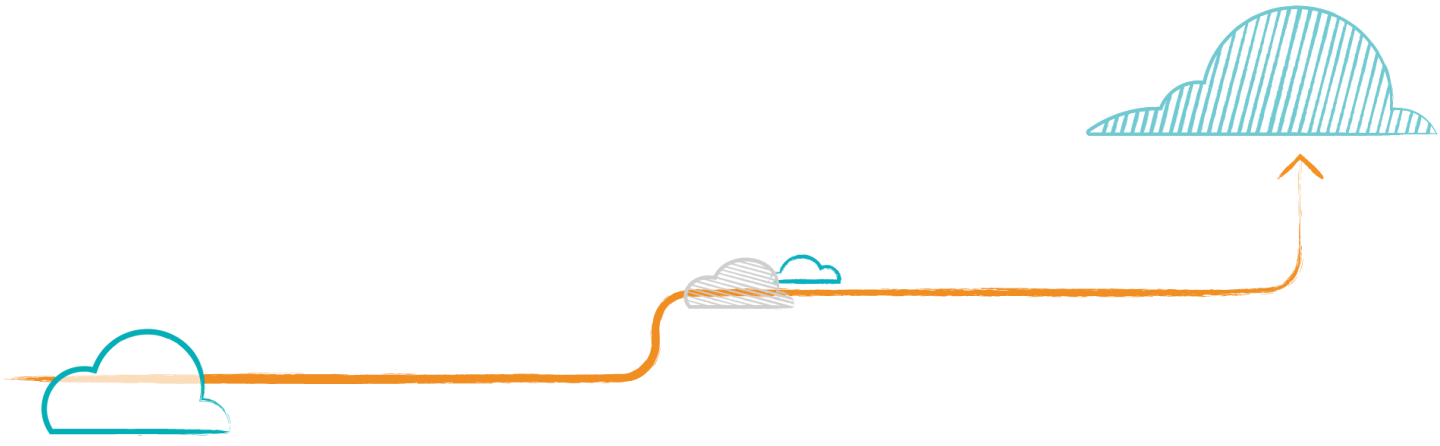
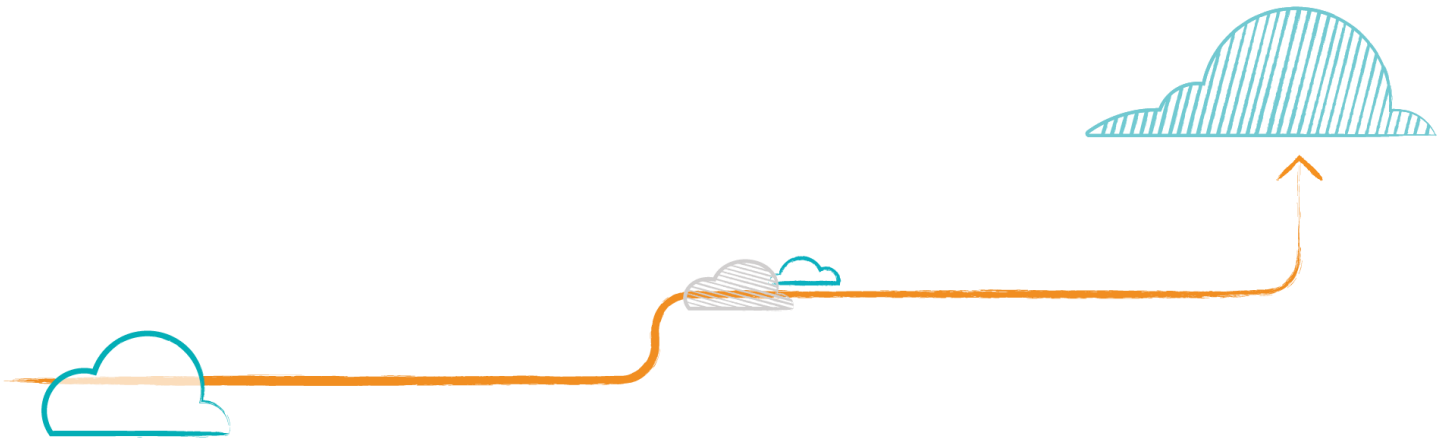
New Relic Practitioner

Lab 04 – Troubleshooting with New Relic





Introduction

This lab is intended to simulate some possible problem scenarios you may find on real world environments. We will be enabling some additional monitoring as well on a distributed application using New Relic agents.

