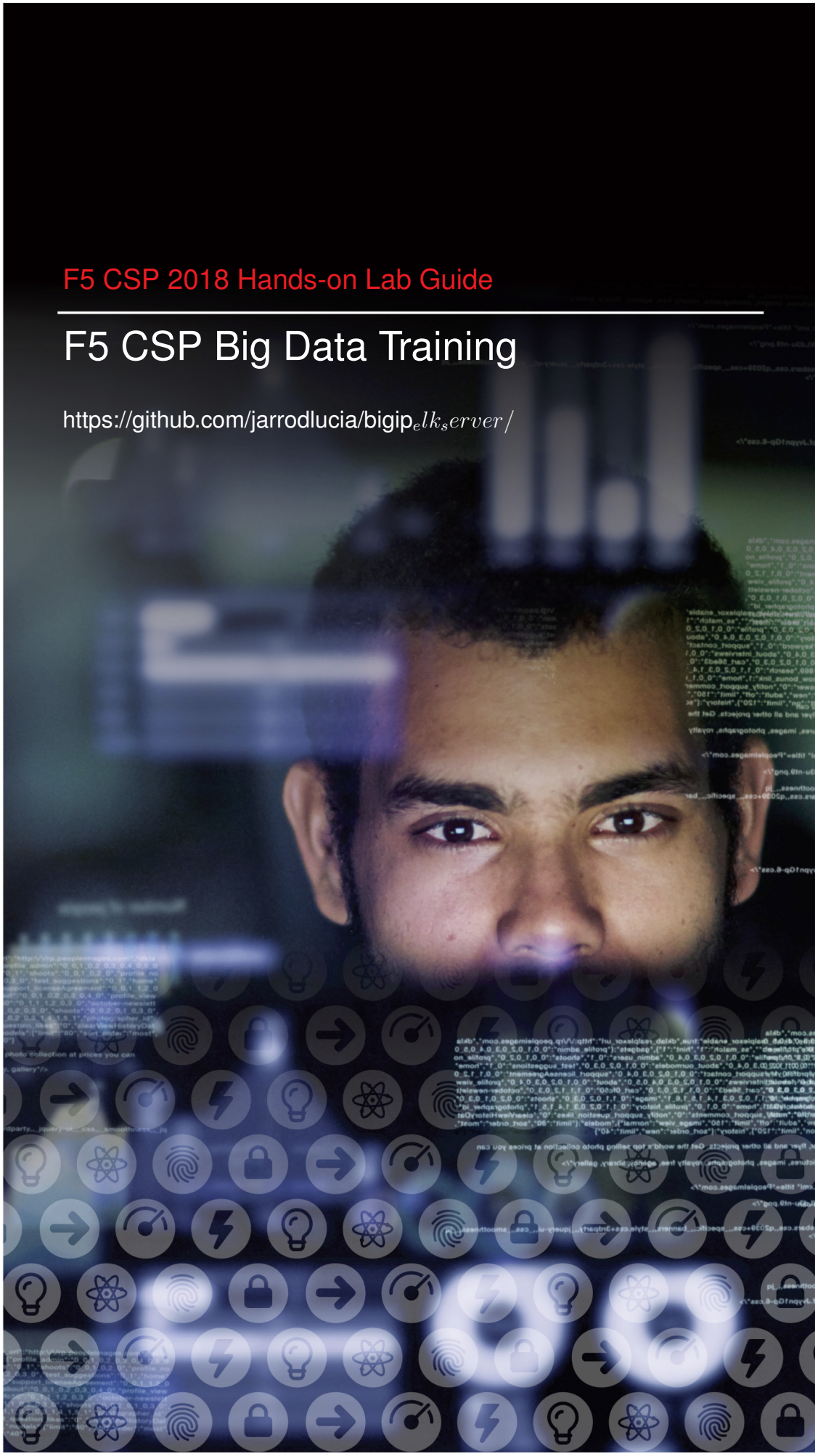
F5 CSP 2018 Hands-on Lab Guide

F5 CSP Big Data Training

https://github.com/jarrodlucia/bigip$_{elk_server}$/



Contents:

1 Welcome

Welcome to F5's Service Provider AVR and Big Data hands on training series. The intended audience for these labs are Service Provider engineers that would like to leverage the power of F5 data visibility and integrate this immense data capability into open source tools such as Elasticsearch, Hadoop and others.

2 Getting Started

Please follow the instructions provided by this documentation to start your lab and access your lab.

Note: All work for this lab will be performed exclusively from the Linux Jumphost and Linux Client Machines. All required access and servies needed to perform classes and labs are provided by the UDF. No installation or interaction with your local system is required.

---

3 Prerequisites

In order to complete this series of training classes you will need to utilize the provided blueprint for the course session. To access the UDF sessions you will need to have the following prerequists met.

- Current Access to UDF
- SSH key of your access machine in UDF
- Windows or MAC ssh client working with UDF

All pre-built environments implement the lab-topology shown below.

3.1 UDF Blueprint

Please follow the instructions provided by your lab instructor to access your lab environment. The lab environment will be delivered via UDF blueprints to each student.

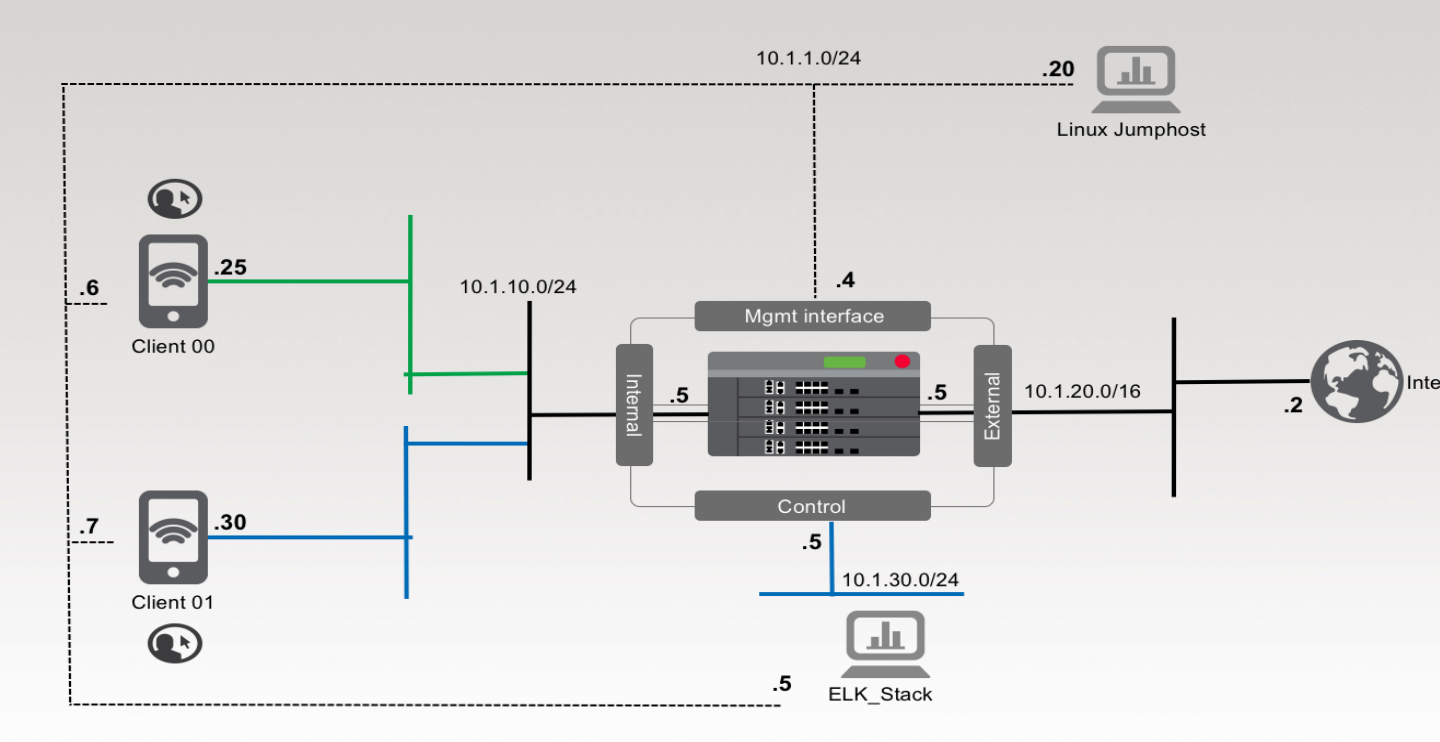
Note: Please deploy and start your lab as soon as you have access to the class as the lab takes some time to boot all the components.

---

4 Lab Topology

The network topology implemented for this lab based on the Service Provider Gi lan path. The focus of the lab is Control Plane programmability and Data Plane elements, so this lab will focus at both parts at different time. The following components have been included in your lab environment:

- 1 x `F5 BIG-IP VE` (v13.0 HF2)
- 1 x Linux Jumphost (ubuntu 16.04 - mate)
- 2 x Linux Clients (ubuntu 16.04 - mate)
- 1 x Linux Server (ubuntu 16.04)



The following table lists VLANS, IP Addresses and Credentials for all components:

Table 4.1: Lab Network Information

Component	VLAN	IP Address	Credentials
Linux Jumphost	Mgmt	10.1.1.20	
BIG-IP	Mgmt	10.1.1.4	admin/admin
	Internal	10.1.10.5	
	External	10.1.20.5	
	Control	10.1.30.5	
Client 00	Mgmt	10.1.1.9	udfclient/S3rv1ceP0weR
	Internal	10.1.10.25	
Client 01	Mgmt	10.1.1.7	udfclient/S3rv1ceP0weR
	Internal	10.1.10.30	
ELK Stack	Mgmt	10.1.1.5	ubuntu/default
	Control	10.1.30.15	

5

Class 1: BIG-IP AVR (BIG-IP Goodness)

This class covers the following topics:

- Module 1
  - REST API Basics
- Module 2
  - F5 BIG-IP AVR
  - Configuring AVR
  - Navigating AVR
  - Modify AVR Reports

Expected time to complete: **30 mins**

5.1 Module 1: REST API Basics

In this module you will learn the basic concepts required to interact with the BIG-IP iControl REST API. Additionally, you will walk through a typical Device navigation.

This is a cut down version of the F5 Programmibility Super Net Ops training.

Note: The Lab Deployment for this lab includes a single BIG-IP devices. For most of the labs we will configuring the BIG-IP device (management IP and licensing have already been completed).

Note: It's beneficial to have GUI/SSH sessions open to BIG-IP devices while going through this lab. Feel free to verify the actions taken in the lab against the GUI or SSH. You can also watch the following logs:

- BIG-IP:
  - /var/log/ltm
  - /var/log/restjavad.0.log

5.1.1 Lab 1.1: Exploring the iControl REST API

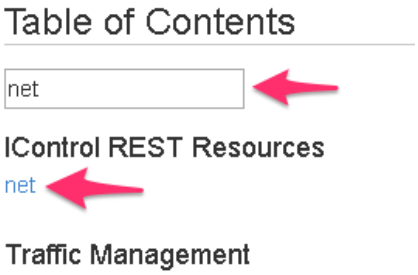
Task 1 – Explore the API using the TMOS Web Interface

In this lab we will explore the API using an interface that is built-in to TMOS. This utility is useful for understanding how TMOS objects map to the REST API. The interfaces implement full Create, Read, Update and Delete (CRUD) functionality, however, in most practical use cases it's far easier to use this interface as a 'Read' tool rather than trying to Create objects directly from it. It's usually far easier to use TMUI or TMSH to create the object as needed and then use this tool to view the created object with all the correct attributes already populated.

1. Open Google Chrome and navigate to the following bookmarks: BIG-IP

   GUI. Bypass any SSL errors that appear and ensure you see the login screen for each bookmark.

1. Navigate to the URL https://10.1.1.4/mgmt/toc (or click the BIG-IP REST TOC bookmark). The '/mgmt/toc' path in the URL is available on all TMOS versions 11.6 or newer.
2. Authenticate to the interface using the default admin/admin credentials.
3. You will now be presented with a top-level list of various REST resources. At the top of the page there is a search box filter worker... that can be used to find items on the page. Type 'net' in the search box



   and then click on the 'net' link under iControl REST Resources:

4. Find the `/mgmt/tm/net/route-domain` **Collection** and click it.
5. You will now see a listing of the **Resources** that are part of the route-domain(s) collection. As you can see the default route domain of `0` is listed. You can also create new objects by clicking the + button. Additionally resources can be deleted using the button or edited using the button.
6. Click the `0` resource to view the attributes of route-domain 0 on the device:

/mgmt/tm/net/route-domain/~Common~0

name	0
partition	Common
fullPath	/Common/0
connectionLimit	0
id	0
strict	enabled

Take note of the full path to the resource. Here is how the path is broken down:

```
/ mgmt / tm / net / route-domain / ~Common~0
| Root | OC | OC  |  Collection  | Resource
*OC=Organizing Collection
```

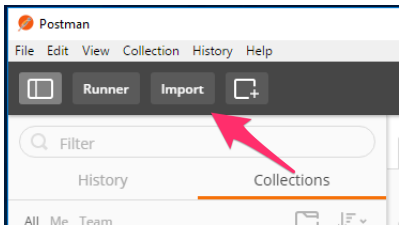
5.1.2 Lab 1.2: REST API Authentication & 'example' Templates

One of the many basic concepts related to interaction with REST API's is how a particular consumer is authenticated to the system. BIG-IP supports two types of authentication: HTTP BASIC and Token based. It's important to understand both of these authentication mechanisms, as consumers of the API will often make use of both types depending on the use case. This lab will demonstrate how to interact with both types of authentication.

Task 1 - Import the Postman Collection & Environment

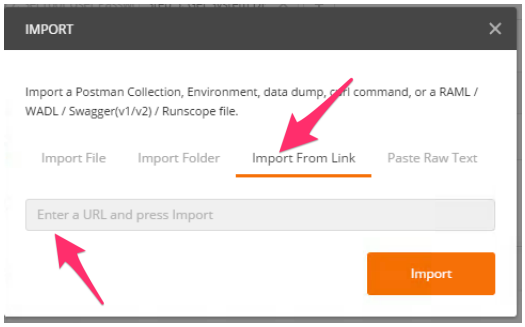
In this task you will Import a Postman Collection & Environment for this lab. Perform the following steps to complete this task:

1. Open the Postman tool by clicking the  icon of the desktop of your Linux Jumphost
2. Click the 'Import' button in the top left of the Postman window

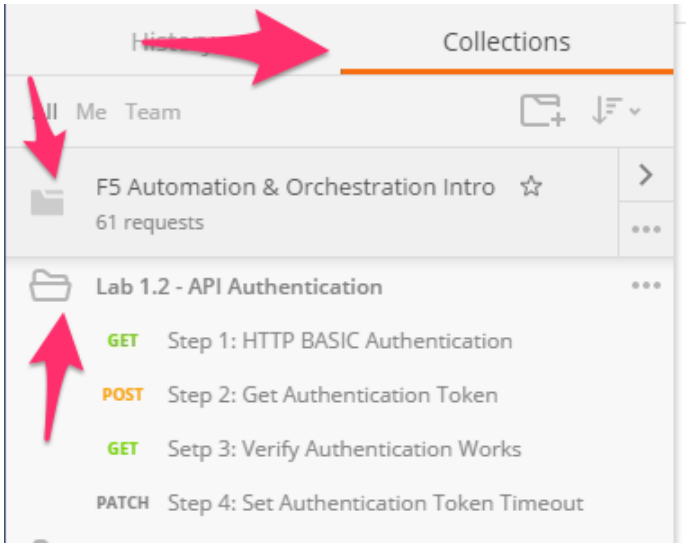


3. Click the 'Import from Link' tab. Paste the following URL into the text box and click 'Import'

   https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/F5_Automation_Orchestration_Intro.postman_collection.json



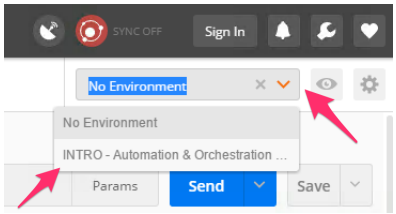
4. You should now see a collection named 'F5 Automation & Orchestration Intro' in your Postman Collections sidebar:



5. Import the Environment file by clicking 'Import' -> 'Import from Link' and pasting the following URL and clicking 'Import':

   `https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/INTRO_Automation_Orchestration_Lab.postman_environment.json`

6. To assist in multi-step procedures we make heavy use of the 'Environments' capability in Postman. This capability allows us to set various global variables that are then substituted into a request before it's sent. Set your environment to 'INTRO - Automation & Orchestration Lab' by using the menu at the top right of your Postman window:



Task 2 – HTTP BASIC Authentication

In this task we will use the Postman tool to send API requests using HTTP BASIC authentication. As its name implies this method of authentication encodes the user credentials via the existing BASIC authentica-

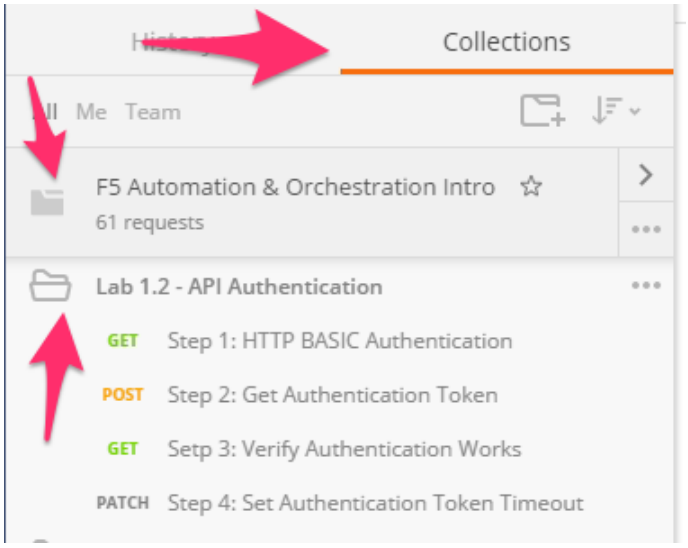
tion method provided by the HTTP protocol. The mechanism this method uses is to insert an HTTP header named 'Authorization' with a value that is built by Base 64 encoding the string `<username>:<password>`. The resulting header takes this form:

```
Authorization:  Basic YWRtaW46YWRtaW4=
```

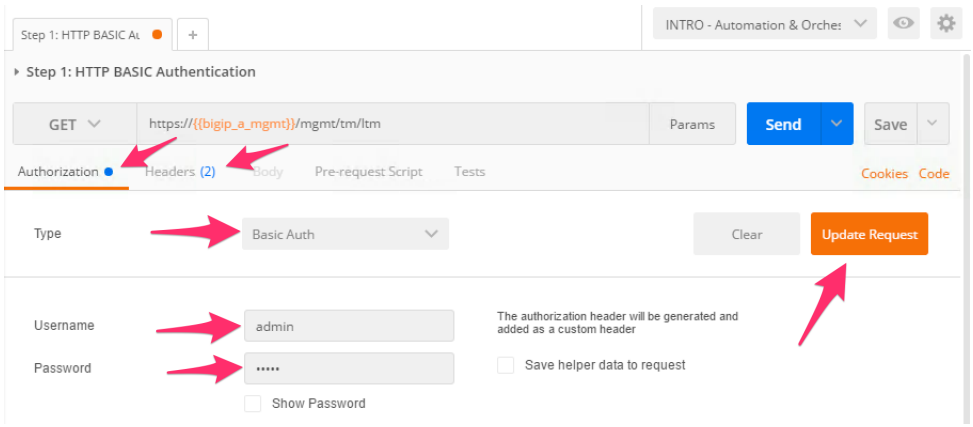
It should be noted that cracking the method of authentication is TRIVIAL; as a result API calls should always be performed using HTTPS (F5 default) rather than HTTP.

Perform the following steps to complete this task:

1. Click the 'Collections' tab on the left side of the screen, expand the 'F5 Automation & Orchestration Intro' collection on the left side of the screen, expand the 'Lab 1.2 – API Authentication' folder:



2. Click the 'Step 1: HTTP BASIC Authentication' item. Click the 'Authorization' tab and select 'Basic Auth' as the Type. Fill in the username and password (admin/admin) and click the 'Update Request' button. Notice that the number of Headers in the Headers tab changed from 1 to 2. This is because Postman automatically created the HTTP header and updated your request to include it. Click the 'Headers' tab and examine the HTTP header:



3. Click the 'Send' button to send the request. If the request succeeds you should be presented with a listing of the `/mgmt/tm/ltm` Organizing Collection.

4. Update the credentials and specify an INCORRECT password. Send the request again and examine the response:

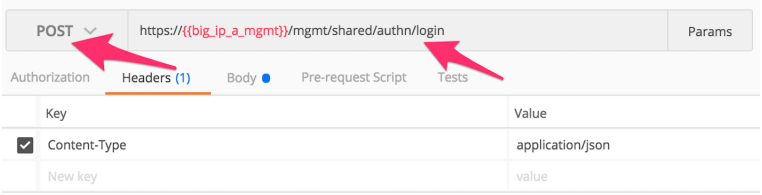
Task 3 – Token Based Authentication

One of the disadvantages of BASIC Authentication is that credentials are sent with each and every request. This can result in a much greater attack surface being exposed unnecessarily. As a result Token Based Authentication (TBA) is preferred in many cases. This method only sends the credentials once, on the first request. The system then responds with a unique token for that session and the consumer then uses that token for all subsequent requests. Both BIG-IP and iWorkflow support token-based authentication that drops down to the underlying authentication subsystems available in TMOS. As a result the system can be configured to support external authentication providers (RADIUS, TACACS, AD, etc) and those authentication methods can flow through to the REST API. In this task we will demonstrate TBA using the local authentication database, however, authentication to external providers is fully supported.

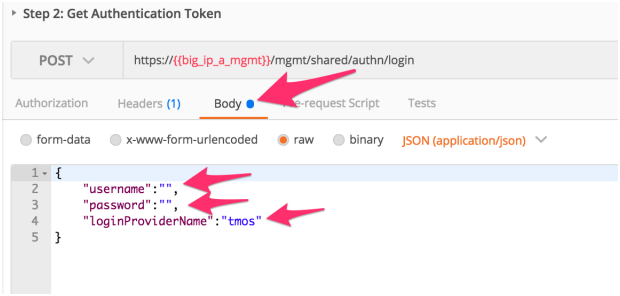
For more information about external authentication providers see the section titled "**About external authentication providers with iControl REST**" in the iControl REST API User Guide available at https://devcentral.f5.com

Perform the following steps to complete this task:

1. Click the 'Step 2: Get Authentication Token' item in the Lab 1.2 Postman Collection
2. Notice that we send a POST request to the `/mgmt/shared/authn/login` endpoint.



3. Click the 'Body' tab and examine the JSON that we will send to BIG-IP to provide credentials and the authentication provider:



4. Modify the JSON body and add the required credentials (admin/admin). Then click the 'Send' button.
5. Examine the response status code. If authentication succeeded and a token was generated the response will have a 200 OK status code. If the status code is 401 then check your credentials:

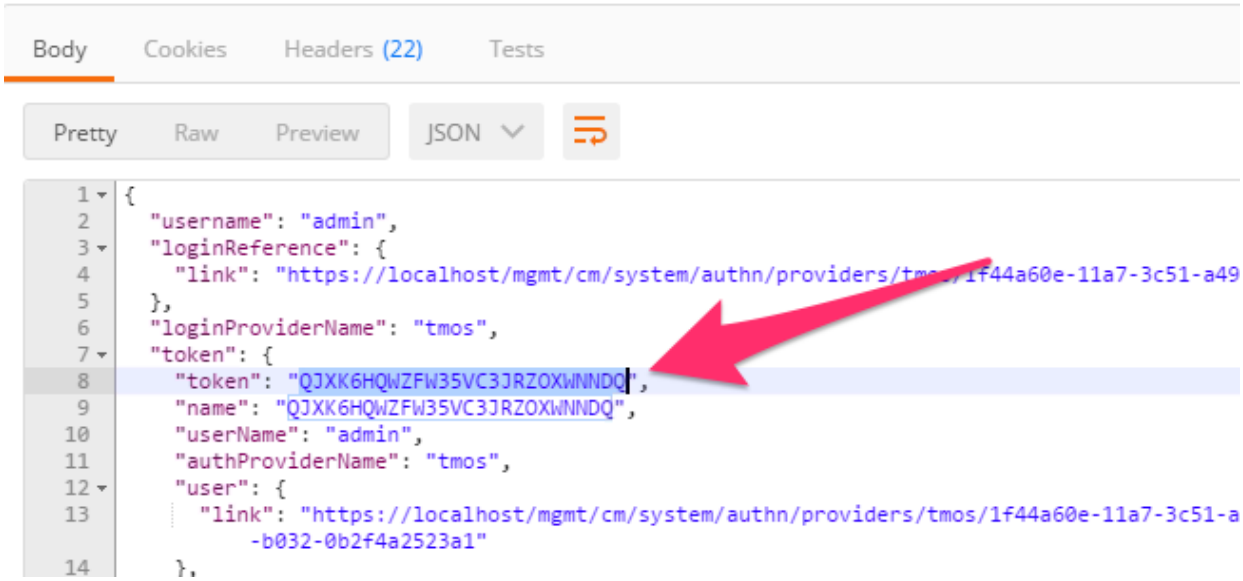
   Successful:

- 

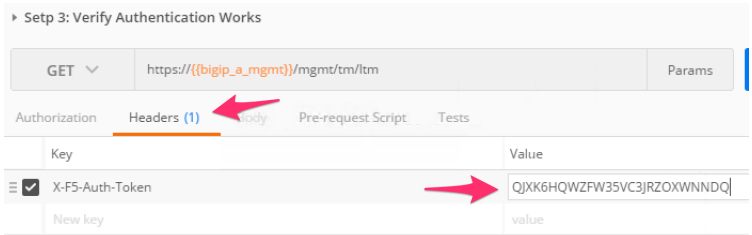
Unsuccessful:

- 

6. Once you receive a 200 OK status code examine the response body. The various attributes show the parameters assigned to the particular token. Find the 'token' attribute and copy it into your clipboard (Ctrl+c) for use in the next step:

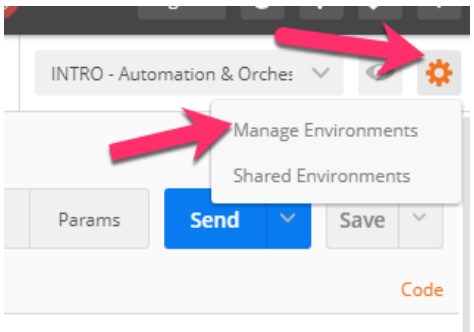


7. Click the 'Step 3: Verify Authentication Works' item in the Lab 1.2 Postman collection. Click the 'Headers' tab and paste the token value copied above as the VALUE for the `X-F5-Auth-Token` header. This header is required to be sent on all requests when using token based authentication.

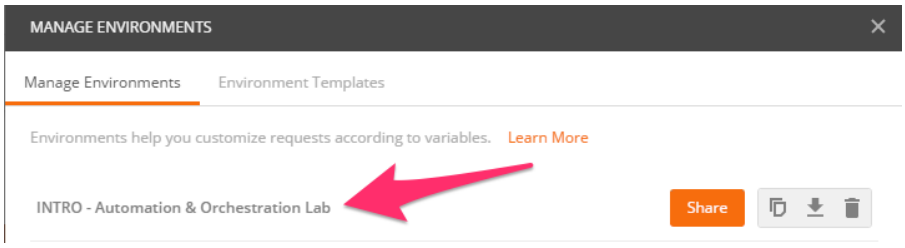


8. Click the 'Send' button. If your request is successful you should see a '200 OK' status and a listing of the `ltm` Organizing Collection.

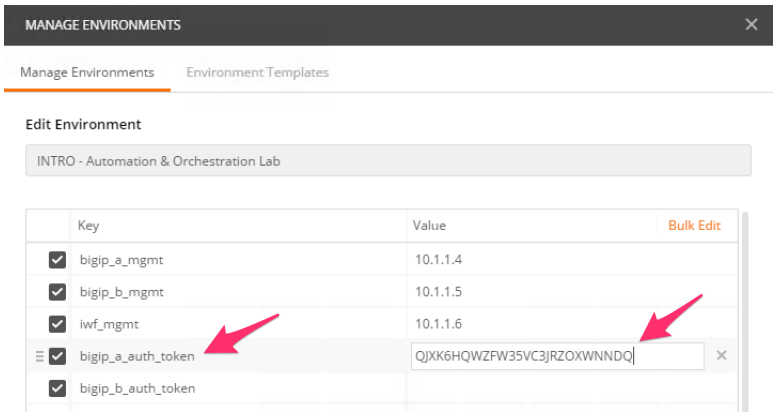
9. We will now update your Postman environment to use this auth token for the remainder of the lab. Click the Environment menu in the top right of the Postman window and click 'Manage Environments':



10. Click the 'INTRO – Automation & Orchestration Lab' item:

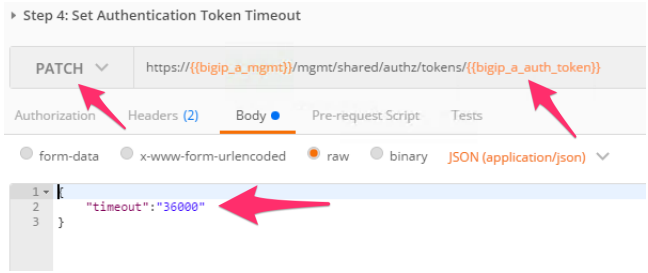


11. Update the value for `bigip_a_auth_token` by Pasting (Ctrl-v) in your auth token:



12. Click the 'Update' button and then close the 'Manage Environments' window. Your subsequent requests will now automatically include the token.

13. Click the 'Step 4: Set Authentication Token Timeout' item in the Lab 1.2 Postman collection. This request will PATCH your token Resource (check the URI) and update the timeout attribute so we can complete the lab easily. Examine the request type and JSON Body and then click the 'Send' button. Verify that the timeout has been changed to '36000' in the response:

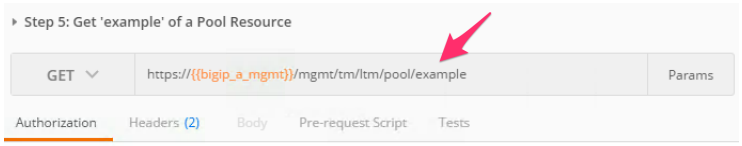


Task 4 – Get a pool 'example' Template

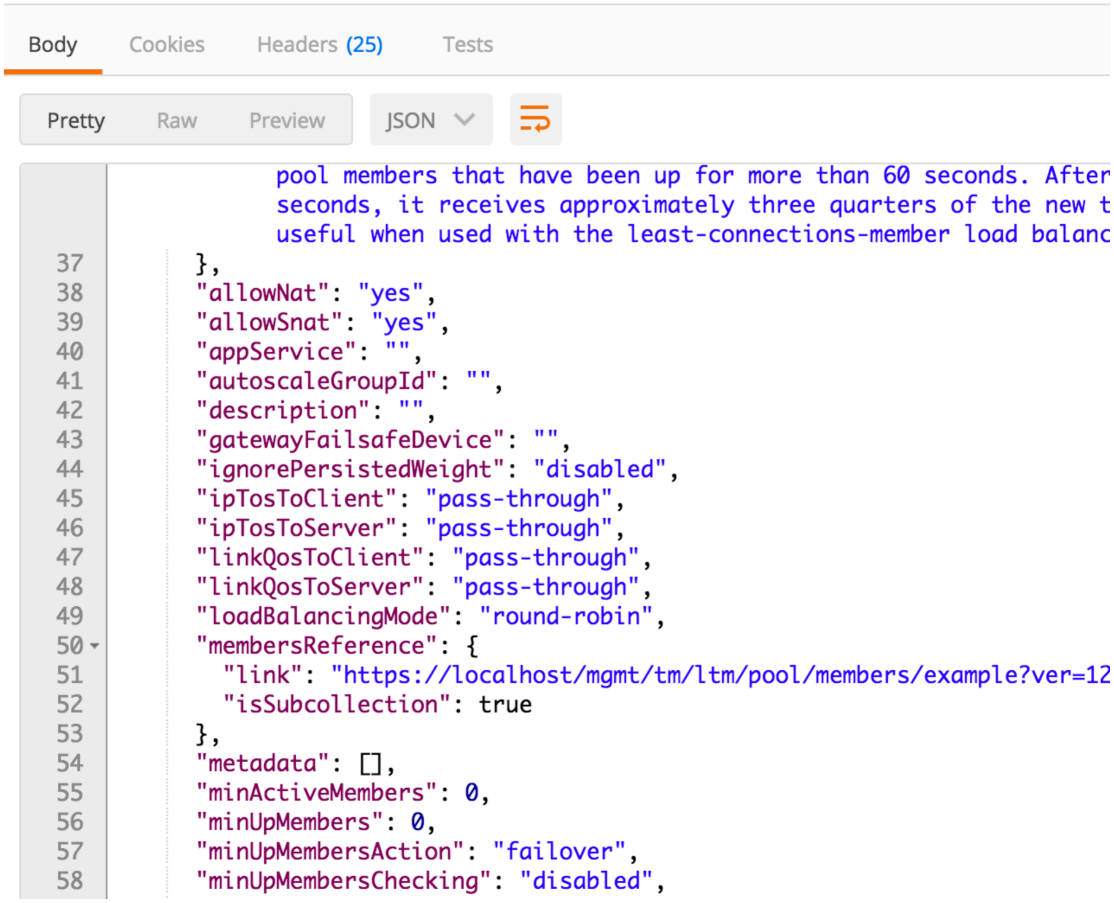
In order to assist with REST API interactions you can request a template of the various attributes of a Resource type in a Collection. This template can then be used as the body of a POST, PUT or PATCH request as needed.

