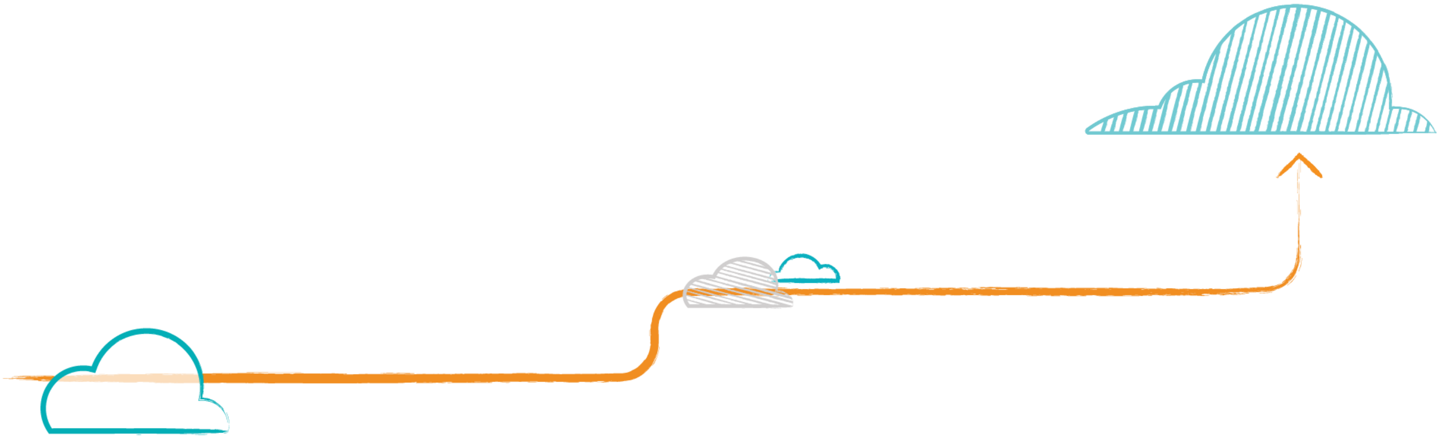
New Relic Practitioner

Installing agents and collecting data



Introduction

This lab is intended to give you experience of deploying the New Relic agents and collecting data from different entities. **Use your own free account License to perform this Lab.**

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

- Please ensure you have received your webserver's IP address and/or Public DNS name!

Exercise 1: Installing the Infrastructure Agent

Objective	*Install the New Relic Infrastructure agent and collect data from your host.* *Manage the lifecycle of the agent.*
Context	*Although installing the agent is an infrequent task, it is the one that causes the most support issues. Unsupported environments, network issues or proxies, permissions problems can all result in missing data or other issues.*

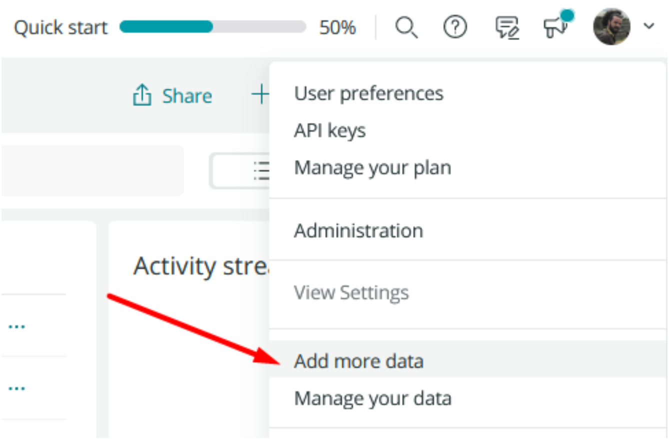
> Pro tip: **Copying and pasting is the safest way to do this installation exercise!**
>
> If you want to install the Infra agent using the online documentation, do it and jump to Exercise 2. The agent installation instructions are present here:
>
> https://docs.newrelic.com/docs/infrastructure/install-configure-manage-infrastructure/linux-installation/install-infrastructure-linux-using-package-manager

Otherwise, follow the next instructions:

Agent Installation using New Relic GUI and Ubuntu command line

In the New Relic GUI you can find all the necessary steps to install the infrastructure agent on a Windows or Linux server. We will use the New Relic interface to create all the necessary server configurations.