Please note to begin the following lab, you must have gone through the following:

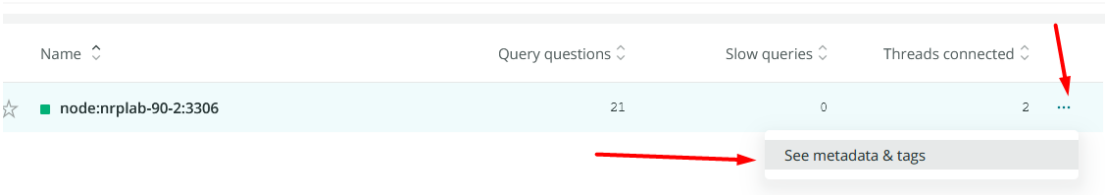
- **Please ensure you have your VM IP and your own New Relic account**

Exercise 1: Creating workloads

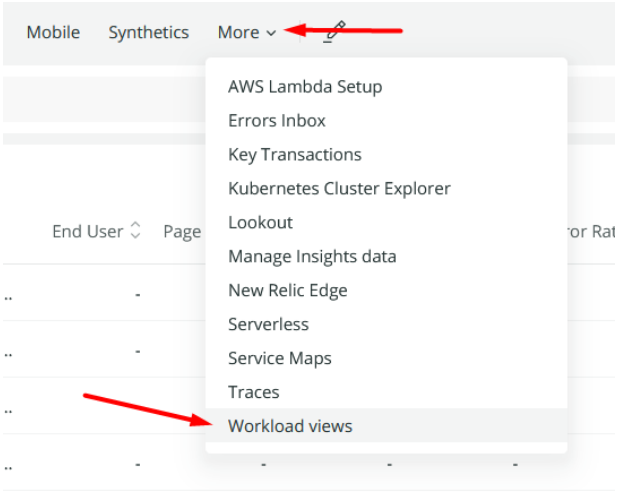
Objective	*Workload monitoring is a powerful tool to understand and observe performance of components from an application and/or environment. We will be configuring 2 workloads to monitor the distinct systems we have.*
Time	15 minutes

For this lab, we will use the New Relic UI uniquely:

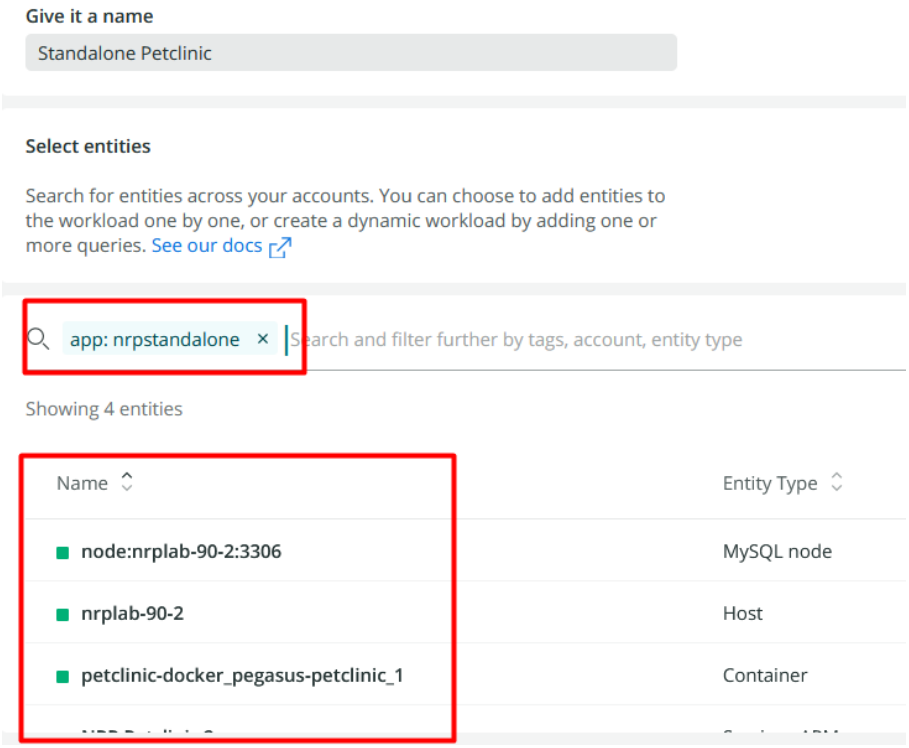
1. In your New Relic account UI, on the top menu click on **Explorer**.
2. Select **MySQL nodes**, click on the "..." icon at the right side of the Petclinic MySQL instance and select **See metadata & tags**.



