

Synapse Bootcamp - Module 20

Automation in Synapse - Exercises

Automation in Synapse - Exercises	1
Objectives	1
Exercises	2
Cron Jobs	2
Exercise 1	2
Adding Triggers	7
Exercise 2	7
Trigger Execution	12
Exercise 3	12

Objectives

In these exercises you will learn:

- How to create and manage Cron jobs
- How to create and manage Triggers

Note: We are constantly updating Synapse and its Power-Ups! We do our best to make sure our course documents (slides, exercises, and answer keys) are up-to-date. However, you may notice small differences (such as between a screen capture in the documents and the appearance of your current instance of Synapse).

If something is unclear or if you identify an error, please reach out to us so we can assist!

Exercises

Cron Jobs

Exercise 1

Objective:

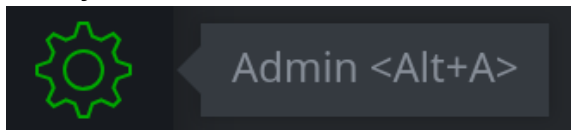
- Create, manage, and inspect cron jobs.

Note: In a **production** instance of Synapse, you should **always** fork a view before creating and testing automation.

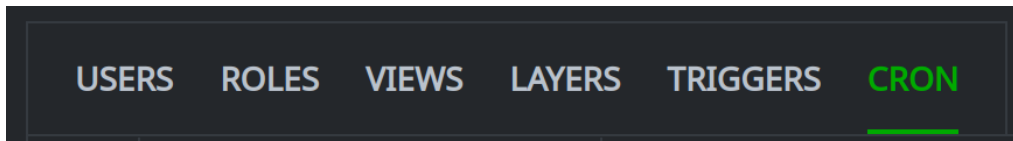
For simplicity, we will do these exercises directly in our **Fork - Synapse Bootcamp** view.

You want to create a **cron job** to ingest the latest Known Exploited Vulnerabilities (KEV) list data on an hourly basis using the **synapse-us-cisa** Power-Up.

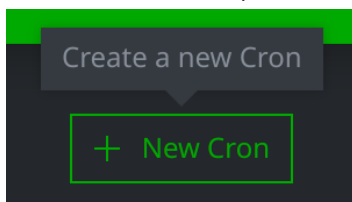
- From your **Toolbar**, select the **Admin Tool**:



- In the **Admin Tool**, click on the **CRON** tab:

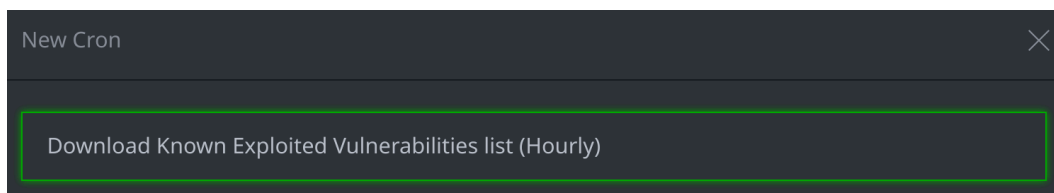


- On the **CRON** tab, click the **+ New Cron** button:



- In the **New Cron** dialog:
 - In the *name* field, enter the following:

Download Known Exploited Vulnerabilities list (Hourly)

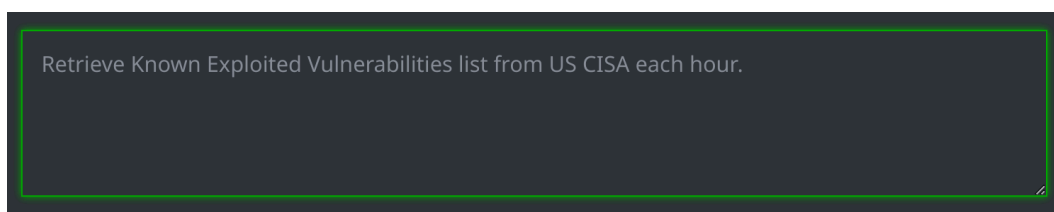


New Cron

Download Known Exploited Vulnerabilities list (Hourly)

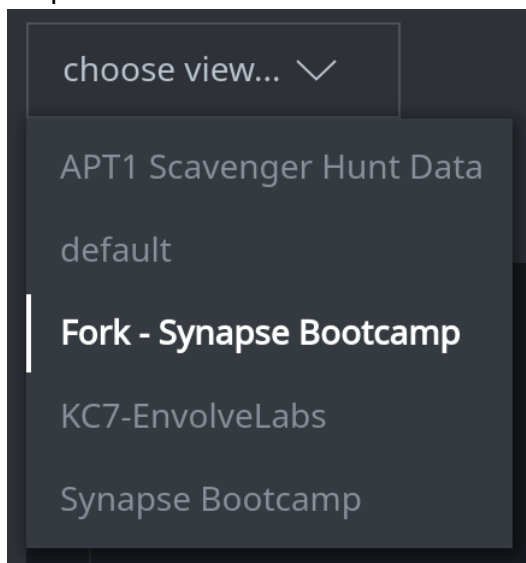
- In the *doc* field, enter the following:

Retrieve Known Exploited Vulnerabilities list from US CISA each hour.