1. Open a SSH session to the Linux VM provided by your instructor, using **putty** or any other SSH client.
2. To start go to New Relic and login into **your** FREE account (**not** the one your instructor provided previously).
3. Click on your name at the top right and select **Add more data**.



4. Under **Host operating systems** select **Ubuntu**.
5. Follow the instructions displayed by the New Relic website, executing the commands on the VM provided by your instructor.
6. If an error is returned at the end of the installation, you can safely ignore and proceed with the next step to check if data is being reported – this happens because we use a proxy configuration on the lab that is filtering the agent check.
7. Verify data is appearing in the New Relic Infrastructure or on the **Entity > Infrastructure > Hosts** at https://one.newrelic.com/ The hostname should be nrplab-* or the IP of your instance separated by "-" instead of ".". If you don´t see your lab host on your account, ask your instructor for assistance.

Exercise 2: Agent Lifecycle

Objective	*Learn how to start, stop, and check the status of the agent*
Context	*The agent needs to be running to collect data, and you need to be able to restart the agent whenever the configuration changes.*

You can start, stop, or restart the agent using:

```
sudo service newrelic-infra <start|stop|restart|status>
```

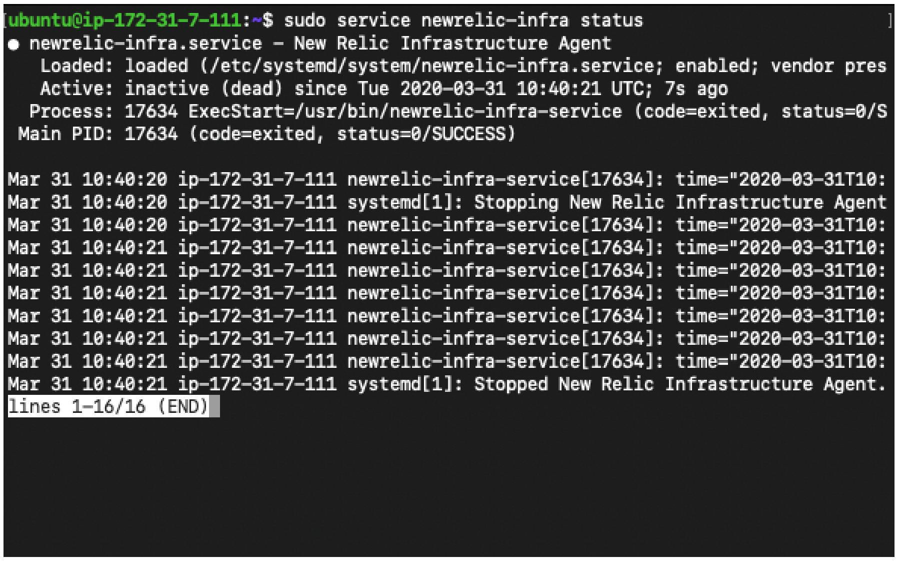
1. Stop the agent if it’s running:

```
sudo service newrelic-infra stop
```

2. Check the status of the agent:

```
sudo service newrelic-infra status
```

Verify that the agent is stopped (it should show the Active state as “Inactive”).



```
[ubuntu@ip-172-31-7-111:~$ sudo service newrelic-infra status                    ]
● newrelic-infra.service - New Relic Infrastructure Agent
   Loaded: loaded (/etc/systemd/system/newrelic-infra.service; enabled; vendor pres
   Active: inactive (dead) since Tue 2020-03-31 10:40:21 UTC; 7s ago
  Process: 17634 ExecStart=/usr/bin/newrelic-infra-service (code=exited, status=0/S
 Main PID: 17634 (code=exited, status=0/SUCCESS)

Mar 31 10:40:20 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:20 ip-172-31-7-111 systemd[1]: Stopping New Relic Infrastructure Agent
Mar 31 10:40:20 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 newrelic-infra-service[17634]: time="2020-03-31T10:
Mar 31 10:40:21 ip-172-31-7-111 systemd[1]: Stopped New Relic Infrastructure Agent.
lines 1-16/16 (END)
```