3. On the **Add new tag** field, type **app:nrpstandalone.** Hit ENTER to add the tag.
4. Repeat steps 1-3 for the **NRP Petclinic** *Service* (the standalone, not the distributed), for the **petclinic-docker_pegasus-petclinic_1** *container,* for both Synthetics checks created and for the **nrplab-XX** *host*.
5. In your New Relic account UI, top menu, click o **More > Workload views**:



6. If you never used Workloads, click on **Create your first workload**.
7. Give it a name, like “Standalone Petclinic”.
8. In the search box, look for “app:nrpstandalone”. You will notice all tagged entities will be listed below.



9. Click on **Add this query**, and then **Create workload**.

10. Check the created workload – it should contain 4 entities.
11. Go back to the explorer and tag with **app:nrpstandalone** the Synthetics monito created on Lab 2.
12. Go back to the **More > Workload views** and click on the created Workload. Notice the Synthetics check will appear because we created a filter over the tag. Other filters are possible as well.

Exercise 2: Creating a dashboard to monitor Petclinic

Objective	*Create a useful dashboard with relevant information from the Petclinic app and the components that are needed for it to work properly*
Time	*20 minutes*

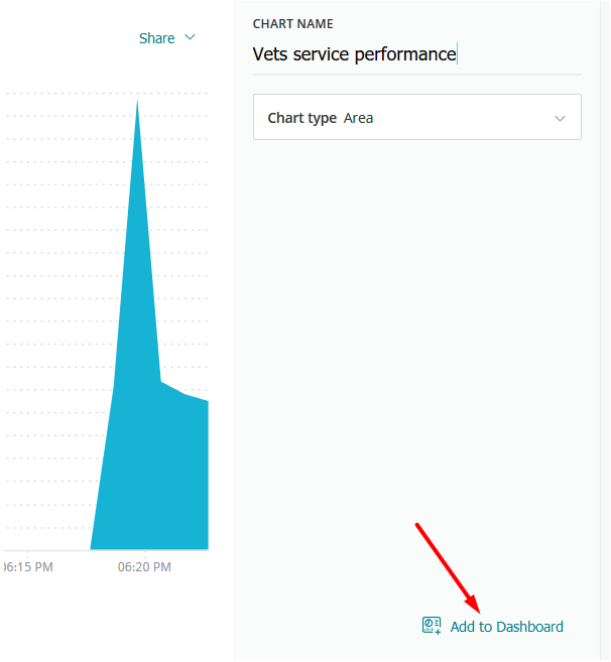
Create a new dashboard and add widgets about the application

Let´s create a dashboard with useful information to monitor the health of our Petclinic app:

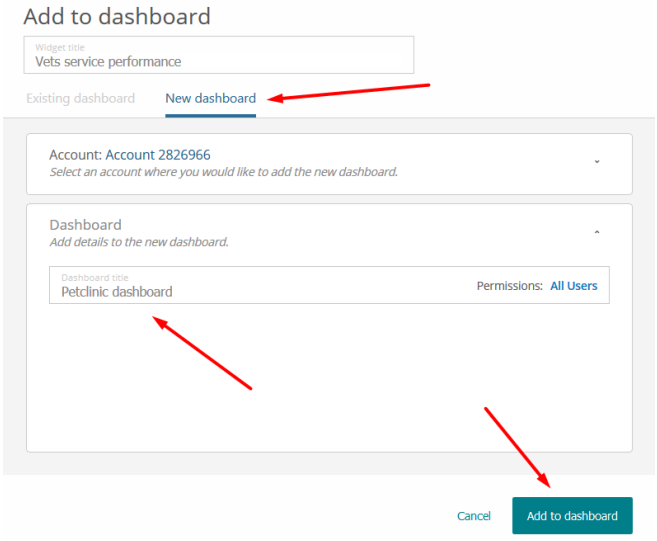
1. Using your own account, click on the top menu called “Query your data”.
2. Type in the below NRQL query, that will look for the Synthetics check duration that we created on the end of lab 03 – specifically, we are monitoring for the Vets service performance and if it is working properly. Replace **Petclinic Owners** by the name of your monitor.

```
FROM SyntheticCheck select average(duration) timeseries where
monitorName='Petclinic Owners'
```

3. Click run, and see the displayed graphic. Feel free to tweak the NRQL as you wish. On the Chart name, type “Owners service performance” and on type, select **Area**. Click on the **Add to dashboard** below.