Perform the following steps:

1. Click the 'Step 5: Get 'example' of a Pool Resource' item in the Lab 1.2 Postman collection
2. Examine the URI. Notice the addition of example at the end of the collection name:



3. Click 'Send' and examine the FULL response. You will see descriptions and then all the attributes for the Pool resource type. The response also shows the default values for the attributes if applicable:



5.2 Module 2: F5 Application Visibility and Reporting

In this module we will explore how to configure and use F5's Application Visibility and Reporting to provide application reporting. Analytics (also called Application Visibility and Reporting) is a module on the BIG-IP® system that you can use to analyze the performance of services and applications. It provides detailed

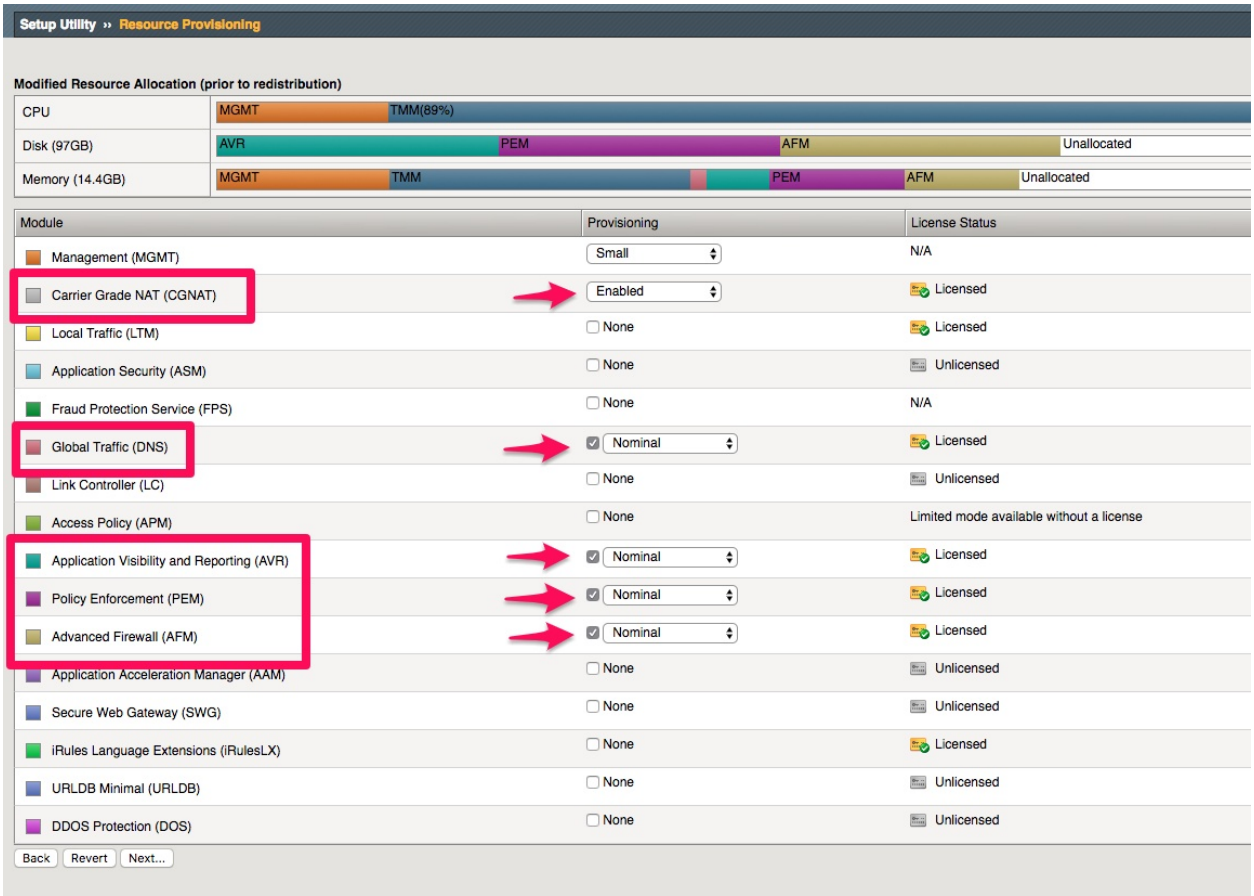
metrics such as transactions per second, server and client latency, request and response throughput, and sessions. You can view metrics for applications, virtual servers, pool members, URLs, specific countries, and additional detailed statistics about application traffic running through the BIG-IP system.

The labs in the module will focus on the high level features of AVR. These will include Analytics profiles, configuration and navigation of the AVR reports and information generated.

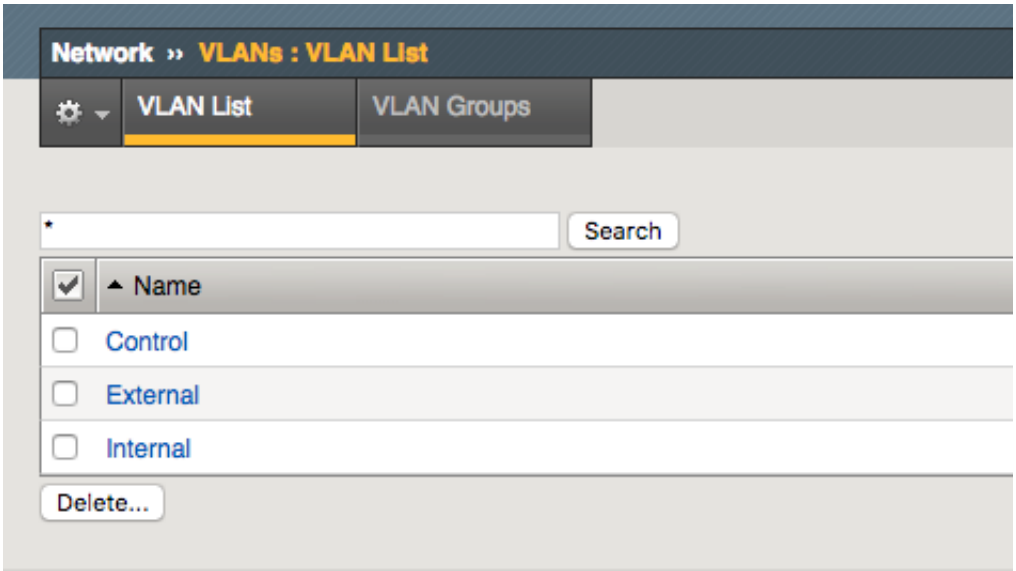
The BIG-IP in the lab is preconfigured with DNS / PEM / and AFM provisioned and configured. Please explore the current f5 config to familarise yourself with this lab.

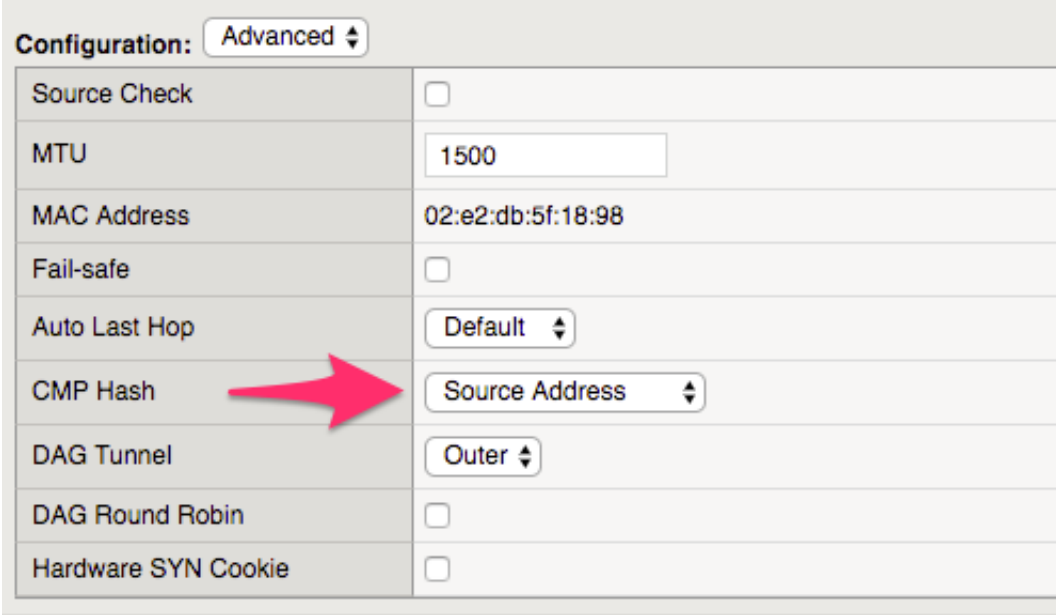
Confirm the following main config items to verify your BIG-IP lab is on working order:

1. Checked Provisioned Modules refelcts the below image (DNS / PEM / AFM and AVR).

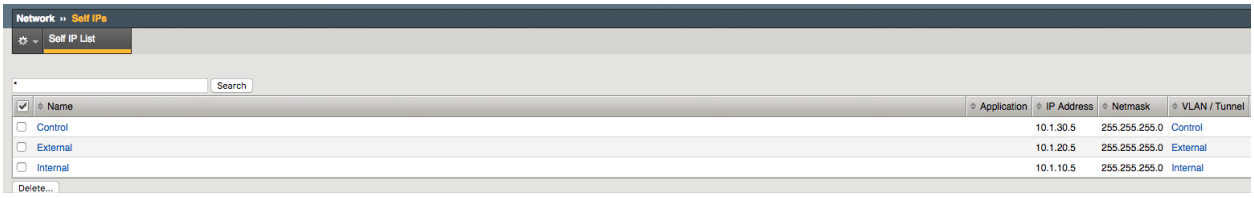


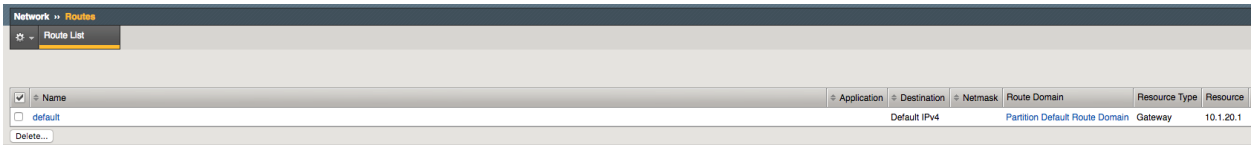
1. Check VLAN setup. Make sure Interval VLAN is set to source (SP DAG).



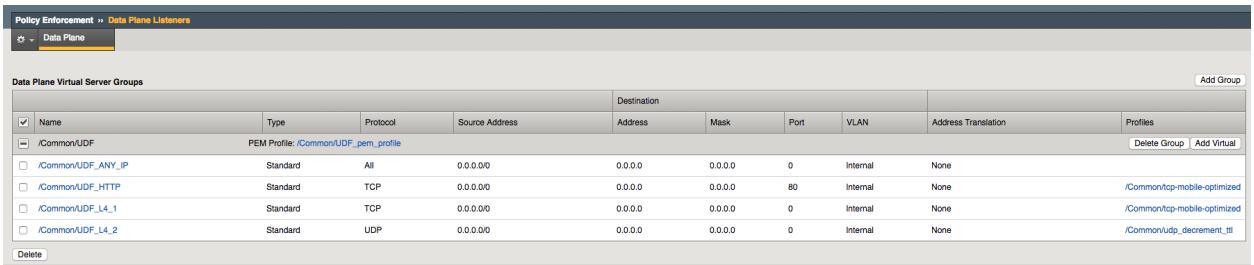


1. Verify SELF-IP's and routes are present.

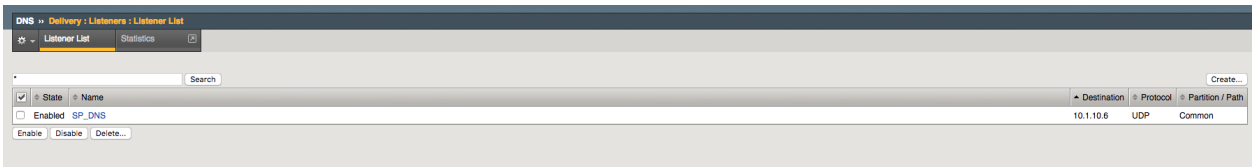




1. Check that PeM Data plane is setup, you should see four PEM data plane VS as below.



1. Check that DNS Listener is configured.



Note: Explore the rest of the configuration. Please look at the DNS setup (cache / monitor) and AFM CGNAT (NAPT) configurations.

5.2.1 Lab 2.1: Configure AVR Profiles

An Analytics profile is a set of definitions that determines the circumstances under which the system gathers, logs, notifies, and graphically displays information regarding traffic to an application or service. The Analytics module requires that you select an Analytics profile for each application you want to monitor. You associate the Analytics profile with one or more virtual servers used by the application / service.

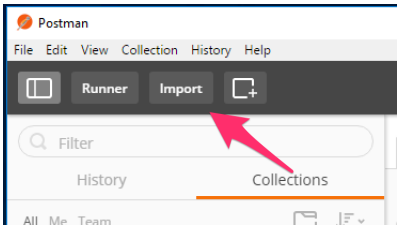
In the Analytics profile, you customize:

- What statistics to collect
- Where to collect data (locally, remotely, or both)
- Whether to capture the traffic itself
- Whether to send notifications.

Task 1 - Import the Postman Collection & Environment

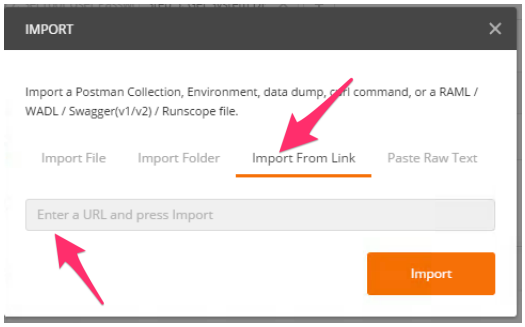
In this task you will Import a Postman Collection & Environment for this lab. Perform the following steps to complete this task:

1. Open the Postman tool by clicking the _____ icon of the desktop of your Linux Jumphost (Postman should be open from previous Lab)
2. Click the 'Import' button in the top left of the Postman window

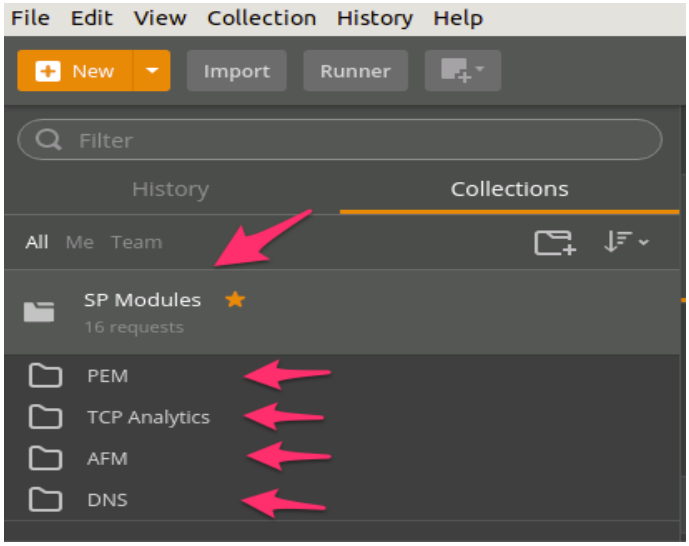


1. Click the 'Import from Link' tab. Paste the following URL into the text box and click 'Import'

   `https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/SPModules.postman_collection.json`

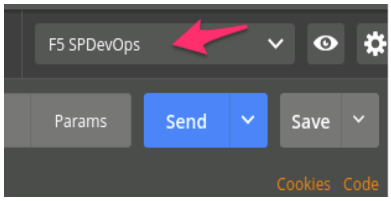


1. You should now see a collection named 'SP Modules' in your Postman Collections sidebar:



2. Import the Environment file by clicking 'Import' -> 'Import from Link' and pasting the following URL and clicking 'Import':

   `https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/F5SPDevOps.postman_environment.json`

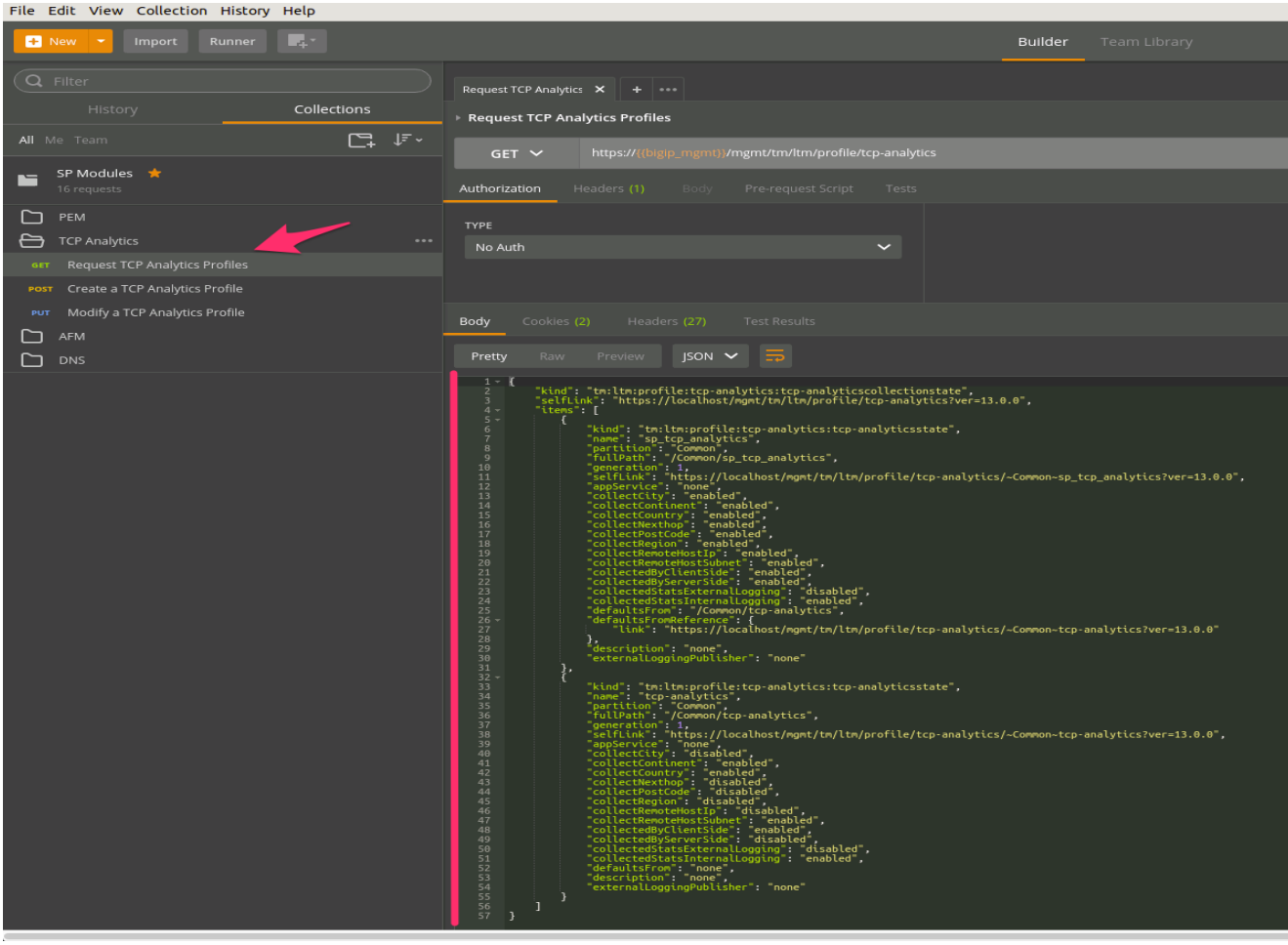


Task 2 – Configure TCP Analytics

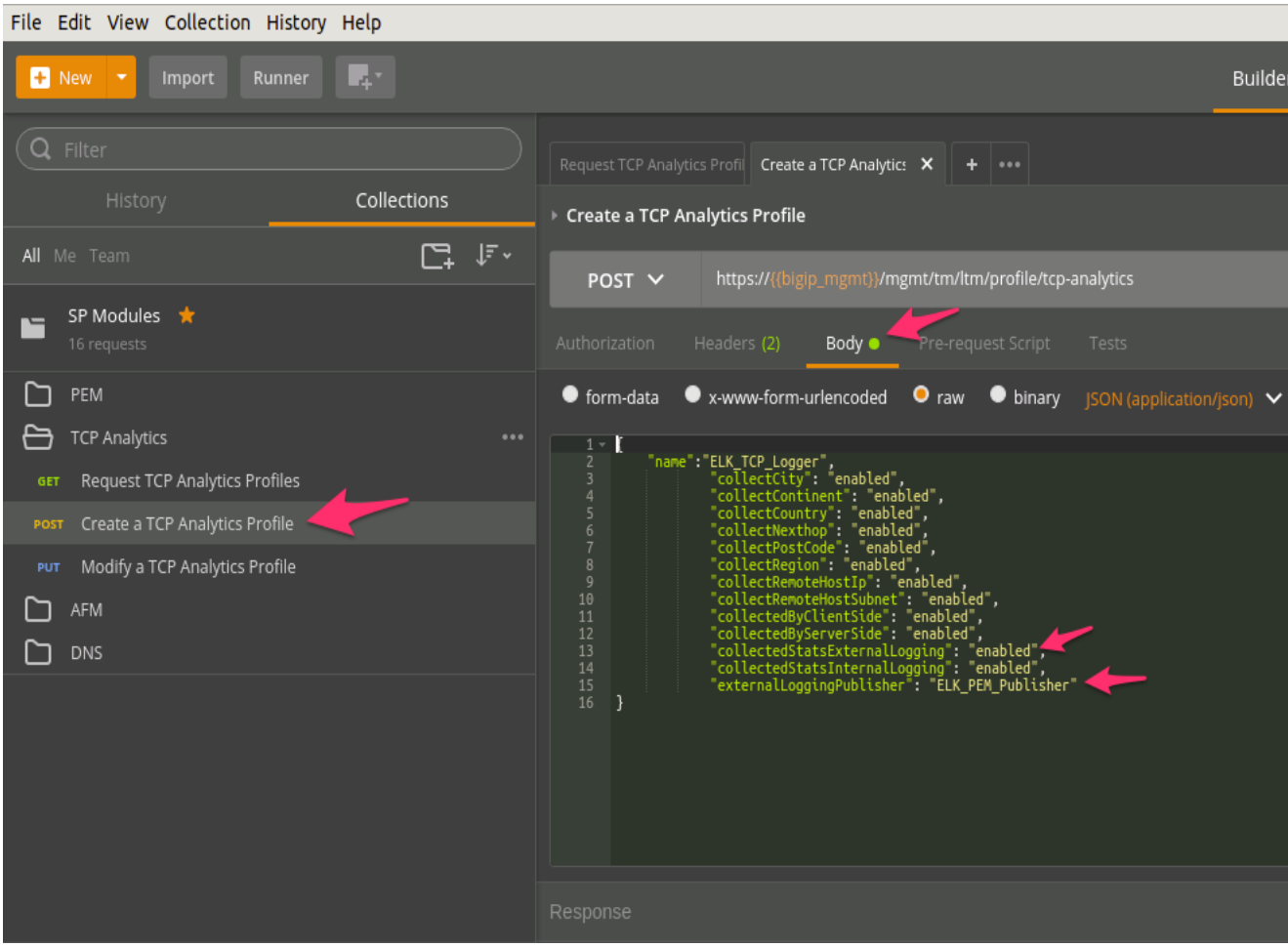
In this task we will query and configure TCP AVR profile. This will be done using REST API (explored in previous Lab)

Perform the following steps to complete this task:

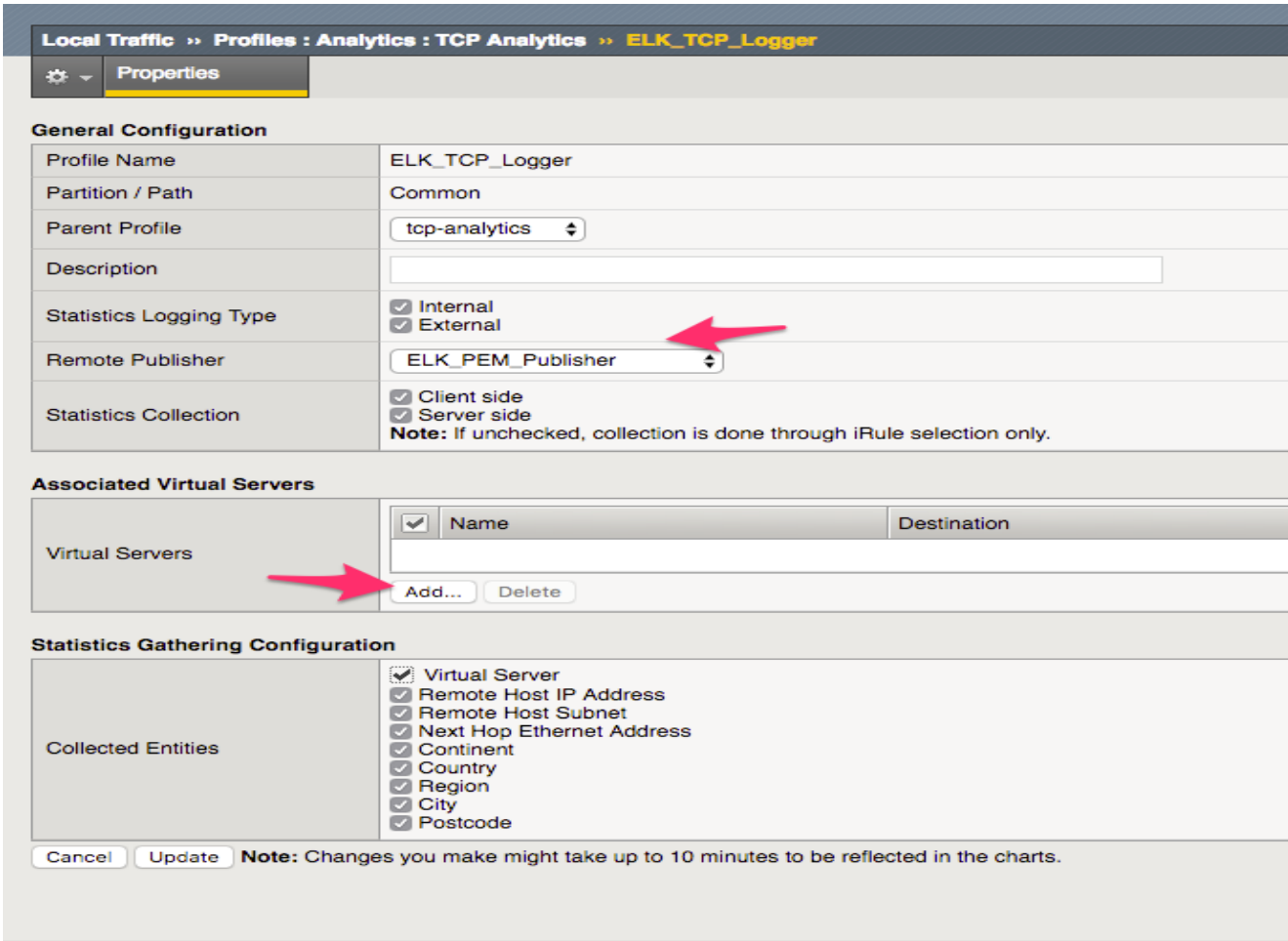
1. Click the 'TCP Analytics' item in the SP Module Postman Collection
2. Notice that we are sending a GET request to the `/mgmt/tm/ltm/profile/tcp-analytics` end-point. Check the body returned and observer the default values.



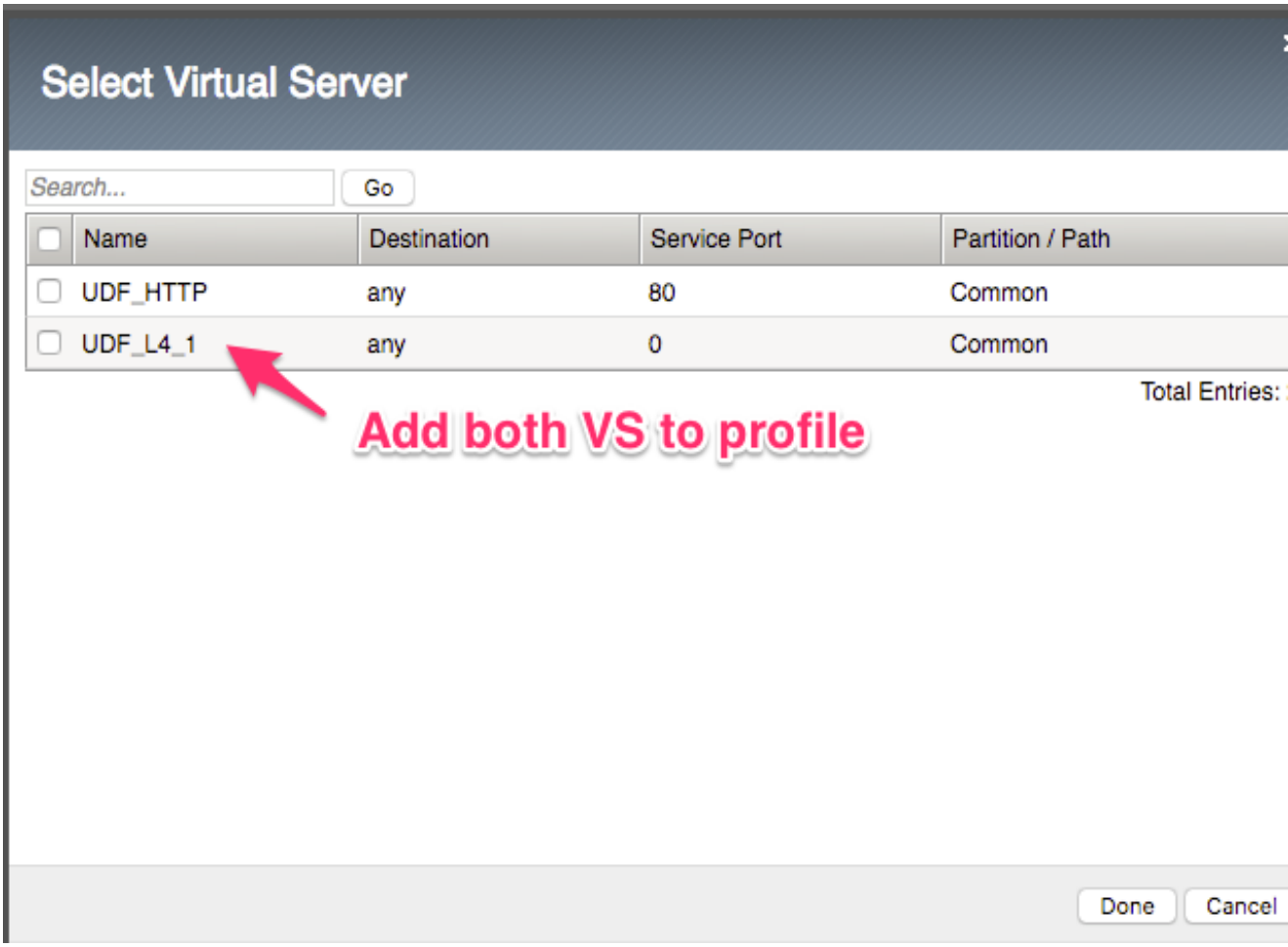
3. Click on the 'Create TCP Analytics Profile' , check the body message for ELK_PEM_Publisher (We will use the PEM index in ELK for logging TCP Optimisation)