3. Start the agent:

```
sudo service newrelic-infra start
```

4. Check the status of the agent again:

```
sudo service  newrelic-infra status
```

5. Restart the agent:

```
sudo service newrelic-infra restart
```

6. Check the status of the agent again:

```
sudo service newrelic-infra status
```

Exercise 3: Installing MySQL OHI

Objective	*Install the MySQL On Host Integration*
Context	*New Relic Infrastructure can collect data from several other software installed on the hots. We will be collecting data from the MySQL instance on the lab environment.*

1. After making sure you have the infrastructure agent installed, install MySQL server and the OHI for MySQL with the following 2 commands:

```
sudo apt-get install mysql-server nri-mysql -y
```

2. Execute the following 2 commands to add a new user to the MySQL instance. This user will be used by New Relic to check the DB statistics:

```
sudo mysql -e "CREATE USER 'newrelic'@'localhost' IDENTIFIED BY 'pegasus';"
```

```
sudo mysql -e "GRANT ALL ON *.* TO 'newrelic'@'localhost';"
```

3. Go to the /etc/newrelic-infra/integrations.d directory, and copy the MySQL sample config file, so we can edit it and change according to our configuration:

```
cd /etc/newrelic-infra/integrations.d
```

```
sudo cp mysql-config.yml.sample mysql-config.yml
```

```
sudo nano mysql-config.yml
```

4. The only thing that needs to be changed is the password field – change it to “pegasus”. Also comment or remove the line starting with **SOCKET:**. The file should look like the following:

```
integration_name: com.newrelic.mysql

instances:
  - name: mysql-server
    command: status
    arguments:
        hostname: localhost
        port: 3306
        username: newrelic
        password: pegasus
        remote_monitoring: true
    labels:
        env: production
        role: write-replica
```

5. Restart the newrelic-infra service:

```
sudo service newrelic-infra restart
```

6. Navigate to the infrastructure section on your New Relic account, click on the menu **Explorer > MySQL nodes**. You should see that your MySQL instance is being listed (it should contain your hostname on it). Click on it and navigate on the graphics. Ask your instructor if you can´t find yours.

7. Generate load to your MySQL using the following command and monitor the load in the MySQL dashboard

```
sudo mysqlslap --user=newrelic \
 --password=pegasus \
 --host=localhost \
 --concurrency=20 \
 --iterations=100 \
 --number-int-cols=5 \
 --number-char-cols=20 \
 --auto-generate-sql \
 –verbose &
```

Ignore any errors that may appear about number of connections being exceeded.

If you are not able to see your OHI application, check the following link for troubleshooting: https://docs.newrelic.com/docs/integrations/host-integrations/troubleshooting/not-seeing-host-integration-data

Exercise 4: Starting your Petclinic app

Objective	Get your Petclinic web app up and running in your webserver
Duration	20 Minutes

We will be using a Spring demo app called Petclinic as a test app and will use it for later labs. The application is executed in a Java Tomcat Docker container. To make the process easier, we will execute a script that downloads and installs the application ad dependencies.

Open up the Terminal app and connect to the IP provided to you previously using SSH.

1. To avoid problems and to speed up the lab, a git projejct have been created with some scripts that will prepare for the lab. Feel free to check the script and git project contents – it is publicly available. Clone the project by doing:

```
cd ~
git clone https://lab.pegasus-consultancy.co.uk/newrelic/nrp-lab.git
```

2. Run the following script to install everything needed to run our test Web Application. The script may take around 5 minutes to complete.

```
sudo ~/nrp-lab/lab2-exercise4.sh
```

3. Now let´s test and deploy our application. Enter the application directory:

```
cd ~/nrp-lab/petclinic-docker
```

4. In this directory, we have a docker-compose.yml file that is already configured with all information needed to start our Petclinic test app. Start it by doing:

```
docker-compose -f docker-compose.yml up -d
```

