET'.
- HTTP Headers:** A section with 'Add HTTP header' and 'Remove HTTP header' buttons, and a table with 'Name' and 'Value' columns.
- Test connection:** A button above a large empty text area.



Configuring an API

jobs app

127.0.0.1:8000/jobs/new

New API configuration

Connection details

Name:

Url:

Run every mins:

Data type:

Http method:

Post data:

```
{
  "random": "true",
  "number": "1"
}
```

Name	Value
------	-------



Configuring an API

The screenshot shows a web browser window titled "jobs app" with the URL "127.0.0.1:8000/jobs/new". The interface includes a "Post data:" field, "Add HTTP header" and "Remove HTTP header" buttons, and a "Content-Type" dropdown menu set to "application/json". A "Test connection" button is highlighted with a red box. Below the button, a JSON response is displayed in a scrollable area, with a red arrow pointing to it. The JSON response contains details about a database event, including timestamps, categories, and source information.

```
{
  "_id": "AVnxugIF1zVXp0K1cf2m",
  "_index": "logstash-example_customer-2017.01.30",
  "_score": 1,
  "_source": {
    "@timestamp": "2017-01-30T17:07:05.000Z",
    "@version": "1",
    "Category": "General",
    "Channel": "Application",
    "EventID": 327,
    "EventReceivedTime": "2017-01-30 18:33:29",
    "EventType": "INFO",
    "Keywords": 36028797018963970,
    "Message": "sychohost (2744) The database engine detached a database (2,
C:\\Windows\\system32\\logfiles\\sum\\SystemIdentity.mdb). (Time=0 seconds) \\r\\n
\\r\\nInternal Timing Sequence: [1] 0.000, [2] 0.000, [3] 0.000, [4] 0.000, [5]
0.000, [6] 0.000, [7] 0.000, [8] 0.000, [9] 0.000, [10] 0.000, [11] 0.000, [12]
0.016. \\r\\nRevised Cache: 0",
    "Opcode": "Info",
    "ProcessID": 0,
    "RecordNumber": 78370,
    "SeverityValue": 2,
    "SourceModuleName": "eventlog",
    "SourceModuleType": "im_mavistalog",
    "SourceName": "ESENT",
    "Task": 1,
    "ThreadID": 0,
    "host": "10.222.195.227",
    "hostname": "bpg-hyper1.amag.local",
    "port": 58138,
    "zook_client": "example_customer",
  }
}
```

Data manning



Configuring an API

```
"@version": "1",
"Category": "General",
"Channel": "Application",
"EventID": 327,
"EventReceivedTime": "2017-01-30 18:33:29",
"EventType": "INFO",
"Keywords": 36028797018963970,
"Message": "sqlhost (2744) The database engine detached a database (2,
C:\\Windows\\system32\\LogFiles\\Sum\\SystemIdentity.mdb). (Time=0 seconds) \\r\\n
\\r\\nInternal Timing Sequence: [1] 0.000, [2] 0.000, [3] 0.000, [4] 0.000, [5]
0.000, [6] 0.000, [7] 0.000, [8] 0.000, [9] 0.000, [10] 0.000, [11] 0.000, [12]
0.016. \\r\\nRevived Cache: 0",
"Opcode": "Info",
"ProcessID": 0,
"RecordNumber": 78370,
"SeverityValue": 2,
"SourceModuleName": "eventlog",
"SourceModuleType": "in_mvistalog",
"SourceName": "ESENT",
"Task": 1,
"ThreadID": 0,
"host": "10.222.195.227",
"hostname": "bpg-hyper1.anaq.local",
"port": 58138,
"rook_client": "example_customer",
```

Data mapping

Add field mapping	Remove field mapping
HostIP	ry("_source").key("host")

Test mapping



Configuring an API

```
{}
{
  "Message": "svchost (2744) The database engine detached a database (2, C:\Windows\system32\logfiles\Sum\SystemIdentity.mdb). (Time=0 seconds) \r\n\r\nInternal Timing Sequence: [1] 0.000, [2] 0.000, [3] 0.000, [4] 0.000, [5] 0.000, [6] 0.000, [7] 0.000, [8] 0.000, [9] 0.000, [10] 0.000, [11] 0.000, [12] 0.016. \r\n\r\nRevived Cache: 0",
  "Opcode": "Info",
  "ProcessID": 0,
  "RecordNumber": 78370,
  "SeverityValue": 2,
  "SourceModuleName": "eventlog",
  "SourceModuleType": "im_msvistalog",
  "SourceName": "EVENT",
  "Task": 1,
  "ThreadID": 0,
  "host": "10.222.195.227",
  "hostname": "bpg-hyper1.smg.local",
  "port": 58138,
  "rook_client": "example_customer",
}
```

