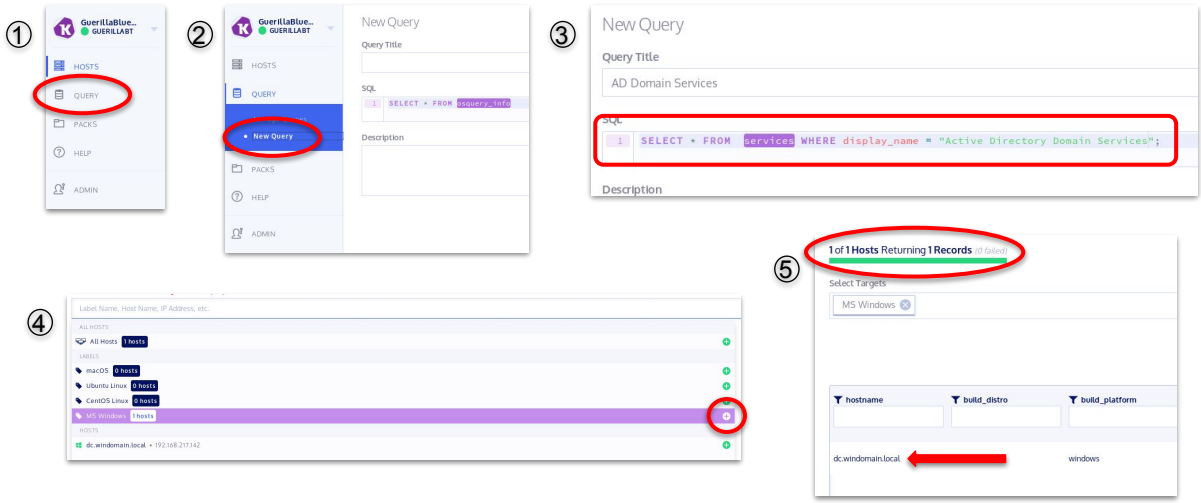
Fleet - Creating a Label



In Fleet, we can create labels to group hosts. Labels are applied to all hosts which return results for queries. In this example, we'll create a query to automatically label and group Windows Domain Controllers in our Fleet instance. Let's first test our query and ensure it returns the expected results:

1) In Fleet, click on **Query**.

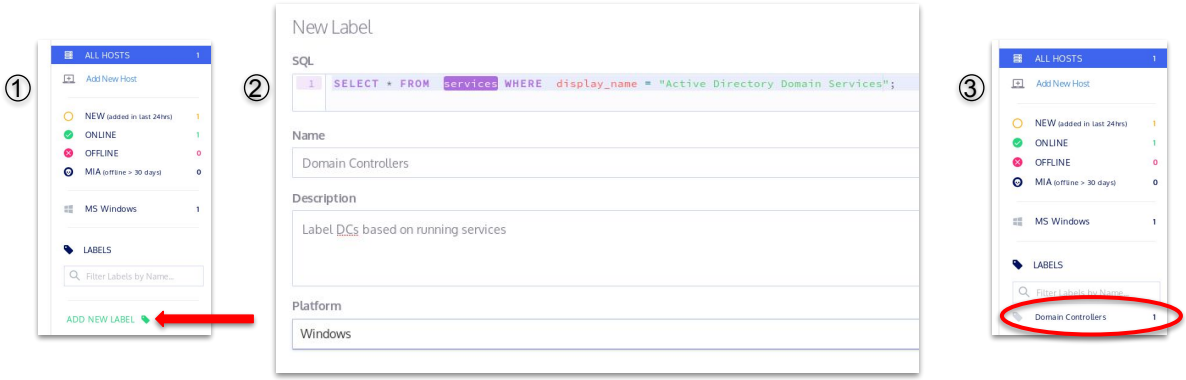
2) Select **New Query**.

3) In the SQL box, enter the query above. The query will return results for all hosts running the **Active Directory Domain Services** service.

