



LOGGING & MONITORING WORKSHOP

Platform Engineering

Platform Engineering

Table of Contents

Clone the workshop	2
from codebase https://github.com/arunma/spring-logmon	2
Launch all the containers	2
on your local machine using docker-compose	2
Create dummy logs using Swagger UI.....	2
Check your logs in Kibana	4
Create a new Index Pattern.....	5
Let's filter for Errors from the logs	6
Try using a richer query criteria	7
Filtering all logs from Controller	8
Visualize	8
Create a pie chart.....	10
End of workshop	12
OS Specific instructions	13
Common Errors and Solutions.....	18

Clone the workshop

from codebase <https://github.com/arunma/spring-logmon>

OS Specific Instructions are available at the bottom of the workshop

The cloned repository has four containers wrapped in a docker-compose.yml

1. maid-account-manager spring application
2. Elasticsearch datastore
3. Logstash log shipper
4. Kibana UI

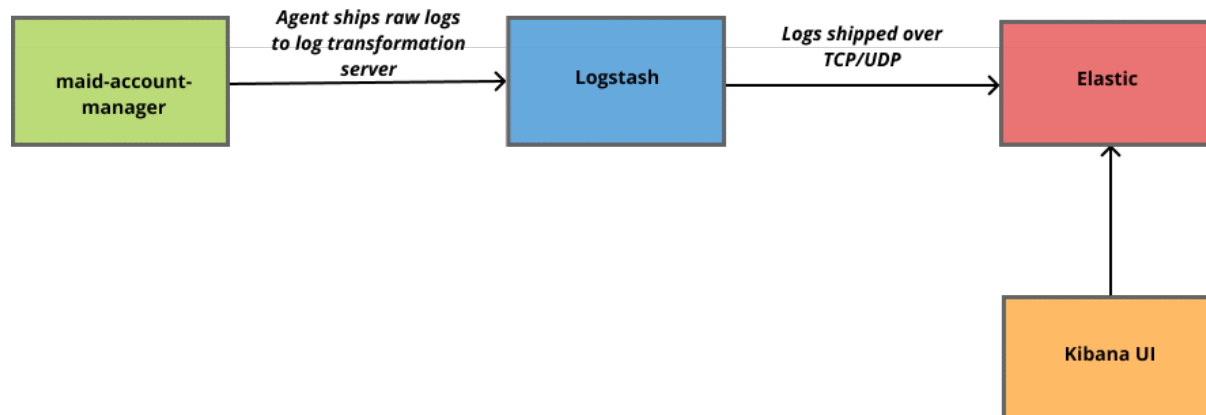
Launch all the containers

on your local machine using docker-compose

```
cd <location of spring-logmon >
```

```
# eg. cd /Users/arunma/projects/spring-logmon
```

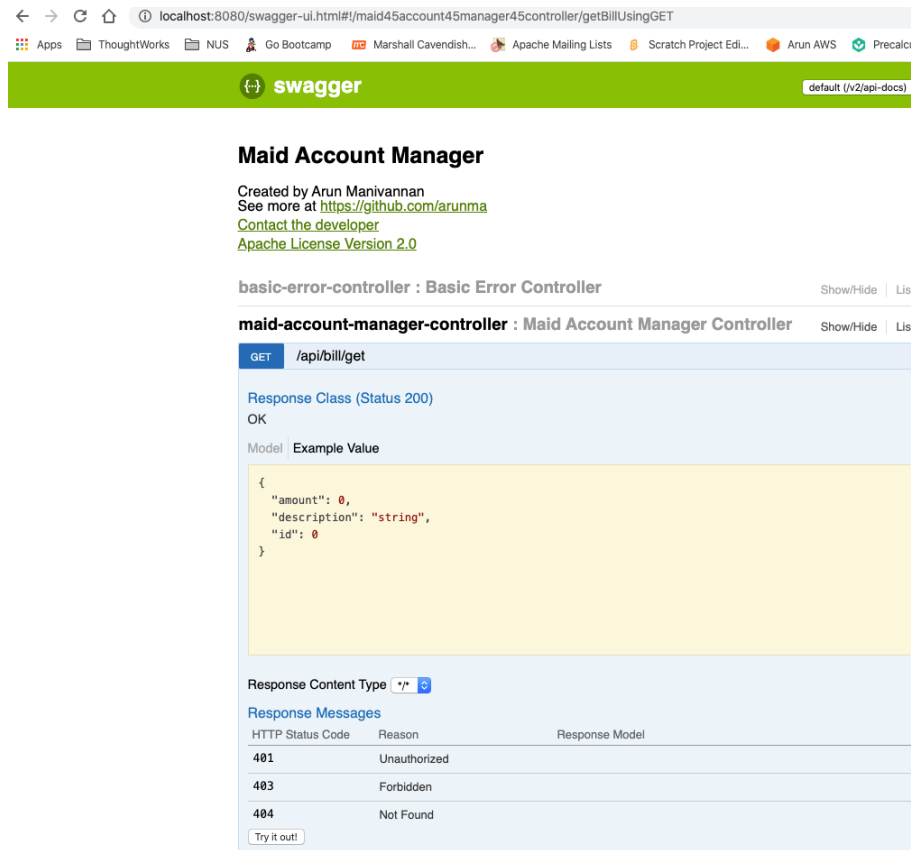
```
docker-compose -f docker/docker-compose.yml up --build
```