4. Verify in BIG-IP TMUI that the new profile was created.



5. Add in the VS manually (This is not available in REST API currently)

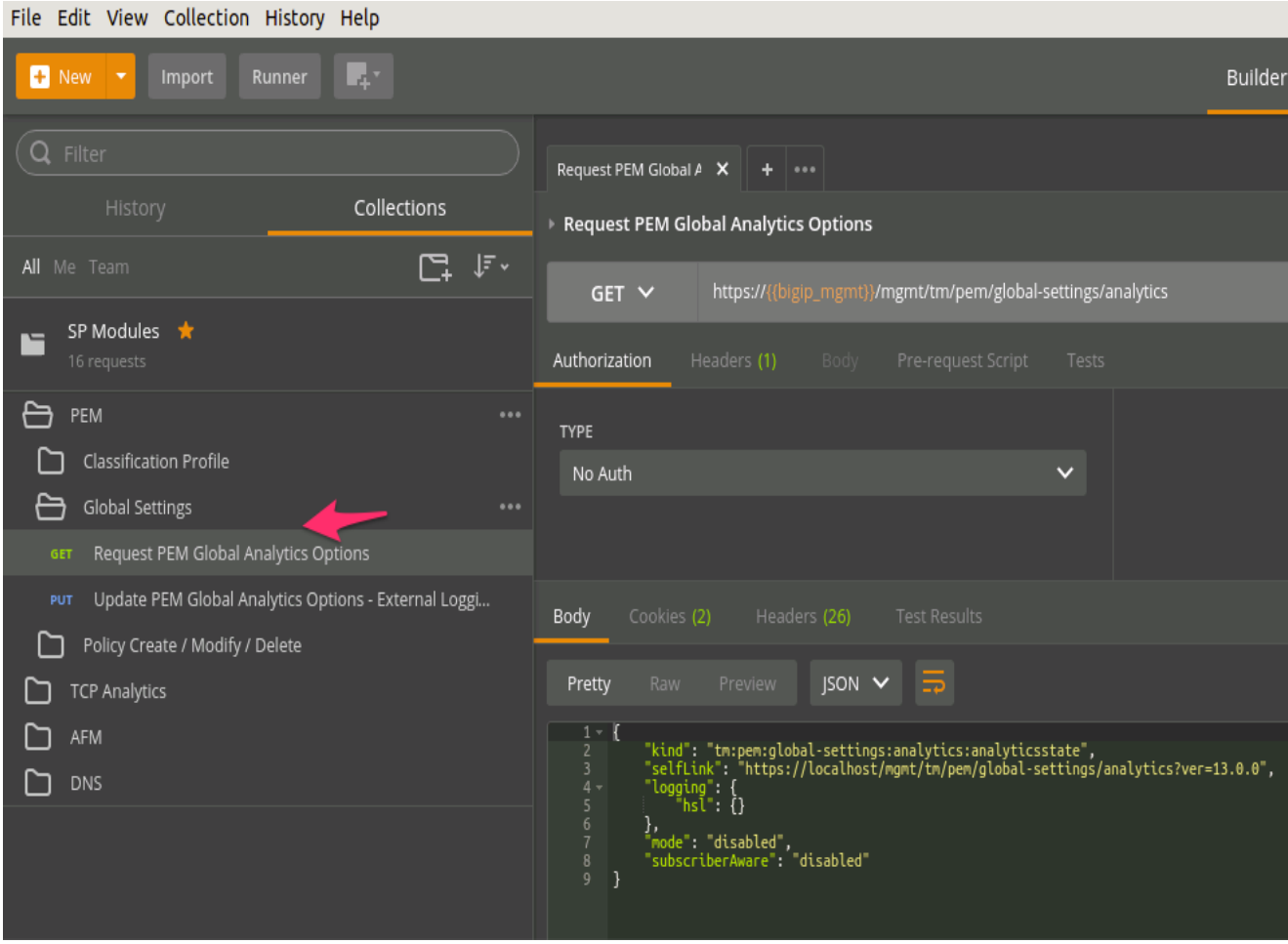


Task 3 – Configure PEM Analytics

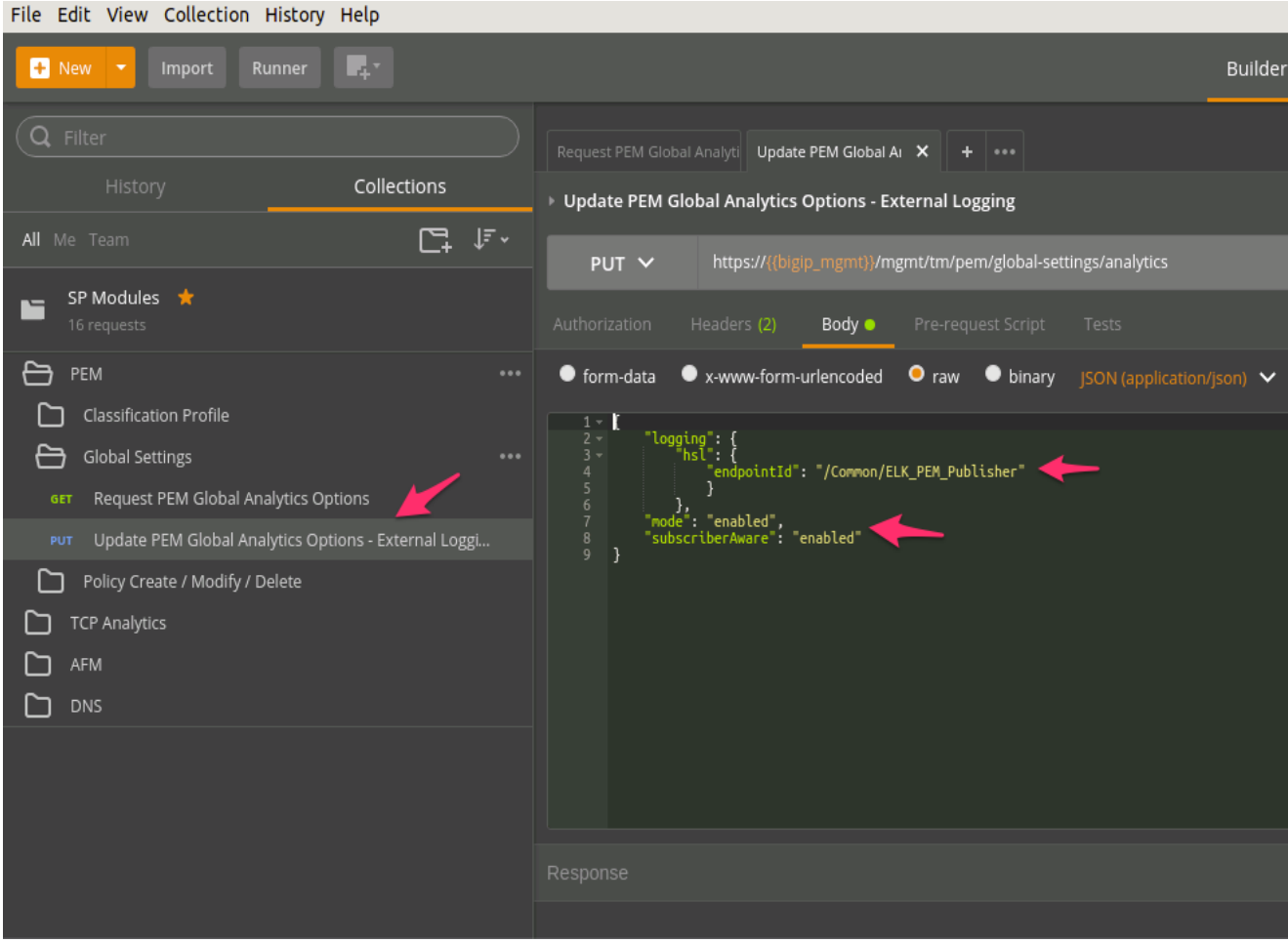
In this task we will query and configure PEM AVR profile. This will be done using REST API (explored in previous Lab)

Perform the following steps to complete this task:

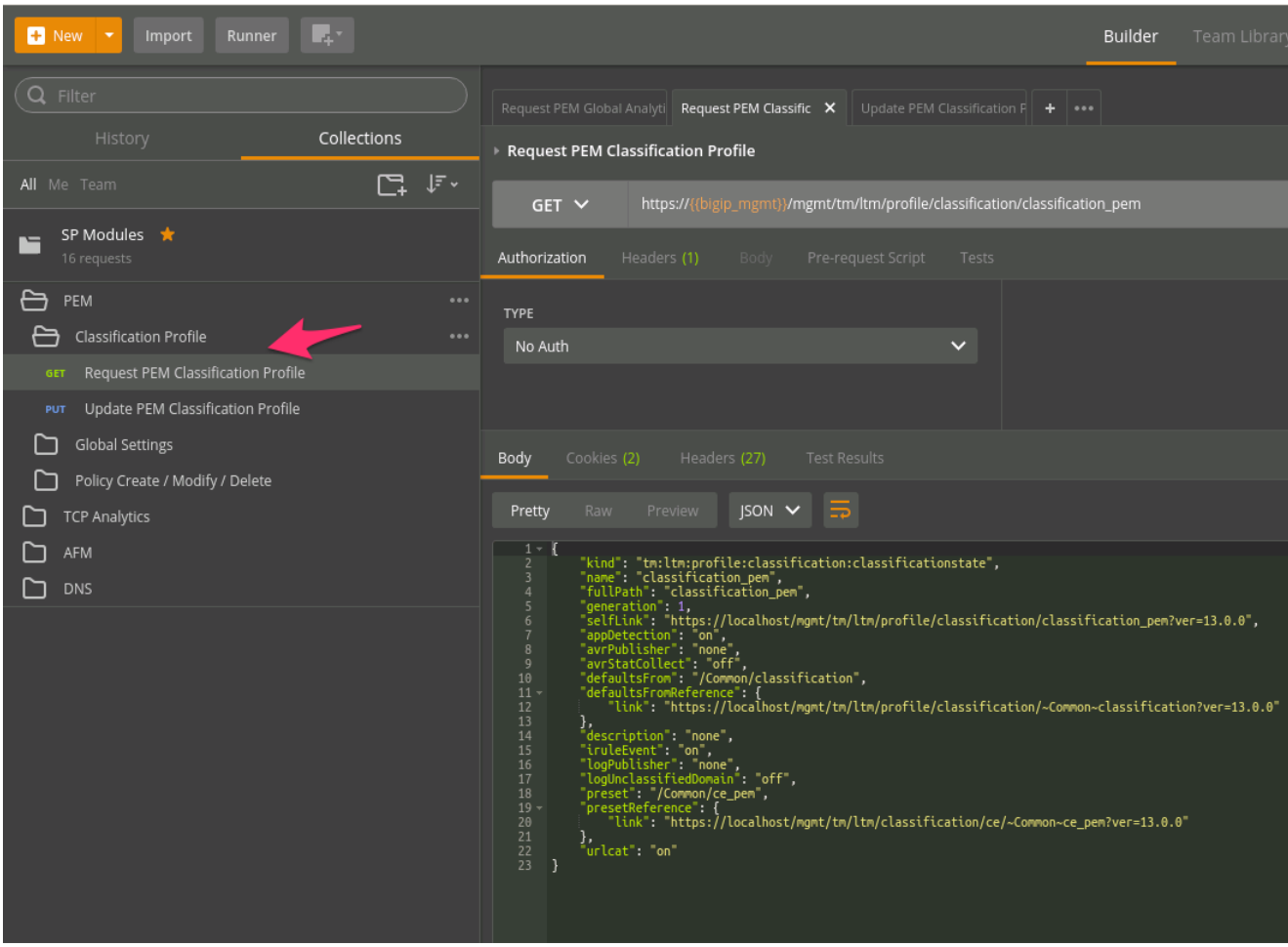
1. Click the 'PEM' item in the SP Module Postman Collection
2. Notice there are two sections we must update Global and Classification. We will do Global first, click on 'Request PEM Global Analytics Options' we are sending a GET request to the `/mgmt/tm/pem/global-settings/analytics` endpoint. Check the body returned and observer the default values.



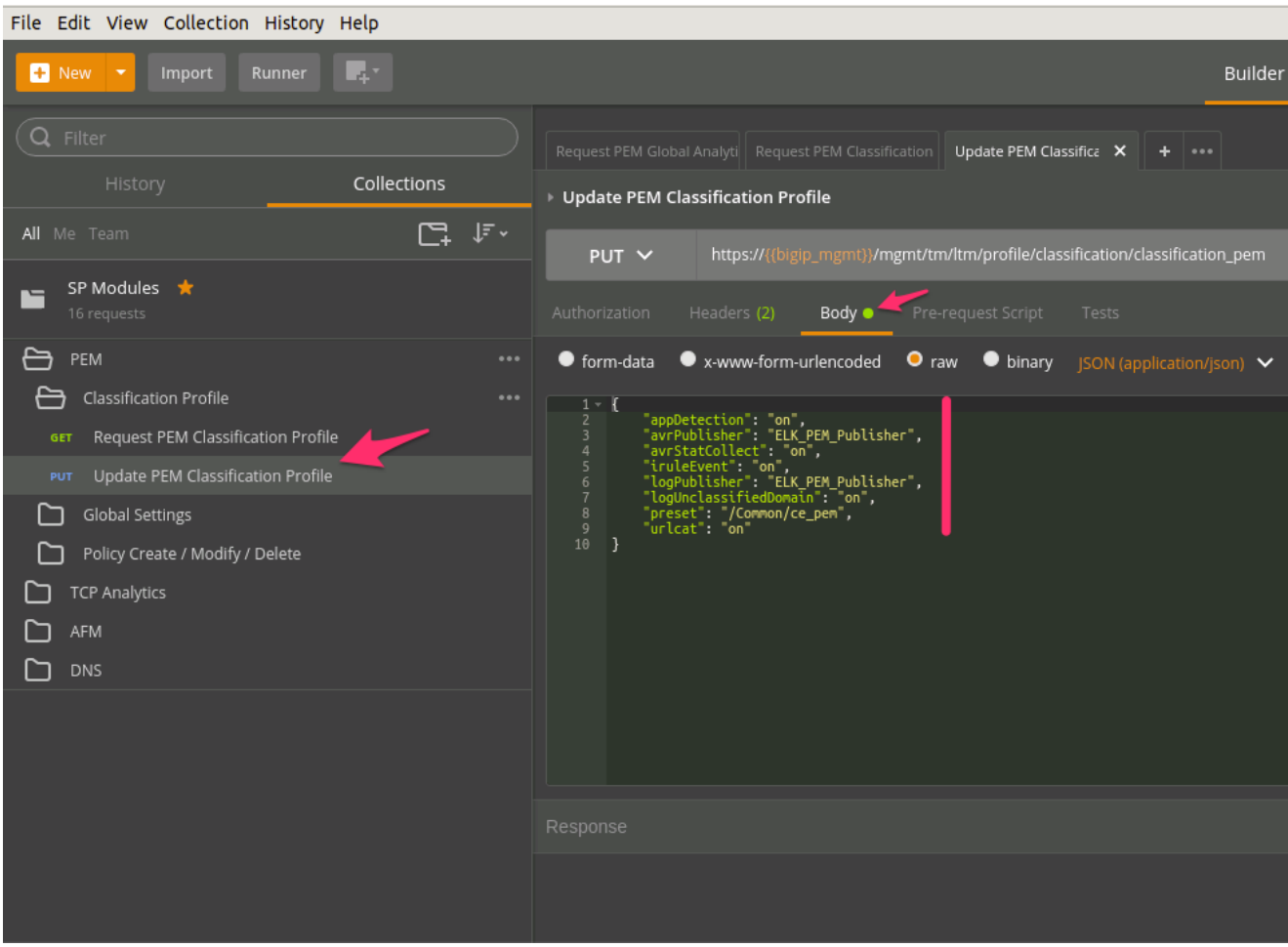
3. Click on the ‘Update PEM Global Analytics Options - External Logging’ , check the body message for ELK_PEM_Publisher.



4. Verify in BIG-IP TMUI that the new updates where changed in PEM global options.
5. Click on 'Request PEM Classification Profile' we are sending a GET request to the `/mgmt/tm/ltm/profile/classification/classification_pem` endpoint. Check the body returned and observer the default values.



6. Click on the ‘Update PEM Classification Profile’ , check the body message for ELK_PEM_Publisher.



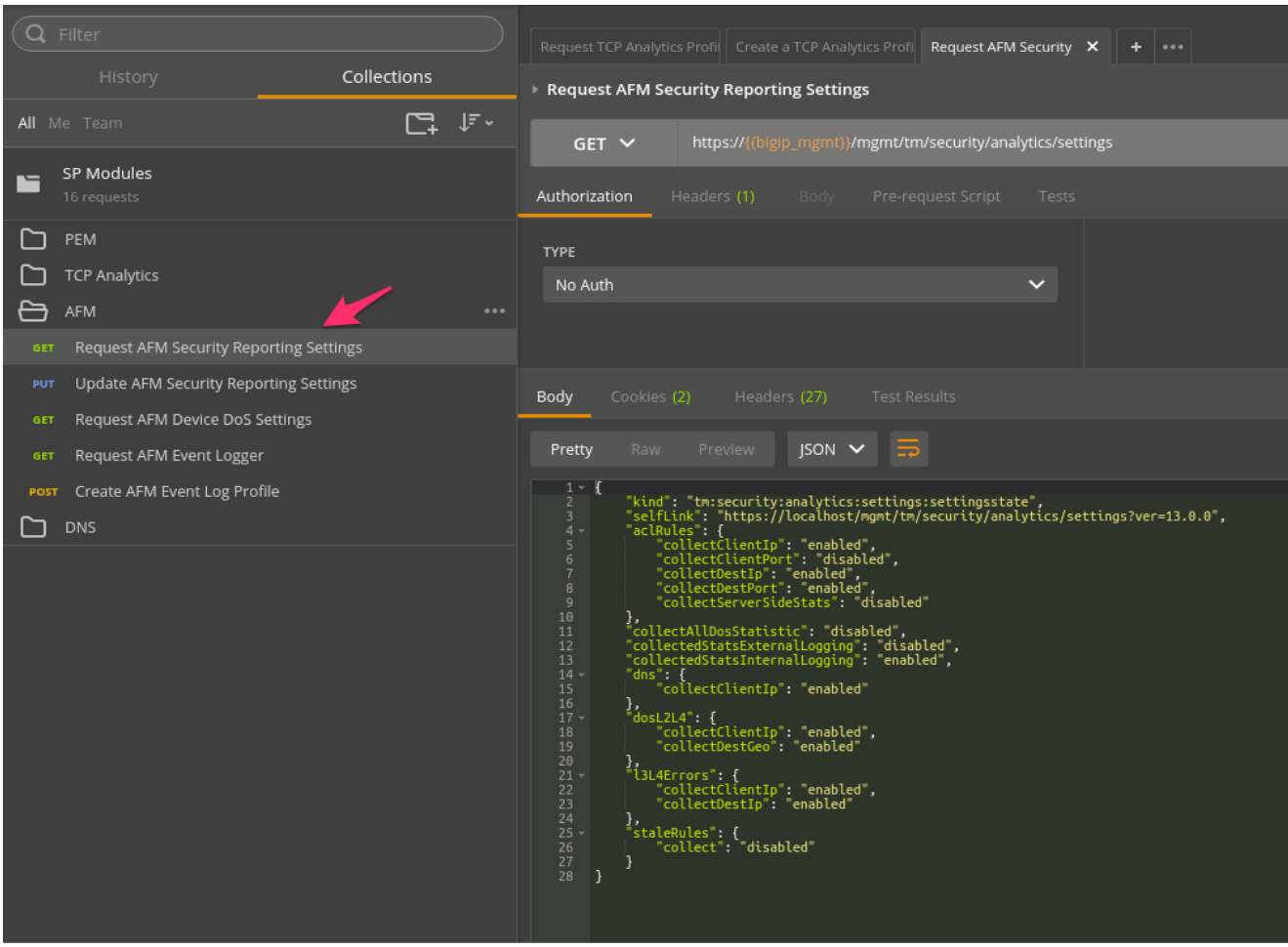
7. Verify in BIG-IP TMUI that the new updates where changed in PEM Classification.

Task 4 – Configure AFM Analytics

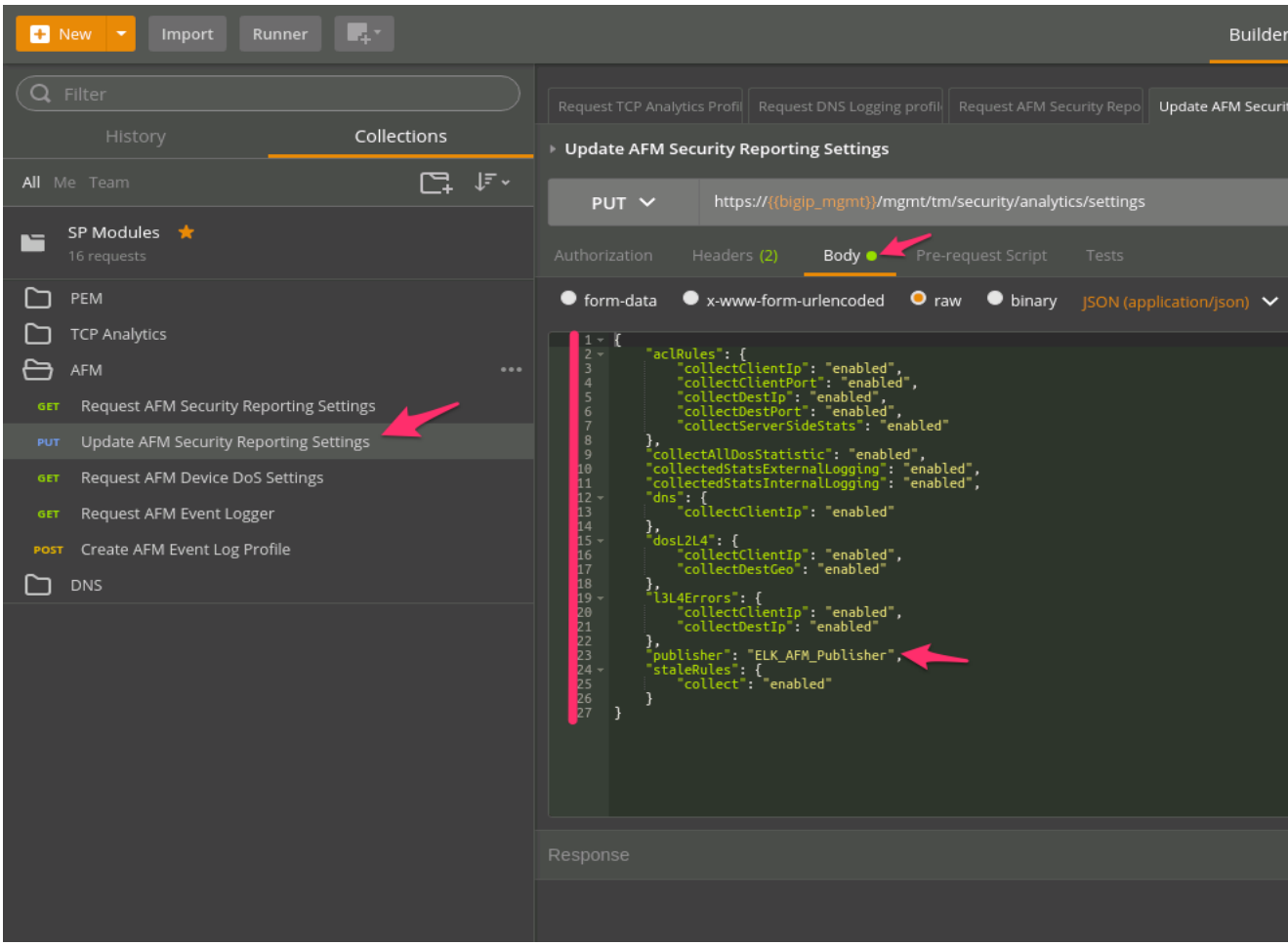
In this task we will query and configure AFM AVR profile and Logging. This will be done using REST API (explored in previous Lab)

Perform the following steps to complete this task:

1. Click the 'AFM' item in the SP Module Postman Collection
2. Notice there are two sections we must update Security Reporting and Event Logging. We will do Security Reporting first, click on 'Request AFM Security Reporting Settings' we are sending a GET request to the `/mgmt/tm/security/analytics/settings` endpoint. Check the body returned and observer the default values.



3. Click on the 'Update AFM Security Reporting Settings' , check the body message for ELK_AFM_Publisher.

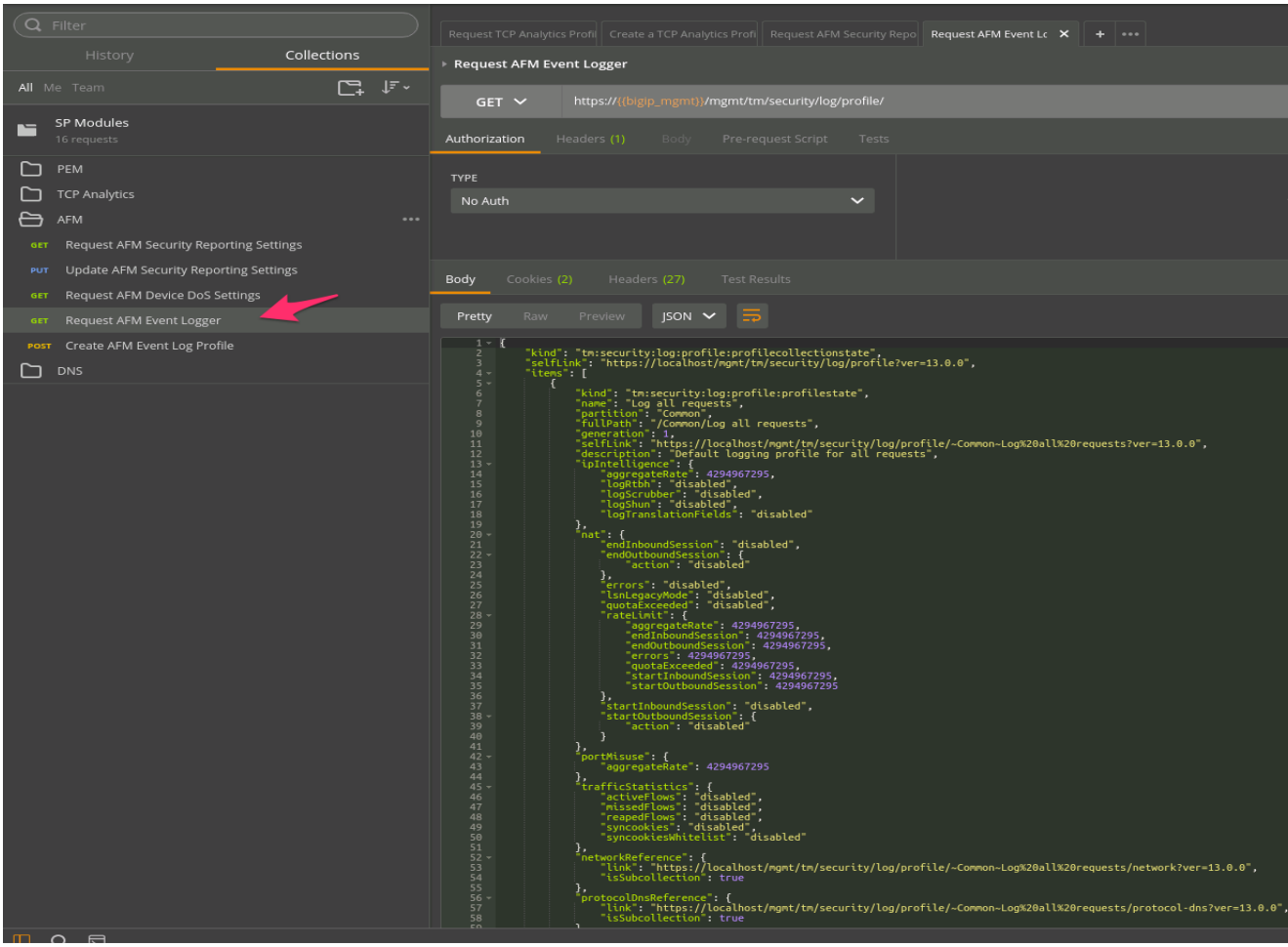


4. Verify in BIG-IP TMUI that the new updates where changed in AFM Report Settings.

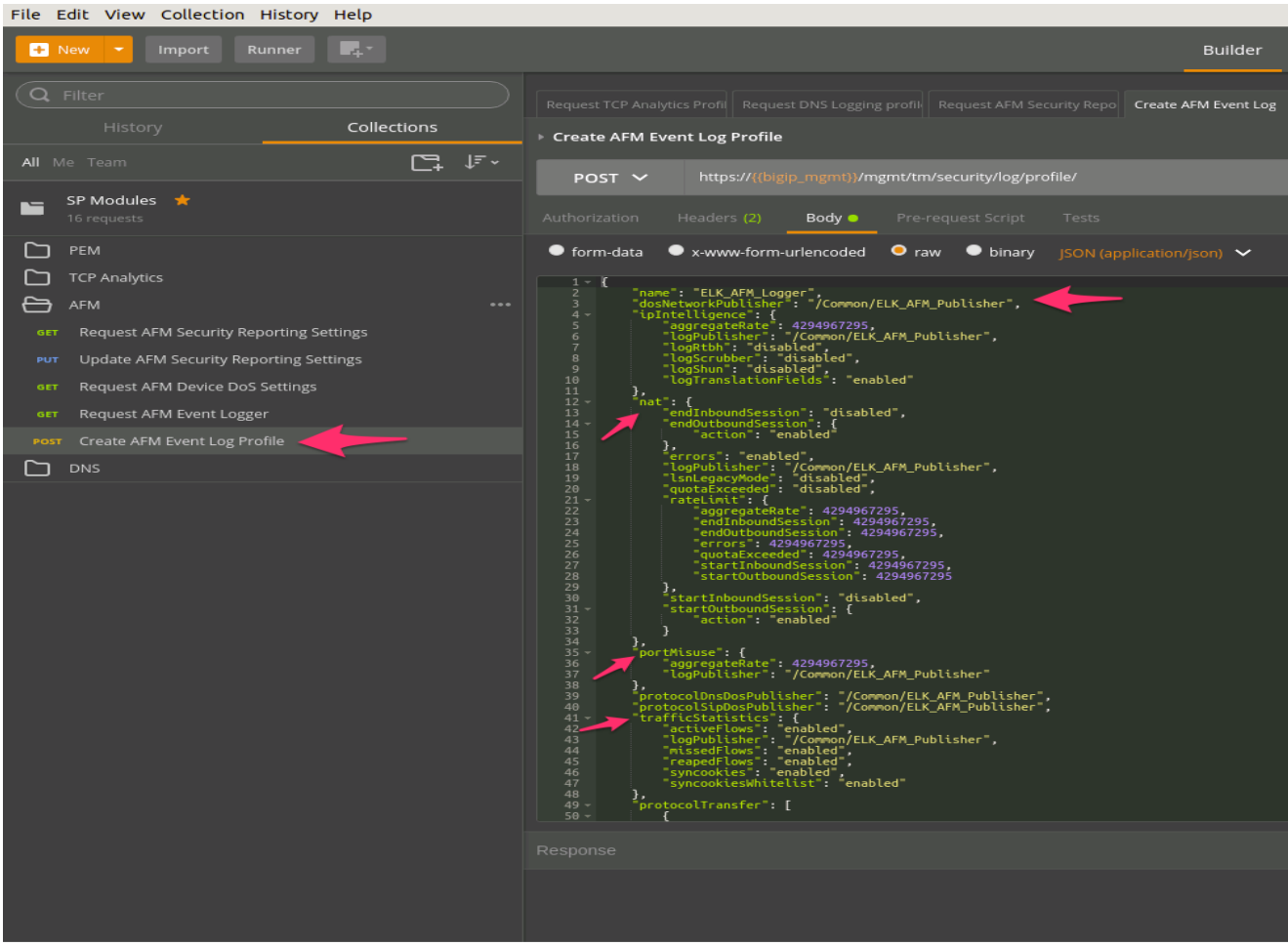
Note: Request AFM Device DoS Settings - Can be used to report on settings currently set, however REST API cannot be used to update these settings at this time.

---

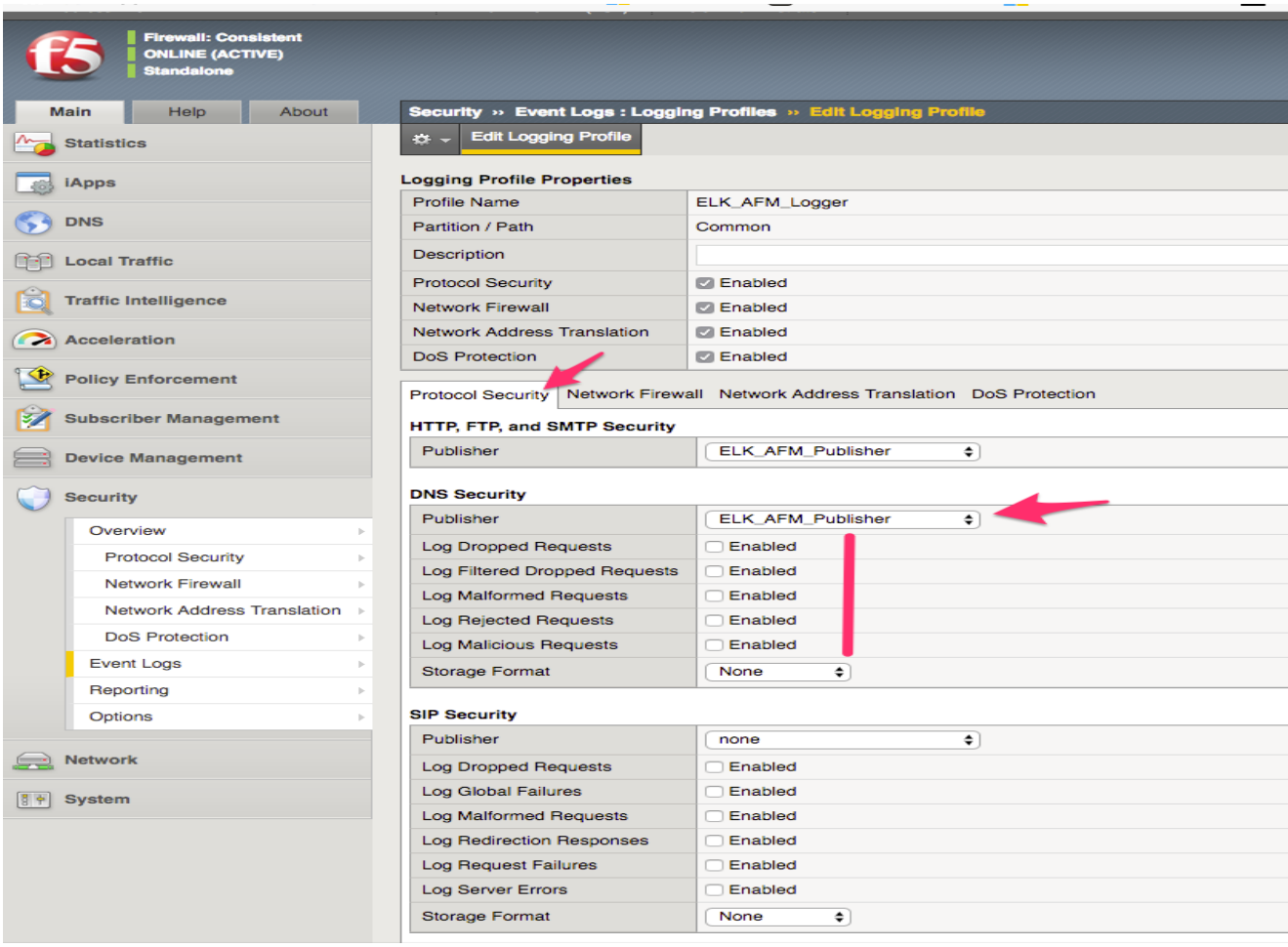
1. Click on 'Request AFM Event Logger' we are sending a GET request to the `/mgmt/tm/security/log/profile/` endpoint. Check the body returned and observer the default values.