5. OPTIONAL: you can take a look on the logs for the application startup:

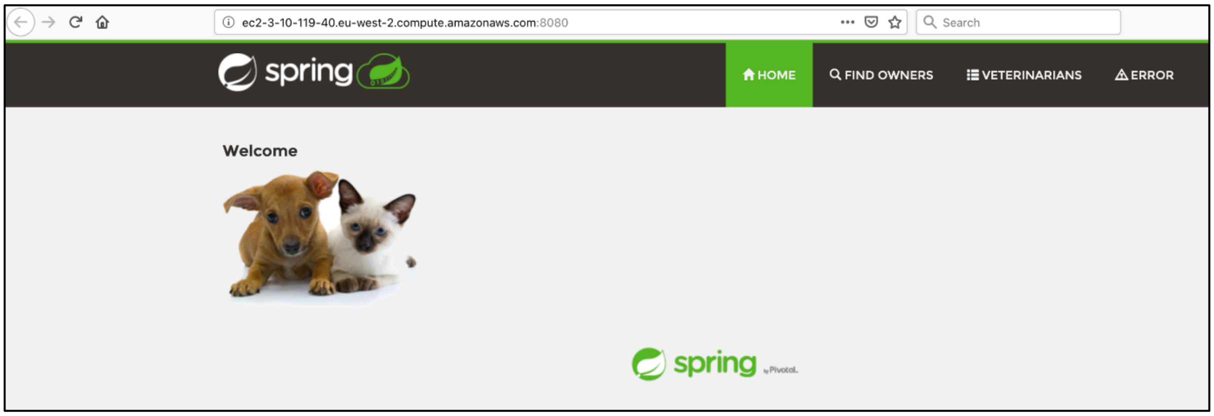
```
docker logs petclinic-docker_pegasus-petclinic_1
```

6. Verify application is running

   In your internet browser type in the following address: http://< Public IP (IPv4) >/petclinic

   Replace < Public DNS (IPv4) > with the Public IP of your instance.

   The Spring Petclinic application page opens.



Explore your website

Right now the application is not instrumented to send data to New Relic. Explore the application by clicking on the links near the top, searching for owners, checking pets, etc.

Exercise 5: Installing a Java New Relic Agent to your Petclinic App

Objective	Get the Petclinic app reporting to your New Relic account
Duration	15 Minutes

Installing the Agent

Follow these steps to download the New Relic agent and configuring it.

1) Let´s stop the petclinic application and download the agent file on the Docker container working directory:

```
cd ~/nrp-lab/petclinic-docker
docker-compose -f docker-compose.yml down
wget download.newrelic.com/newrelic/java-agent/newrelic-agent/current/newrelic.jar
```

2) At this directory, there is a `Dockerfile` that describes how to create the petclinic application container. Let´s add the agent file to the container and build a new image.

   Open the Dockerfile

```
nano Dockerfile
```

   Add the following line at the end of the file:

```
COPY newrelic.jar /newrelic-agent/newrelic.jar
```

   Lastly, let´s build a new image with the New Relic agent embedded on it:

```
docker build . -t pegasus-petclinic:2.4.5-newrelic
```

3) Now we need change the docker-compose.yml file to tell Tomcat to start with the agent and to send the instrumentation information to the agent using environment variables.
   First edit the docker-compose file:

```
nano docker-compose.yml
```

Now we need to change the docker-compose.yml file. On YML files, indentation is EXTREMELY important. Edit the file changing:

- “image” to “pegasus-petclinic:2.4.5-newrelic”
- Add the “environment:” section, in the same level as “ports:”
- JAVA_OPTS variable to "-javaagent:/newrelic-agent/newrelic.jar"
- Add the NEW_RELIC_LICENSE_KEY environment variable with your license key (Ingestion API key) as a value
- Add the NEW_RELIC_APP_NAME environment variable with a name for your application (use a meaningful name as this will appear on New Relic UI).

Your file should look like this:

```
version: '3.1'
services:
  pegasus-petclinic:
    extra_hosts:
      - mysql:172.17.0.1
    image: pegasus-petclinic:2.4.5-newrelic
    restart: always
    ports:
      - 80:8080
    environment:
      - JAVA_OPTS="-javaagent:/newrelic-agent/newrelic.jar"
      - NEW_RELIC_LICENSE_KEY=<PUT_YOUR_NR_KEY_HERE>
      - NEW_RELIC_APP_NAME=NRP Petclinic
```