Create dummy logs using Swagger UI

Visit the Swagger UI url on your localhost at <http://localhost:8080/swagger-ui.html>

Expand the maid-account-manager-controller and click “Try it out” a few times.



The screenshot shows the Swagger UI for the 'Maid Account Manager' API. The browser address bar indicates the URL is `localhost:8080/swagger-ui.html#/maid45account45manager45controller/getBillUsingGET`. The Swagger logo is visible in the top left, and a 'default (/v2/api-docs)' button is in the top right. The main heading is 'Maid Account Manager', followed by metadata: 'Created by Arun Manivannan', 'See more at <https://github.com/aronma>', 'Contact the developer', and 'Apache License Version 2.0'. Below this, two controllers are listed: 'basic-error-controller : Basic Error Controller' and 'maid-account-manager-controller : Maid Account Manager Controller'. The 'maid-account-manager-controller' is selected, showing a 'GET /api/bill/get' endpoint. The 'Response Class (Status 200)' is 'OK'. An 'Example Value' is shown in a yellow box:

```
{  "amount": 0,  "description": "string",  "id": 0}
```

. Below the example, the 'Response Content Type' is set to 'application/json'. A 'Response Messages' section contains a table of HTTP status codes and reasons.

HTTP Status Code	Reason	Response Model
401	Unauthorized	
403	Forbidden	
404	Not Found	

A 'Try it out!' button is located at the bottom left of the response messages section.

The MaidAccountController's implementation creates exception roughly 40% of the time.

```
nl x MaidAccountManagerController.java x logback-spring.xml x application.properties x MaidApplication.java x

import java.time.Month;
import java.time.format.TextStyle;
import java.util.Date;
import java.util.Locale;
import java.util.Random;
import java.util.concurrent.ThreadLocalRandom;

@RestController
@RequestMapping(MaidAccountManagerController.BASE_URL)
public class MaidAccountManagerController {

    public static final String BASE_URL = "/api/bill";
    private Logger log = LoggerFactory.getLogger(MaidAccountManagerController.class);

    @GetMapping(path = "/get")
    @ResponseStatus(HttpStatus.OK)
    public MaidUsageBill getBill() {
        double random = new Random().nextDouble();
        if (random < 0.4) {
            log.error("Random Exception for value " + random, new RuntimeException("Random Exception is thrown @ " + new Date()));
        }
        String randomMonth = Month.of(ThreadLocalRandom.current().nextInt( origin: 1, bound: 12)).getDisplayName(TextStyle.FULL, Locale.US);
        String randomYear = String.valueOf(ThreadLocalRandom.current().nextInt( origin: 2000, bound: 2020));
        int id = Math.abs(ThreadLocalRandom.current().nextInt());
        double amount = Math.ceil(ThreadLocalRandom.current().nextDouble( origin: 500, bound: 1200));
        log.info("Returning Maid Usage for the period {}, {} with Id {} and Amount {}", randomMonth, randomYear, id, amount);
        return new MaidUsageBill(
            id,
            amount,
            String.format("Version 2 of the API - Payment for %s %s", randomMonth, randomYear)
        );
    }
}
```