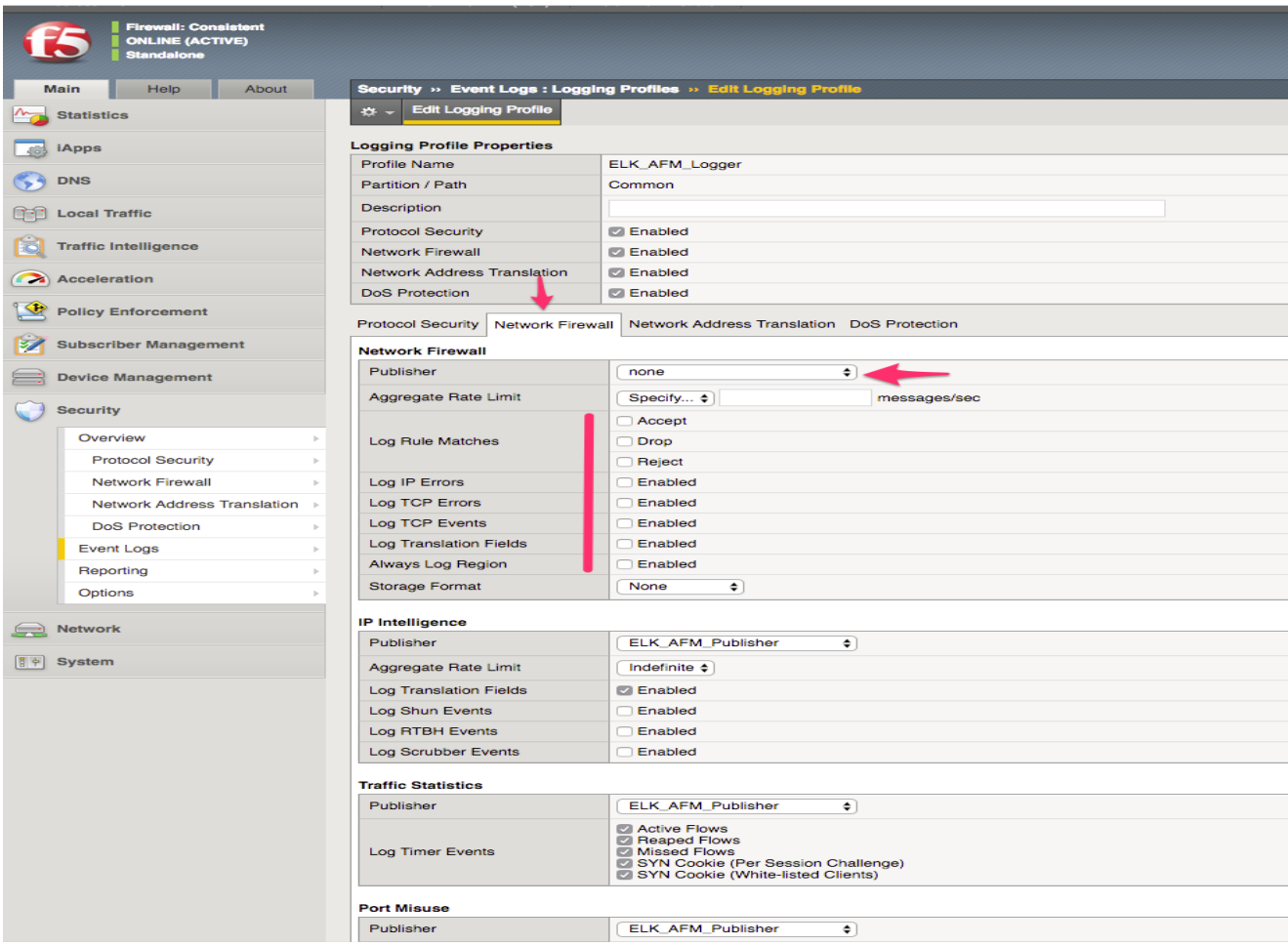
2. Click on the ‘Create AFM Event Log Profile’ , check the body message for ELK_AFM_Publisher.



3. Addtional Steps are required for AFM as not all REST commands can configure all sections at this time. Go to TMUI on BIG-IP and navigate to Security / Event Logs / Logging Profiles. Change Publishers and tick events to log.



Update Network Firewall tab and click update.

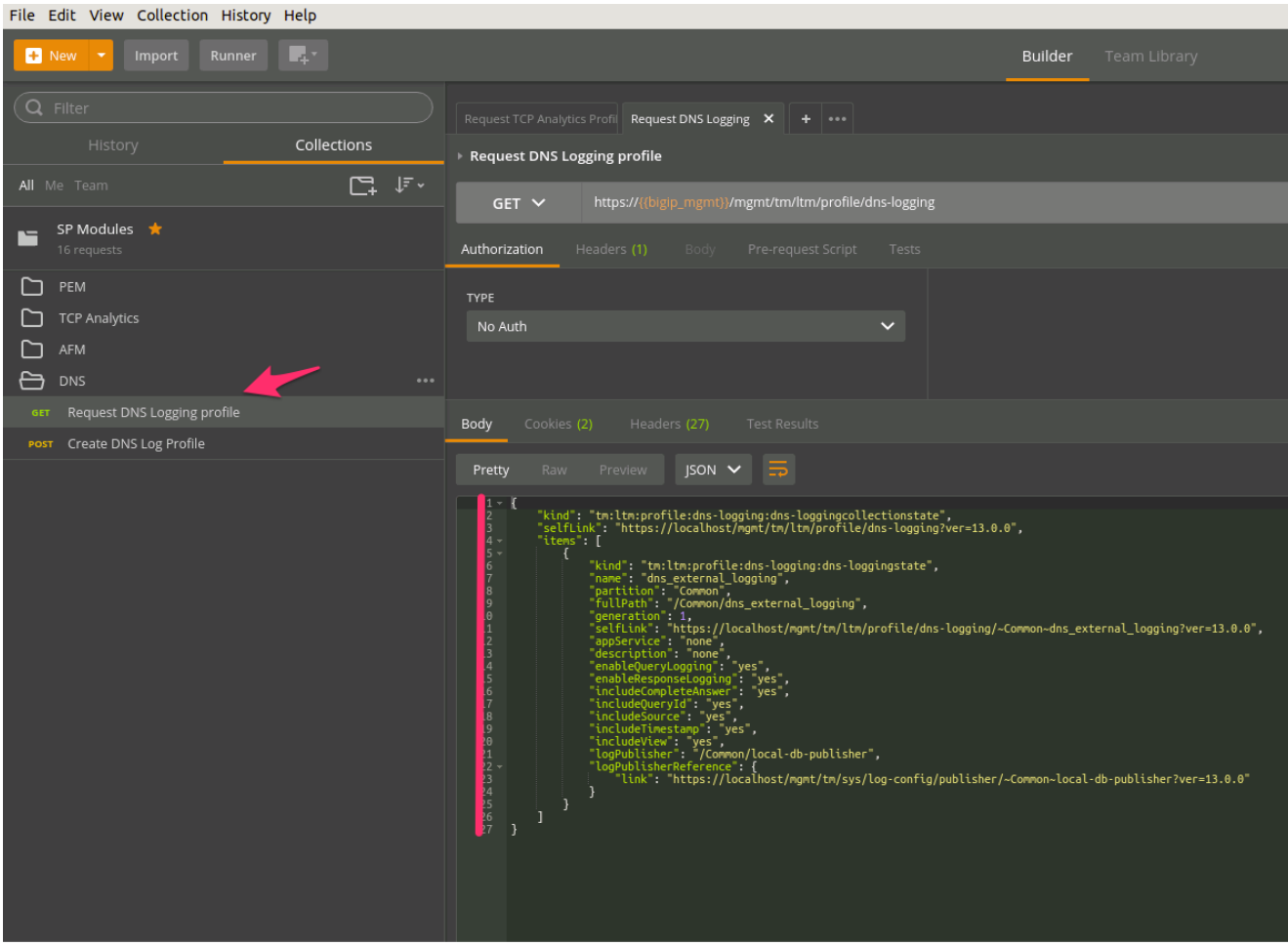


Task 5 – Configure DNS Analytics

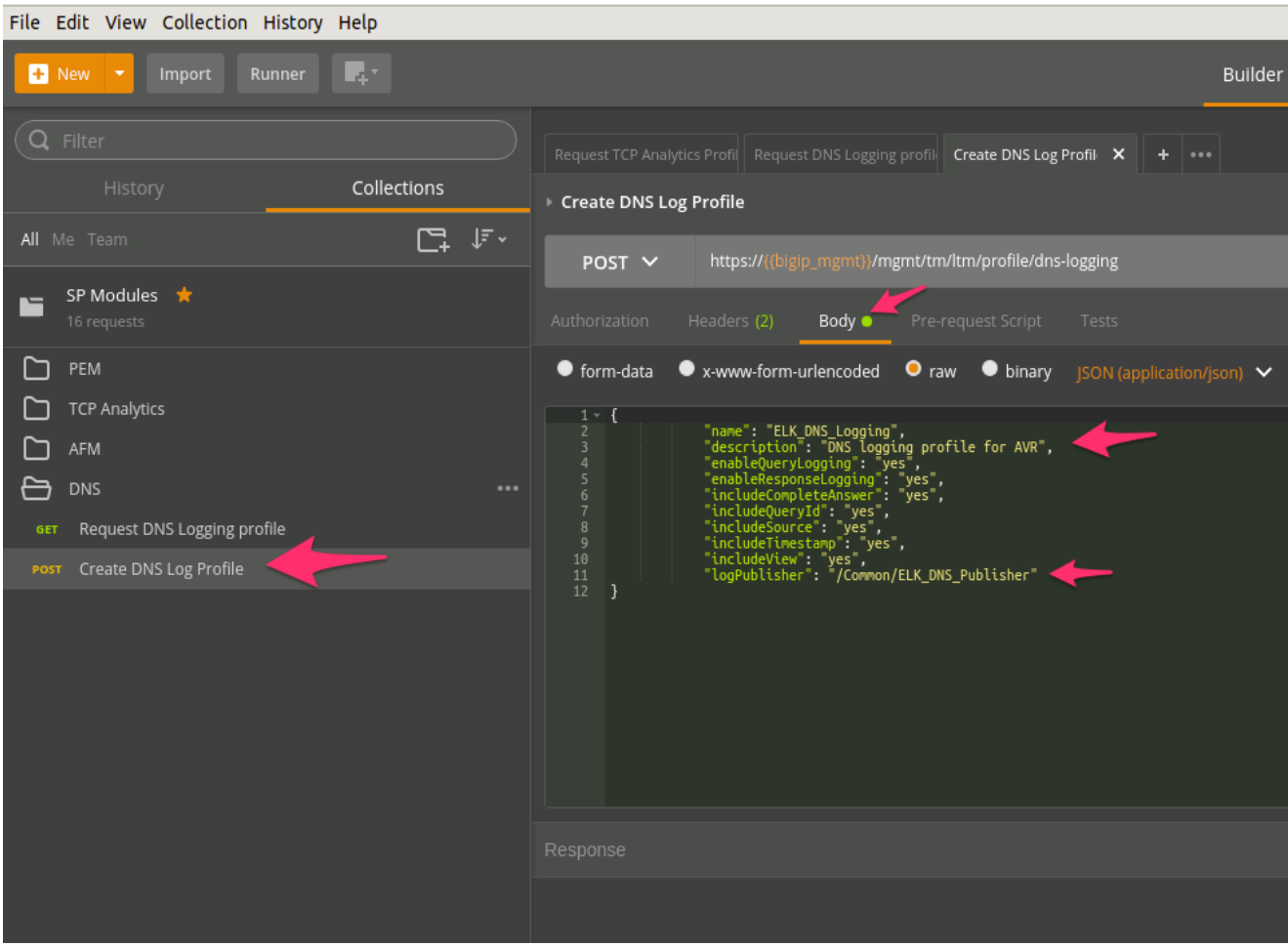
In this task we will query and configure DNS AVR profile. This will be done using REST API (explored in previous Lab)

Perform the following steps to complete this task:

1. Click the 'DNS' item in the SP Module Postman Collection
2. Notice that we are sending a GET request to the `/mgmt/tm/ltm/profile/dns-logging` endpoint. Check the body returned and observer the default values.



3. Click on the ‘Create DNS Log Profile’ , check the body message for ELK_DNS_Publisher.



4. Verify in BIG-IP TMUI that the new profile was created.

5.2.2 Lab 2.2: Access Clients and Generate Traffic

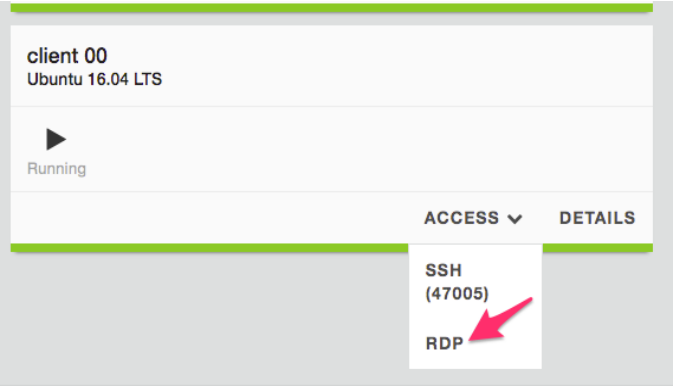
In this lab you will walk through re-configuring the Clients to USE the F5 for traffic. This will generate traffic for PEM / DNS / and AFM for AVR and logging to ELK Stack.

Task 1 - Configure Client Netwoking & Traffic Generation

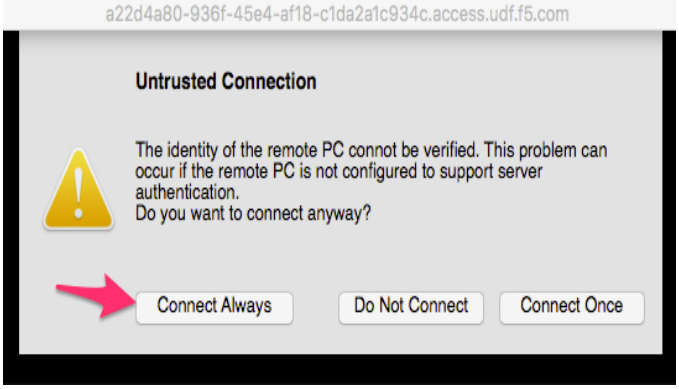
In this task we will configure and use the Client UDF machines. These Clients are required to be reconfigured to utilise the network and DNS from the F5..

Perform the following steps to complete this task:

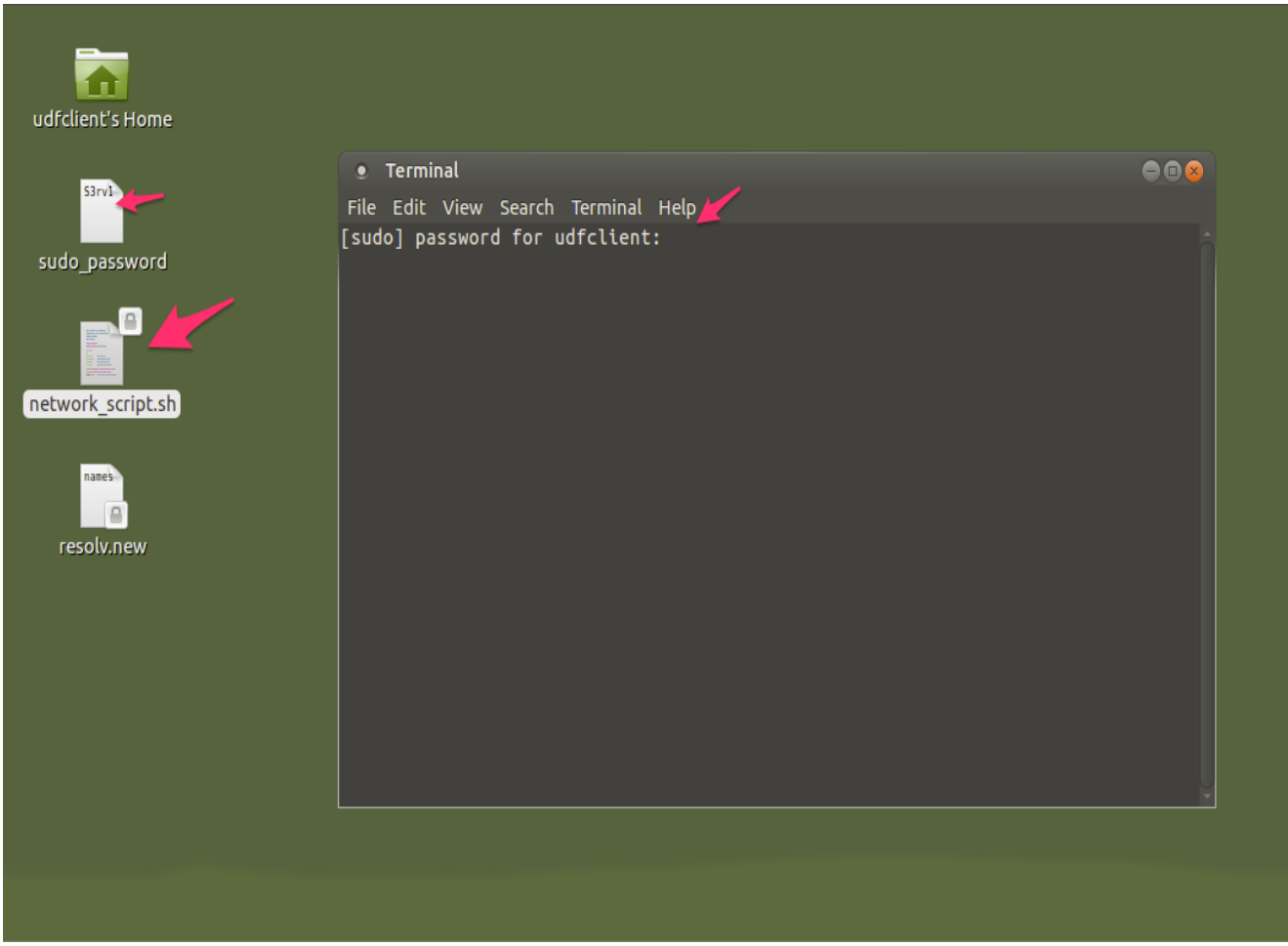
1. Click the on the RDP access for UDF for each client.



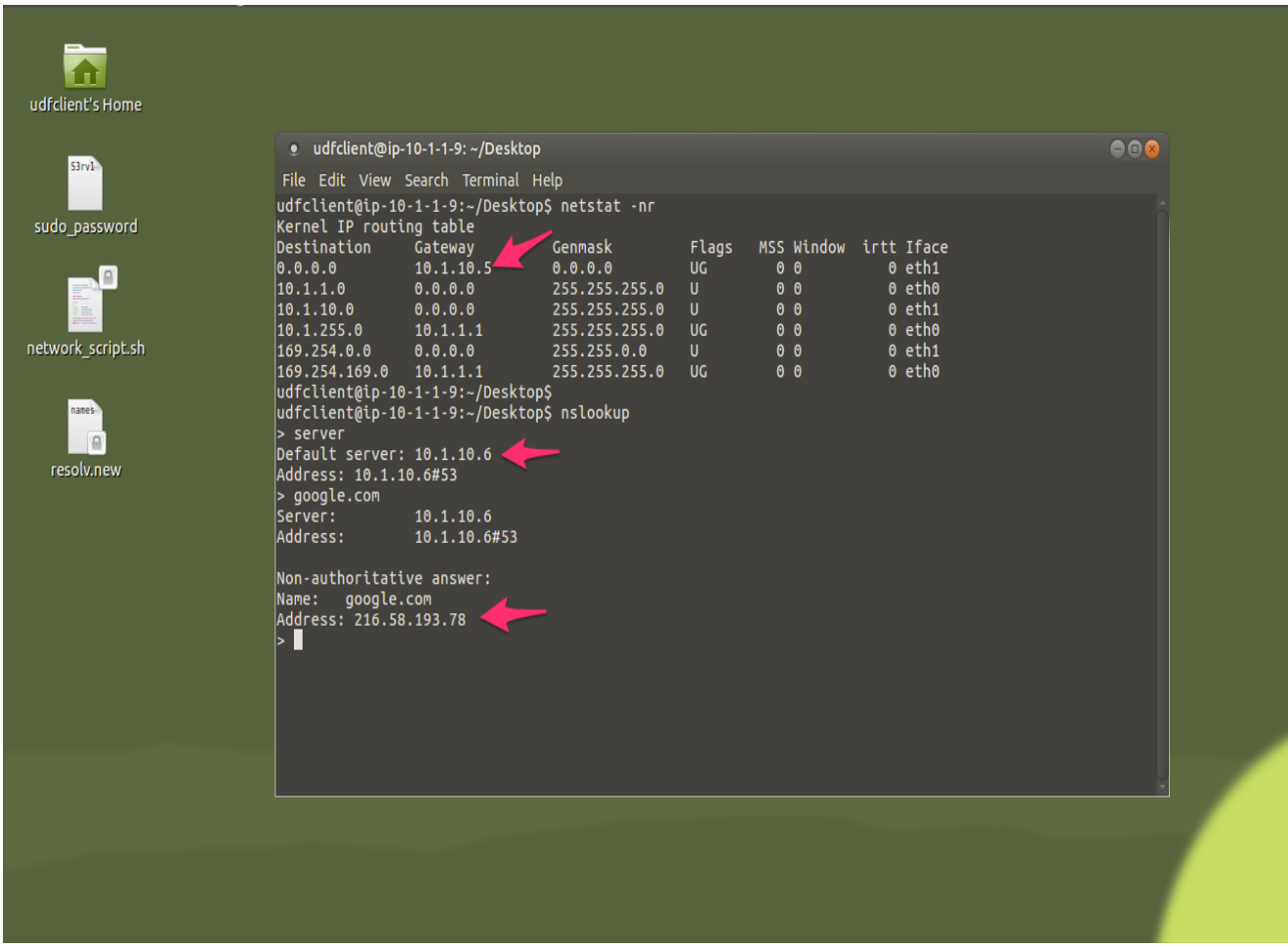
Accept warning always



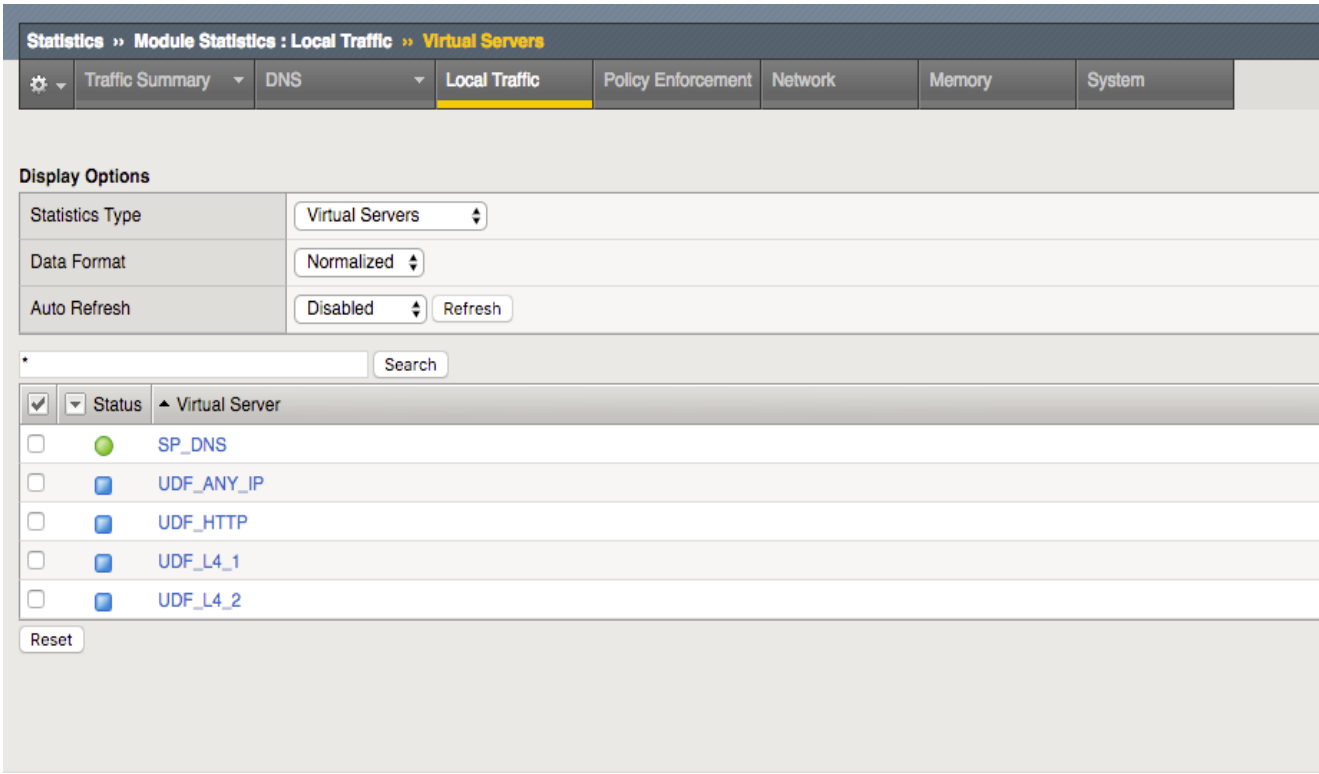
2. Click on the networking script (this will prompt for Sudo password)



3. Once the script has completed, check netstat -nr and nslookup to verify you have traffic passing the F5.



4. Verify in BIG-IP TMUI that you see traffic on the F5 VS's



5. Apply the same fix for the other client.
6. Once both clients are fixed, generate traffic by opening applications and webpages (Leave the applications open so traffic generation continues)

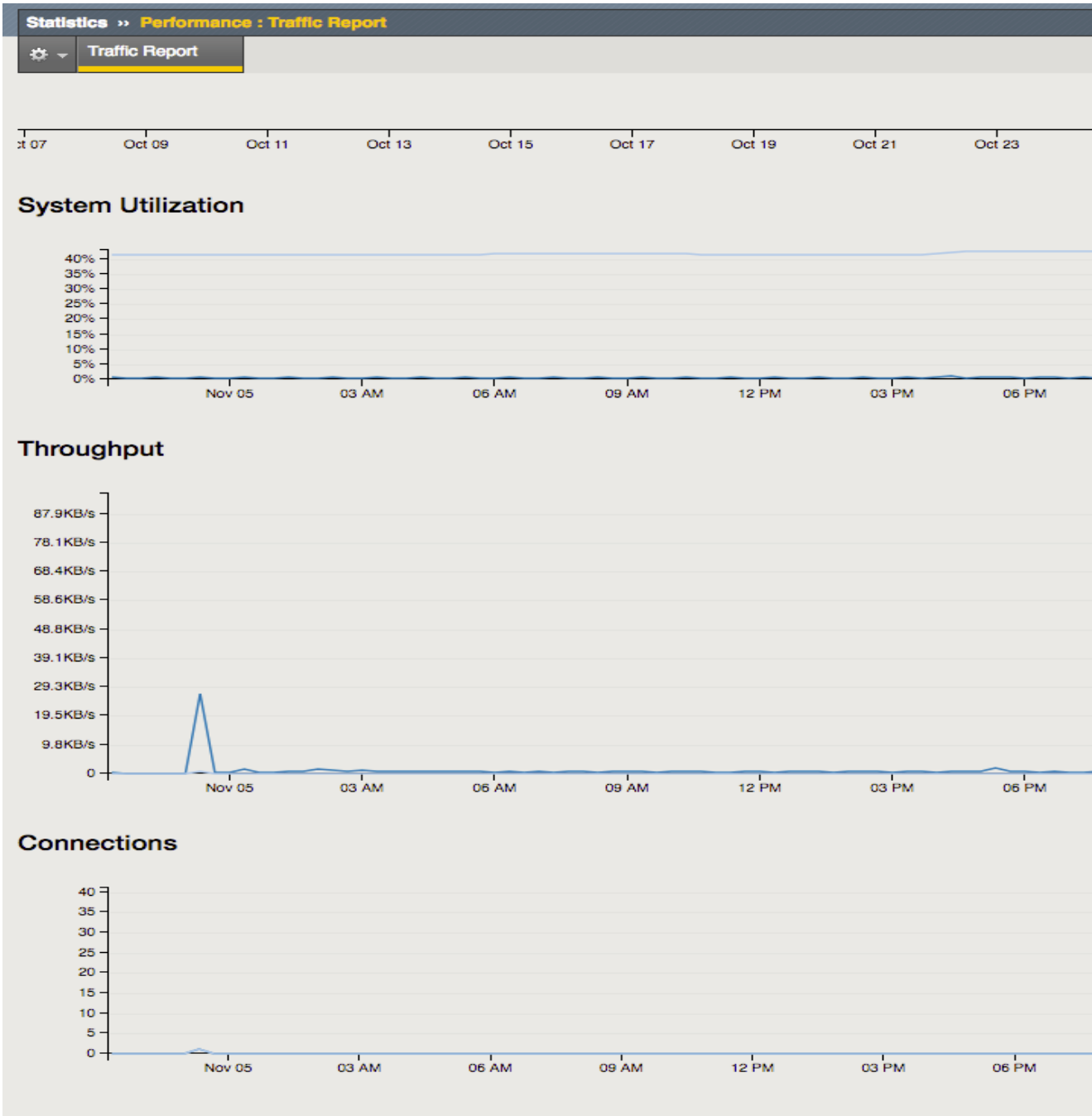
5.2.3 Lab 2.3: Navigating AVR

Navigating and viewing AVR reports.

Task 1 – BIG-IP Performance Report

Perform the following steps to complete this task:

1. Navigate to Performance Report under Statistics.

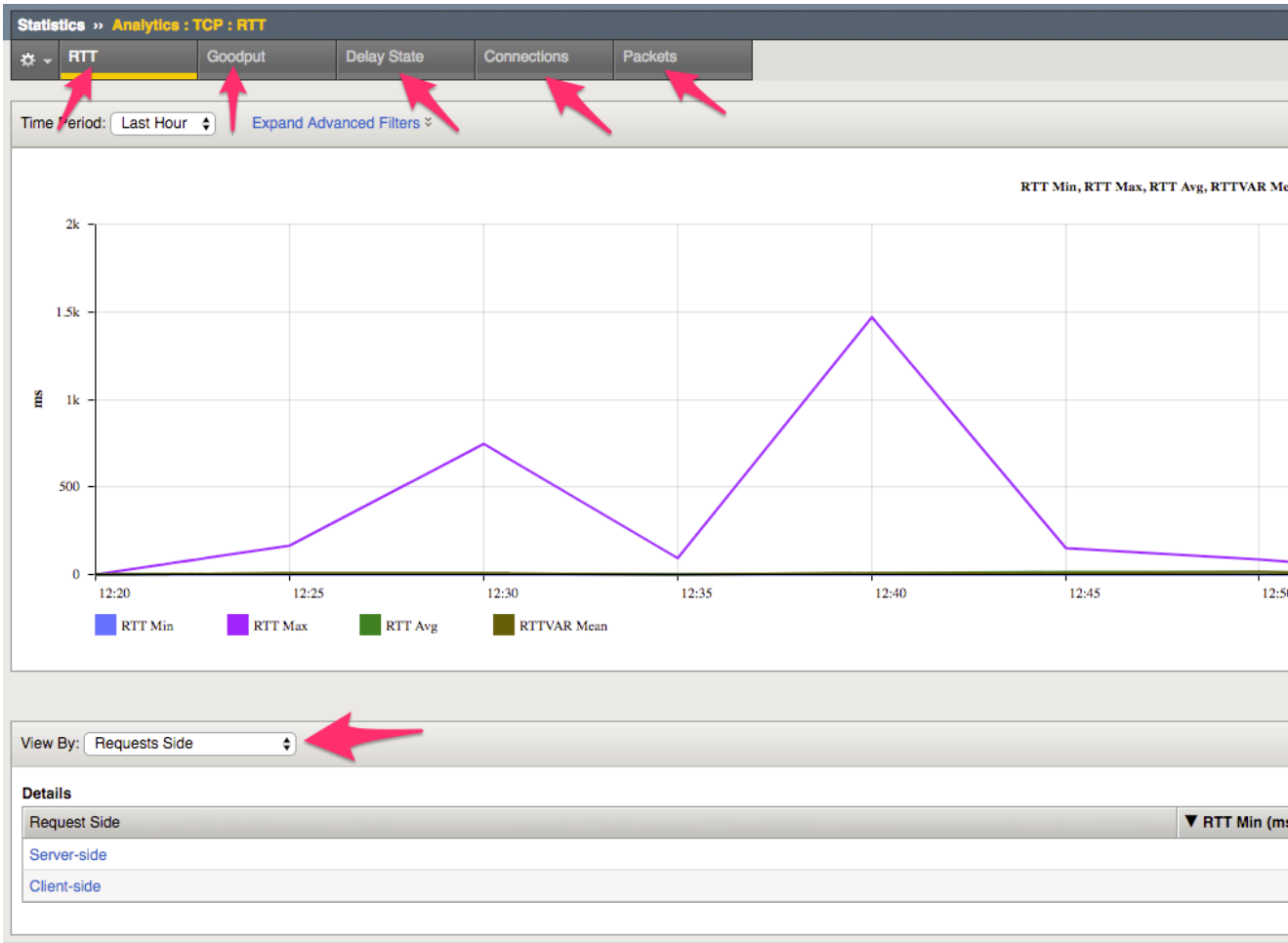


Explore the interface with the sliding bar, and tick and untick options.