4. Select **New dashboard,** type a name for it (like “Petclinic dashboard”) and click on **Add to dashboard**.

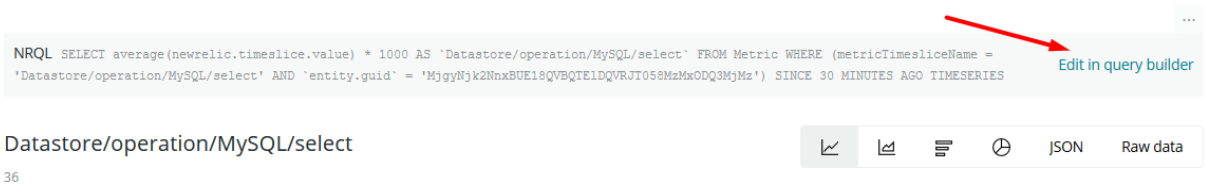


5. Type a new NRQL query, that will look for the number of requests done to our system in the last minute – let´s also set a **warning** and **critical** threshold so we know when our site is being overused beyond our SLAs. Replace **NRP PetClinic** by the name of your monitored app.

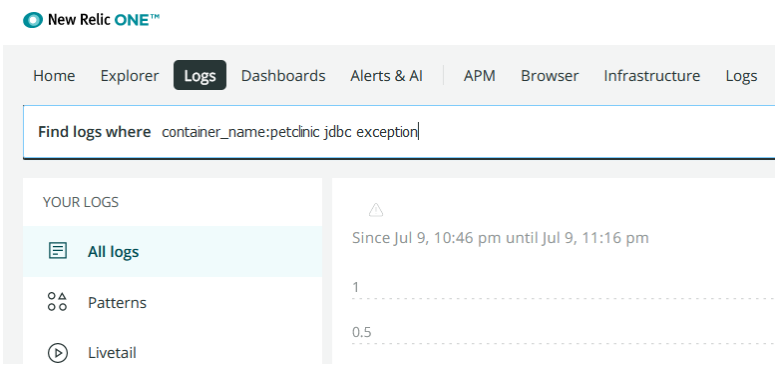
```
FROM Transaction select count(*) as Requests where appName='NRP PetClinic' since 1 minute ago
```

There are other ways of doing the same verification, so feel free to think about it and change the query to anything else that also works the same way.

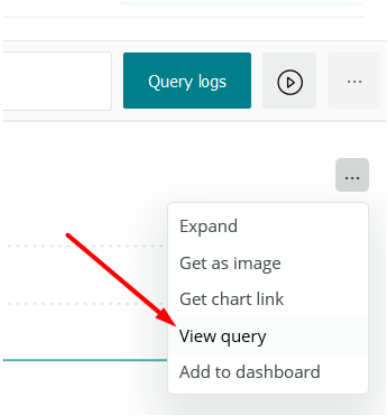
6. Click run, and see the displayed graphic. On the Chart name, type “Petclinic accesses” and on type, select **Billboard –** also configure a **Warning** threshold of **200** and a **Critical** threshold of **300**. Click on the **Add to dashboard** below.
7. Leave selected **Existing dashboard** and make sure the previously created dashboard is selected, and click **Add to dashboard**
8. Now let´s add some metrics to the dashboard. Close the query builder**,** go to the **Entity Explorer**, select your Petclinic application, and on the left menu go to the **Metrics explorer**. Select the Datastore/operation/MySQL/select metric which will give us thee average select query performance for the entire application. Observe you could have selected a “Select” metric from other transactions as well if you want to be very granular.
9. This will create a NRQL query on top of the graph. Hover the mouse pointer over it and click on the **Edit on query builder** option that will appear.



10. On the chart name, type “Select performance in seconds” and click on **Add dashboard,** selecting again the dashboard we are creating.
11. Now let´s monitor some logs. We have logs from the Petclinic application, retrieved from Kubernetes, so let´s look for any error that may appear. Close the query builder and on the top menu, click on **Logs**.
12. On the **Find logs where** box type “container_name:petclinic jdbc exception”. Click on **Query logs**.



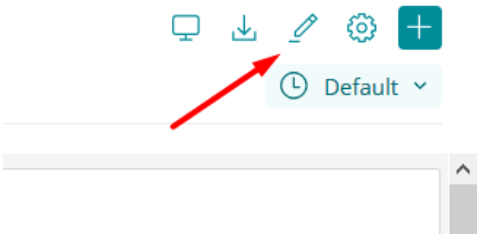
13. You will notice that no log appears, which is normal – we are looking for JDBC database connection errors, and now there isn´t any – our app and DB are working fine. Click on the "…" icon on the right side of the graphic and click on the **View query** option, selecting after this the **Open in query builder** button.