Check your logs in Kibana

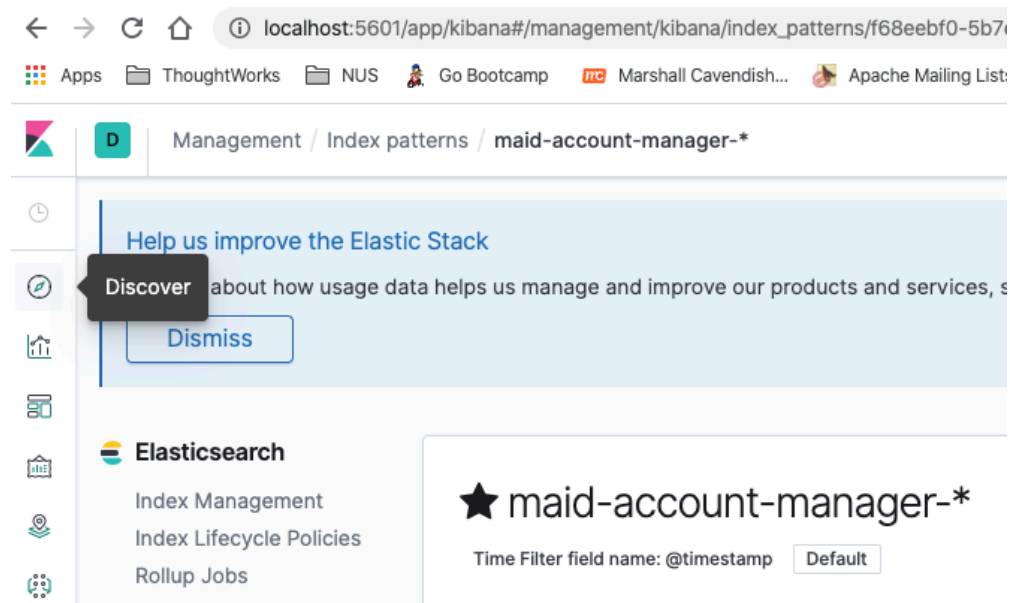
Once enough logs have been created, go to the Kibana UI at <http://localhost:5601/app/kibana>

Create a new Index Pattern

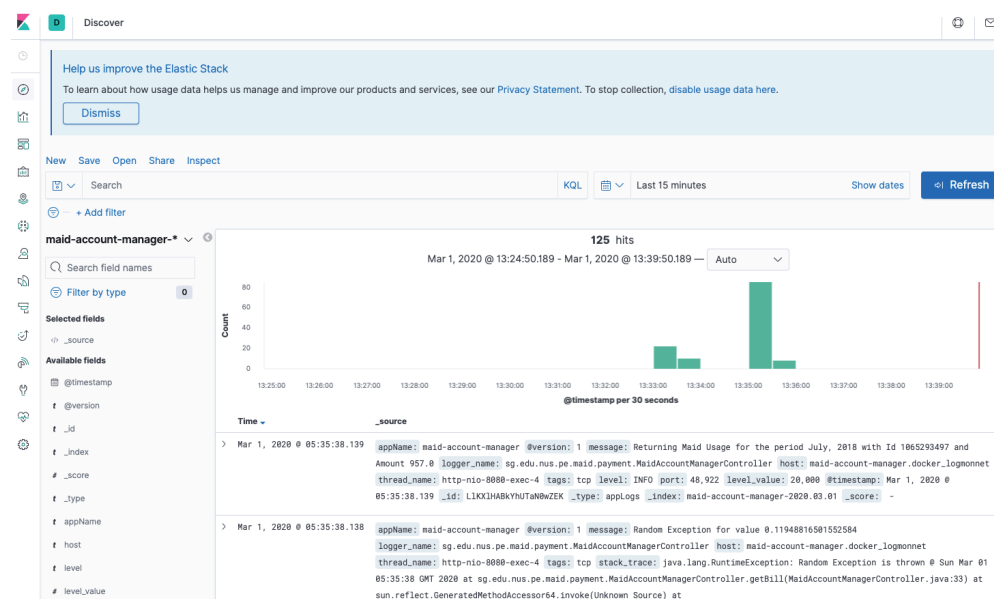
The screenshot shows the Kibana 'Create index pattern' page. The browser address bar indicates the URL: localhost:5601/app/kibana#/management/kibana/index_pattern?_g=(). The left sidebar shows the 'Management' section with 'Index patterns' selected. The main content area is titled 'Create index pattern' and includes a sub-header 'Step 1 of 2: Define index pattern'. A text input field contains the index pattern 'maid-account-manager-*', which is highlighted with a red box. Below the input field, a success message states: 'Success! Your index pattern matches 1 index.' The index pattern 'maid-account-manager-2020.03.01' is listed below the message. The 'Rows per page' is set to 10.