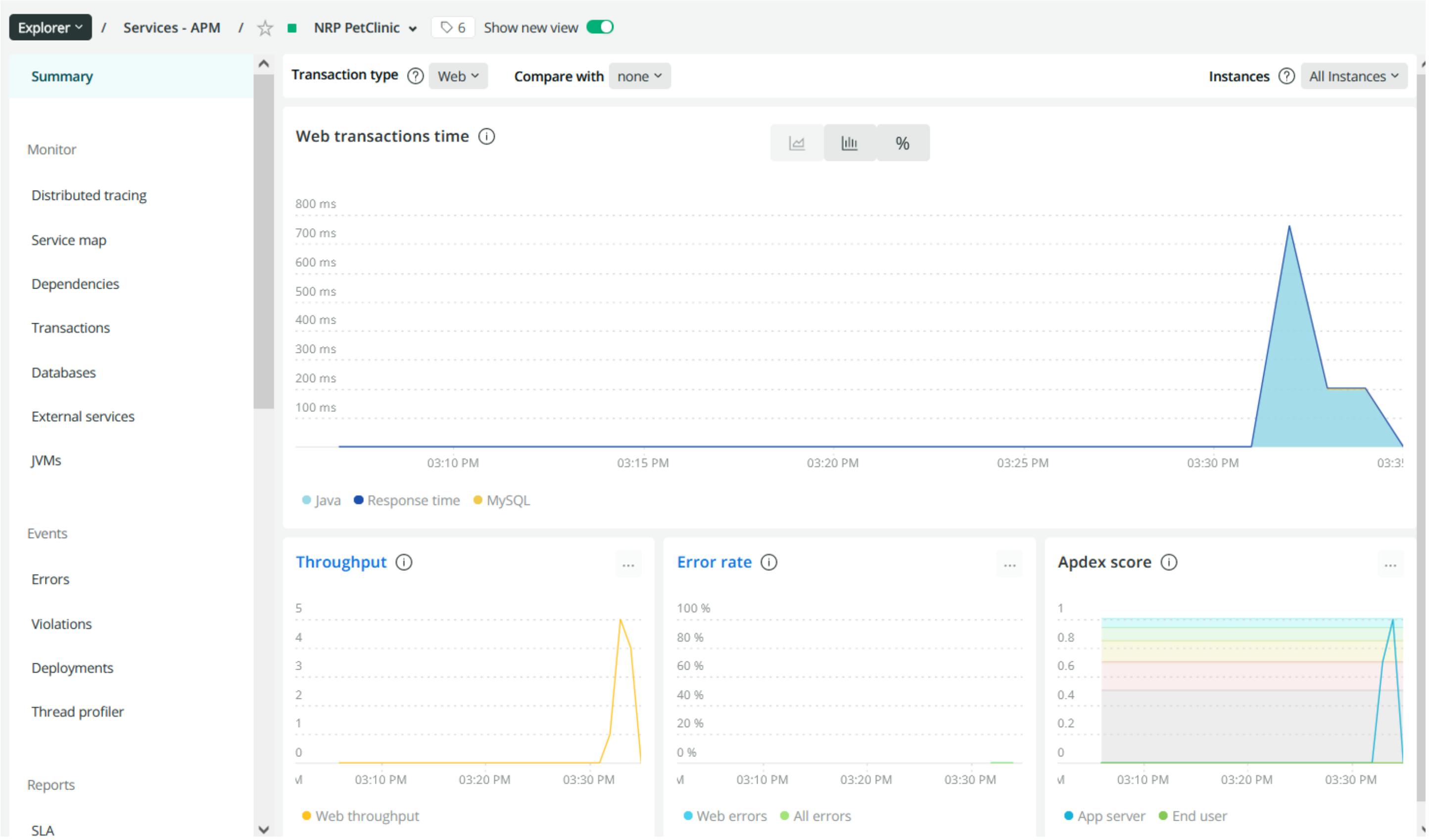
Save the file and exit. Start the petclinic application again:

```
docker-compose -f docker-compose.yml up -d
```

Check the container logs again to check if no errors are observed:

```
docker logs petclinic-docker_pegasus-petclinic_1
```

4) On your web browser, go to the Petclinic application (http://<YOUR_LAB_IP>/petclinic) and navigate over the pages to generate some load.
5) On your New Relic account, click on APM. You should see your application on the list. Click on it and observe the default dashboard displaying the load generated by you.