14. This will take us back to the Query builder with a NRQL query to create the graphic we just saw. Let´s change the query a bit to reflect a hard number on how many errors we are seeing. Remove the "SINCE XXXXXX", "UNTIL XXXXXXX" and "TIMESERIES MAX", replace it by "since 1 minute ago", and also add "as JDBCErrors" after the count(*) statement. The final query should look like the below:

```
SELECT count(*) as JDBCErrors from Log WHERE `service_name` = 'nrplab-81-2.petclinic' AND message LIKE '%jdbc%' AND message LIKE
 '%error%' since 1 minute ago
```

15. The query should show a 0 (unless you already have problems). Name it “JDBC errors”, set the **Warning** threshold to 1 and **Critical** to 5, and add it to your dashboard.
16. Close the query builder and click on the **Dashboards** menu on the top. Select your dashboard. You can edit the dashboard and reorganize the widgets.



17. Another thing we can add is a custom event about some aspect of the Petclinic app. For example, there is no event about Services in Linux, but we always can extend the New Relic collection with custom events. We will be creating a Custom Event type called **Service check**, sending the API payload from a curl command and checking the status of the MySQL service. Go to the **New Relic UI > API keys > Insights insert keys** as done on Lab 3.
18. Edit the lab4-exercise2.sh.

```
nano ~/nrp-lab/lab4-exercise2.sh
```

19. Replace the 4th line by the curl command you have on the **Inights insert keys** UI, replacing the json filename for **lab4-exercise2.json** and also **YOUR_KEY_HERE** by your Insert key. Your file should look like this:

The payload, that is decoded on the command, is the following (just so you know what you are sending there is no need to use the below text for anything):

```
[
        {
```

```
      "eventType":"Petclinic services",
      "Service": "mysql",
      "Status":"$(systemctl show -p SubState --value mysql)",
      "Hostname":"$HOSTNAME"
      }
]
```

20. Test your script:

```
~/nrp-lab/lab4-exercise2.sh
```

21. You should see an output with a ID and the word **"success":true** on it. Now let´s add the script to cron so it runs every minute. Run the command **`crontab -e`** and place the following line at the end of the crontab file and saving with CTRL+O and CTRL+X:

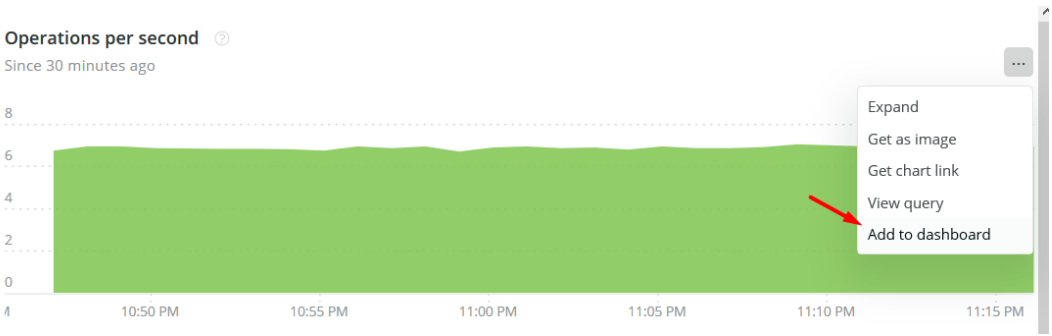
```
* * * * * ~/nrp-lab/lab4-exercise2.sh
```

NOTE: the five * are needed and there must be a space between them. Ignore any other information the file may have.