Task 2 – AVR TCP Optimisation

Perform the following steps to complete this task:

1. Navigate to Analytics TCP Statistics.



2. Explore the different display options by clicking around the dashboard.

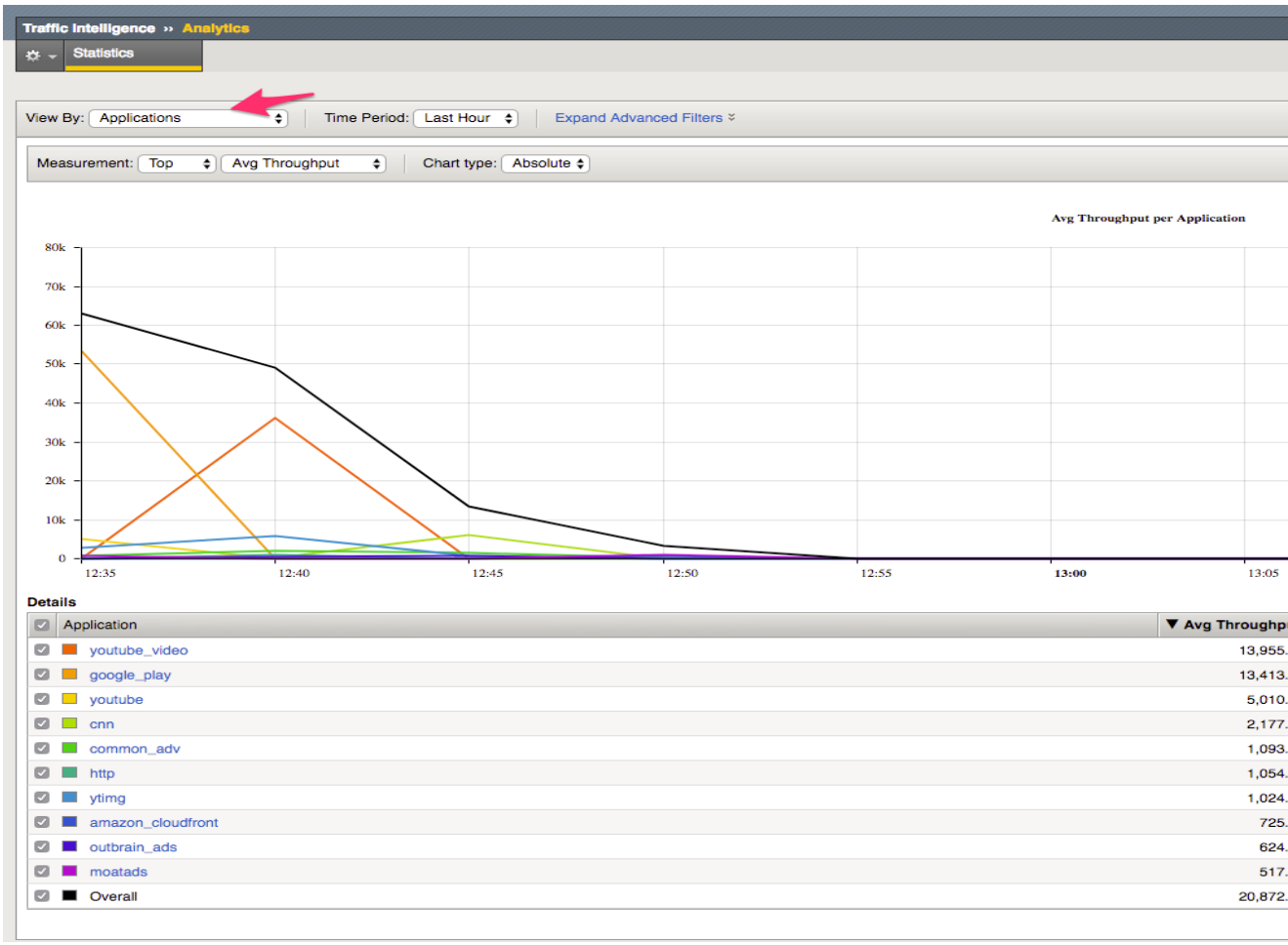
   See the following link for further TCP AVR information:

   https://support.f5.com/kb/en-us/products/big-ip_analytics/manuals/product/analytics-implementations-12-1-0/9.html

Task 3 – AVR Traffic Classification

Perform the following steps to complete this task:

1. Navigate to Traffic Classification Analytics.

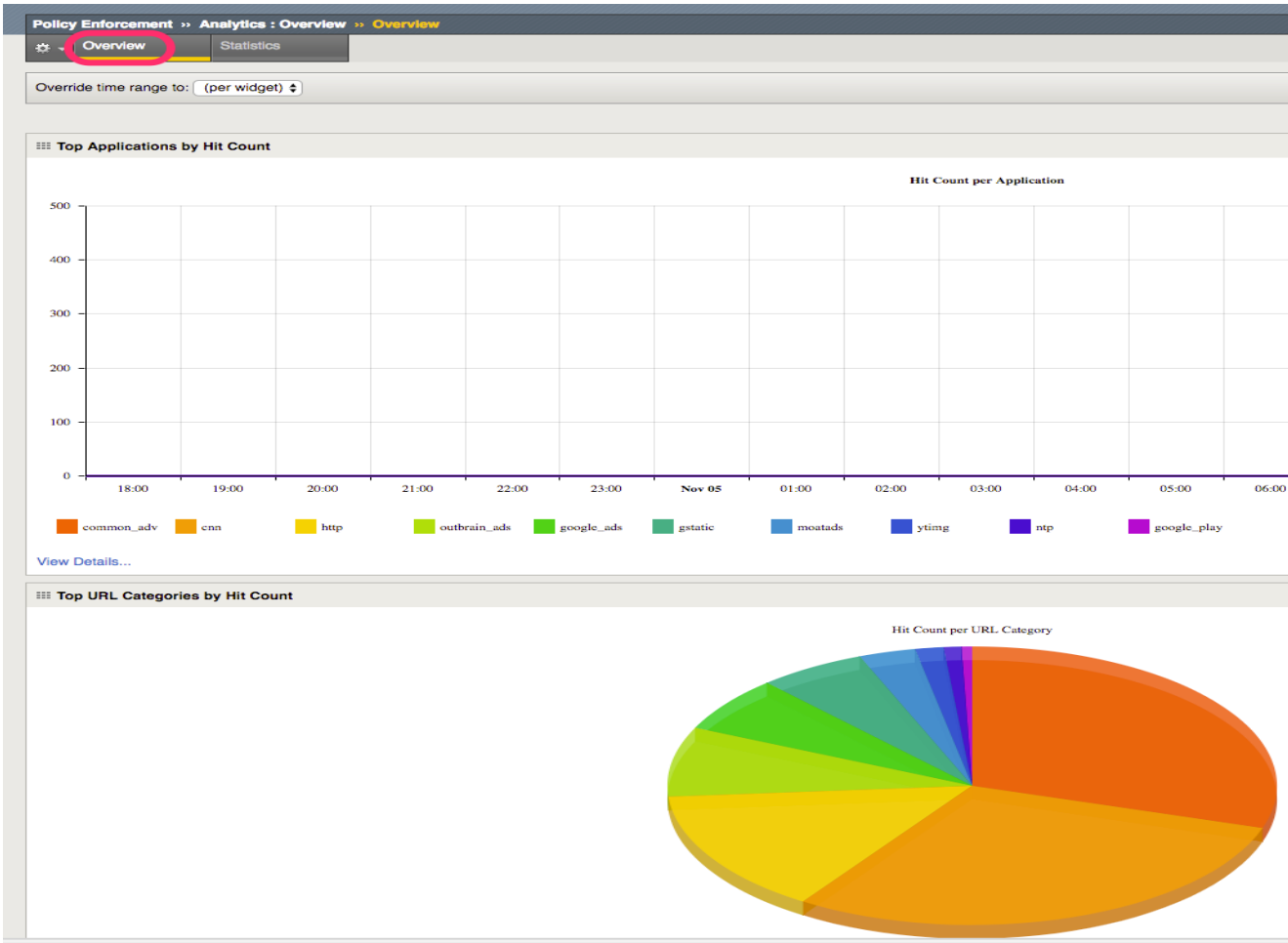


2. Explore the different display options by clicking around the dashboard.

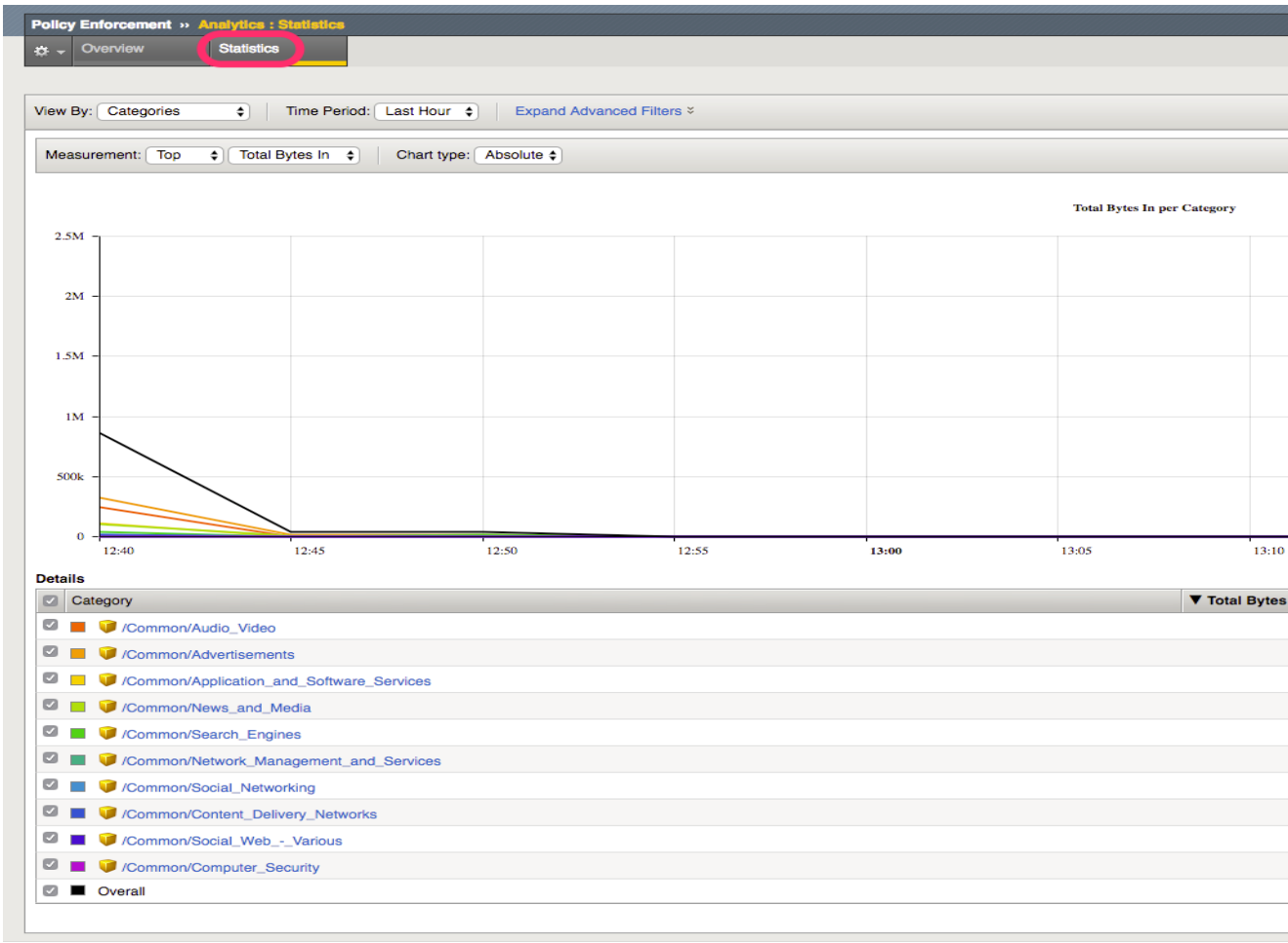
Task 4 – PEM Analytics Report

Perform the following steps to complete this task:

1. Navigate to Policy Enforcement Analytics Overview.



2. Navigate to Policy Enforcement Analytics Statistics.



Explore the different screens and options available for display. See the following link for further AVR information:

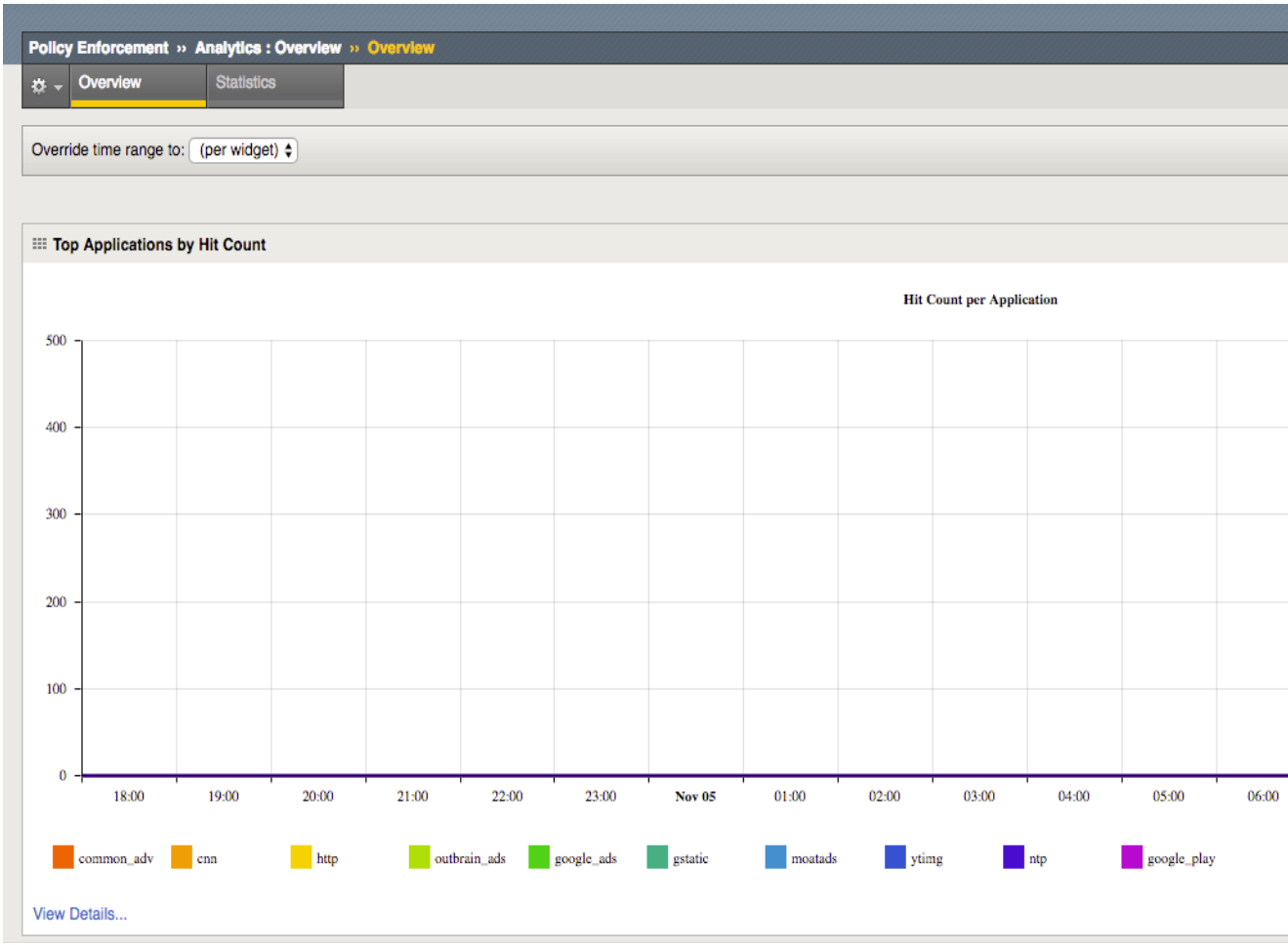
https://support.f5.com/kb/en-us/products/big-ip-pem/manuals/product/pem-implementations-13-0-0.html

Task 5 – Modify PEM AVR Dashboard / Export AVR Report

In this task we will modify and add widgets to default dashboard, and export an Analytics dashboard to a PDF report.

Perform the following steps to complete this task:

1. Navigate to Policy Enforcement Analytics.



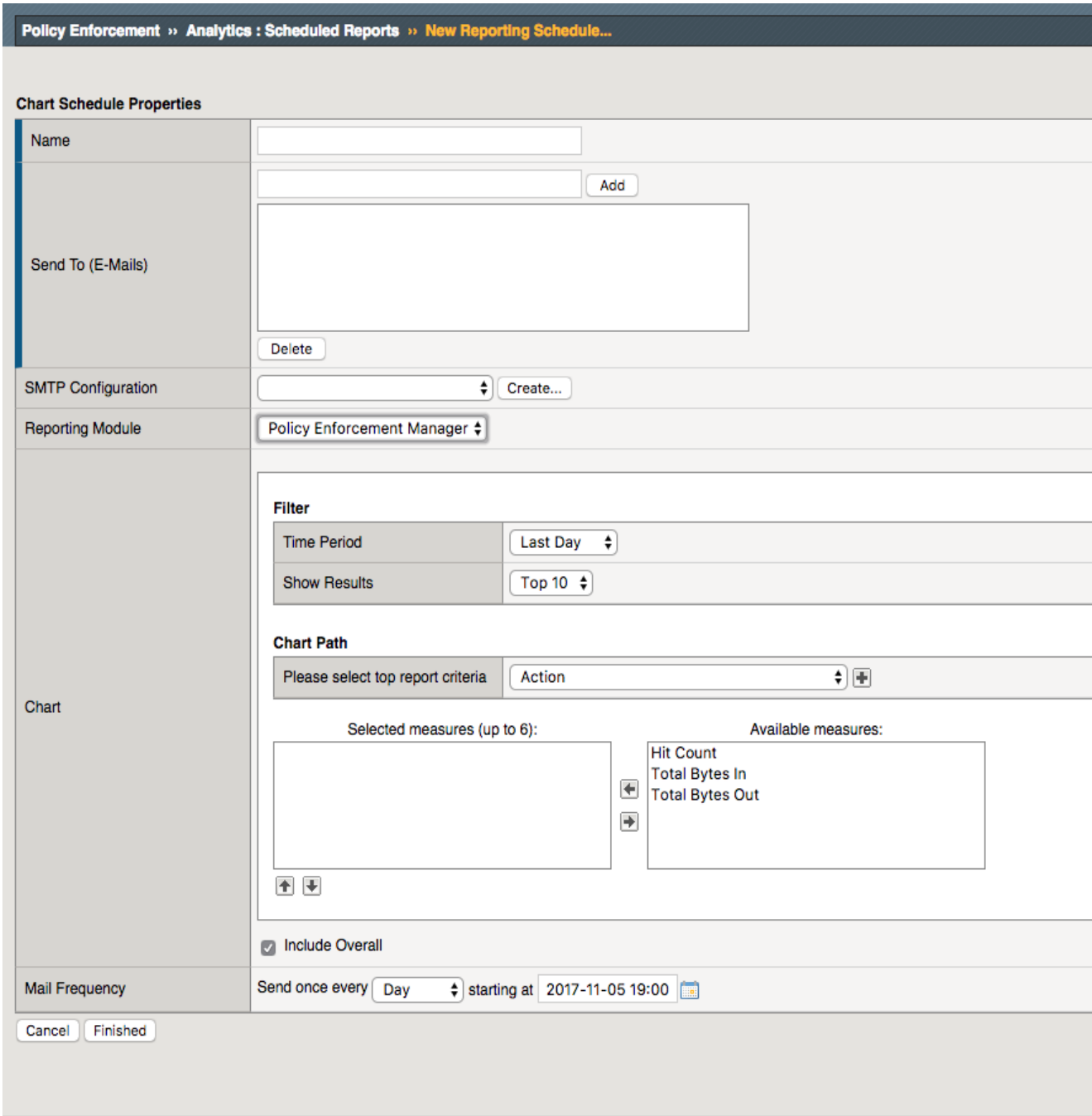
2. Click on Add Widget
3. Create a New Wdiget of your choice.
4. Explore the options within the Dashboard widgets for display
5. Click on Export, select PDF to generate report.

Task 6 - PEM Scheduled Reports

In this task we will configure a Scheduled PEM report.

Perform the following steps to complete this task:

1. Navigate to Policy Enforcement Analytics Scheduled Reports.



2. Explore the options for scheduled reporting.

6 Class 2: Introduction to ELK Stack (ELK Coolness)

This class covers the following topics:

- ELK Stack Overview
- ELK Stack build on Ubuntu
- F5 logging to ELK Stack
- ELK Stack:
  - Indexes
  - Navigation
  - Searches
  - Visualisations
  - Dashboards

Expected time to complete: **1.5 hours**

6.1 Module 1: ELK Stack Build Ubuntu Server

ELK stack from the previous module is made up of three key components:

- Logstash,
- Elasticsearch,
- Kibana.

There are many ready made ELK stack services that can be used:

- Docker https://elk-docker.readthedocs.io/
- AWS https://aws.amazon.com/elasticsearch-service/
- Azure https://azuremarketplace.microsoft.com/en-us/marketplace/apps/bitnami.elk?tab=Overview
- Elastic Cloud https://www.elastic.co/cloud

However, it is important to understand how the ELK stack is build, the configuration files and their purposes.

This module will guide you through the installation of ELK stack onto a ubuntu server.

External Reference Documentation:

https://www.elastic.co/guide/index.html

6.1.1 Lab 1.1: Install the Ubuntu Base

In this lab you will walk through installing the ubuntu base ready for ELK stack

Task 1 - GIT Clone Repo onto the Server

git clone https://github.com/jarrodlucia/bigip_elk_server <directory of choice>

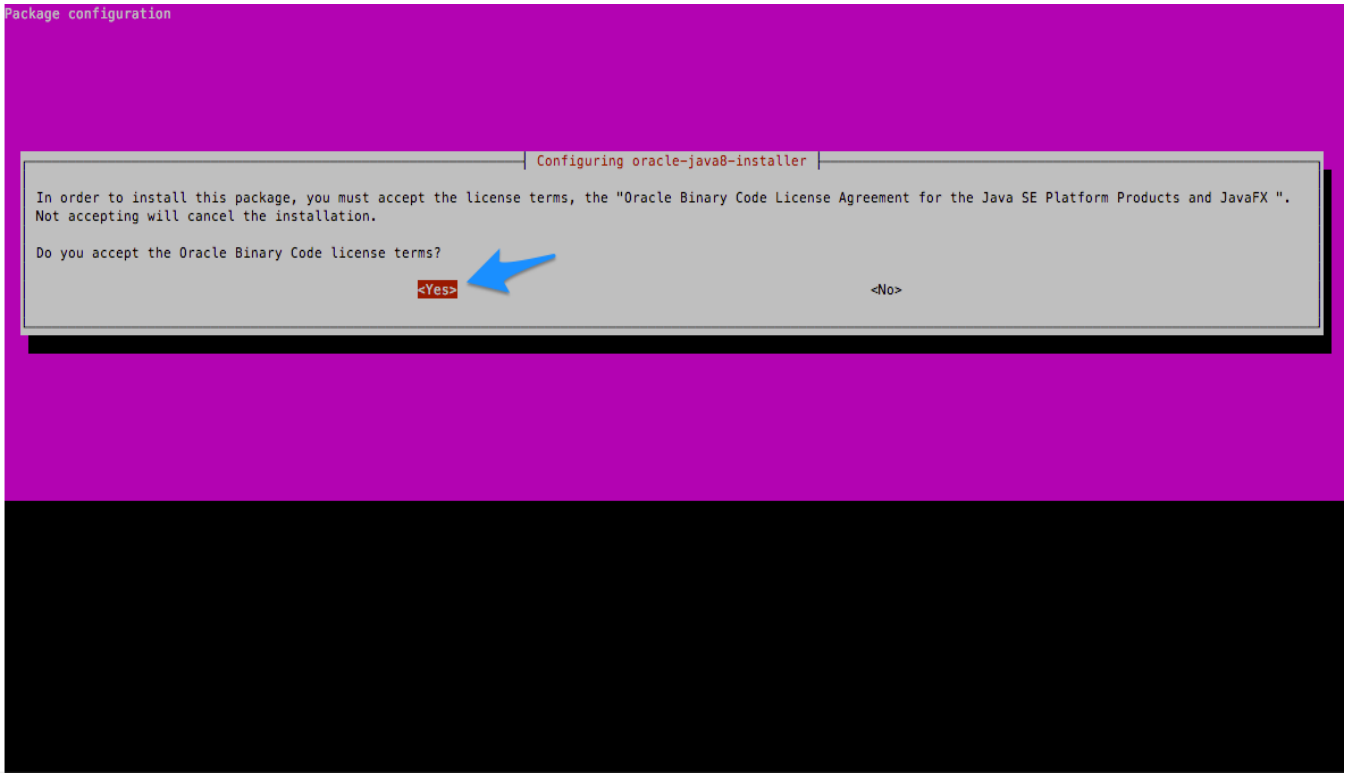
Task 2 - Install additional software required for ELK Stack

```
sudo apt-get install software-properties-common
sudo apt install curl
```

Task 3 - Add and install Java

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

Accept the Oracle License Agreement



Fix Below for Java8 Error (If Required)

```
sudo apt-get -y install oracle-java8-installer
sudo sed -i 's|JAVA_VERSION=8u144|JAVA_VERSION=8u152|' oracle-java8-installer.*
sudo sed -i 's|PARTNER_URL=http://download.oracle.com/otn-pub/java/jdk/8u144-b01/
↪090f390dda5b47b9b721c7dfaa008135/|PARTNER_URL=http://download.oracle.com/otn-pub/
↪java/jdk/8u152-b16/aa0333dd3019491ca4f6ddbe78cdb6d0/|' oracle-java8-installer.*
sudo sed -i 's|SHA256SUM_TGZ=
↪"e8a341ce566f32c3d06f6d0f0eeea9a0f434f538d22af949ae58bc86f2eeaae4"|SHA256SUM_TGZ=
↪"218b3b340c3f6d05d940b817d0270dfe0cfd657a636bad074dcabe0c111961bf"|' oracle-java8-
↪installer.*
sudo sed -i 's|J_DIR=jdk1.8.0_144|J_DIR=jdk1.8.0_152|' oracle-java8-installer.*
sudo apt-get -y install oracle-java8-installer
```

6.1.2 Lab 1.2: Install Elasticsearch

This lab will install the Elasticsearch component, It is recommended to install Elasticsearch as the first module.

Task 1 Install Repo and Keys

1. Download and install the public signing key:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

1. Save the repository definition to /etc/apt/sources.list.d/elastic-5.x.list:

```
echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a /
↪etc/apt/sources.list.d/elastic-5.x.list
sudo apt-get update
```

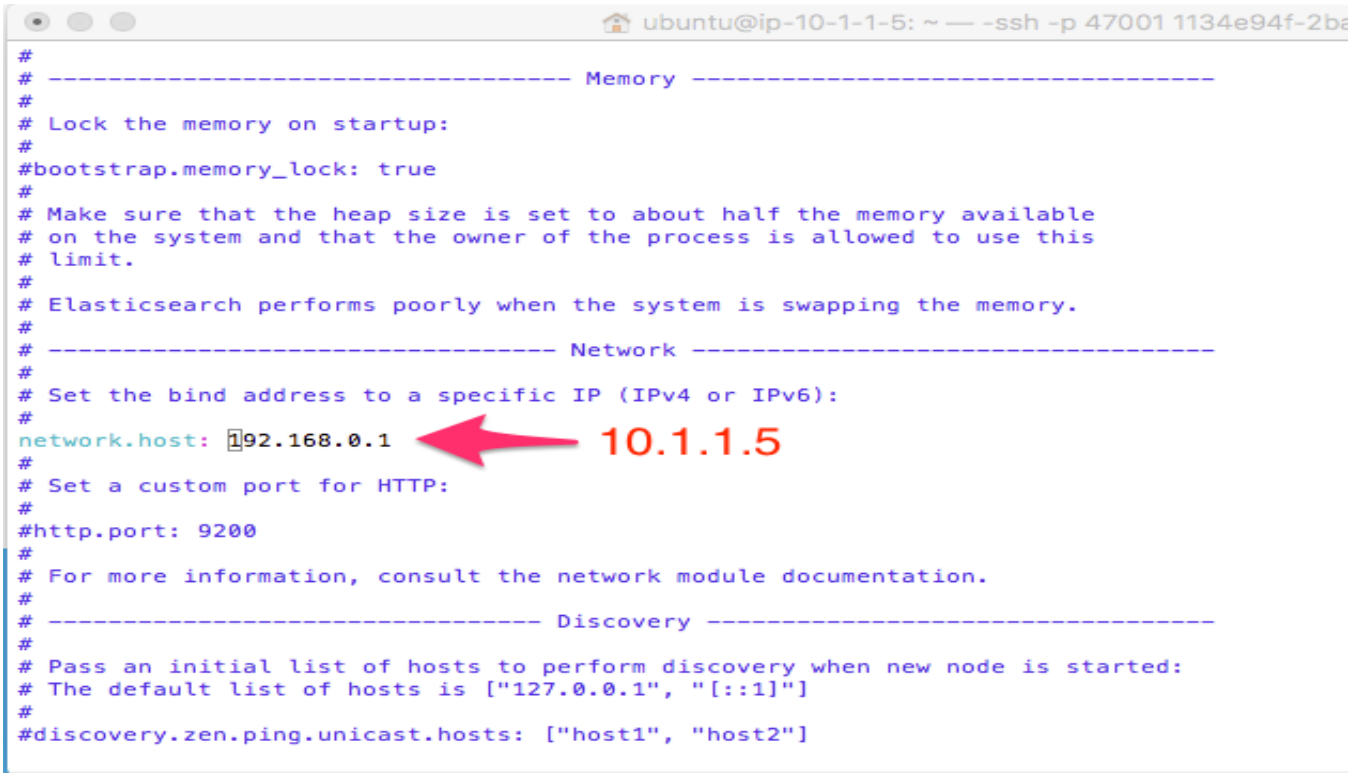
Task 2 Install elasticseach and setup system

1. Install Elasticsearch

```
sudo apt-get install elasticsearch
```

1. Edit config file to change bind address to Host address 10.1.1.5

```
sudo vi /etc/elasticsearch/elasticsearch.yml
```



1. Install additional plugins

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install ingest-geoip
```

1. Restart Elastic Search

```
sudo systemctl restart elasticsearch
```

1. Configure the system to start at boot

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
```

1. Checking Start / Stop / Status

```
sudo systemctl start elasticsearch.service
sudo systemctl stop elasticsearch.service
sudo systemctl status elasticsearch.service
```

```
ubuntu@ip-10-1-1-5:~$ sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-11-06 11:24:11 UTC; 17s ago
     Docs: http://www.elastic.co
 Main PID: 4425 (java)
   CGroup: /system.slice/elasticsearch.service
           └─4425 /usr/bin/java -Xms2g -Xmx2g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:

Nov 06 11:24:11 ip-10-1-1-5 systemd[1]: Starting Elasticsearch...
Nov 06 11:24:11 ip-10-1-1-5 systemd[1]: Started Elasticsearch.
ubuntu@ip-10-1-1-5:~$
```

6.1.3 Lab 1.3: Install Kibana

In this lab we will install Kibana

Task 1 Install Kibana

1. Install Kibana

```
sudo apt-get install kibana
```

2. Change config file to set Outside IP address

```
sudo vi /etc/kibana/kibana.yml
```

Note: Kibana is served by a back end server. This setting specifies the port to use. Server port is set as default Kibana Port 5601. Server host should be set to the UDF Management IP address 10.1.1.5 as we will be accessing this via the Linux Jumphost. The URL of the Elasticsearch instance to use for all your queries.