Retrieve Known Exploited Vulnerabilities list from US CISA each hour.

- Click the **choose view** button and select **Fork - Synapse Bootcamp** from the dropdown list:



choose view... ▼

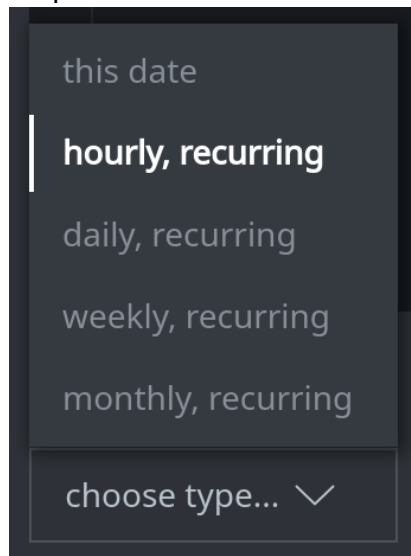
- APT1 Scavenger Hunt Data
- default
- Fork - Synapse Bootcamp**
- KC7-EnvolveLabs
- Synapse Bootcamp

- In the **Storm editor window**, enter the following Storm command:

```
us.cisa.kev.sync
```

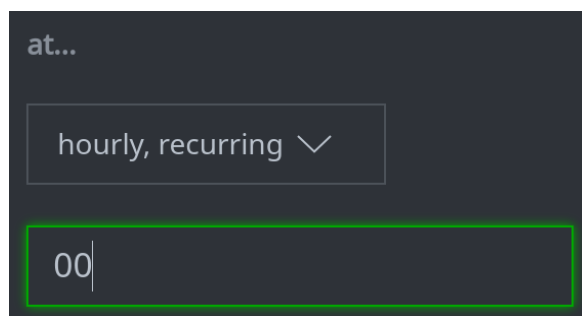
```
1 us.cisa.kev.sync
```

- Click the **choose type** button and select **hourly, recurring** from the dropdown menu:

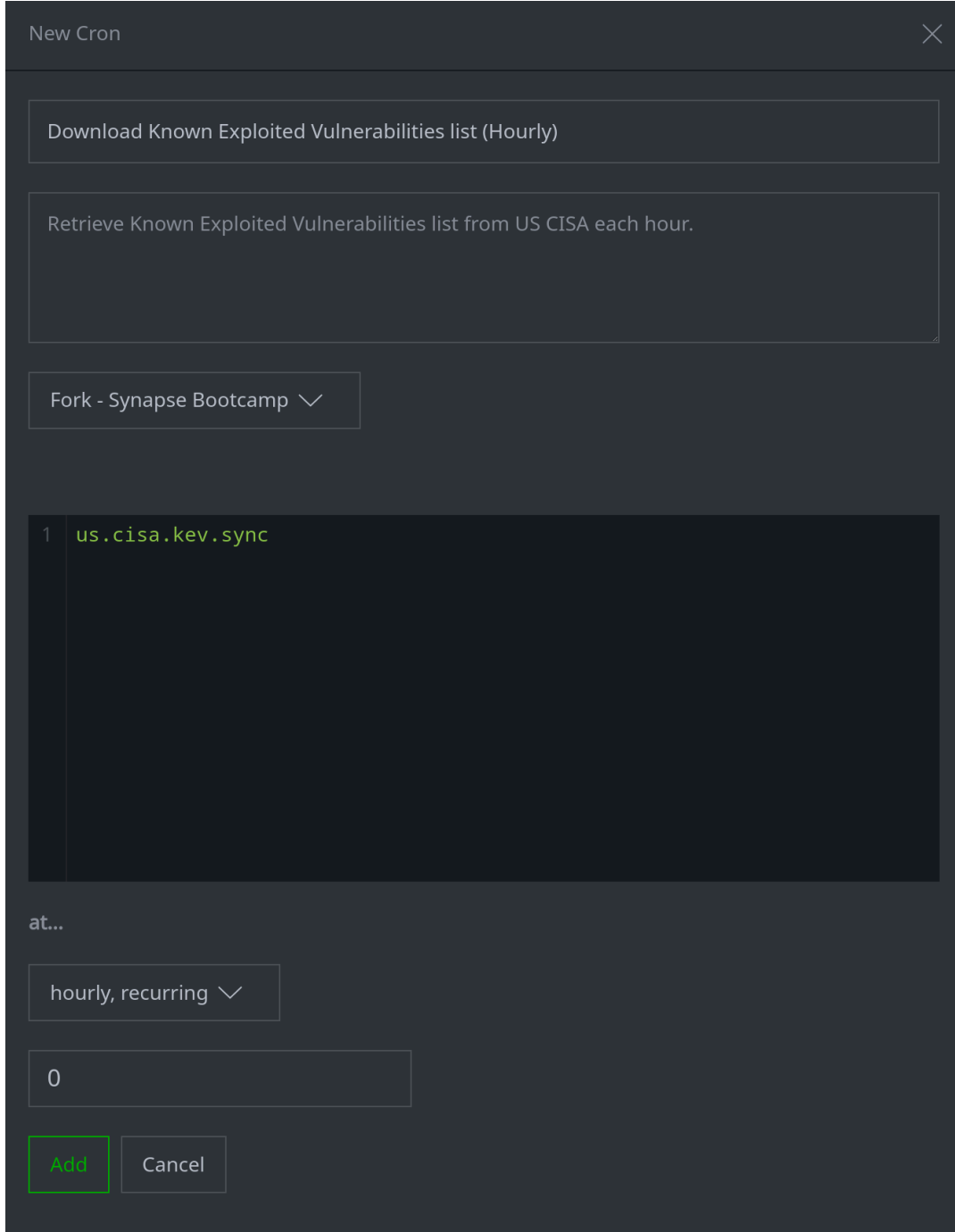


- In the **MM** field, enter the following time:

```
00
```



- Your **New Cron** dialog should look like this:



New Cron

Download Known Exploited Vulnerabilities list (Hourly)

Retrieve Known Exploited Vulnerabilities list from US CISA each hour.

Fork - Synapse Bootcamp

```
1 us.cisa.kev.sync
```

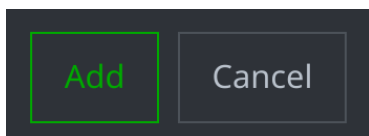
at...

hourly, recurring

0

Add Cancel

- Click the **Add** button to create the cron job:



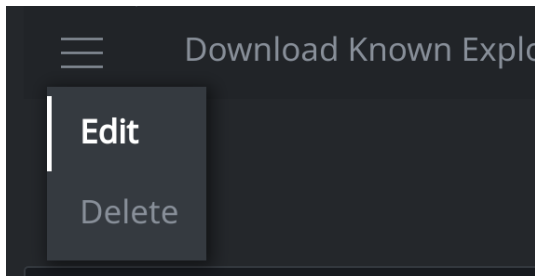
Add Cancel

Question 1: What does your newly added cron job look like?

Question 2: Is the cron job enabled or disabled by default?

You decide that you want to download the Known Exploited Vulnerabilities list once a day instead of every hour.

- Click the **hamburger menu** next to your cron job and select **Edit**:



Question 3: Are you able to modify your existing cron job to make this change?

Adding Triggers

Exercise 2

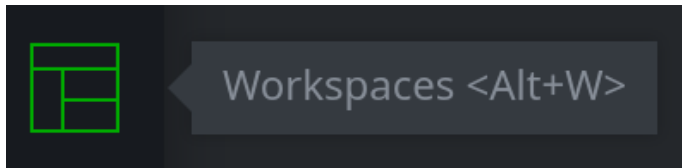
Objective:

- **Create a trigger to perform basic enrichment of IP addresses when they are created.**

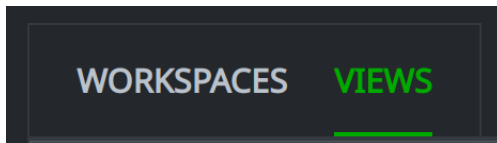
Every time you create an **inet:ipv4** node, you run several Node Actions to enrich the IP address with basic Power-Ups such as **synapse-maxmind** (for AS / geolocation data) and **synapse-nettools** (for DNS PTR and netblock registration data).

You want to create a trigger to automate this process instead of running the actions manually.