4) In the **Select Targets** box, select **MS Windows** by clicking on the plus sign to the right.

5) Select **RUN**. You should see results returned for the host **dc.windomain.local**. This is great, because this a domain controller.

Fleet - Creating a Label



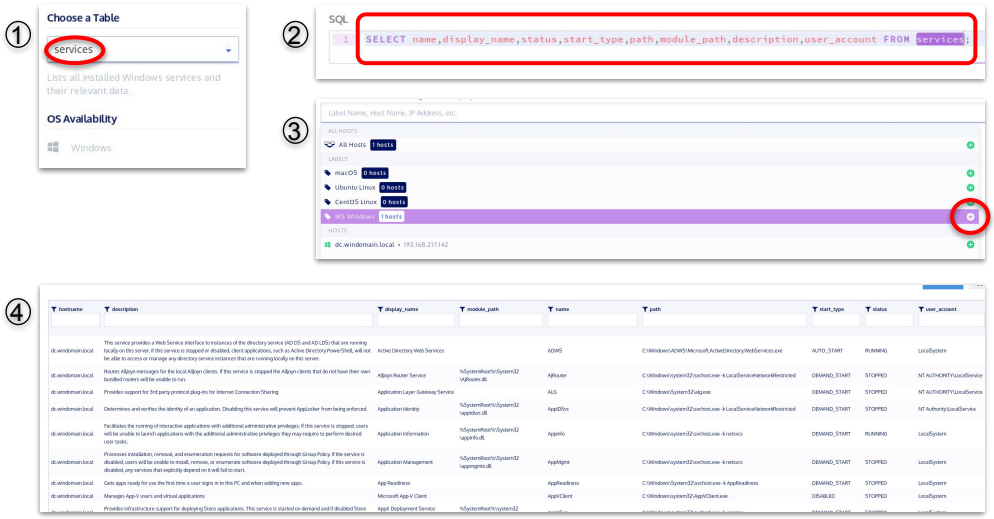
Now that we've verified our query returns the expected results, we're ready to create a label. Return to the HOSTS view by clicking the **HOSTS** link at the top-left of the browser window.

1) Click **ADD NEW LABEL** on the right on the browser window.

2) Fill out the inputs using our query from before and the information above and click **SAVE LABEL**.

3) After a moment you should see the **Domain Controllers** label populate with a value of one.

Fleet - Creating a Pack



Next, let's create a pack. Packs allow us to schedule queries and log results to a file. This will allow us to ship the results into ELK or another SIEM. In the top-left of the browser window, select **Queries** > **New Query**. We're going to test a query first.

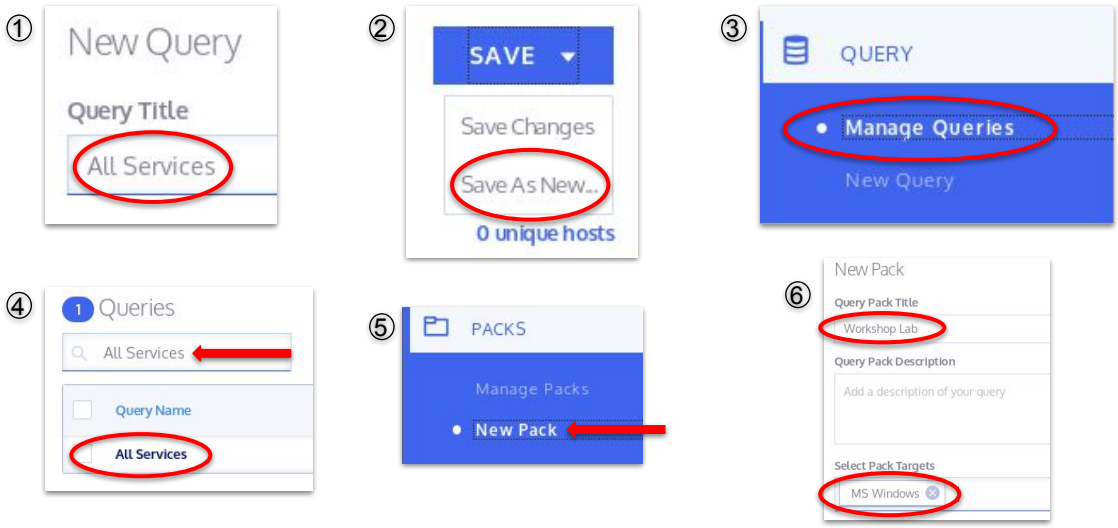
1) For this pack, we'll create a query to return all running services. In the top-right we can see there is a list of all the tables available to us. If we select the **services** table, we can see a description of the 'columns' in the service.