21. Now go back to the **Query your data,** and enter the following NRQL query:

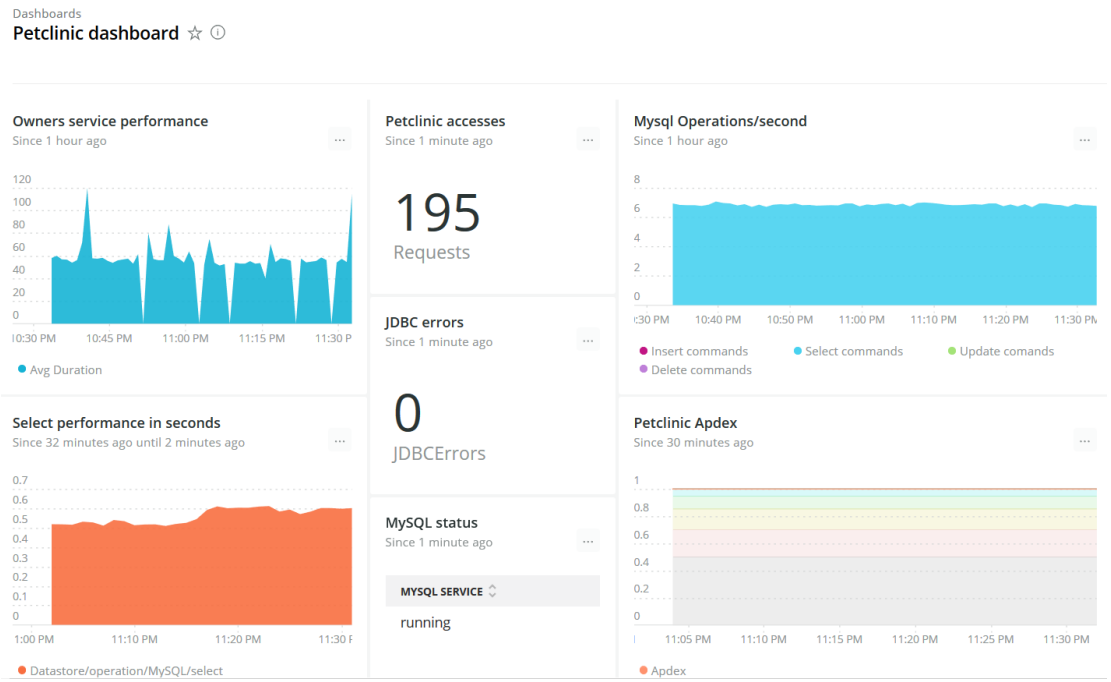
```
FROM `Petclinic Services` select uniques(Status) as 'Mysql Service' since 1 minute ago
```

22. Name it as "Mysql Status" and add to the dashboard as well.
23. Now let´s add a pre-built widget t our dashboard. Go to the **Entity explorer** and look for your MySQL instance – double click it and check the dashboard.
24. On the top widget, click on the "..." icon and select **Add to dashboard.**



25. Select your dashboard and name the widget "MySQL Operations/second".

26. Now go back to the **Entity explorer** and select your Petclinic application. Under the Apdex graph on the top right, click on **Add to dashboard**. Name it Petclinic Apdex and add to your dashboard as well. Notice we could do it with any other graph or information we see out-of-the-box.
27. Go back to your dashboard, edit it clicking on the pencil icon at the top right and save your changes. After some rearrangement, it can be looking very similar to this:



28. Don´t worry if is not exactly the same, but you should have the same information as you see here. Now you have very consistent information about the performance of one specific part of the app, if it is accessible from the outside world, how many requests it is servicing, if it have any errors and if the related services are up and running on real time.
29. Let´s just add one more verification. Go to **Query your data** again and let´s keep track of the free space on our server´s disk. Try to build the query yourself and add it to the dashboard. HINT: the event type that have this kind of information is the “SystemSample”. If needed, ask your instructor for assistance.

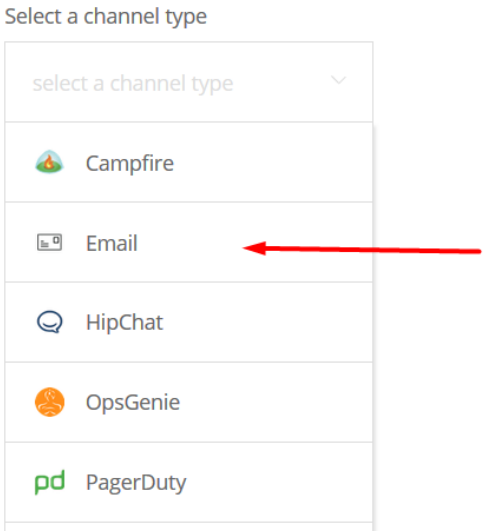
Exercise 2: Creating alerts for our application and environment

Objective	*At this exercise we will create some alerts to let us know if there is any issue with our monitored system.*
Time	*10 minutes*

Create an alert for the Synthetics monitor

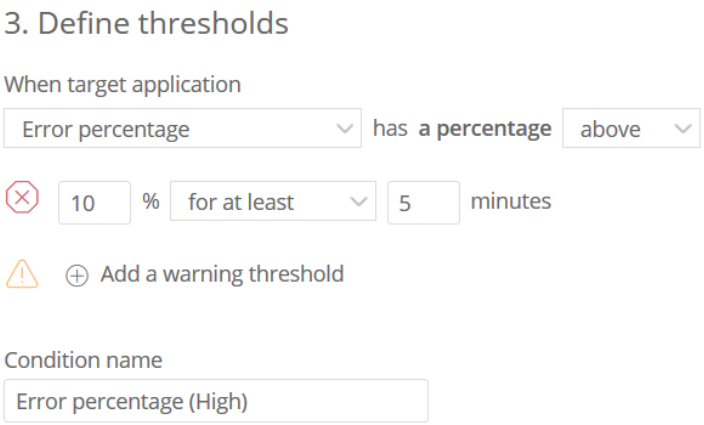
Let´s create a communication channel and alert:

1. On the top menu, click on **Alerts & AI**. ON the left menu, select **Notification channels.**
2. Click on the top right **Add channel** option. Select **Email** from the list.