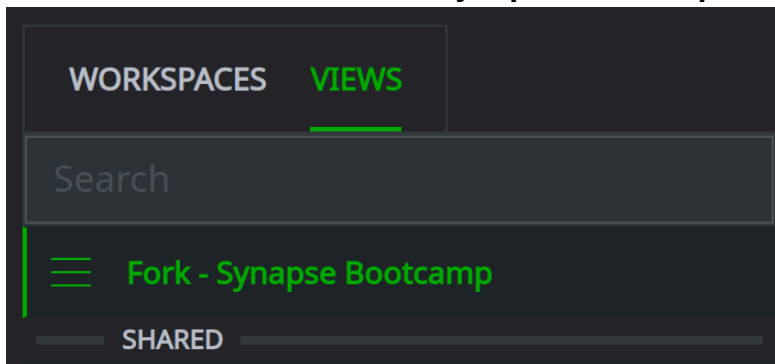
- From your **Toolbar**, select the **Workspaces Tool**:



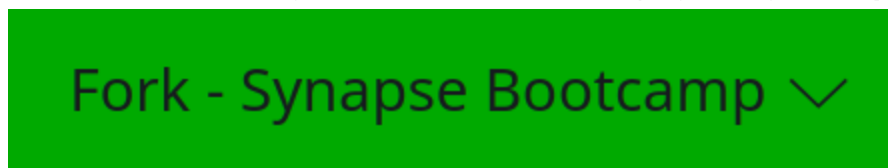
- Select the **VIEWS** tab:



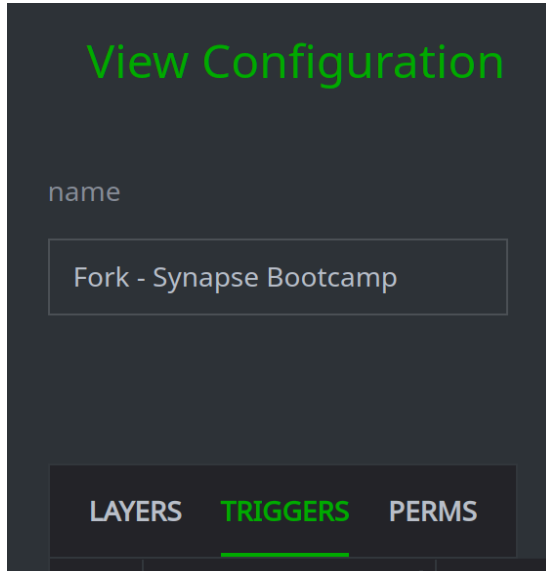
- From the **list** view, select **Fork - Synapse Bootcamp**:



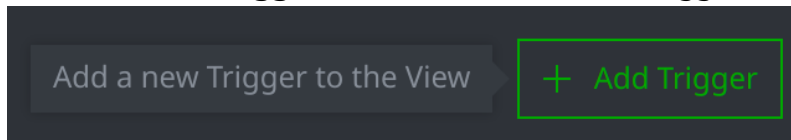
Note: this should be your **current** view (as displayed in your **Top Bar**):



- In the **View Configuration** panel, select the **TRIGGERS** tab:

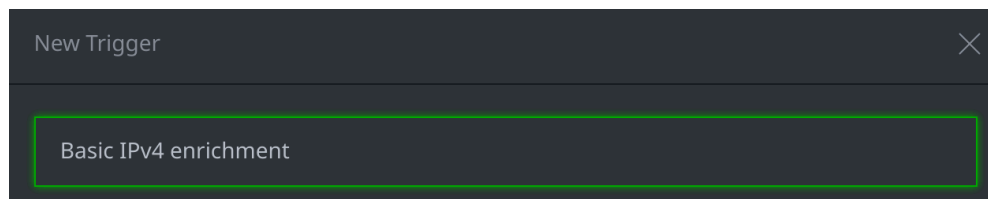


- Click the **+ Add Trigger** button to create a new trigger:



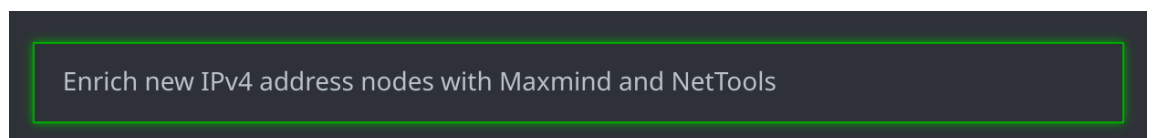
- In the **New Trigger** dialog:
 - In the *name* field, enter the following:

Basic IPv4 enrichment

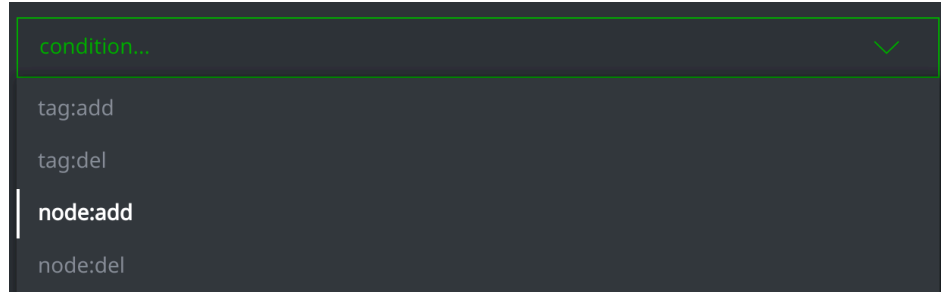


- In the *doc* field, enter the following:

Enrich new IPv4 address nodes with Maxmind and NetTools

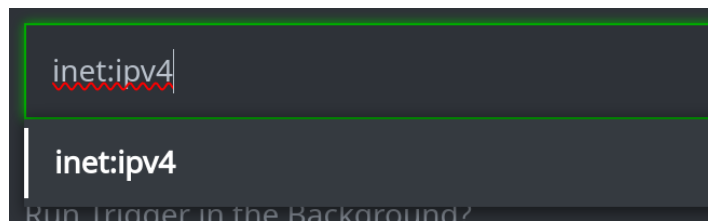


- Click the **condition...** button and choose **node:add** from the dropdown menu:



- In the *form* field, enter the following:

```
inet:ipv4
```

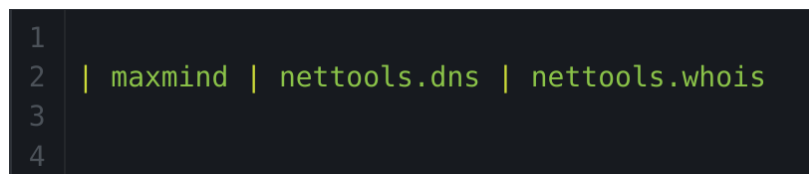


- Set the **Run Trigger in the Background?** toggle to **ON**:

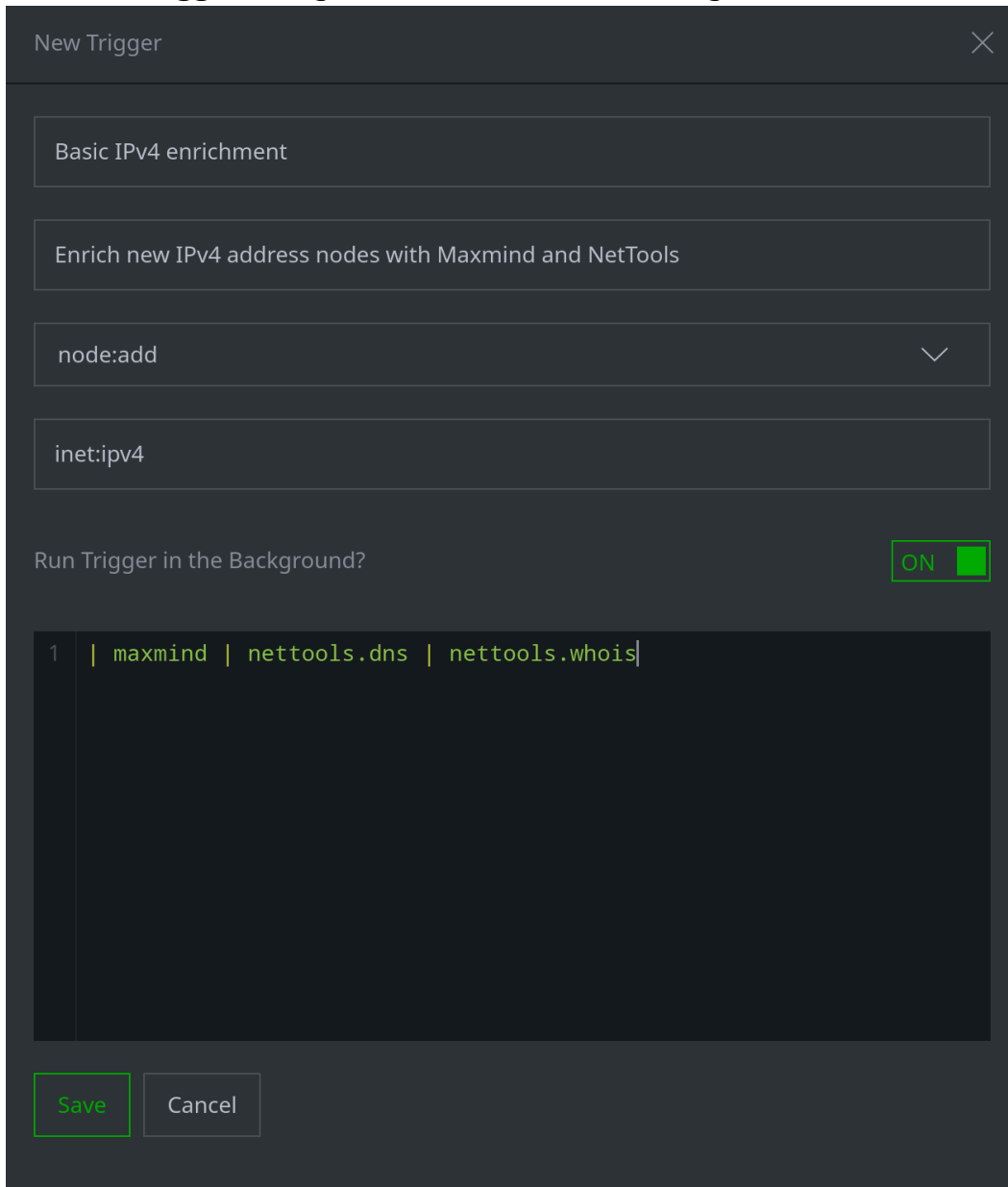


- In the **Storm editor window**, enter the following set of Storm commands:

```
| maxmind | nettools.dns | nettools.whois
```



- Your **New Trigger** dialog should look like the following:



New Trigger

Basic IPv4 enrichment

Enrich new IPv4 address nodes with Maxmind and NetTools

node:add

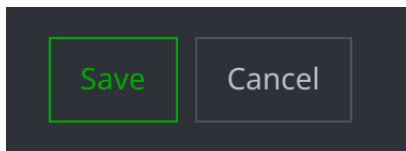
inet:ipv4

Run Trigger in the Background? ☒ ON

```
1 | maxmind | nettools.dns | nettools.whois|
```

Save Cancel

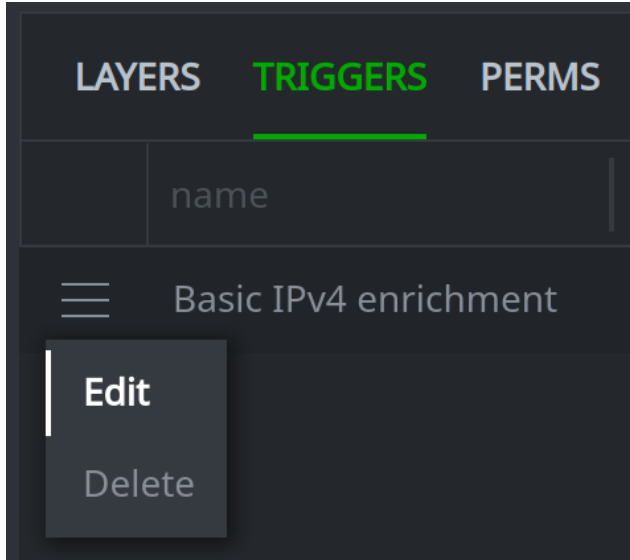
- Click the **Save** button to create the trigger:



Save Cancel

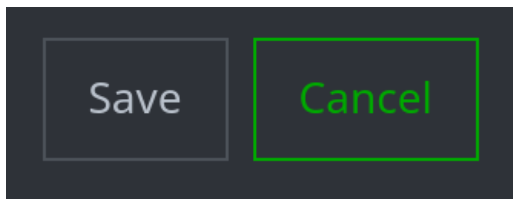
Question 1: Is the trigger enabled or disabled by default?

- Click the **hamburger menu** next to your new trigger and select **Edit**:



Question 2: What elements of the trigger can be changed after it has been saved?

- Click the **Cancel** button to close the **Edit Trigger** dialog:



Trigger Execution

Exercise 3

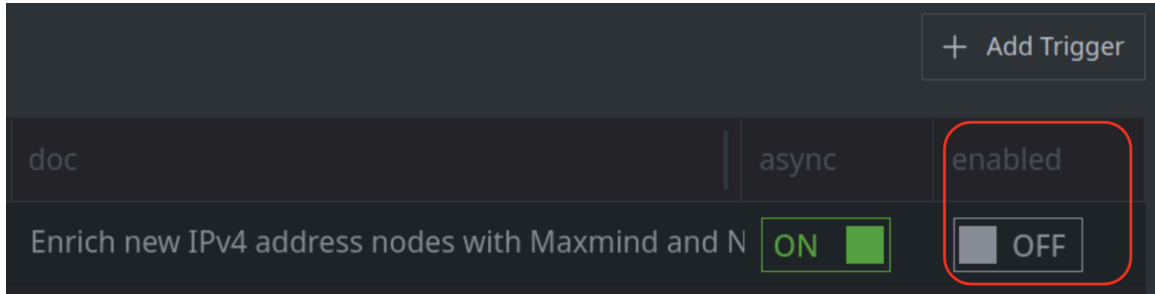
Objective:

- Observe trigger behavior by creating an IPv4 node when the trigger is disabled and when it has been enabled.