2) We can select to return specific columns in our query. Enter the query as shown above, or modify it as you wish.

3) In the **Select Targets** box, select **MS Windows** by clicking on the plus sign to the right.

4) Select **RUN**. You should see results returned for the host **dc.windomain.local**. This is great, because this a domain controller.

Fleet - Creating a Pack



Now we'll save our query so we can access it later when we create our pack.

1) Still in the same window as the query you just ran, give your query a title. Call it **All Services**.

2) Click **Save** > **Save As New…**.

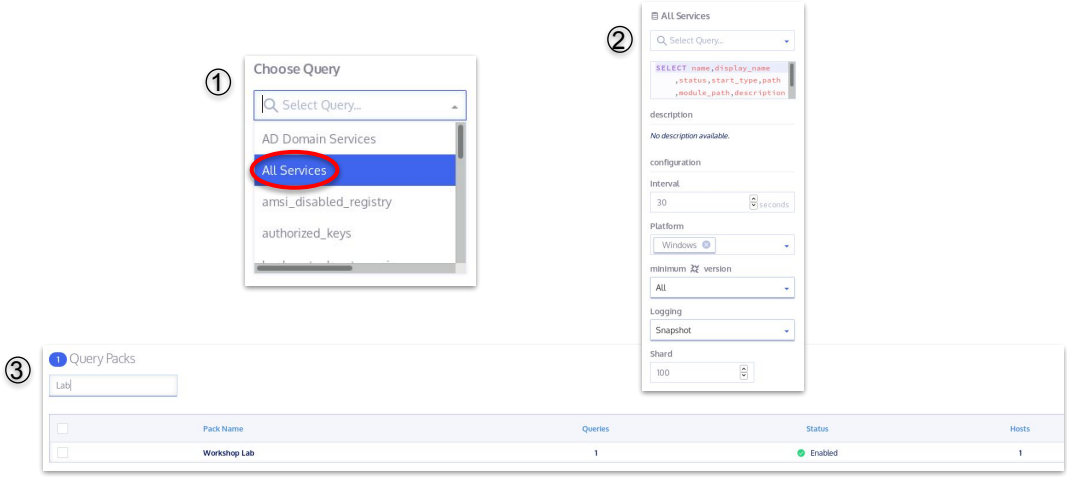
3) Let's check that it's saved. In the top-left of the browser window, select **Manage Queries** > **New Query**.

4) There's a search bar where you can search through your saved queries. Type **All Services** and you should see your query as the only result.

5) Now we'll create our Pack. In the top-left menu, select **Packs** > **New Pack**.

6) Give your pack a title, and select **MS Windows** as the target.

Fleet - Creating a Pack



Now we'll assign a query to our Pack.

1) In the top-right, select **All Services** (or whatever you named the query you just created.)

2) Go ahead and configure the bar on the left as shown above. The **interval** defines how often the query will run. We'll set it to 30 seconds for this lab just to quickly get results back, but that's probably far too frequent for anything you'd do in production. The **platform** is Windows and the **shard** should be set to 100. The shard value can be used to only query a percentage of the hosts targeted by the Pack each time it runs. Save your settings.

3) In the query bar, you should be able to search for your new Pack by name and see that it is **enabled** and one host is targeted by it.