- server.port: 5601
- server.host: “10.1.1.5”
- elasticsearch.url: “http://10.1.1.5:9200”

```
ubuntu@ip-10-1-1-5: ~ — -ssh -p 47001 1134e94f-2ba0-4477-899a-f7b1549a1581.access.udf.f5.com — 15

# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "10.1.1.5"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only affects
# the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwarding requests
# to Kibana. This setting cannot end in a slash.
#server.basePath: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://10.1.1.5:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "discover"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
#elasticsearch.username: "user"
#elasticsearch.password: "pass"

"/etc/kibana/kibana.yml" 104L, 4644C
```

1. Kibana restart

```
sudo systemctl restart kibana.service
```

2. To configure Kibana to start automatically when the system boots up, run the following commands:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service
```

3. Kibana Control

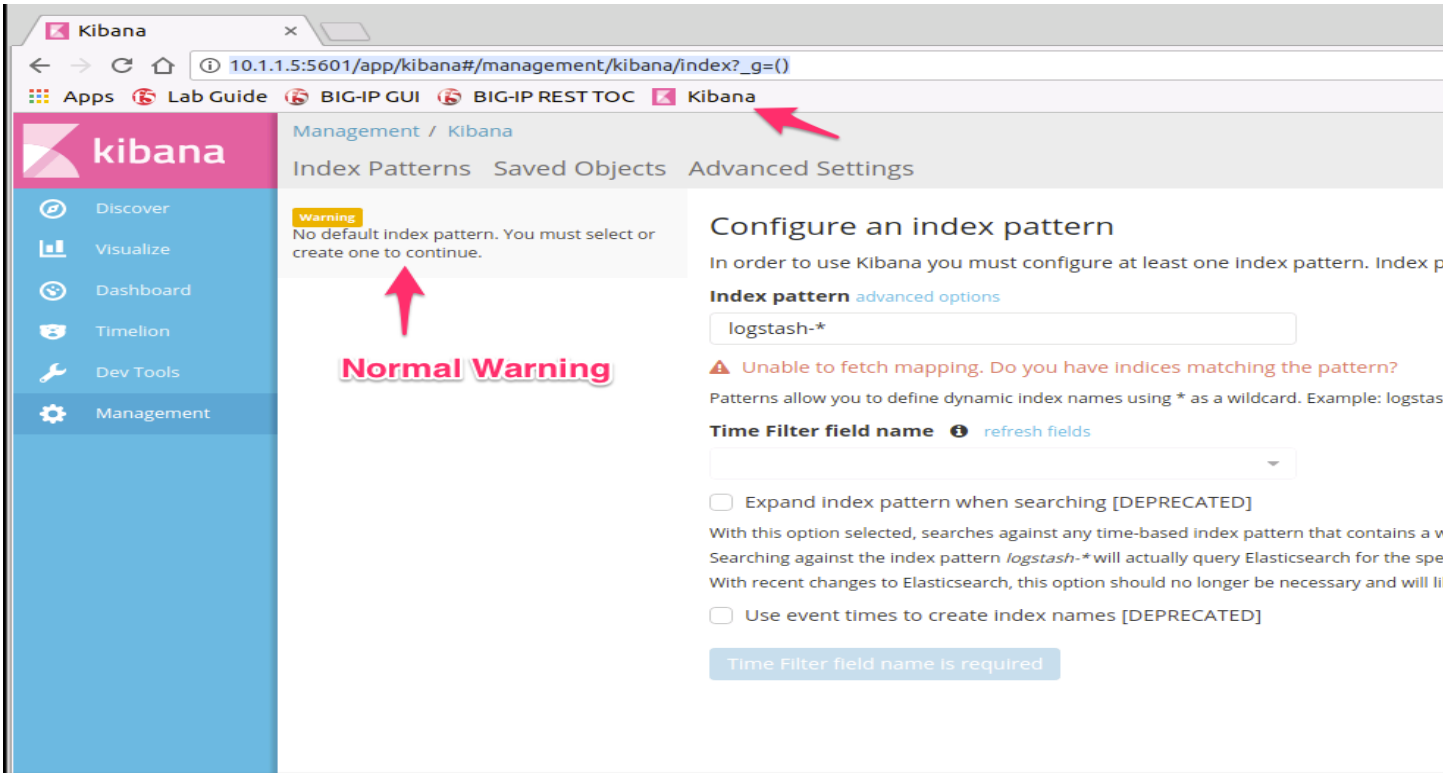
```
sudo systemctl start kibana.service
sudo systemctl stop kibana.service
```

4. Check Kibana is running via command-line:

```
ubuntu@ip-10-1-1-5: ~ — -ssh -p 47001 1134e94f-2ba0-4477-899a-f7b154
~
~
~
~
~
~
~
~
~
~
~
~
~
ubuntu@ip-10-1-1-5:~$ sudo systemctl status kibana.service
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-11-06 11:49:38 UTC; 47s ago
 Main PID: 4782 (node)
   CGroup: /system.slice/kibana.service
           └─4782 /usr/share/kibana/bin/../node/bin/node --no-warnings /usr/share/kibana/bin/.

Nov 06 11:49:40 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:40Z","t
Nov 06 11:49:40 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:40Z","t
Nov 06 11:49:40 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:40Z","t
Nov 06 11:49:40 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:40Z","t
Nov 06 11:49:41 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:41Z","t
Nov 06 11:49:41 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:41Z","t
Nov 06 11:49:41 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:41Z","t
Nov 06 11:49:46 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:46Z","t
Nov 06 11:49:47 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:47Z","t
Nov 06 11:49:47 ip-10-1-1-5 kibana[4782]: {"type":"log","@timestamp":"2017-11-06T11:49:47Z","t
ubuntu@ip-10-1-1-5:~$
```

1. Access Kibana via Linux Jumpbox to verify access



6.1.4 Lab 1.4: Install Logstash

Install Logstash

Task 1 - Install Logstah

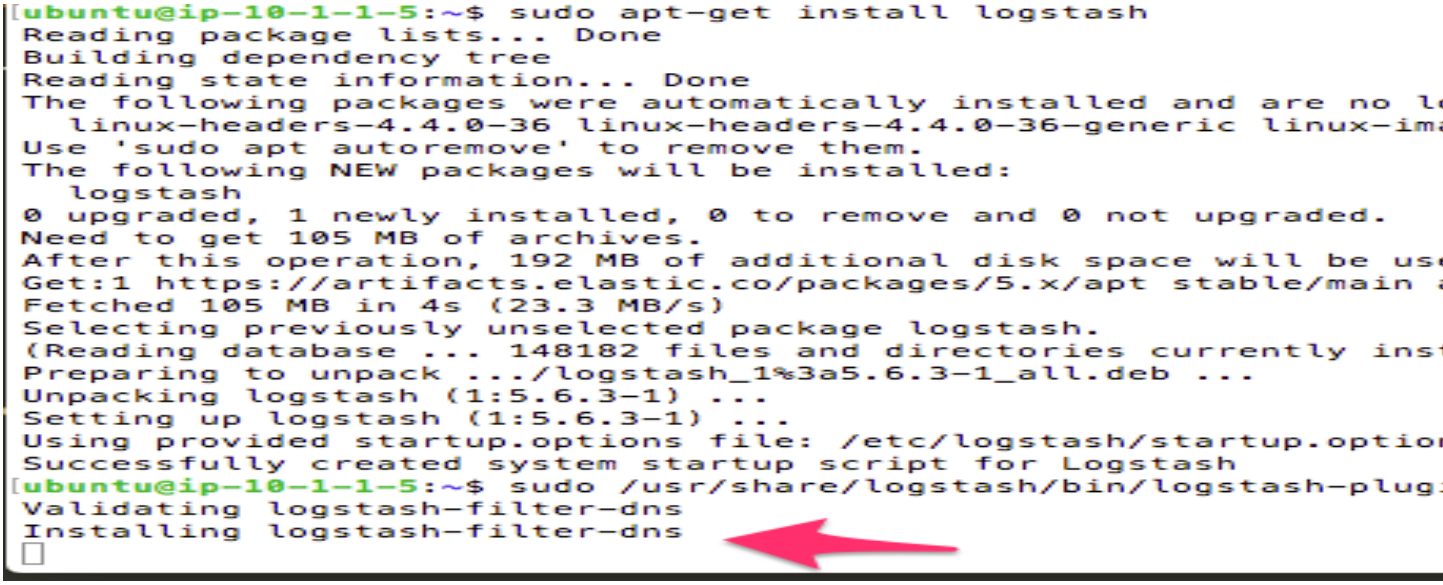
1. Install Logstash

```
sudo apt-get install logstash
```

1. Install Additional Plugins

```
sudo /usr/share/logstash/bin/logstash-plugin install logstash-filter-dns
sudo /usr/share/logstash/bin/logstash-plugin install logstash-filter-geoip
```

Note: Be patient with plugin install it can take a few moments



1. Copy or Create new file to Directory /etc/logstash/conf.d/

```
sudo cp <git clone directory>/config_files/logstash.conf /etc/logstash/conf.d/
↪logstash.conf
sudo vi /etc/logstash/conf.d/logstash.conf
```

1. Logstash restart

```
sudo systemctl restart logstash.service
```

1. Check logstash started correctly with no errors from logstash.conf file

```
ubuntu@ip-10-1-1-5:~$ tail -f /var/log/logstash/logstash-plain.log
[2017-11-06T12:50:24,971][INFO ][logstash.outputs.elasticsearch] Running health check to see if an
p://localhost:9200/, :path=>"/"}
[2017-11-06T12:50:24,976][WARN ][logstash.outputs.elasticsearch] Restored connection to ES instanc
[2017-11-06T12:50:24,980][INFO ][logstash.outputs.elasticsearch] Using mapping template from {:pat
[2017-11-06T12:50:24,981][INFO ][logstash.outputs.elasticsearch] Attempting to install template {:
1, "settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>{"_default_"=>{"_all"=>{"enabled"=>tr
=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>f
pe"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword
{"type"=>"date", "include_in_all"=>false}, "@version"=>{"type"=>"keyword", "include_in_all"=>false
>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"}, "longitude"=>{"type
[2017-11-06T12:50:24,985][INFO ][logstash.outputs.elasticsearch] Installing elasticsearch template
[2017-11-06T12:50:25,001][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output {:class=
9200"]}
[2017-11-06T12:50:25,007][INFO ][logstash.filters.geoip   ] Using geoip database {:path=>"/usr/sha
oip-4.3.1-java/vendor/GeoLite2-City.mmdb"}
[2017-11-06T12:50:25,037][INFO ][logstash.filters.geoip   ] Using geoip database {:path=>"/usr/sha
oip-4.3.1-java/vendor/GeoLite2-City.mmdb"}
[2017-11-06T12:50:25,044][INFO ][logstash.pipeline        ] Starting pipeline {"id"=>"main", "pipe
ch.delay"=>5, "pipeline.max_inflight"=>500}
[2017-11-06T12:50:25,188][INFO ][logstash.pipeline        ] Pipeline main started
[2017-11-06T12:50:25,212][INFO ][logstash.agent           ] Successfully started Logstash API endp
```

1. To configure Logstash to start automatically when the system boots up, run the following commands:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable logstash.service
```

1. Logstash Control

```
sudo systemctl start logstash.service
sudo systemctl stop logstash.service
sudo systemctl status logstash.service
```

logstash.conf

```
input {
    tcp {
        port => 5516
        type => afm
    }
    tcp {
        port => 5515
        type => dns
    }
    tcp {
        port => 5514
        type => pem
    }
}

filter {
    if [type] == 'pem' {
        kv {
          source => "message"
         field_split => ","
       }
    }
    if [type] == 'afm' {
        kv {
          source => "message"
```

```
            field_split => ","
        }
          geoip {
              source => "SourceIp"
              target => "SourceIp_geo"
              add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}"]
              add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
          }
          geoip {
              source => "DestinationIp"
              target => "DestinationIp_geo"
              add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}"]
              add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
          }
          mutate {
              convert => [ "[geoip][coordinates]", "float"]
          }
      }
      if [type] == 'dns' {
          kv {
            source => "message"
            field_split => ","
        }
      }
  }

  output {
      if [type] == 'pem' {
        elasticsearch {
        hosts => ["10.1.1.5:9200"]
        index => "pem-%{+YYYY.MM.dd}"
        template_name => "pem"
      }
      }
      if [type] == 'afm' {
        elasticsearch {
        hosts => ["10.1.1.5:9200"]
        index => "afm-%{+YYYY.MM.dd}"
        template_name => "afm"
      }
      }
      if [type] == 'dns' {
        elasticsearch {
        hosts => ["10.1.1.5:9200"]
        index => "dns-%{+YYYY.MM.dd}"
        template_name => "dns"
      }
      }
      stdout {}
  }
```

6.1.5 Lab 1.5: Configure elasticsearch templates

Templates are used to create mappings between logstash and elasticsearch. Without the mappings elasticsearch will create automatic mappings however these will be elasticsearch's best guess as to the field. In most cases this will default to `text`. This means many of the fields such as IP address's will be searchable but not able to be used in Visualisations.

Upload elasticsearch templates and mappings. There are multiple way this can be achieved. The most common ways are cURL and a REST based program such as POSTMAN. Feel free to use whichever method you are most comfortable with.

Note: **RECOMMENDATION** Use cURL for the uploading of the templates with json file. POSTMAN is useful for Elasticsearch management once the template are in place.

Task 1 Option1 - Install module templates in Elasticsearch via cURL

1. Install Index Templates into Elastic Search for the required modules

   cd `<git clone directory>/json/` **git clone directory from Lab 1**

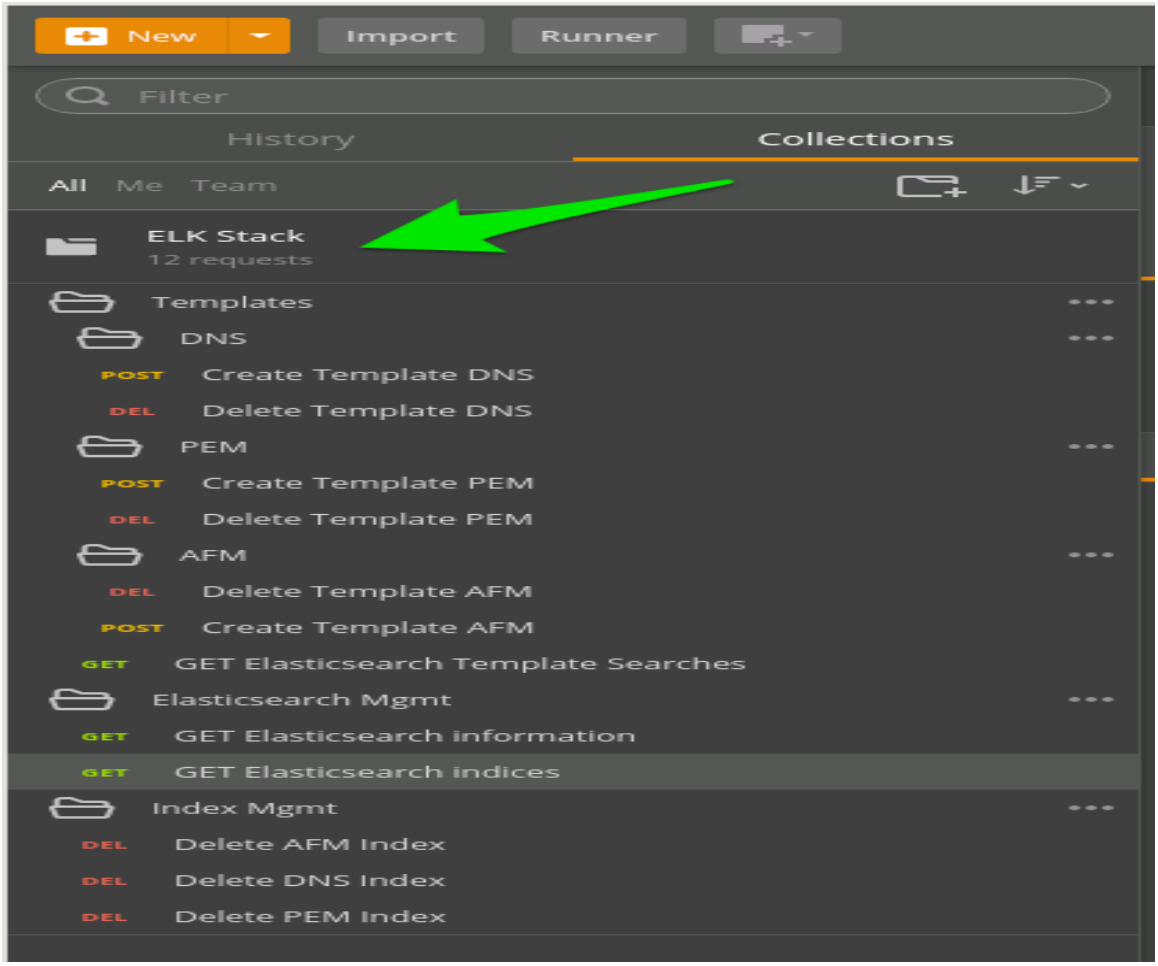
```
curl -XPUT http://localhost:9200/_template/pem?pretty -d @pem_mapping.json
curl -XPUT http://localhost:9200/_template/afm?pretty -d @afm_mapping.json
curl -XPUT http://localhost:9200/_template/dns?pretty -d @dns_mapping.json
```

Task 1 Option1 - Install module templates in Elasticsearch via POSTMAN

1. Import ELK Postman Collection and Environment
2. Click the 'Import from Link' tab. Paste the following URL into the text box and click 'Import'

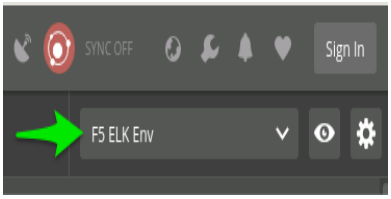
   https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/ELKStack.postman_collection.json

3. You should now see a collection named 'F5 ELK' in your Postman Collections sidebar:



4. Import the Environment file by clicking 'Import' -> 'Import from Link' and pasting the following URL and clicking 'Import':

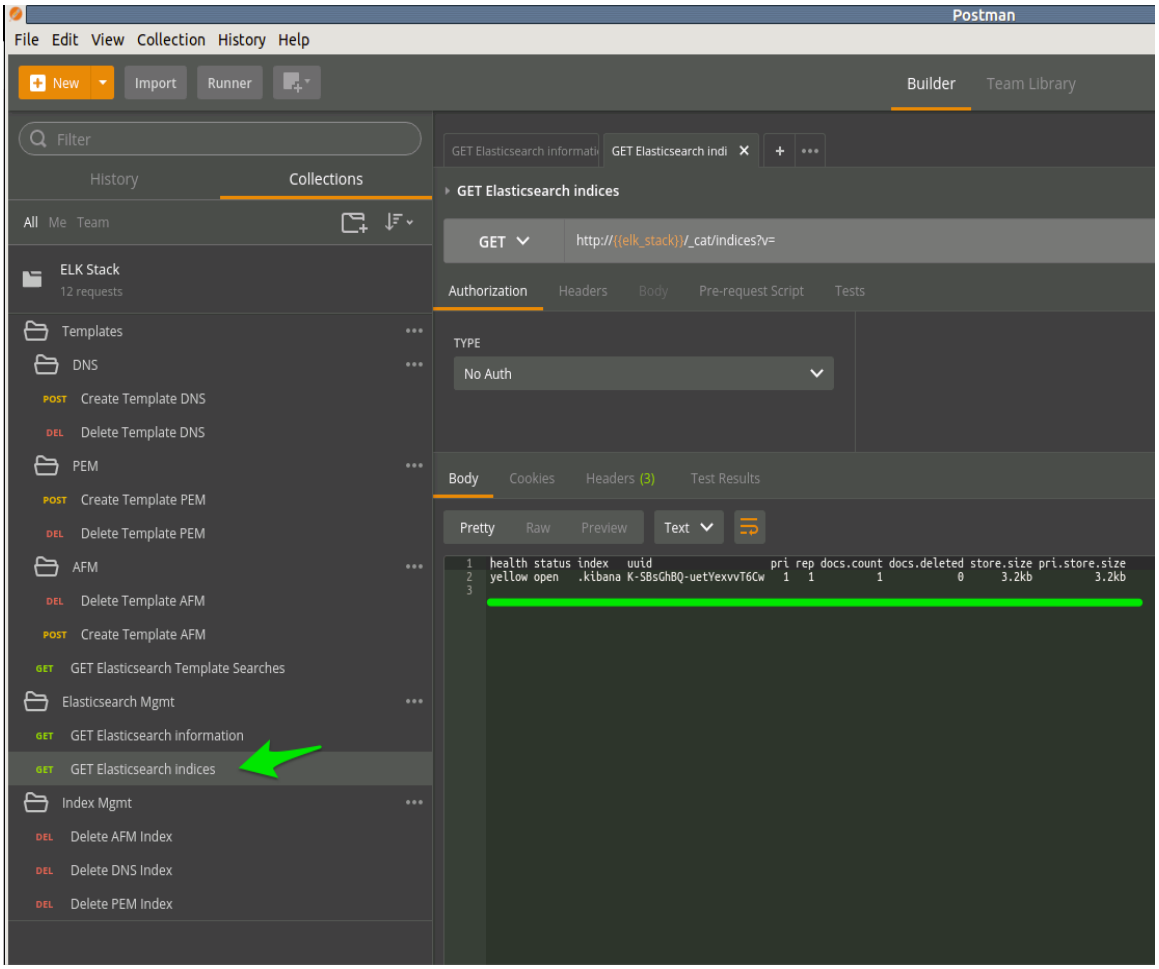
   `https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/postman_collections/F5ELKEnv.postman_environment.json`



1. Click on GET Elasticsearch information, **HIT SEND**.

You should see cluster information regarding elasticsearch

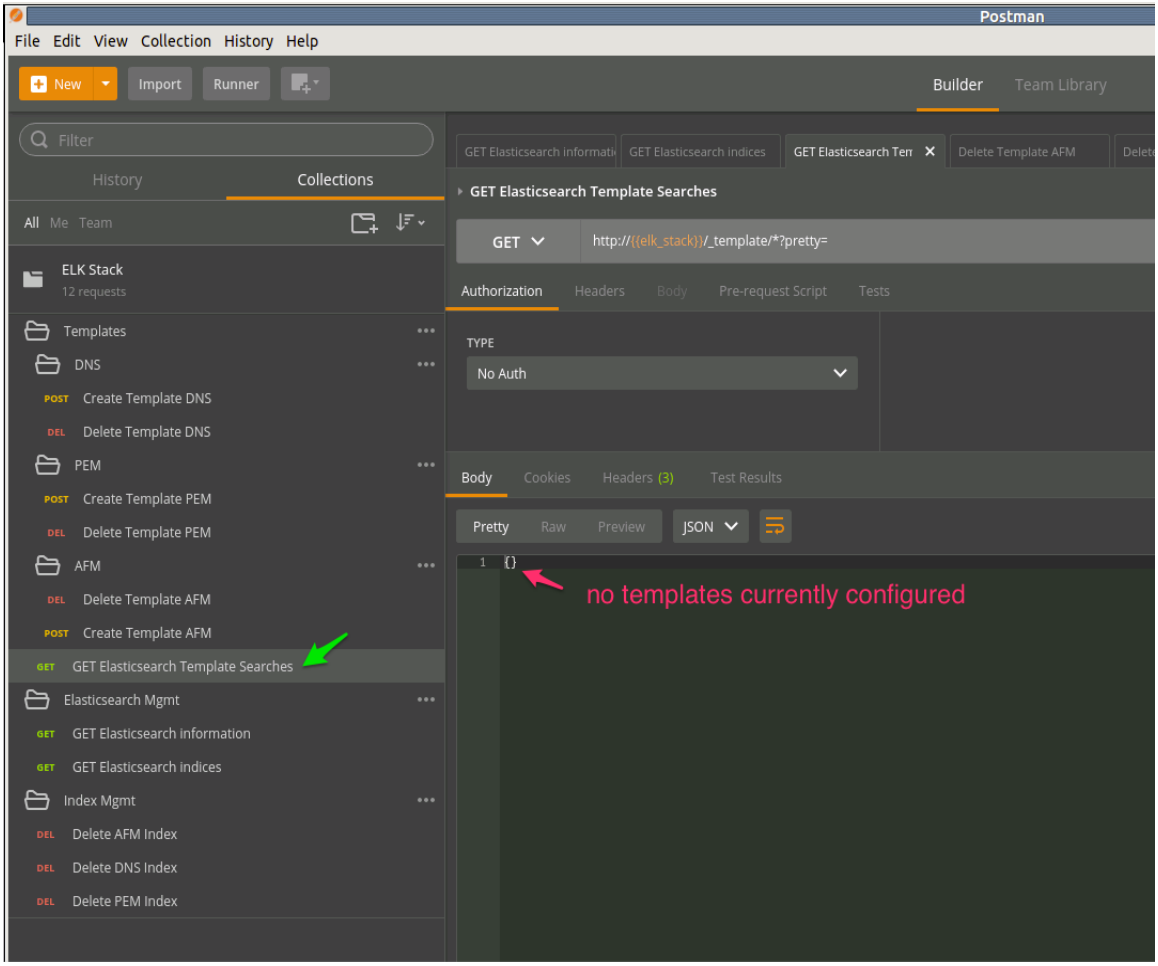
1. Click on GET Elasticsearch indices, **HIT SEND**.



You should see the current index's and information regarding each index.

We will use this command to observe the creation of new indexes

1. Click on GET Elasticsearch Template Searches, **HIT SEND**

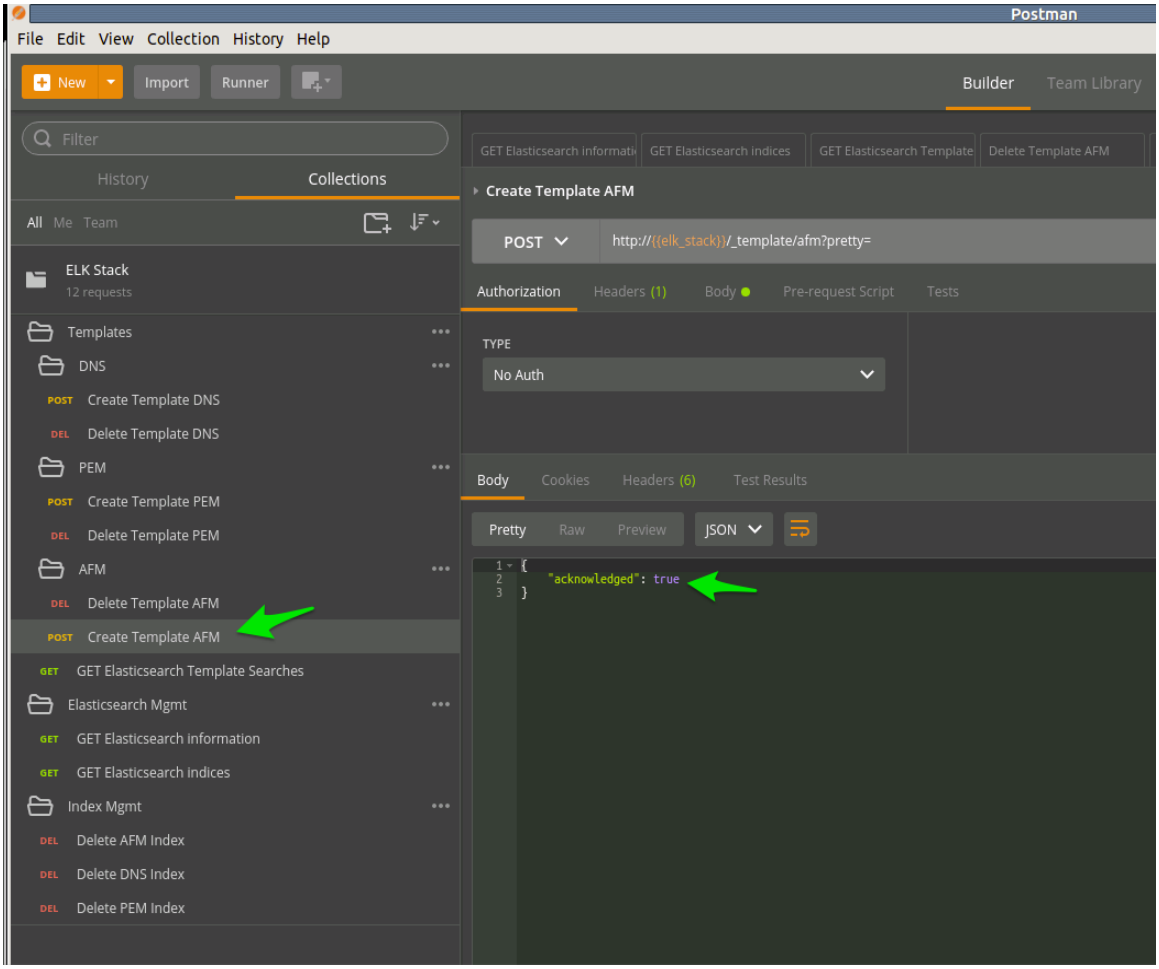


You should see any current templates listed.

Note: New Install will **NOT** contain any templates showing {}

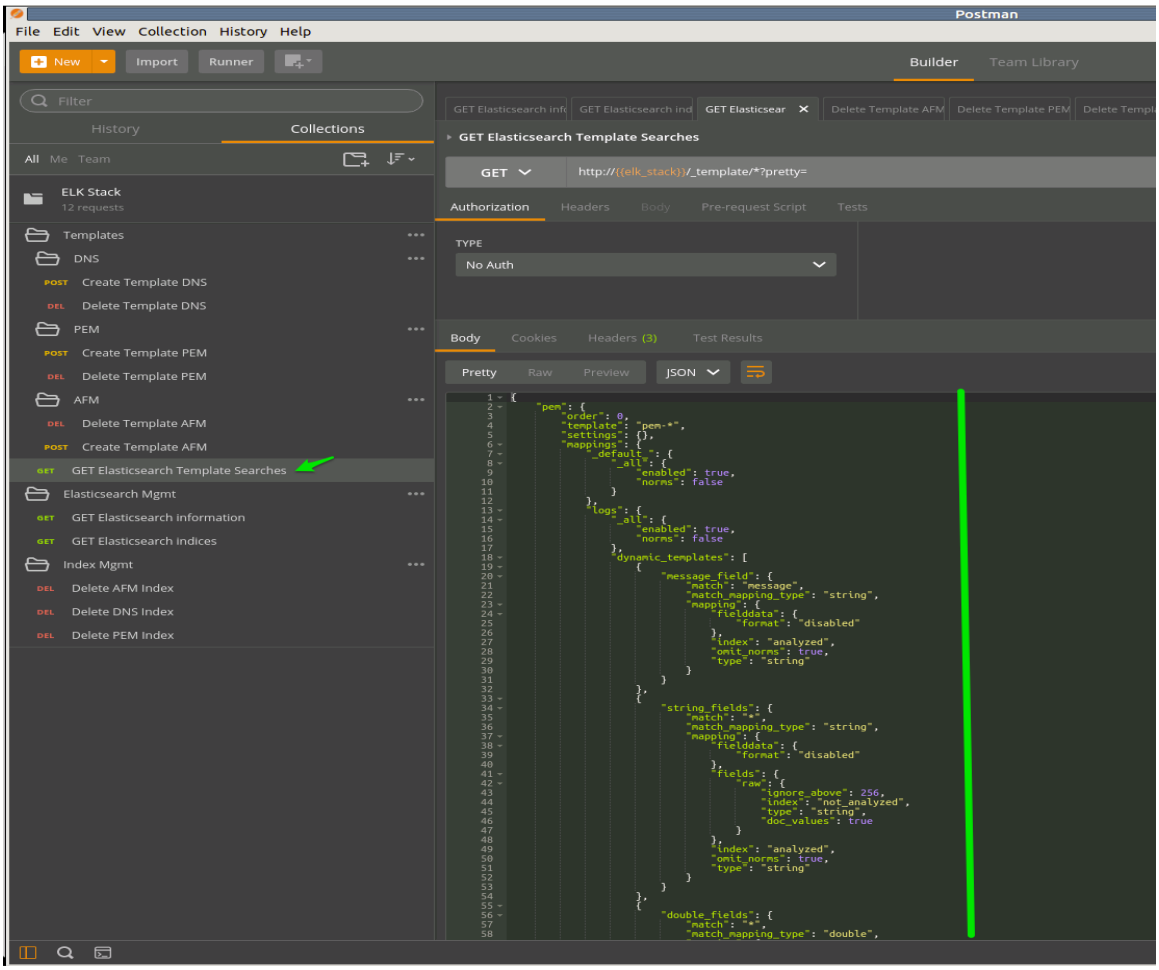
---

1. Click on Create Template AFM + PEM + DNS **Install all templates**



Note: Create all templates from the POSTMAN collection

1. Verify templates created and exist. Click on GET Elasticsearch Template Searches



Note: Look through the template JSON outputted by POSTMAN. Verify and check that the three templates created are present.