The screenshot shows the Kibana 'Create index pattern' page, Step 2 of 2: Configure settings. The main content area is titled 'Create index pattern' and includes a sub-header 'Step 2 of 2: Configure settings'. A text input field contains the index pattern 'maid-account-manager-*'. Below the input field, a success message states: 'Success! Your index pattern matches 1 index.' The index pattern 'maid-account-manager-2020.03.01' is listed below the message. The 'Rows per page' is set to 10. The 'Time Filter field name' is set to '@timestamp'. The 'Include system indices' checkbox is checked. The 'Create index pattern' button is visible at the bottom right.

Click the discovery button



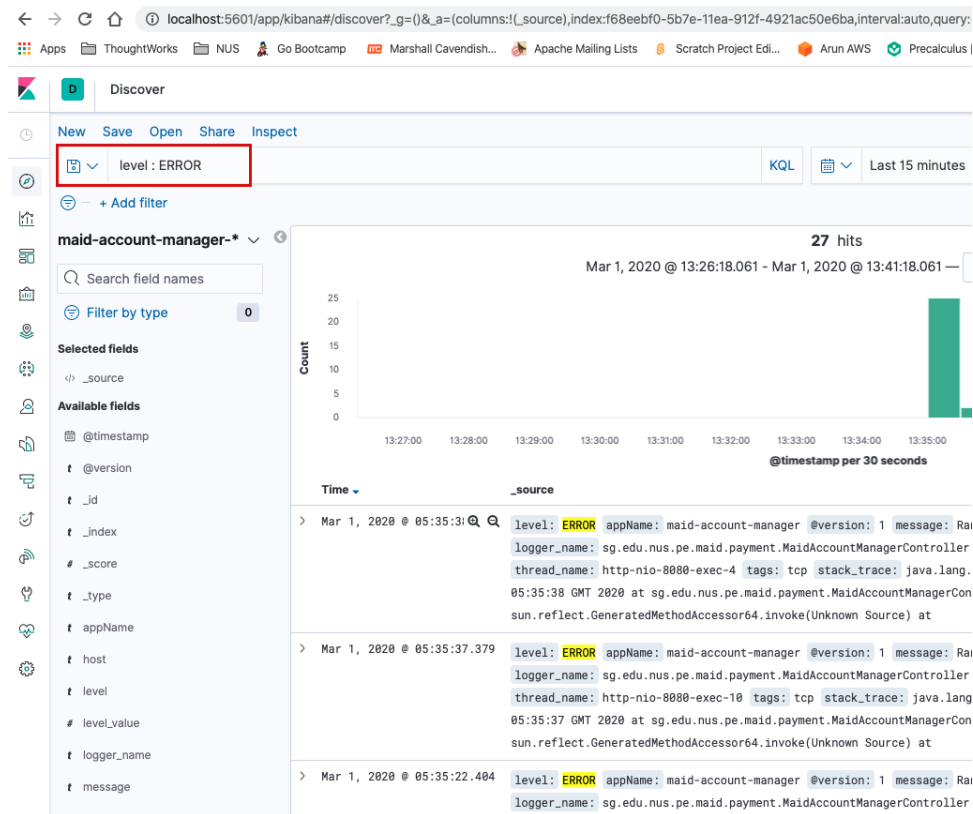
You should see all the logs already captured



Let's filter for Errors from the logs

In the "Search" bar, issue the following search criteria. You can also see the count of errors in the histogram.