6) In order to generate some load automatically, there is a Jmeter script that will be using the petclinic website. To use it, run the following command:

```
~/nrp-lab/jmeter_start.sh
```

7) Check your application in New Relic, and you should notice a consistent usage of around 200 requests / min. This is the script using your system.

Exercise 6: Enabling log collection using Fluentd

Objective	Enable log collection on a Linux machine using Fluentd.
Duration	15 Minutes

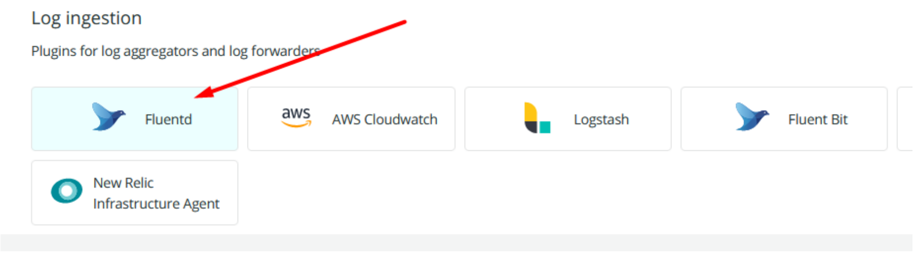
Installing Fluentd

Instead of using the New Relic Infra agent to collect logs, we will follow these steps to install Fluentd and td-agent (open source tools used by developers and infra administrators) on a Linux machine and forward OS logs and also Docker logs.

1) Install the td-agent and fluent gem package to the Linux VM – we will use another script prepared to simplify the process (you can take a look on it to check the commands that actually perform the installation):

```
sudo ~/nrp-lab/lab2-exercise6.sh
```

2) On the New Relic UI, go to **Add more data**. Under **Log ingestion** select **Fluentd**.



Log ingestion
Plugins for log aggregators and log forwarders
Fluentd
aws AWS Cloudwatch
Logstash
Fluent Bit
New Relic Infrastructure Agent

3) On Step 1, select **td-agent**. Copy the generated command and run it on your Linux VM with **sudo** before the command.

```
sudo td-agent-gem install fluent-plugin-newrelic
```

4) Go back to the new Relic UI. On **Step 2** provide the file paths we will be monitoring**: /var/log/syslog,/var/log/auth.log**
5) Click on **Generate config block**.

6) The block generated need to be pasted on the td-agent.conf file, replacing all its content. To do so, clean and edit the file:

```
sudo truncate -s 0 /etc/td-agent/td-agent.conf
sudo nano /etc/td-agent/td-agent.conf
```

7) Copy the config block from the New Relic UI and paste on the file. Save it and exit the text editor.
8) Restart the td-agent service:

```
sudo service td-agent restart
```

9) After a few minutes, you will be able to see logs on the New Relic interface, at the **Logs** section. To check if you are seeing actual data, try to login to your monitored Linux VM with a non-existent username (like "superhacker", you will receive an authentication error) and see the error being logged (you can search logs for this string: "Invalid user").
10) Now let´s add our petclinic application logs. As it is a docker based application, we need to tell fluentd to accept logs from containers. For this, edit the td-agent.conf file:

```
sudo nano /etc/td-agent/td-agent.conf
```

11) Add the following block to the end of the file (don´t remove the existing configuration):

```
<source>
  @type forward
  port 24224
  bind 0.0.0.0
</source>
```

12) Save and exit. Restart the td-agent service.

```
sudo service td-agent restart
```

13) Let´s edit the docker-compose file for our app to enable log forwarding:

```
cd  ~/nrp-lab/petclinic-docker
nano docker-compose.yml
```

14)Add the lines in **bold** at the end of the file. The configuration should look like this:

```
version: '3.1'
services:
  pegasus-petclinic:
    extra_hosts:
      - mysql:172.17.0.1
    image: pegasus-petclinic:2.4.5-newrelic
    restart: always
    ports:
      - 80:8080
    environment:
      - JAVA_OPTS="-javaagent:/newrelic-agent/newrelic.jar"
      - NEW_RELIC_LICENSE_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      - NEW_RELIC_APP_NAME=NRP Petclinic
    logging:
      driver: "fluentd"
```

15)NOTE: Identation Is IMPORTANT on YML files, “logging:” should be on the same column as “environment:” and driver should be 2 spaces ahead of logging. Restart the petclininc application:

```
docker-compose -f docker-compose.yml down
docker-compose -f docker-compose.yml up -d
```

16)Navigate to the New Relic UI, and check on **Logs** if the petclinic application logs are being displayed. As we have the test load generator, you should see logs regarding the application start and SQL queries that are being executed (same output as if you run docker logs <CONTAINER_NAME>).