Data mapping

Add field mapping	Remove field mapping
HostIP	key("_source").key("host")
Description	key("_source").key("Mes")

Test mapping



Configuring an API

The screenshot shows a web browser window with the URL `127.0.0.1:8000/jobs/new`. The page title is "jobs app". The main content area is titled "Data mapping" and contains two sections:

Data mapping

Field	Mapping
HostIP	<code>!y("_source").key("host")</code>
Description	<code>key("_source").key("Mes</code>
AlertID	<code>key("_id")</code>

Test mapping

```
{
  "Description": "sychohost (2744) The database engine detached a database (2,
  Cr:\\Windows\\system32\\logfiles\\sum\\SystemIdentity.mdb). (Time=0 seconds) \\r\\n
  \\r\\nInternal Timing Sequence: [1] 0.000, [2] 0.000, [3] 0.000, [4] 0.000, [5]
  0.000, [6] 0.000, [7] 0.000, [8] 0.000, [9] 0.000, [10] 0.000, [11] 0.000, [12]
  0.016. \\r\\nRevised Cache: 0",
  "AlertID": "AVnxugIF1zVKp0X1of2m",
  "HostIP": "10.222.195.227"
}
```

A red arrow points to the "Description" field in the JSON output.



Configuring an API

jobs app

127.0.0.1:8000/jobs/new

HostIP	sy("_source").key("host")
Description	source().key("Message")
AlertID	key("_id")

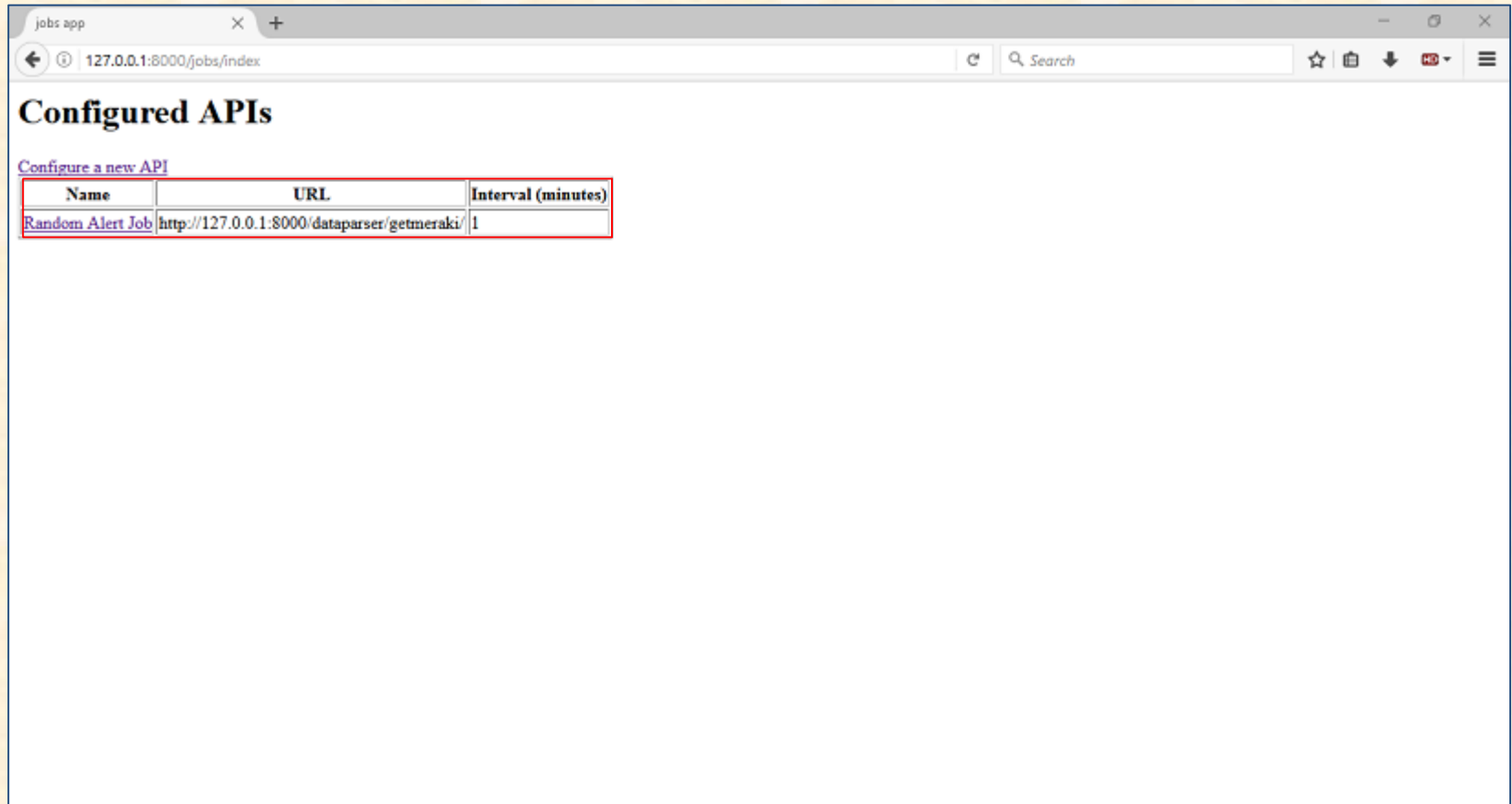
Test mapping

```
{
  "Description": "svchost (2744) The database engine detached a database (2,
  C:\\Windows\\system32\\logfiles\\sum\\SystemIdentity.mdb). (Time=0 seconds) \\r\\n
  \\r\\nInternal Timing Sequence: [1] 0.000, [2] 0.000, [3] 0.000, [4] 0.000, [5]
  0.000, [6] 0.000, [7] 0.000, [8] 0.000, [9] 0.000, [10] 0.000, [11] 0.000, [12]
  0.016. \\r\\nRevived Cache: 0",
  "AlertID": "AVnxugIF1zVXp0x1cf2m",
  "HostIP": "10.222.195.227"
}
```