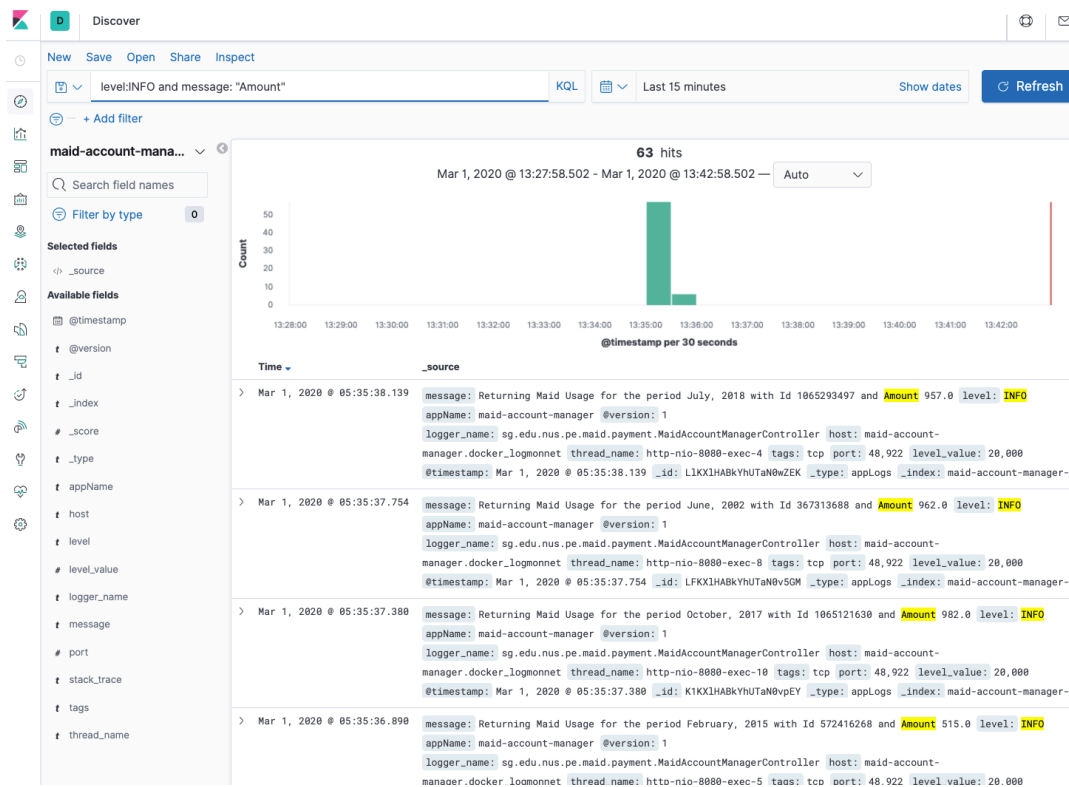
`level:ERROR`



Try using a richer query criteria

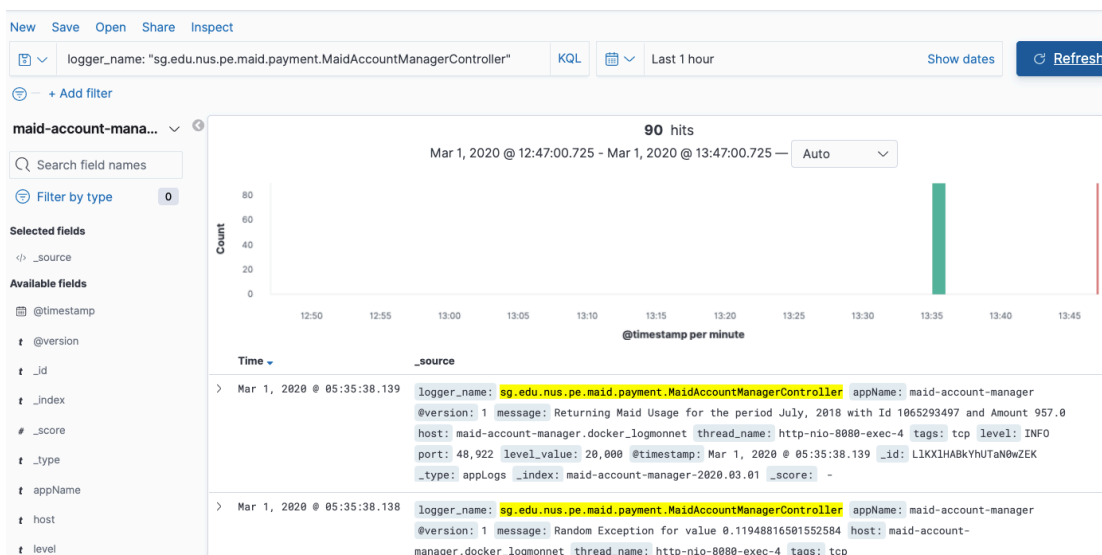
Filter INFO level logs that has "Amount" in the log message