Fleet - Logging Query Results

①

```
analyst@elk:~
File  Edit  View  Search  Terminal  Help
[analyst@elk ~]$ tail /var/log/kolide/osquery_result.log
{"name":"pack/Workshop Lab/All Services","hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:21:09 2019 UTC","unixTime":
1561616469,"epoch":0,"counter":0,"logNumericsAsNumbers":false,"decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"},"columns":{"description":"Enables the detection, download, and installation of updates for Windows and other programs. If this service is disabled,
 users of this computer will not be able to use Windows Update or its automatic updating feature, and programs will not be able to use the Windows Update
Agent (WUA) API.","display_name":"Windows Update","module_path":"%systemroot%\\system32\\wuaueng.dll","name":"wuauserv","path":"C:\\Windows\\system32\\svc
host.exe -k netsvcs","start_type":"DISABLED","status":"STOPPED","user_account":"LocalSystem"},"action":"added"}
{"name":"pack/Workshop Lab/All Services","hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:21:09 2019 UTC","unixTime":
1561616469,"epoch":0,"counter":0,"logNumericsAsNumbers":false,"decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"},"columns":{"description":"Creates and manages user-mode driver processes. This service cannot be stopped.","display_name":"Windows Driver Foundatio
n - User-mode Driver Framework","module_path":"%SystemRoot%\\System32\\WUDFSvc.dll","name":"wudfsvc","path":"C:\\Windows\\system32\\svchost.exe -k LocalSy
stemNetworkRestricted","start_type":"DEMAND_START","status":"RUNNING","user_account":"LocalSystem"},"action":"added"}
{"name":"pack/Workshop Lab/All Services","hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:21:09 2019 UTC","unixTime":
1561616469,"epoch":0,"counter":0,"logNumericsAsNumbers":false,"decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"},"columns":{"description":"Provides authentication and authorization services for interacting with Xbox Live. If this service is stopped, some appli
cations may not operate correctly.","display_name":"Xbox Live Auth Manager","module_path":"%SystemRoot%\\System32\\XblAuthManager.dll","name":"XblAuthMana
ger","path":"C:\\Windows\\system32\\svchost.exe -k netsvcs","start_type":"DEMAND_START","status":"STOPPED","user_account":"LocalSystem"},"action":"added"}
{"name":"pack/Workshop Lab/All Services","hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:21:09 2019 UTC","unixTime":
1561616469,"epoch":0,"counter":0,"logNumericsAsNumbers":false,"decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
```

②

```
analyst@elk:~
File  Edit  View  Search  Terminal  Help
[analyst@elk ~]$ tail /var/log/kolide/osquery_status.log
{"hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:22:58 2019 UTC","unixTime":"1561616578","severity":"0","filename":"
scheduler.cpp","line":"105","message":"Executing scheduled query pack/Workshop Lab/All Services: SELECT name,display_name,status,start_type,path,module_pa
th,description,user_account FROM services;","version":"3.4.0","decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"}}
{"hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:23:25 2019 UTC","unixTime":"1561616605","severity":"0","filename":"
scheduler.cpp","line":"105","message":"Executing scheduled query pack/Workshop Lab/All Services: SELECT name,display_name,status,start_type,path,module_pa
th,description,user_account FROM services;","version":"3.4.0","decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"}}
{"hostIdentifier":"399D4D56-E6F8-E906-F69C-A3F6C387244D","calendarTime":"Thu Jun 27 06:23:53 2019 UTC","unixTime":"1561616633","severity":"0","filename":"
scheduler.cpp","line":"105","message":"Executing scheduled query pack/Workshop Lab/All Services: SELECT name,display_name,status,start_type,path,module_pa
th,description,user_account FROM services;","version":"3.4.0","decorations":{"host_uuid":"399D4D56-E6F8-E906-F69C-A3F6C387244D","hostname":"dc.windomain.l
ocal"}}
```

You can view the logs that are created in **/var/log/kolide/**.