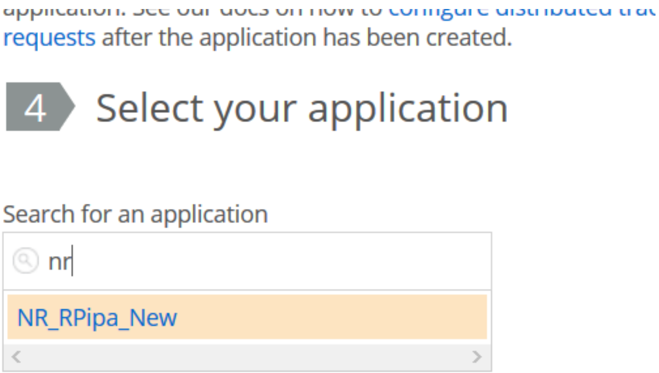
Exercise 7: Installing the Browser Agent

Objective	New Relic Browser represents the reality of your end user experience. In this exercise you will try to use New Relic Browser automatically on the Petclinic application.
Duration	5 Minutes

As our test application is using Tomcat as a framework, as soon as we have enabled the APM agent, the browser agent have been enabled as well – but we will be running over the steps to enable the browser monitoring anyway.

Installing the Browser Agent using the Browser UI instructions

1. From the New Relic UI, click on the **Add more data** option. Under **Browser metrics** select **New Relic Browser.**
2. Leave the **Enable via New Relic APM** option selected.
3. Scroll to the bottom to **2: Select your Application**
4. Search for your Application



application. See our docs on how to configure distributed trac requests after the application has been created.

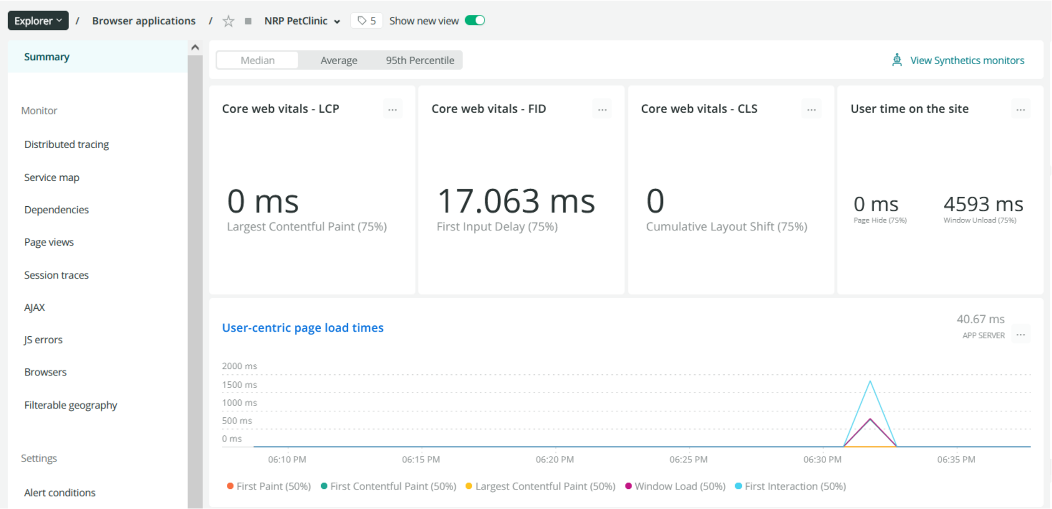
4 Select your application

Search for an application

nr

NR_RPipa_New

5. Click **Enable.**
6. Now navigate to your petclinic application and generate some Browser load. IMPORTANT: if you have a AdBlocker, disable it – those tools usually block any data collection from a client browser.
7. On the New Relic UI, click on **Browser**.
8. You should see some traffic.