3. Enter your email (or any email where you want to receive the New Relic alerts). Click **Create channel.**
4. On the left menu, click **Policies**, and then on **New alert policy.**
5. Under **Policy name** type “Petclinic errors”.
6. Scroll down and click on **Create alert policy.** On the next screen, **Create a condition**.

7. Leave **APM, Application Metric** and **Scope to the application** selected. Click on **Next, select entities.**
8. Select your application and click on **Next, define thresholds.**
9. On the **When target application** dropdown, change it to **Error percentage.** On the Critical threshold (red X), configure it to alert on 10%, for at least 5 minutes.



10. Click on **Create condition.**
11. On the **Notification channels** tab, click **Add notification channels.** Click **Email** and then select the email you included previously. Click on **Update policy.**
12. By doing this, you will receive an email when more than 10% of the transactions are errors on your app, which will happen in the next exercises. Feel free to create other policies monitoring other aspects of the application.

Exercise 3: Breaking the system and troubleshooting

Objective	*At this exercise we will break the system, and will perform a GUIDED solution for the issue.*
Time	*30 minutes*

Execute the break script 1

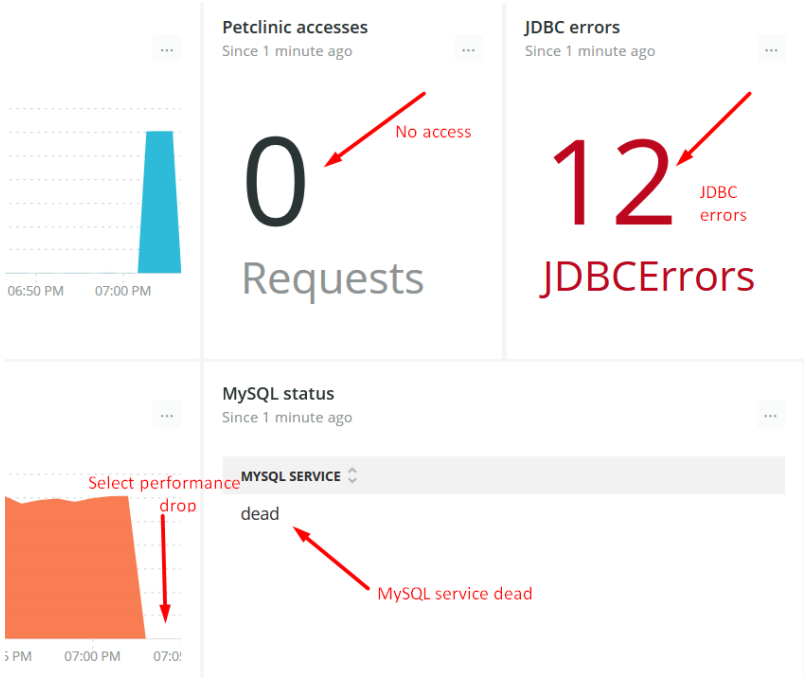
Let´s proceed and break the system:

13. Type in the below command that is a script downloaded when we cloned the nrp-lab project.

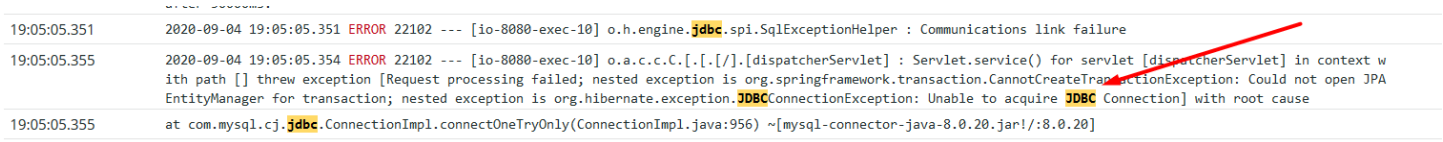
```
~/nrp-lab/lab4-break1.sh
```

NOTE: You could check the script to know what was the problem generated, but for the exercises try not to sneak into it.