1) The results of the queries can be found in **/var/log/kolide/osquery_result.log**.

2) The status of query executions can be found in **/var/log/kolide/osquery_status.log**.

These are both configured in Fleet and the settings can be viewed in **/opt/kolide-fleet/docker-compose.yml**.

Fleet - Filebeat Config

①

```
File  Edit  View  Search  Terminal  Help
[analyst@elk ~]$ ls /etc/filebeat/
fields.yml  filebeat.reference.yml  filebeat.yml  modules.d
[analyst@elk ~]$
```

②

```
File  Edit  View  Search  Terminal  Help
[analyst@elk ~]$ ls /etc/filebeat/modules.d/
apache2.yml.disabled        icinga.yml.disabled    kibana.yml.disabled    nginx.yml.disabled        suricata.yml.disabled
auditd.yml.disabled         iis.yml.disabled       logstash.yml.disabled  osquery.yml.disabled      system.yml.disabled
elasticsearch.yml.disabled  iptables.yml.disabled  mongodb.yml.disabled   postgresql.yml.disabled   traefik.yml.disabled
haproxy.yml.disabled        kafka.yml.disabled     mysql.yml.disabled     redis.yml.disabled
[analyst@elk ~]$
```

③

```
File  Edit  View  Search  Terminal  Help
[analyst@elk ~]$ cat /etc/filebeat/modules.d/osquery.yml.disabled
- module: osquery
  result:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

    # If true, all fields created by this module are prefixed with
    # `osquery.result`. Set to false to copy the fields in the root
    # of the document. The default is true.
    #var.use_namespace: true
[analyst@elk ~]$
```

Now we want to configure filebeat to pick up the osquery_results.log file and ship the contents to ELK. Filebeat comes with several modules that can be quickly configured. We'll be running Filebeat from the CentOS VM.

1) Filebeat resides in **/etc/filebeat** on the CentOS VM.

2) Filebeat's modules reside in **/etc/filebeat/modules.d**. As you can see by the file extension, they're all disabled by default.

3) Take a look at the contents of the osquery module.

Fleet - Filebeat Config

①

```
- module: osquery
  result:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: /var/logs/kolide/osquery_result.log

    # If true, all fields created by this module are prefixed with
    # `osquery.result`. Set to false to copy the fields in the root
    # of the document. The default is true.
    var.use_namespace: false
```

②

```
[analyst@elk ~]$ sudo mv /etc/filebeat/modules.d/osquery.yml.disabled /etc/filebeat/modules.d/osquery.yml
[analyst@elk ~]$
```

③

```
[analyst@elk ~]$ vim /etc/filebeat/filebeat.yml
```

1) Edit the osquery.yml.disabled file to match the settings above. (Don't forget to sudo.)

2) Rename the file to enable it. We just need to remove the **disabled** extension.

3) Next, we'll edit **filebeat.yml** config. This is the general config. (**NOTE**: you'll need to prepend this line with sudo.)

Fleet - Filebeat Config

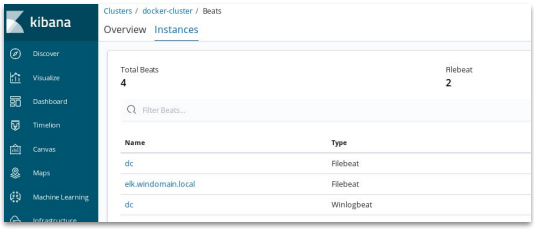
①

```
[analyst@elk filebeat]$ sudo cat filebeat.yml
filebeat.modules:
- module: osquery

output.logstash:
  hosts: ["localhost:5045"]

xpack.monitoring:
  enabled: true
  elasticsearch:
    hosts: ["http://localhost:9200"]
[analyst@elk filebeat]$
```

②

```
[analyst@elk ~]$ sudo filebeat modules list
Enabled:
osquery

Disabled:
apache2
auditd
elasticsearch
haproxy
icinga
iis
iptables
kafka
kibana
logstash
mongodb
mysql
nginx
postgresql
redis
suricata
system
traefik
[analyst@elk ~]$
```