6.1.6 Lab 1.6: Send Logs to ELK Stack

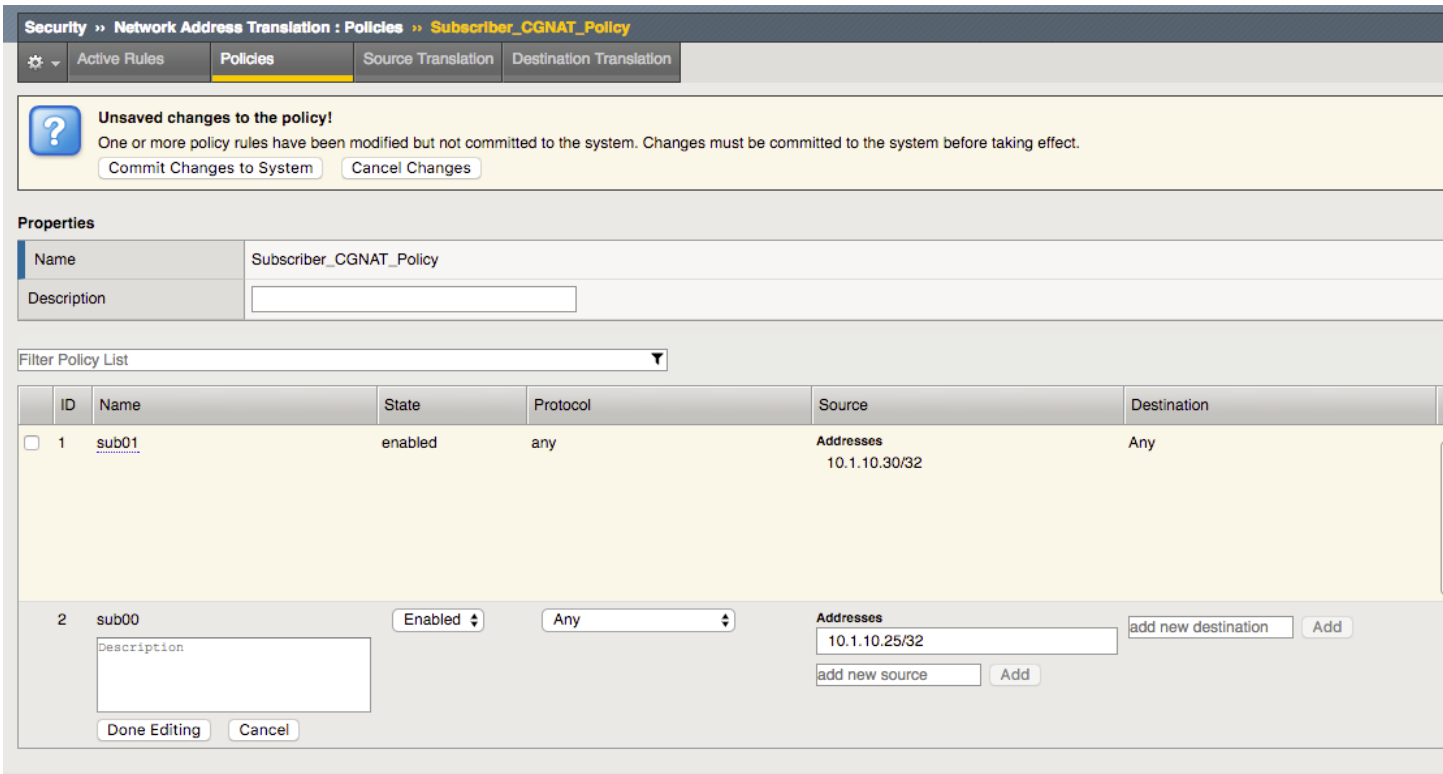
Configure f5 for logging to new ELK stack

Check that data is arriving at ELK stack

Task 1 - Confirm BIG-IP is sending logs to ELK Stack

1. Confirm via TMUI that the setup from **Class 1 Lab 2.1**

Update AFM Reporting to include what was not included in previous lab.



Note:

Make sure the correct port is allocated as per previous Logstash configuration

- Pool = tcp server:5514 - PEM
- Pool = tcp server:5515 - DNS
- Pool = tcp server:5516 - AFM/CGNAT

---

1. Confirm Data is arrinving on server

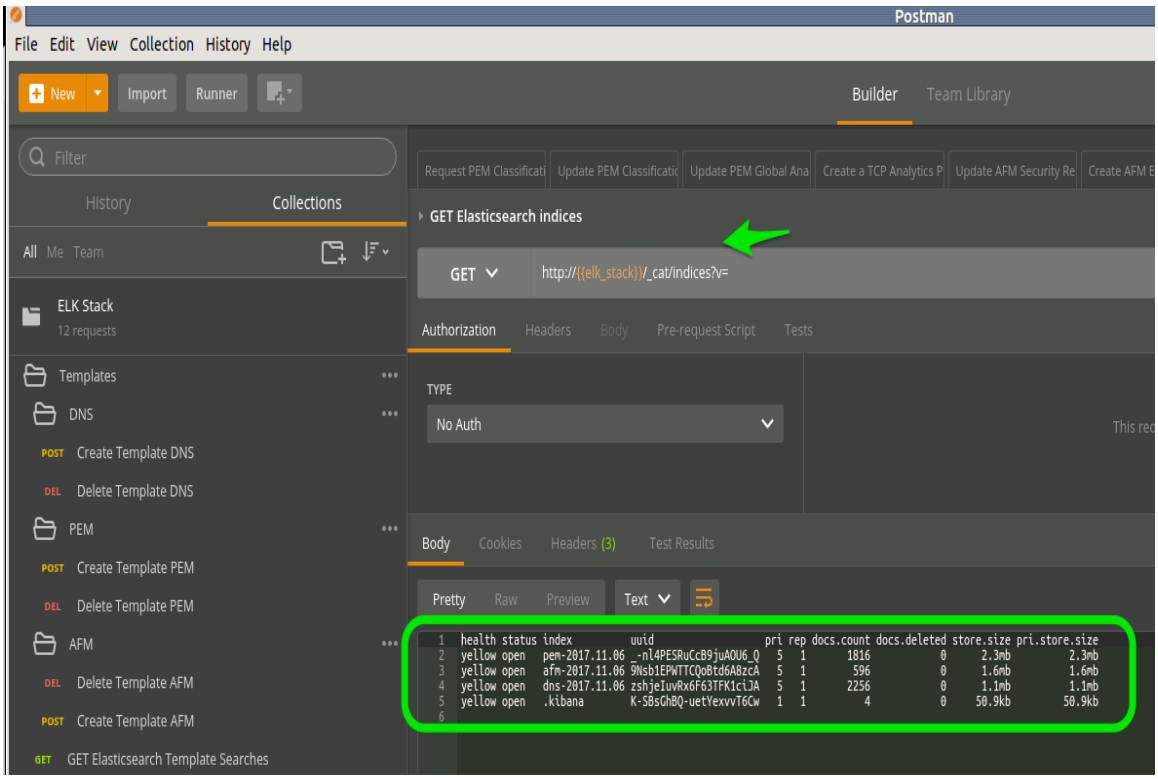
sudo tcpdump -i eth1 port 5514

1. Check that Data is arriving in the Index

curl ‘localhost:9200/_cat/indices?v’

```
ubuntu@ip-172-31-9-140:~$ curl 'localhost:9200/_cat/indices?v'
health status index          uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   afm-2017.10.30 KsbHmgC5T_-Pb6zCiWq04A   5   1      99099            0       96mb           96mb
yellow open   afm-2017.10.20 vNiFLuhoT3yZcCQlQlLNng   5   1      91377            0     93.4mb         93.4mb
yellow open   pem-2017.11.06 WxNj_Bn6SOqgXka4pqboRw   5   1     107905            0     68.8mb         68.8mb
yellow open   dns-2017.10.23 6V_oD6vgS5CI_FNZKQ81mw   5   1      23192            0      6.7mb          6.7mb
yellow open   pem-2017.11.01 6eBHsrsaR9W9ZswMBi1jQg   5   1     241562            0    154.5mb        154.5mb
yellow open   dns-2017.10.26 e8pb_CnwQg2Kb_wTuDkS9A   5   1      23354            0      6.2mb          6.2mb
yellow open   afm-2017.11.01 gsYtMwETQcyPXWEWuIQkVg   5   1     124940            0    125.7mb        125.7mb
yellow open   pem-2017.10.25 jczYTnKQQ5myhMONIGqe3Q   5   1     197893            0    134.5mb        134.5mb
yellow open   pem-2017.10.24 klMyNpnHQJakQhaipQcdEQ   5   1     164781            0    112.4mb        112.4mb
yellow open   dns-2017.10.27 FIls_BWLRQuyNsDQQYJkgQ   5   1      42133            0     12.2mb         12.2mb
yellow open   afm-2017.10.29 AeT2IKi3S_OuxS0QKylkkg   5   1     109143            0    107.4mb        107.4mb
yellow open   pem-2017.10.20 n4u6ln-hRcO5dK0ly_Uj5w   5   1      97583            0       76mb           76mb
yellow open   dns-2017.10.19 2gkvW3ibRLyxsHeKBejqTw   5   1      33939            0      9.7mb          9.7mb
yellow open   dns-2017.10.20 Xudl6KAFTEqIKlj7oATUGg   5   1      42111            0     12.7mb         12.7mb
yellow open   pem-2017.10.31 GCMJjdPTSp-kFhUkLayG2A   5   1     184746            0    135.6mb        135.6mb
yellow open   afm-2017.10.21 _SS3GGdBSAOEvXMRWZexAA   5   1     120292            0    120.8mb        120.8mb
yellow open   dns-2017.10.25 Ujj6kT2IR_uZ02ZlpT9GAA   5   1      30585            0      8.5mb          8.5mb
yellow open   afm-2017.10.25 oSVqB6cvSuy2-0zvD_EHCQ   5   1     119819            0    111.9mb        111.9mb
yellow open   pem-2017.11.02 rlOXsTJSR0KhLmUuzWLxLA   5   1     360479            0    222.5mb        222.5mb
yellow open   dns-2017.10.31 y144oZxYRDqONMIN5W56dA   5   1      35611            0     10.8mb         10.8mb
yellow open   dns-2017.11.02 BSRK9Yj6QNmYwWBAnSEGwA   5   1      19036            0        6mb            6mb
yellow open   afm-2017.11.02 2oZ_ckfvSYSZgM0AvwURJQ   5   1     105112            0    121.9mb        121.9mb
yellow open   dns-2017.11.06 47rN2rM7QyScu7WvPWihZA   5   1       4558            0      2.4mb          2.4mb
yellow open   dns-2017.10.24 p_GXqXSLRNaeACEkr3IEkQ   5   1      25447            0      7.1mb          7.1mb
yellow open   .kibana        GiC8XwNBTGKXnsw-fWyevg   1   1         58           20    208.4kb        208.4kb
```

or via POSTMAN



6.1.7 Lab 1.7: Create Index and Import Pre-Configured

Index's are elasticsearchs way of storing `documents` in `shards`. When index's are created the mapping templates we uploaded before are used to map each of the fields to a type. This is only done once when the index is created

---

Note: If mappings are changed are updates required the "index" will have to be deleted, the template deleted and mapping changed and template added. At this point re-creating the index will remap to the new template

---

This Lab will focus on creating the index's for each module based on logstash in **Lab4**

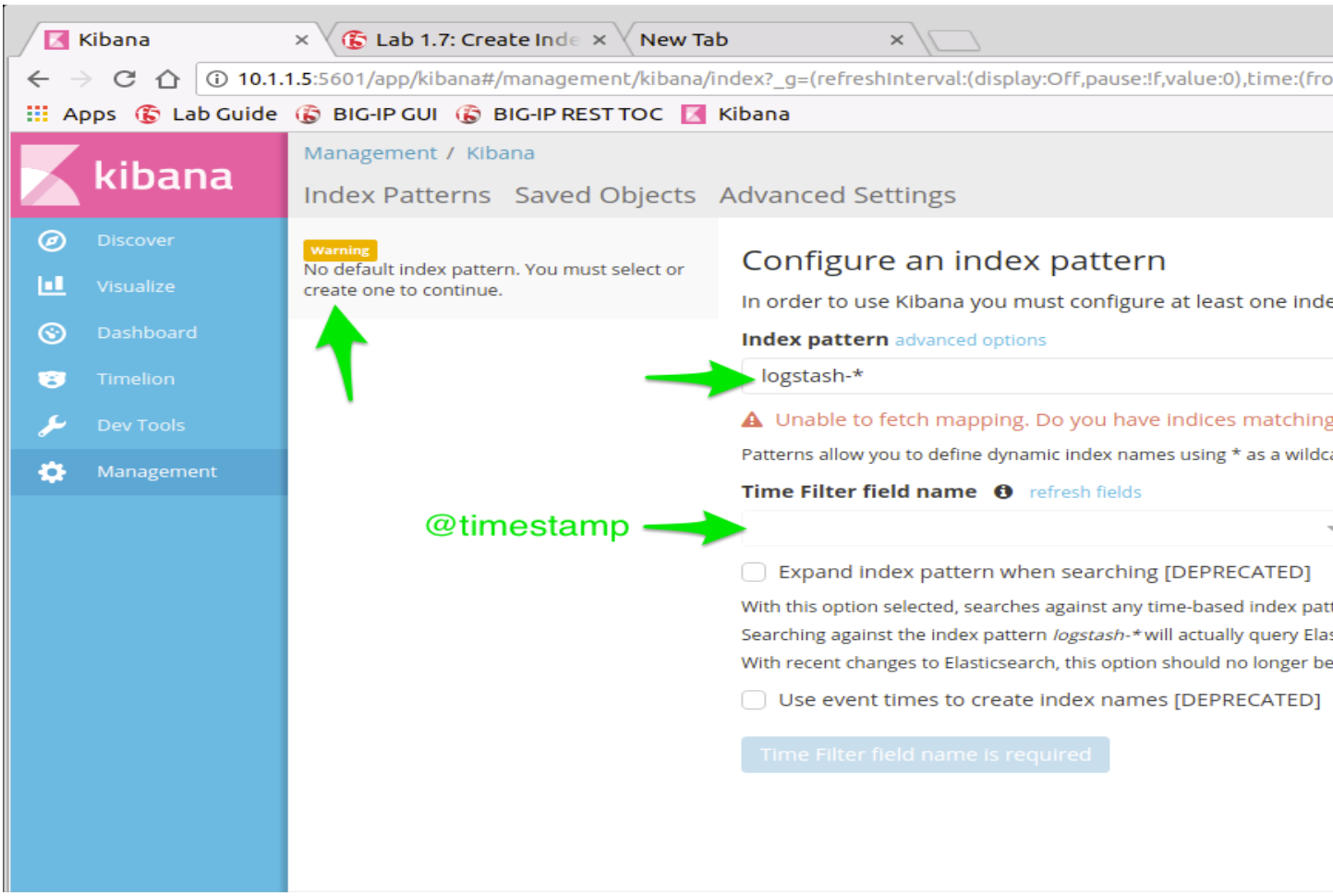
We will import the prepared f5 module json kibana searches / virtuals / and dashboards.

Task 1 - Create Kibana Index's

1. Configure Indexes in Kibana

Configure the first and default index

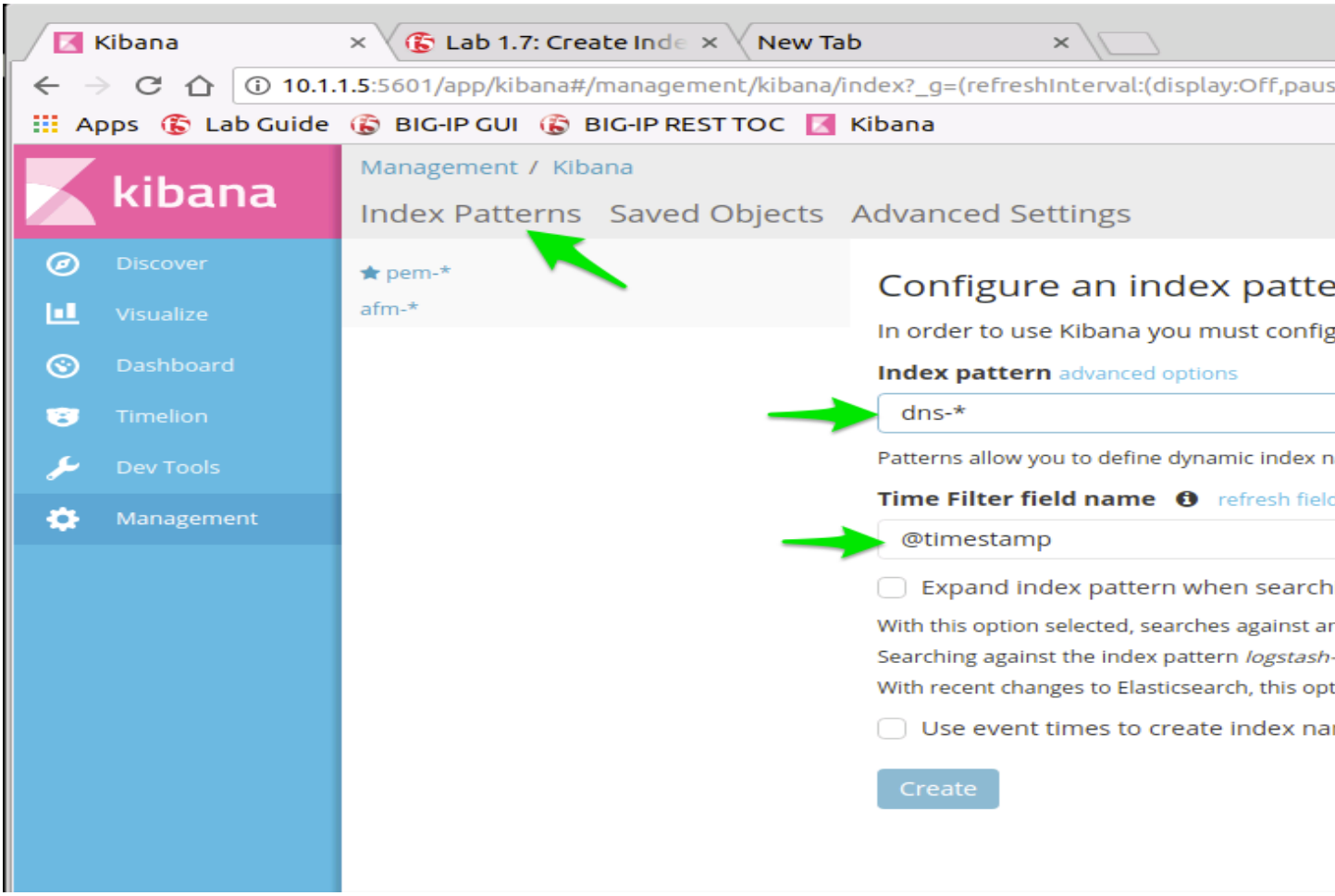
- index pattern = `pem-*`
- select `@timestamps`



- index pattern = `afm-*`
- select `@timestamps`

Follow PEM example above for AFM

- index pattern = `dns-*`
- select `@timestamps`



Task 2 - Import preconfigured Kibana json's

Searches / Visualisation and Dashboards

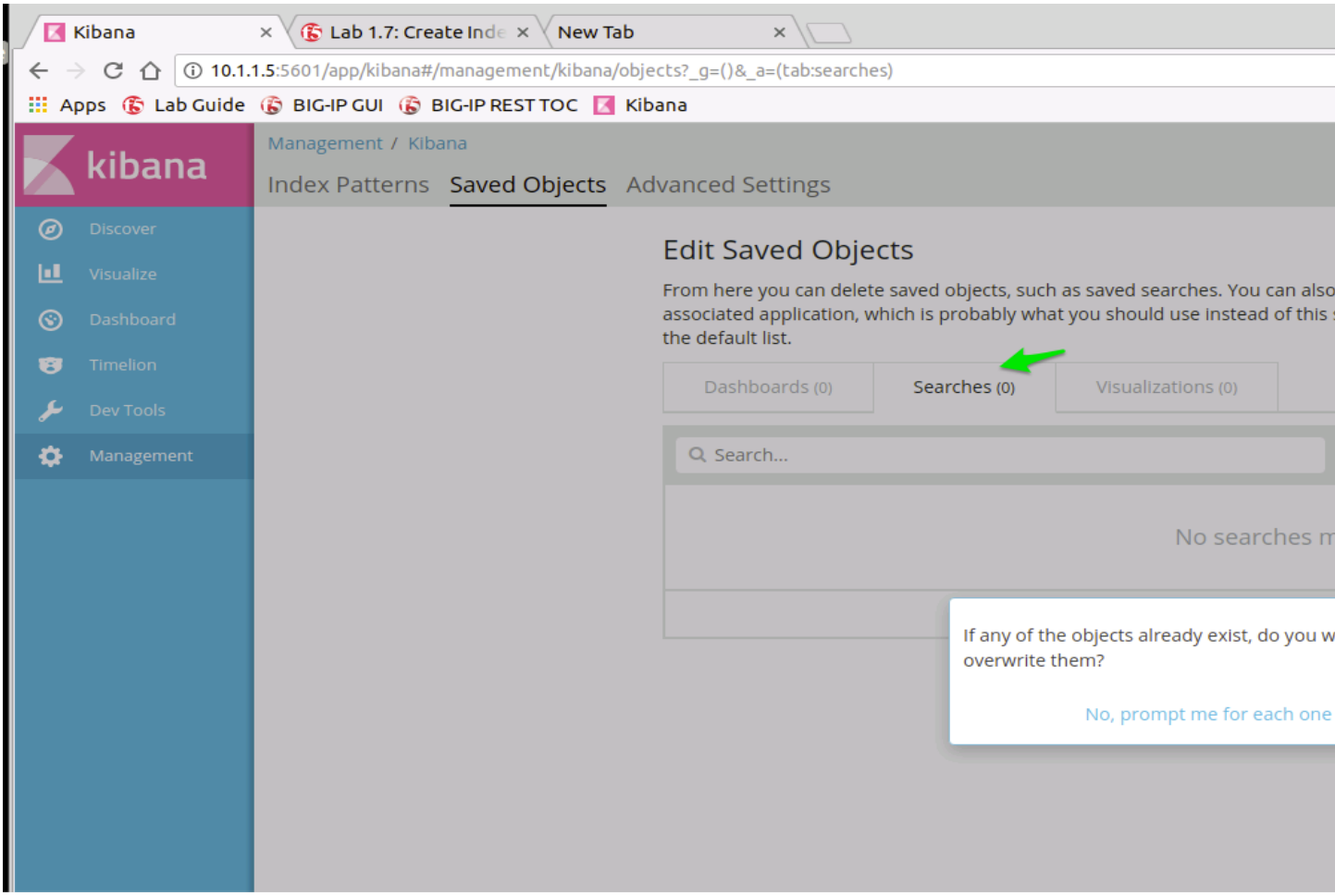
1. Import object data into Kibana

Import the JSON files in the following order:

- Searches
- Visualisations
- Dashboards

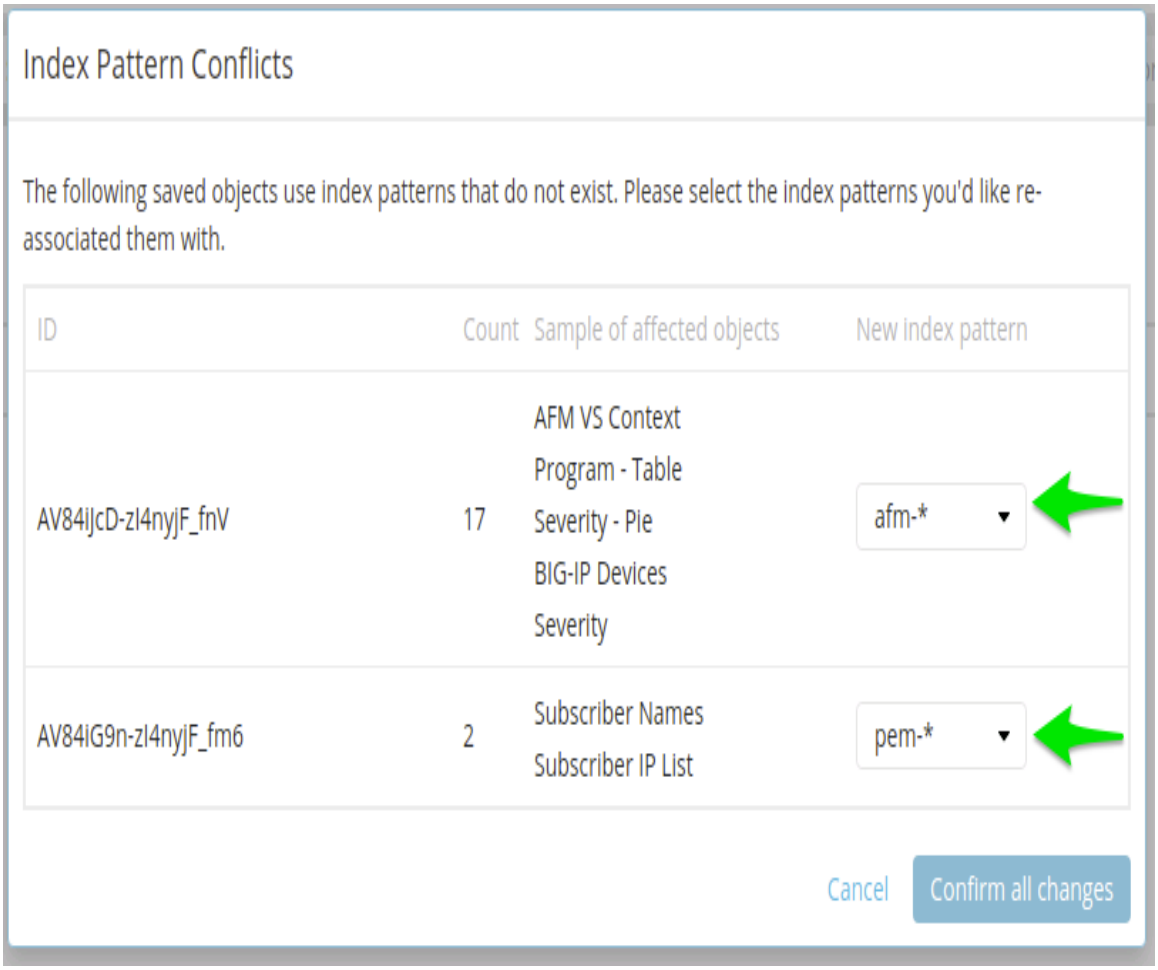
Searches

```
https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/
json/elk_searches.json
```



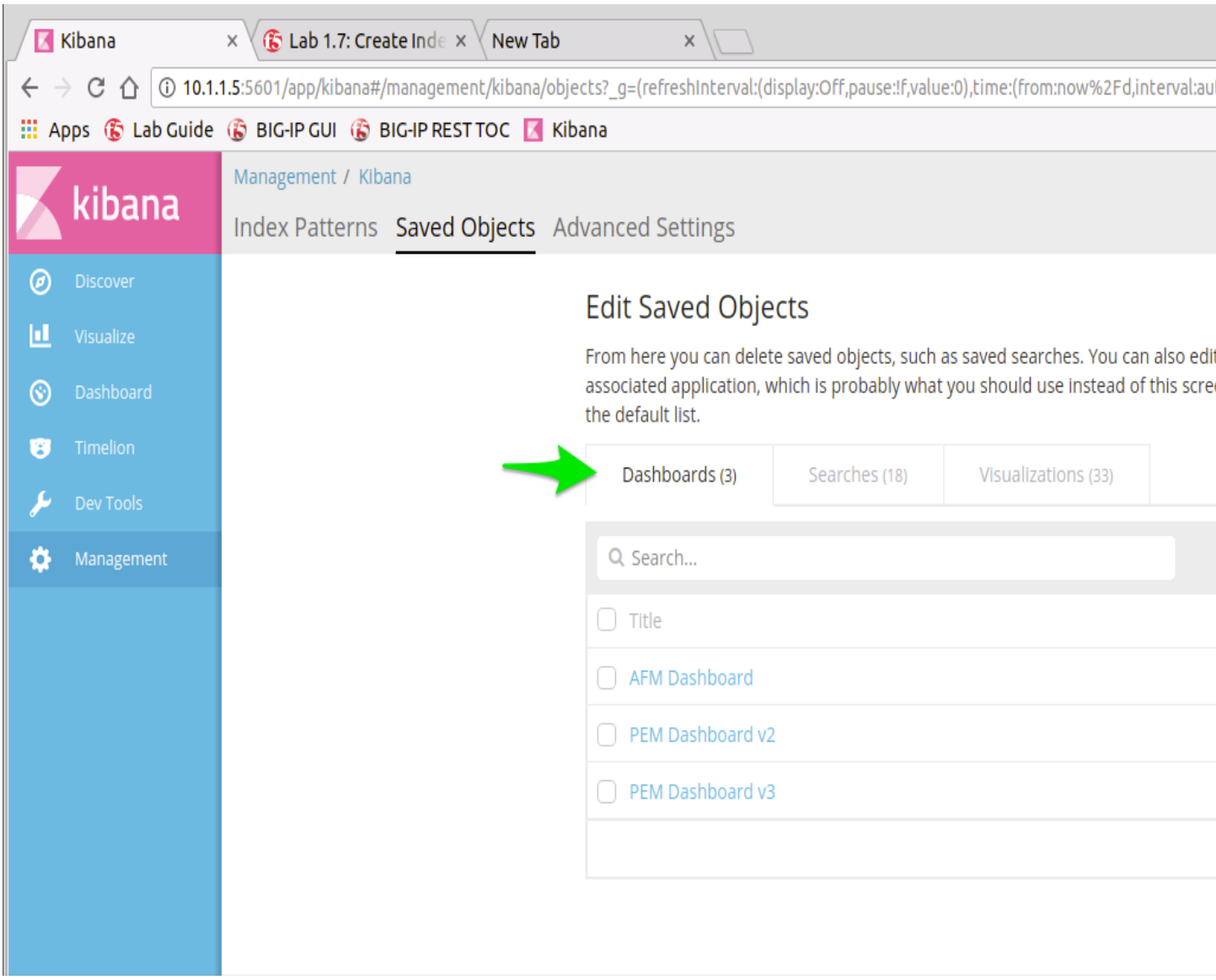
Visuals

```
https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/
json/elk_visualisations.json
```



Dashboards

```
https://raw.githubusercontent.com/jarrodlucia/bigip_elk_server/develop/
json/elk_dashboards.json
```



Note: The JSON files have been placed in the IN_CASE_OF_EMERGENCY folder on the desktop

6.2 Module 2: Kibana and Visualisation

Coolness of Kibana interface

- Navigation
- Searching
- Creating Searches
- Creating Visualisations
- Creating Dashboards

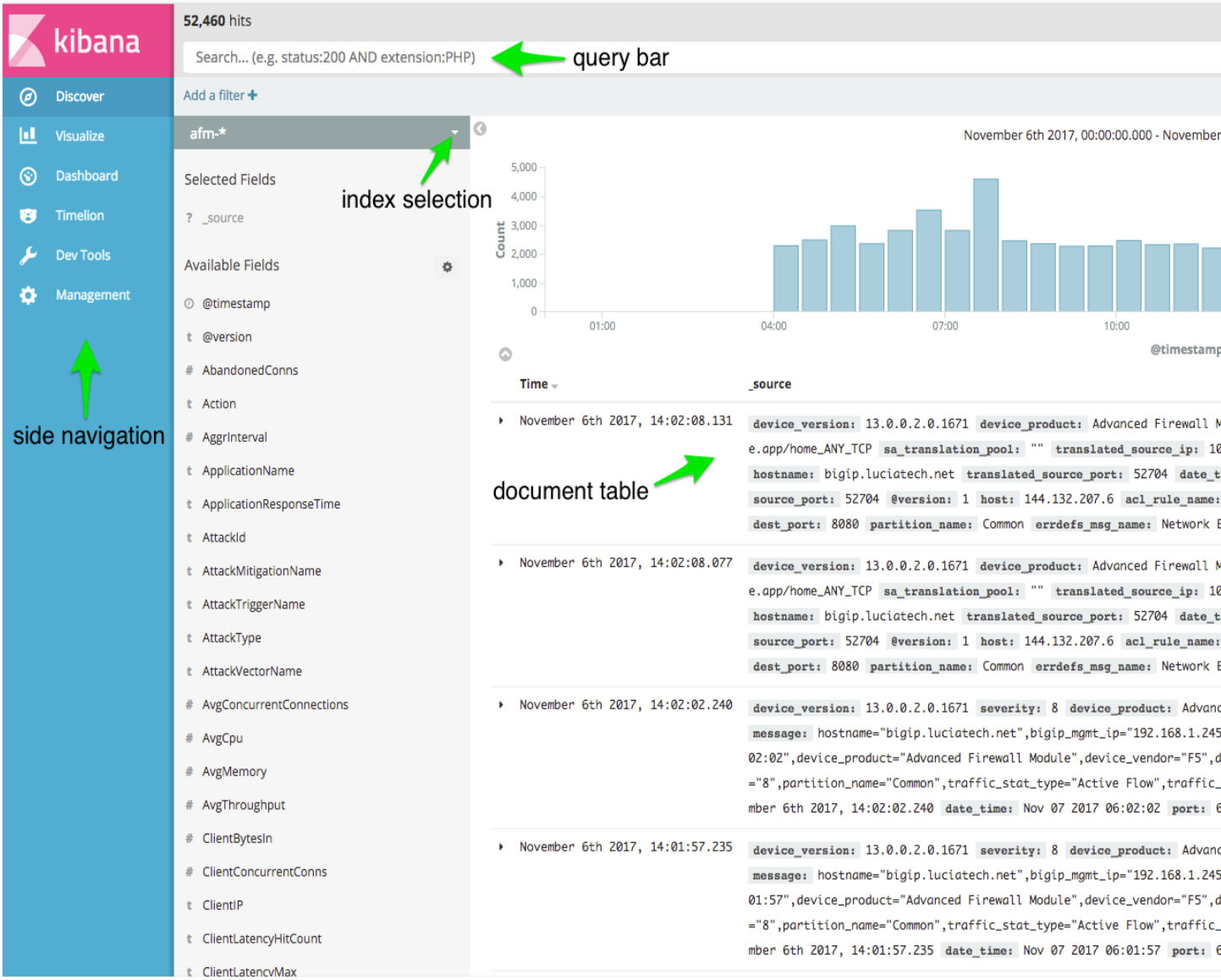
6.2.1 Lab 2.1 – Kibana Interface & Search

Kibana is the interface to elasticsearch and makes visualisationa and dashboards available. It allows REST API calls for development of additional Cutomer interfaces.

This lab will look at the look and feel of the Kibana interface, and some key navigation hints and tips.

Task 1 - Kibana Interface Explantion

This task will focus on explation of the Kibana interface and navigating different aspects of the interface.



Try changing the following:

- Time Range
- Index
- Dashboards

Note: Take your time to explore each of the interface elements.