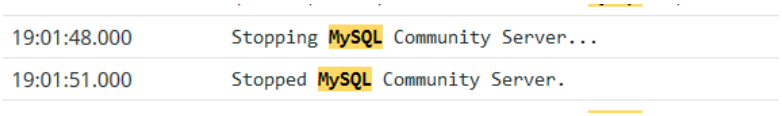
14. Go to the created dashboard and wait around 1 or 2 minutes and observe the information displayed. Also check the Petclinic application and verify it is not working properly (for example, you can´t search for owners or register pet visits).
15. Just looking at the dashboard, could you determine what is the issue? Even if you know already, explore the **Synthetics monitor** you created on Lab 03 and the **Petlinic app** service to check how the issue is affecting it.
16. Go back to the dashboard you created. For now, you should be seeing this:



17. Go to **Logs** and search for “container_name:/petclinic-docker_pegasus-petclinic_1 jdbc exception”. You may notice errors on JDBC connection.



18. Let´s audit the server. Query your logs for “mysql”. Checking the recent events, you may notice the MySQL server was stopped.



19. Now that we determined that the MySQL service is stopped (it happened intentionally), let´s restart it. You can use the command below or use your Linux knowledge to restart MySQL:

```
~/nrp-lab/lab4-fix1.sh
```

20. Check if all is back to normal and your dashboard is reporting a normal behavior again for the app.

Execute the break script 2

Let´s proceed and break the system again:

1. Type in the below command that is a script downloaded when we cloned the nrp-lab project.

```
~/nrp-lab/lab4-break2.sh
```

   NOTE: You could check the script to know what was the problem generated, but for the exercises try not to sneak into it.
2. Go to the created dashboard and wait around 1 or 2 minutes and observe the information displayed. Also check the Petclinic application and verify it is not working properly (for example, you can´t search for owners or register pet visits).

3. Now this problem will not have a guidance. Is not exactly the same as previous one, but you can use the same information to try to determine the issue. Search for events, logs, and any other information that may be helpful. PRO TIP: you can check Lab 2, exercise 4 where we configured the DB for the Petclinic application for some guidance on the fix once you find the issue.
4. You can fix the issue by running:

```
~/nrp-lab/lab4-fix2.sh
```

5. Also, you can ask your instructor for some assistance on resolving the issue if you can´t find it.

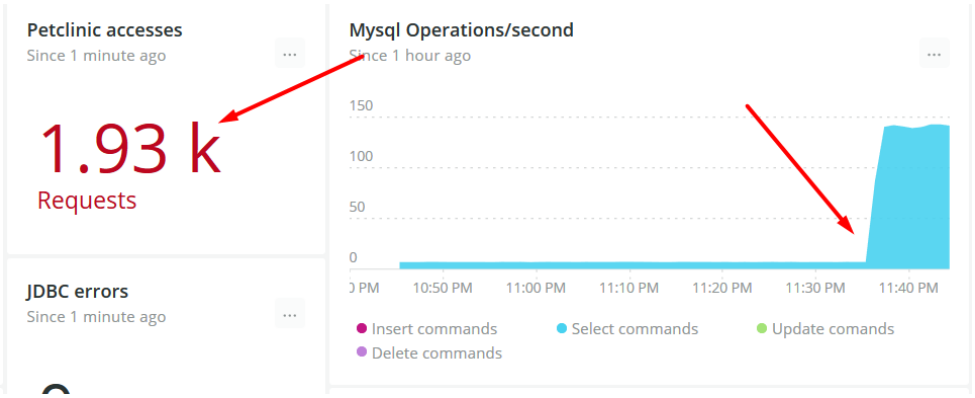
Execute the break script 3

Let´s proceed and break the system again:

1. Type in the below command that is another script downloaded when we cloned the nrp-lab project.

```
~/nrp-lab/lab4-break3.sh
```

   NOTE: Again, you could check the script to know what was the problem generated, but for the exercises try not to sneak into it.
2. Go to the created dashboard and wait around 1 or 2 minutes and observe the information displayed. Also check the Petclinic application and verify if it is working or not.
3. Now this may not be an actual problem, depending on the load our servers can handle. Hopefully, you will notice an increase in the requests and a lot more movement on the MySQL side, but the application should be behaving well – maybe with a minor gap in the Apdex or performance. Think of this as an access spike that could lead to an action or not.



4. We don´t need to do any action to fix the issue, just kill the bots that are using our system:

```
~/nrp-lab/lab4-fix3.sh
```