Save



Configuring an API



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/jobs/index'. The page title is 'Configured APIs'. Below the title is a link 'Configure a new API'. A table is displayed with the following data:

Name	URL	Interval (minutes)
Random Alert Job	http://127.0.0.1:8000/dataparser/getneraki/	1



Configuring an API

jobs app

127.0.0.1:8000/jobs/results/1

[Back](#)

Data pulled by Random Alert Job (latest 0)

Normalizing data from <http://127.0.0.1:8000/dataparser/getmeraki/> every 1 minute(s).

Alert mapping:

```
{\"Description\": \"key(\\\"_source\\\") .key(\\\"Message\\\")\", \"AlertID\": \"key(\\\"_id\\\")\", \"HostIP\": \"key(\\\"_source\\\") .key(\\\"host\\\")\"}
```

Time Fetched	Raw	Normalized
This job has not been run yet.		



Configuring an API

jobs app

127.0.0.1:8000/jobs/results/1

Data pulled by Random Alert Job (latest 1)

Normalizing data from <http://127.0.0.1:8000/dataparser/getmeraki/> every 1 minute(s).

Alert mapping:

```
{"Description": "key(\"_source\").key(\"Message\")", "AlertID": "key(\"_id\")", "HostIP": "key(\"_source\").key(\"host\")"}
```

Time Fetched	Raw
Feb. 20, 2017, 9:38 p.m.	<pre>{ "_id": "AVnxugLdBSFinaeev4Q2", "_index": "logstash-example_customer-2017.01.30", "_score": 1, "_source": { "@timestamp": "2017-01-30T23:33:35.263Z", "@version": "1", "dst_ip": "98.138.199.240", "dst_port": 443, "host": "6.134.255.156", "hostname": "Toyota_Avon_Campus_MX100", "mac_addr": "00:18:0A:C7:F5:B5", "message": "<134> 1485819213.848837171 Toyota_Avon_Campus_MX100 flows src=10.231.112.60 dst=98.138.199.240 mac=00:18:0A:C7:F5:B5 protocol=top sport=63", "module": "flows", "pattern": "allow all", "protocol": "top", "rook_client": "example_customer", "src_ip": "10.231.112.60", "src_port": 63715, "timestamp": "2017-01-30T23:33:33.848Z", "type": "meraki" }, "_type": "meraki" }</pre>



Configuring an API

The screenshot shows a web browser window with the URL `127.0.0.1:8000/jobs/results/1`. The page title is "Data pulled by Random Alert Job (latest 2)". Below the title, there is a text block: "Normalizing data from http://127.0.0.1:8000/dataparser/getmeraki/ every 1 minute(s).".