Core web vitals - LCP: 0 ms Largest Contentful Paint (75%)

Core web vitals - FID: 17.063 ms First Input Delay (75%)

Core web vitals - CLS: 0 Cumulative Layout Shift (75%)

User time on the site: 0 ms Page Hide (75%) 4593 ms Window Unload (75%)

40.67 ms APP SERVER

9. Explore the information displayed, like traces, browsers and geography.

Exercise 8: Creating a Synthetics monitor

Objective	New Relic Synthetics provides manufactured end-user experience testing. In this exercise you will find out how to test end user experiences proactively, ensuring your site is always open for business, that critical paths continue to work, and gathering a wealth of useful data for subsequent analysis.
Duration	10 minutes

We will create a simple browser monitor to demonstrate Synthetics testing. Execute this step at your account.

1. On the New Relic UI, click on **Add more data.** Select New Relic Synthetics.
2. Select **Page Load performance** (**Simple Browser)** for your monitor type.
3. Enter an appropriate name for the monitor.
4. Enter the URL for your Petclinic application.
5. Set the schedule to once every 5 minutes.
6. Click on **Select locations.** Select any locations you wish.
7. Click **Save Monitor**

Wait for the monitor to run, and then verify that you can see successful results in the Synthetics overview.

Exercise 9: (CHALLENGE) Instrumenting a PHP application

Objective	New Relic can collect data from many other types of applications – PHP is used a lot as well. This Lab is quiz, so let´s use the knowledge acquired so far to check what is needed to instrument a Wordpress application.
Duration	10 minutes

This lab is intended for you to instrument the public Wordpress docker container with New Relic agents:

1. Execute the following script to create the database needed for Wordpress (you can check the commands opening the script file):

```
sudo ~/nrp-lab/lab2-exercise9.sh
```

2. Enter the Wordpress directory at out nrp-lab project:

```
cd ~/nrp-lab/wordpress-docker
```

3. Open the `Dockerfile` file. Answer these questions:
    a. From what image we will be starting to implement our instrumented image?
    b. What Ubuntu packages installed are needed for the New Relic instrumentation?
    c. What is the name of the New Relic agent configuration file and where it will be copied to when we build the Docker image?
4. Open the `newrelic.ini` file and check the contents. Answer these questions:
    a. What are the environment variables needed for the New Relic instrumentation to work?

   b. Can you think of any other configuration we could add as an environment variable? You can check the possible ones at https://docs.newrelic.com/docs/agents/php-agent/configuration/php-agent-configuration/

5. Now let´s build our image running the following command:

```
docker build . -t docker-wordpress:1.0
```

6. Wait for the image creation to complete.
7. We have prepared a `docker-compose.yml` file on the same directory. Open it, change it accordingly (remember the environment variables observed on the newrelic.ini file) and save.
8. Using the knowledge you have acquired so far, try to execute the following:
   a. Start the Wordpress container using the `docker-compose.yml` configuration
   b. Instrument the `docker-compose.yml` file with your API key and a suitable App Name.
   c. Start the container.
9. If all goes well, you should be able to access the Wordpress application on http://<YOUR_LAB_IP>:8000/ - interact with the application by initializing it and accessing the webpages. Also confirm Browser monitoring is working, and if not, try to enable it.
10. Check if the application reports data to New Relic. If needed, ask your instructor for assistance.

Exercise 10: (OPTIONAL) Connecting your New Relic account with AWS

Objective	New Relic can collect data from all AWS services running on one specific account. This Lab is optional, as it requires you to have a AWS account – you can create one just for testing with this lab.
Duration	10 minutes

This lab is intended for you to connect your New Relic account with your AWS account. You will need:

11. Access to an AWS account. If you don´t have one, create one at https://console.aws.amazon.com . Take advantage of the Free Tier to create instances or Lambda functions. If you are new to AWS, you can ask your instructor for some guidance.
12. Follow the instructions on the **AWS Integration** documentation: https://docs.newrelic.com/docs/integrations/amazon-integrations/get-started/connect-aws-infrastructure, enabling at least the **EC2** monitoring.
13. Check if you can see the AWS account information on the **Infrastructure > AWS**, or on the **Entity explorer.**