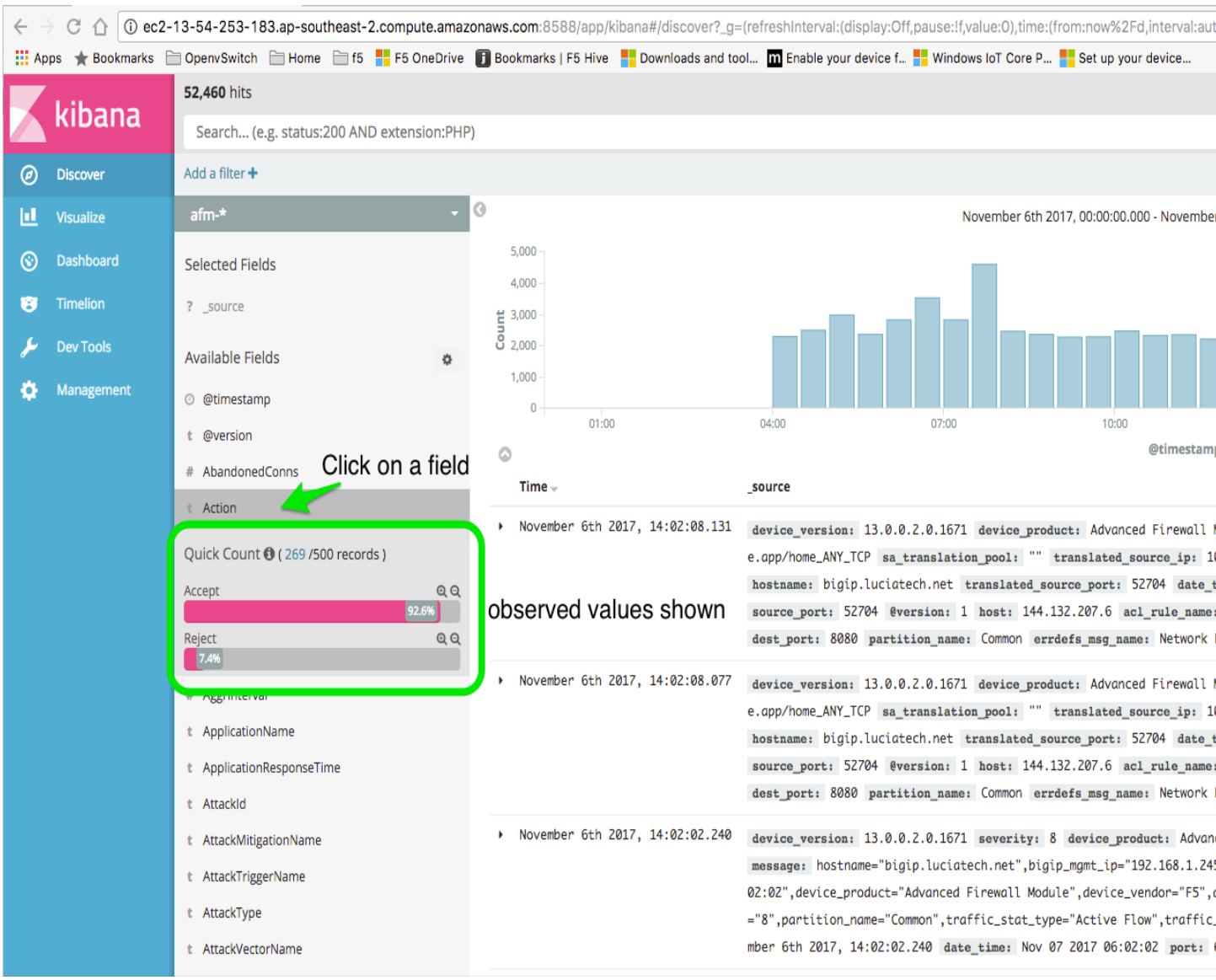
Task 2 - Searching Kibana

In this task we will use two example search types to see how Kibana uses elasticsearch. These example searches will be the following:

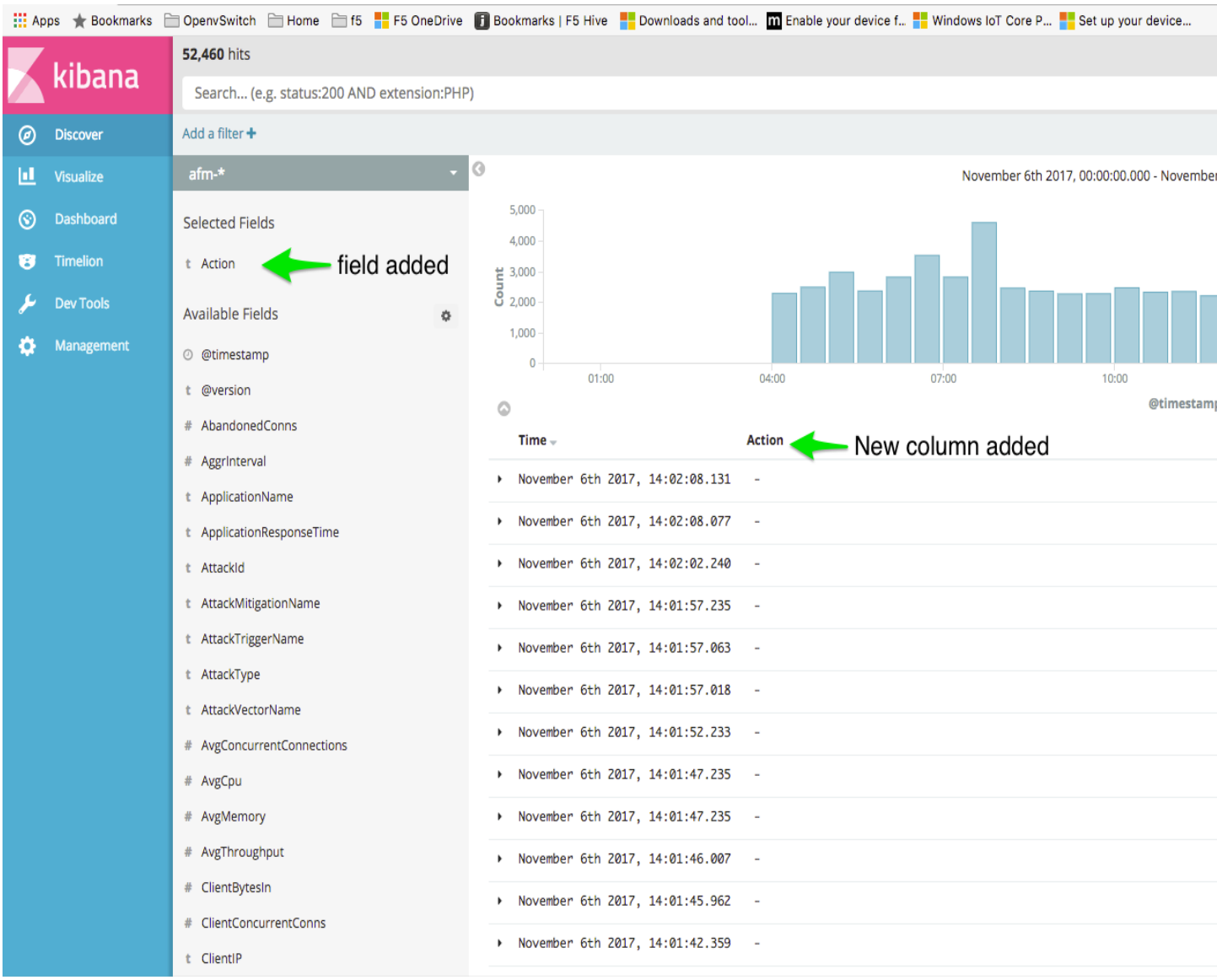
- Field Search
- Query Bar

Field Search Field searching is very useful in Kibana and can be used to see types of data and values that elasticsearch is indexing. To conduct field searching conduct the following:

1. Click on a field
2. Examine the expanded field, note the values that elasticsearch is indexing

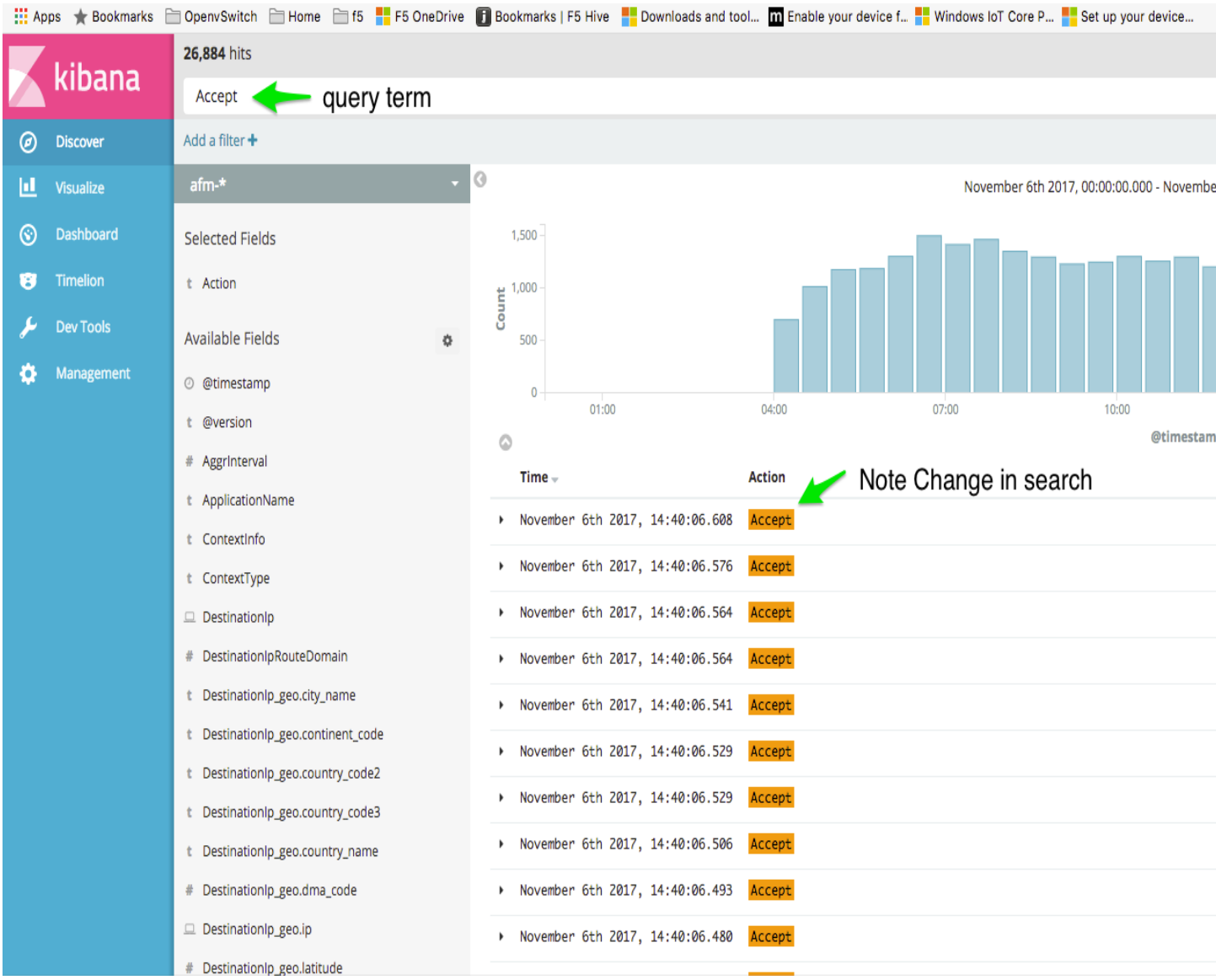


1. Click the add button.
2. Notice the field is in the Selected Field section.



Note: Take time to explore multiple field add to Selected field and build up a set of interesting columns.

Query Bar This type if searching is searching all data fields not only Selected fields as we did previously.



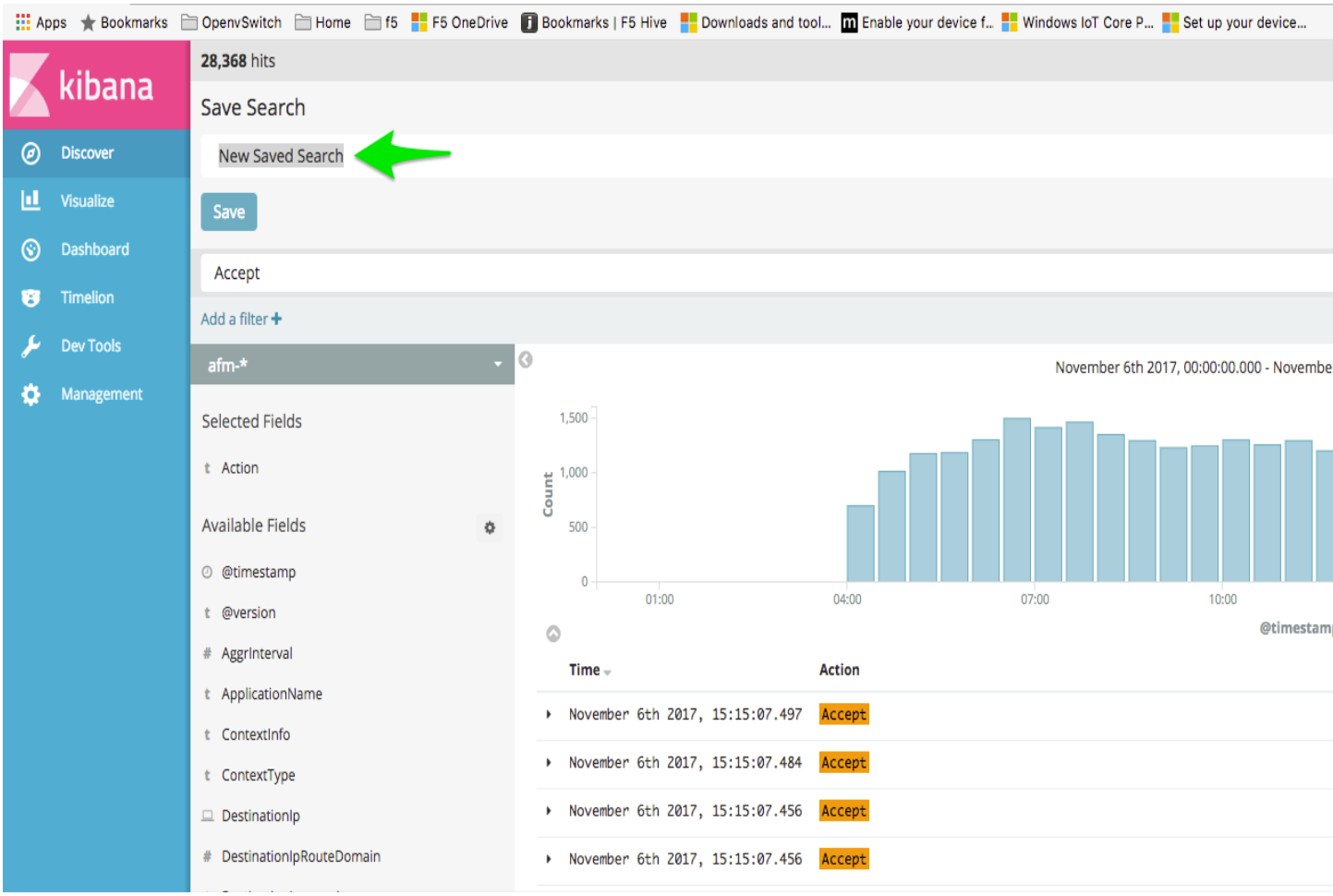
Note: Take time to explore multiple field add to Selected field and use Query terms to see the results.

6.2.2 Lab 2.2 – Creating Kibana Usefulness

This Lab will focus on creation of three key components of Kibana for useful display of information, namely: - Searches - Visualisations - Dashboards

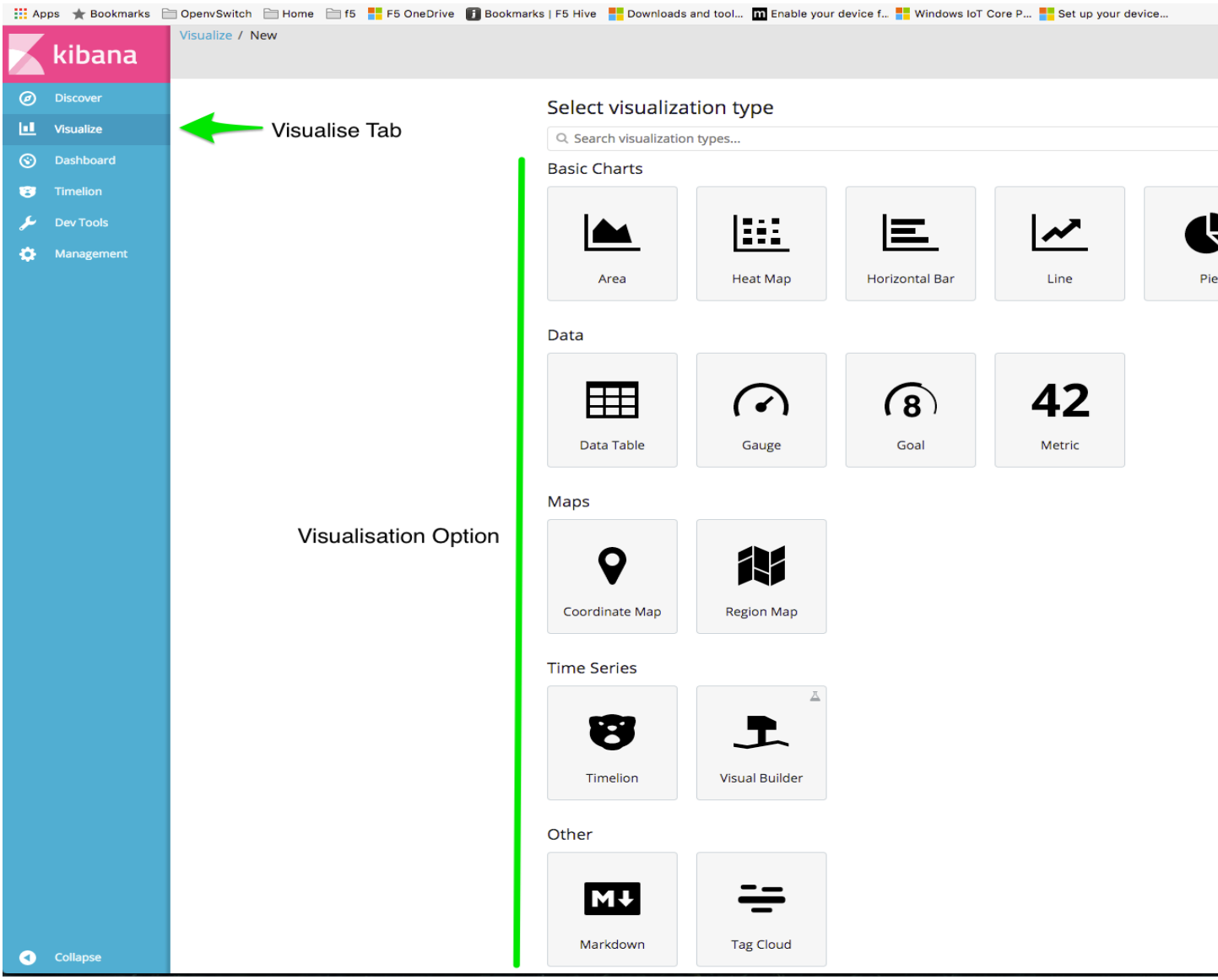
Task 1 - Creating Searches

Create and Save a 3 x search based on the previous lab.

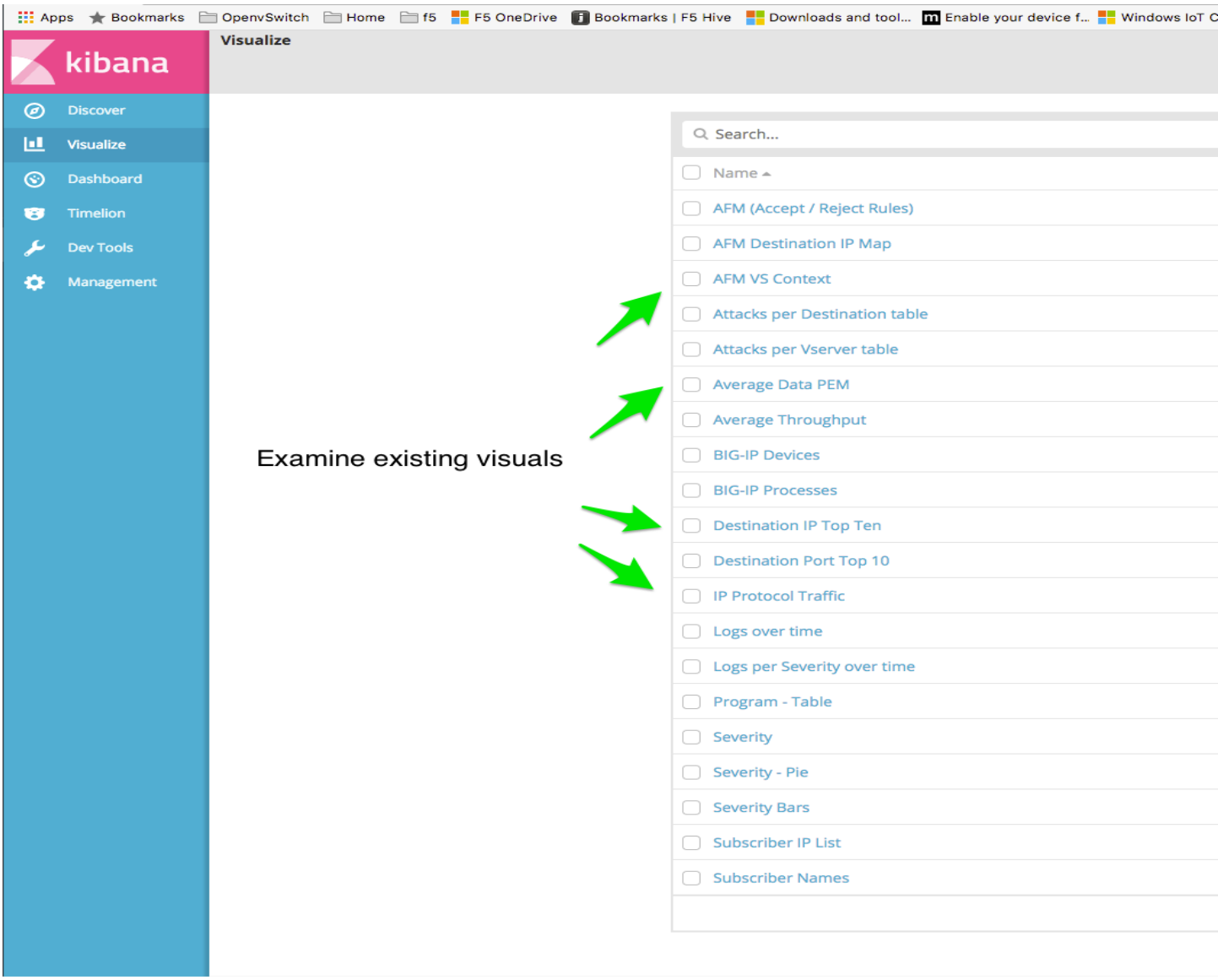


Task 2 - Creating Visualisations

Create and Save a 3 x visualisation based on the above search or the previous lab.

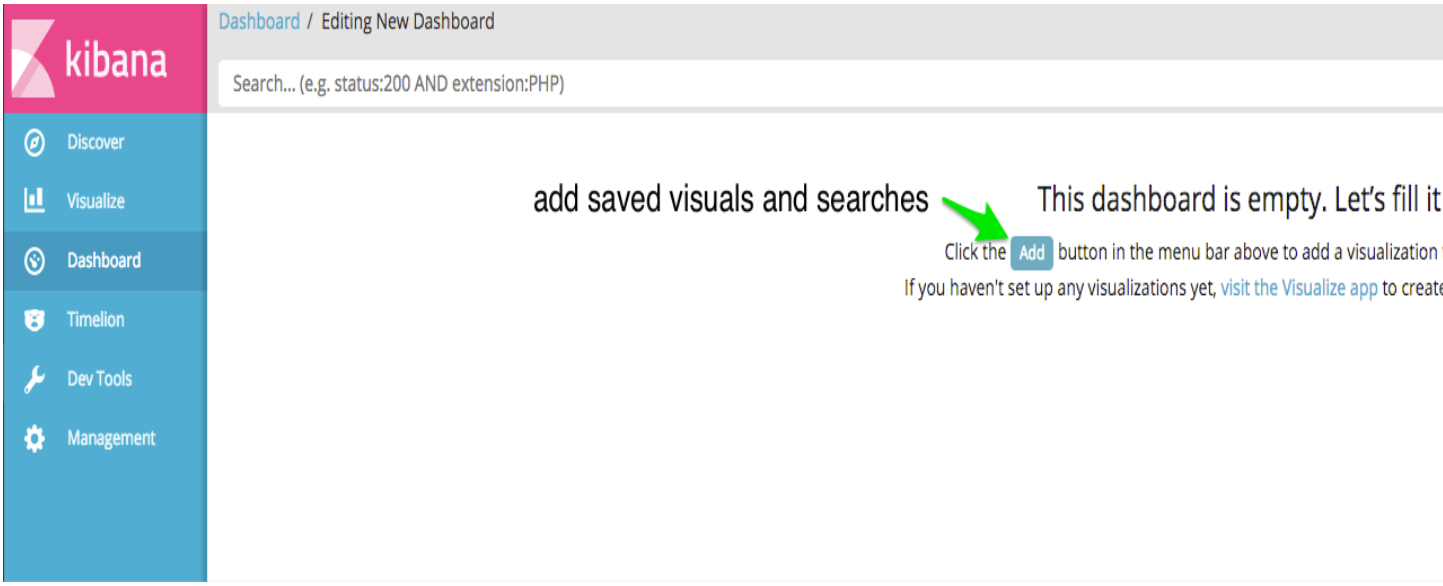


Examine existing Visualisations to understand how some of the different visualisation are constructed.



Task 3 - Creating Dashboards

Create a dashboard from your 3 visualisations created above.


kibana
Dashboard / Editing New Dashboard
Search... (e.g. status:200 AND extension:PHP)
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
add saved visuals and searches
This dashboard is empty. Let's fill it
Click the Add button in the menu bar above to add a visualization
If you haven't set up any visualizations yet, visit the Visualize app to create

7 HOWTOs: Index

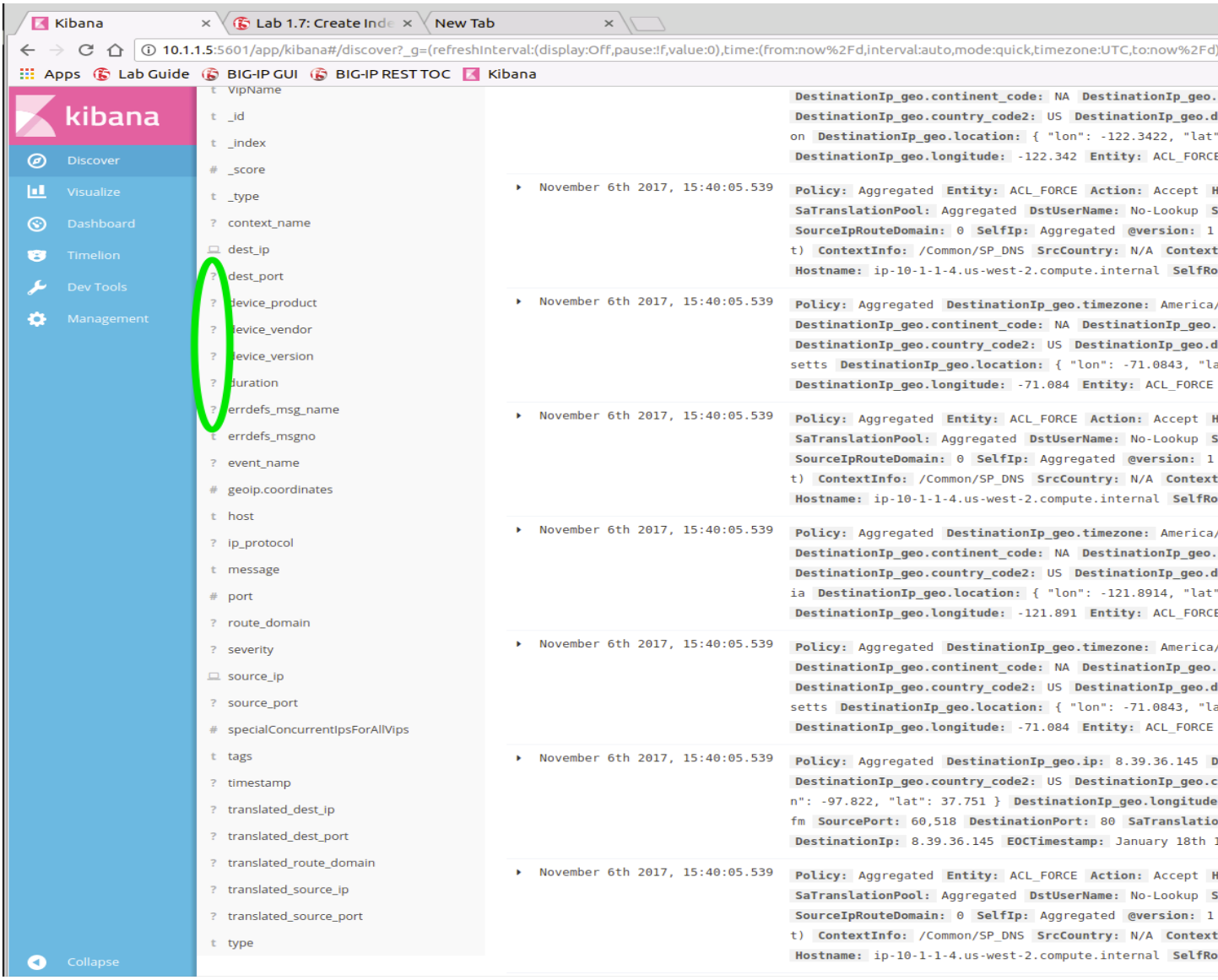
This section contains useful HOWTOs

7.1 HOWTO - how to do stuff

Twill put extra stuff into here

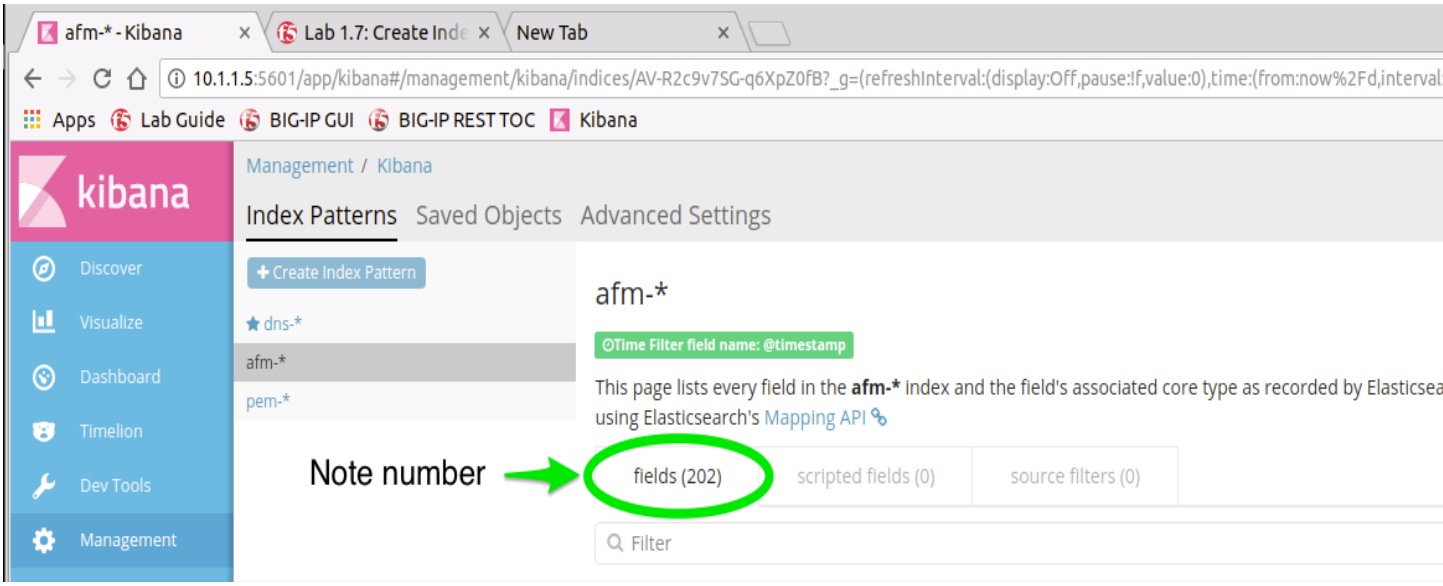
7.1.1 Task 1 – Update unknown index fields

At times new fields may appear in the index field based on software version or addiitonal logging from irules. It will be requried to update the index to make these fields usable.



Note: Notice the ? symbol next to the field.

Update by clicking on the refresh button



Note the increased change

7.1.2 Task 2 - Manual Index Changes

Index changes in json can be done manually if importing from another system.

1. Create a new search or visualisation
2. Export the new search json
3. Open the json and copy the index id
4. Open the json to be imported and paste the updated index id

WE MAKE APPS GO

FASTER.
SMARTER.
SAFER.

F5 Networks, Inc. | f5.com

US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com
©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. These training materials and documentation are F5 Confidential Information and are subject to the F5 Networks Reseller Agreement. You may not share these training materials and documentation with any third party without the express written permission of F5.