Alert mapping:

```
{ "Description": "key(\"_source\").key(\"Message\")", "AlertID": "key(\"_id\")", "HostIP": "key(\"_source\").key(\"host\")" }
```

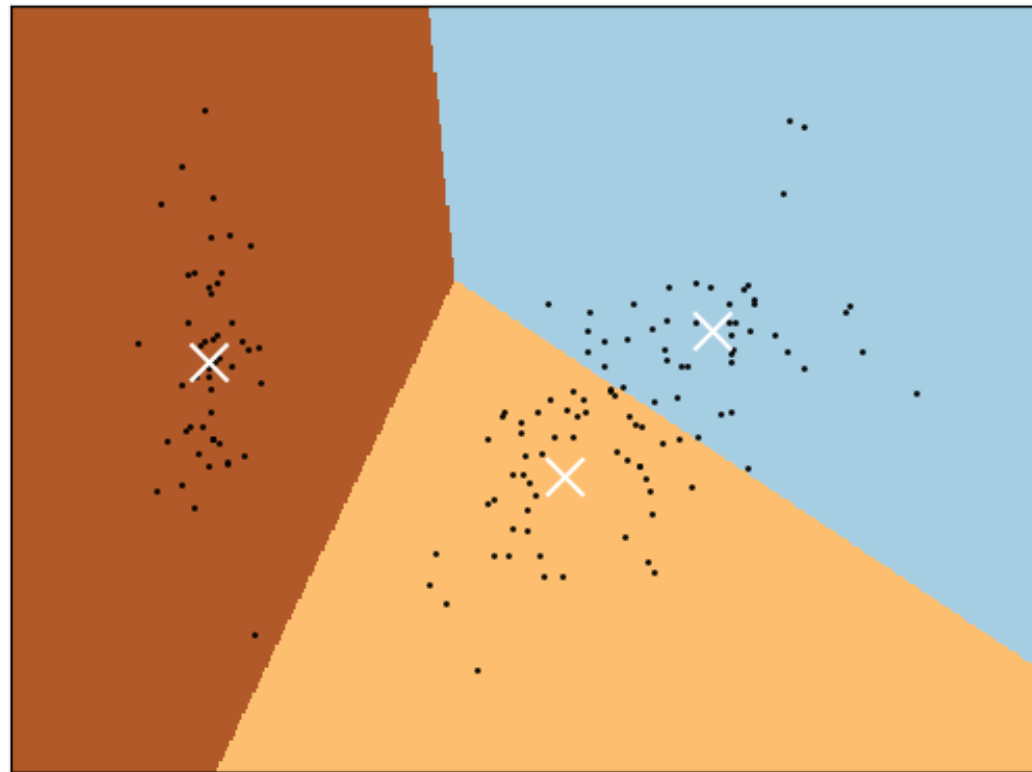
A table is displayed with the following data:

Time Fetched	
Feb. 20, 2017, 10:01 p.m.	<pre>{ "_id": "AVnxugIFlzVXp0K1of3q", "_index": "logstash-example_customer-2017.01.30", "_score": 1, "_source": { "@timestamp": "2017-01-30T15:40:42.000Z", "@version": "1", "Category": "Special Logon", "Channel": "Security", "EventID": 4672, "EventReceivedTime": "2017-01-30 18:33:30", "EventType": "AUDIT_SUCCESS", "Keywords": "-9214364837600035000", "Message": "Special privileges assigned to new logon.\r\n\r\nSubject:\r\n\r\nSecurity ID:\t\t\t5-1-5-21-2582712155-689091070-3522365680-500\r\n\r\nAccount Name: \r\n\r\nOpcode: "Info", "OpcodeValue": 0, "PrivilegeList": "SeSecurityPrivilege\r\n\r\n\t\t\tSeBackupPrivilege\r\n\r\n\t\t\tSeRestorePrivilege\r\n\r\n\t\t\tSeTakeOwnershipPrivilege\r\n\r\n\t\t\tSeDebugPrivilege", "ProcessID": 736, "ProviderGuid": "{54849628-5478-4994-A5BA-3E3B0328C30D}", "RecordNumber": 10063942, "SeverityValue": 2, "SourceModuleName": "eventlog", "SourceModuleType": "im_msvisualog", "SourceName": "Microsoft-Windows-Security-Auditing", "SubjectDomainName": "AMAG", "SubjectLogonId": "0x73a379faf9". } }</pre>



ML Clustering into Cases

Scikit-Learn Clustering on Test Data



What's left to do?

- Make UI design cohesive, get Rook feedback
- Use Django “Channels” library to update React/Redux UI in real time
- Finish ML, append to data normalization flow
- Build out support for as many APIs as we can



Questions?

?

?

?

?

?

?

?

?

?