First we will create a new IPv4 with the trigger **disabled**.

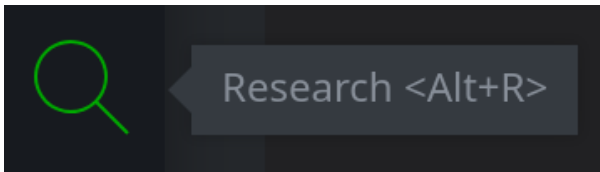
Disable your trigger

- In the **Workspaces Tool**, use the toggle switch to **disable** your trigger:



Create an IPv4

- From your **Toolbar**, select the **Research Tool**:



- In your **Storm Query Bar**, enter the following and press **Enter** to create a new IPv4:

```
[ inet:ipv4=8.8.16.1 ]
```

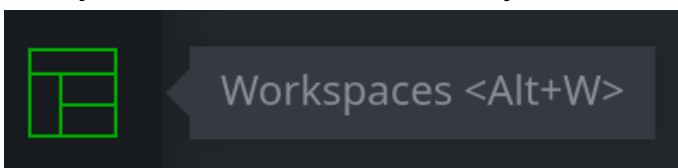
Question 1: What properties are present on the new **inet:ipv4** node?

Question 2: Did your trigger fire?

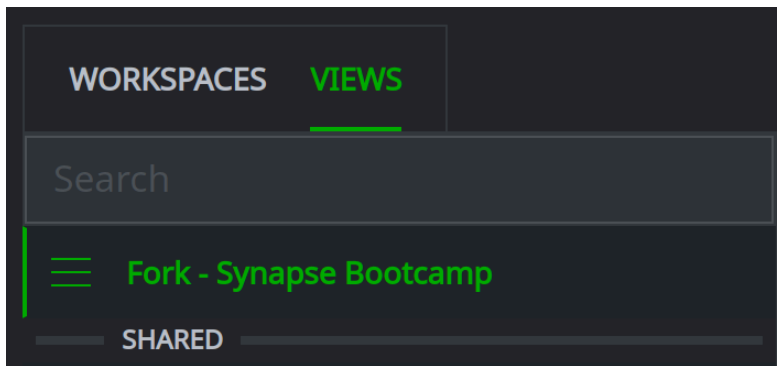
Now we will create a new IPv4 with the trigger **enabled**.

Enable your trigger

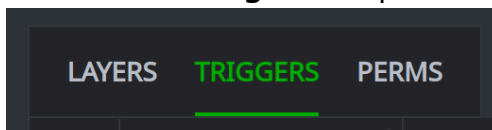
- From your **Toolbar**, select the **Workspaces Tool**:



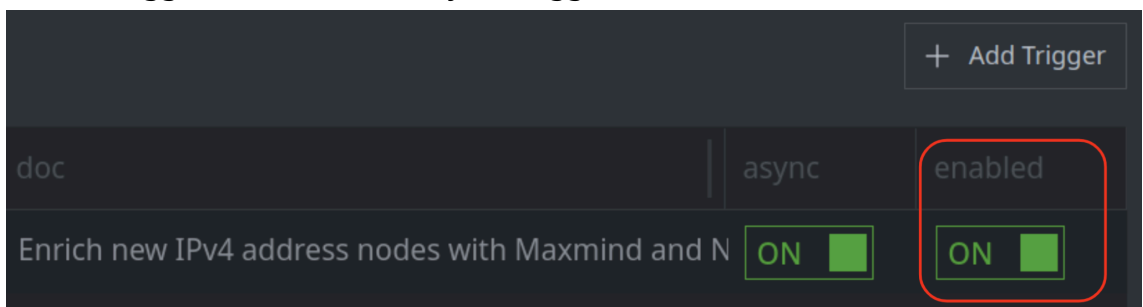
- On the **VIEWS** tab, select the **Fork - Synapse Bootcamp** view:



- In the **View Configuration** panel, select the **TRIGGERS** tab:

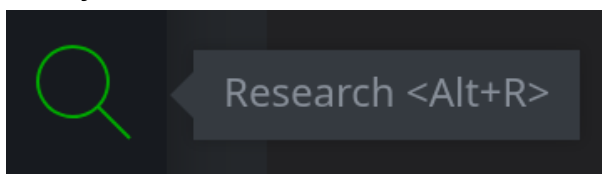


- Use the toggle switch to **enable** your trigger:



Create an IPv4

- From your **Toolbar**, select the **Research Tool**:



- In your **Storm Query Bar**, enter the following and press **Enter** to create another new IPv4:

```
[ inet:ipv4=8.8.16.44 ]
```

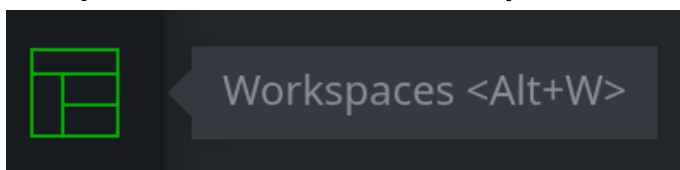
Question 3: What properties appear to be present on the new **inet:ipv4** node?

Question 4: Did your trigger fire? (**Hint:** you may need to re-run / refresh your query.)

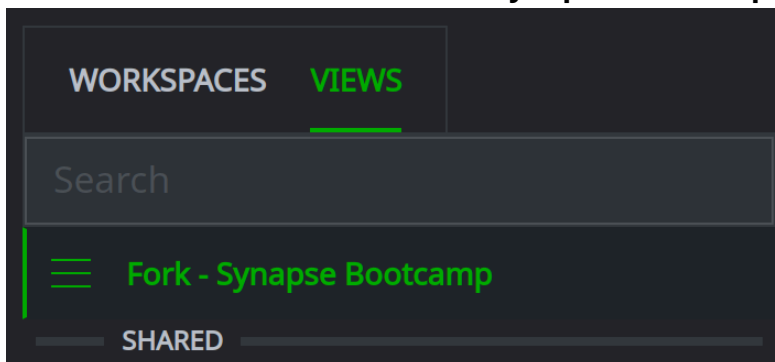
Our trigger ran **asynchronously** (in the background) so we had to refresh our query to see the new data added by the trigger. We will **disable** the async processing to see the difference.

Turn off async processing

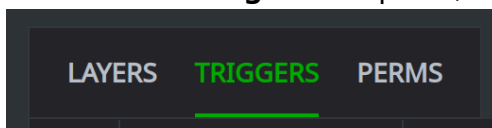
- From your **Toolbar**, select the **Workspaces Tool**:



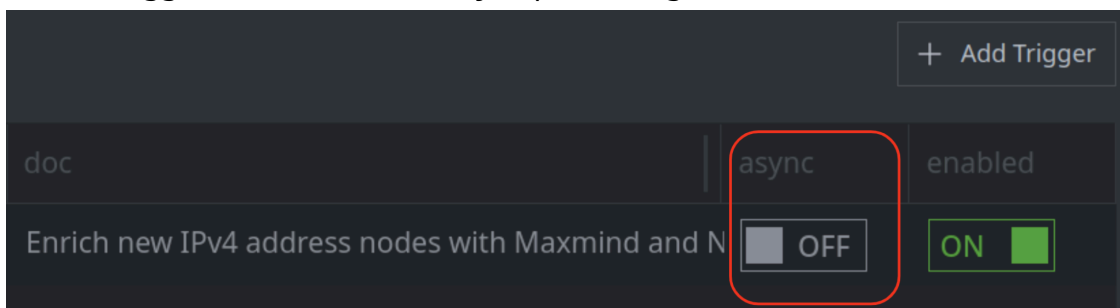
- On the **VIEWS** tab, select the **Fork - Synapse Bootcamp** view:



- In the **View Configuration** panel, select the **TRIGGERS** tab:

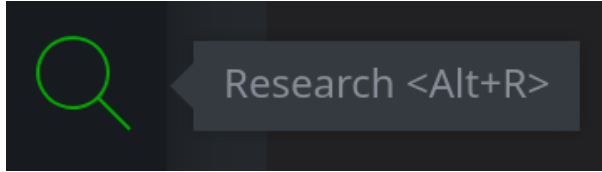


- Use the toggle switch to set the **async** processing to **off**:



Create an IPv4

- From your **Toolbar**, select the **Research Tool**:



- In your **Storm Query Bar**, enter the following and press **Enter** to create another new IPv4:

```
[ inet:ipv4=8.8.16.253 ]
```

Question 5: What properties are present on the new `inet:ipv4` node?

Now that you have tested your trigger, you want to see if the original IPv4 that you created has been updated.

- In the **Research Tool**, enter the following in your **Storm Query Bar** and press **Enter** to lift your **original** IPv4:

```
inet:ipv4=8.8.16.1
```

Question 6: Have any additional properties been set on your original IP since you enabled the trigger? Why or why not?