③



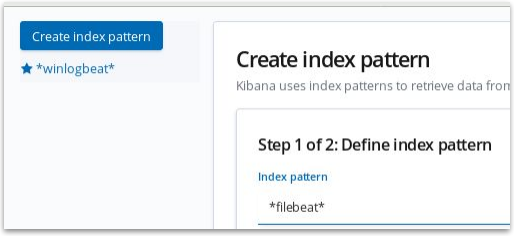
1) If you want to backup the existing filebeat.yml, you can run **mv filebeat.yml filebeat.yml.bak**. Create a new **filebeat.yml** to match the config above. Go ahead and restart the filebeat service: **sudo systemctl restart filebeat**

2) Run **sudo filebeat modules list** and you should see osquery is enabled.

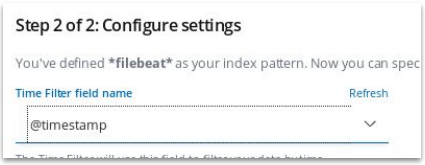
3) Since we configured monitoring, you should be able to navigate to **Monitoring** > **Beats** in Kibana and see that **elk.windomain.local** is now sending beats.

Fleet - Filebeat Config

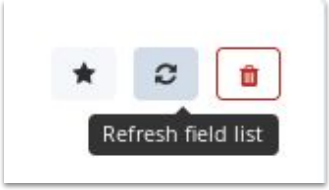
①



②



③



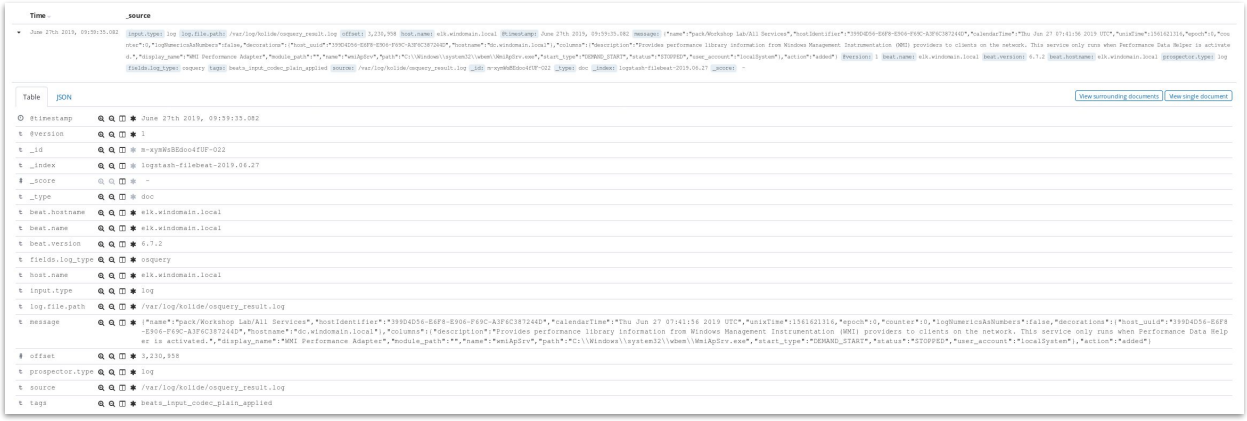
Next, we need to create an index pattern so we can search our results in Kibana.

1) Navigate to **Management** > **Index Patterns** and click on **Create index pattern**. Create the pattern ***filebeat*** or choose another pattern that matches the filebeat indices and click **Next**.

2) Next, select the **@timestamp** timestamp and create the index pattern.

3) Click the **refresh** symbol in the upper right to refresh the field list for this index.

Fleet - Filebeat Config



If you go to the **Discover** tab and choose ***filebeat*** from the drop down, you should now see results with **fields.log_type: osquery**.