`level:INFO` and `message: "Amount"`



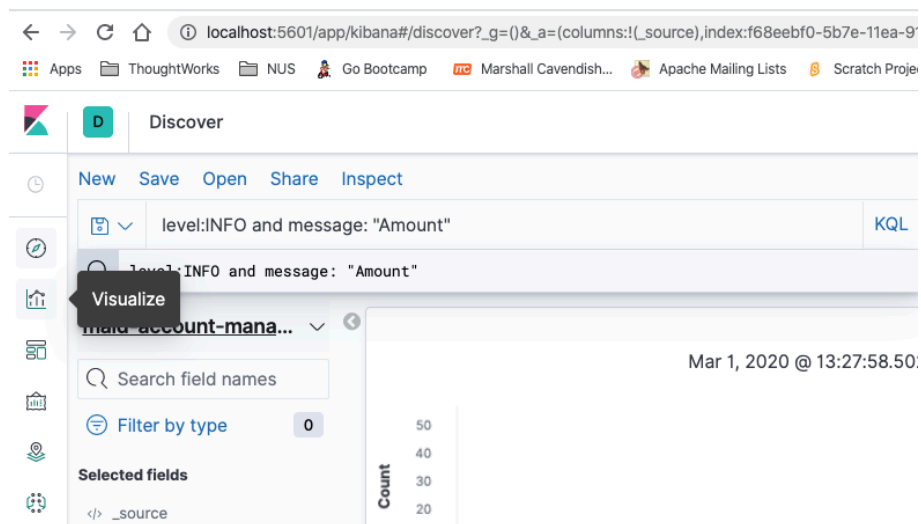
Filtering all logs from Controller

logger_name: "sg.edu.nus.pe.maid.payment.MaidAccountManagerController"

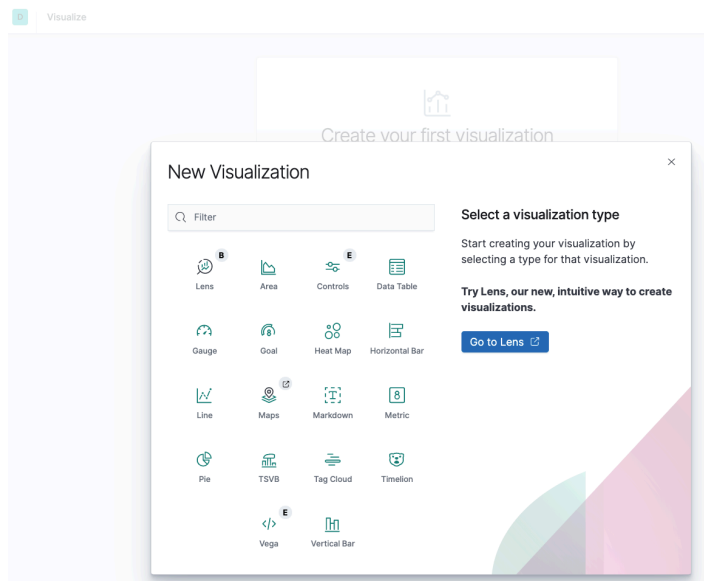


Looks like we have 90 log messages generated from the MaidAccountManagerController. Let's visualize how much of these are errors and how much are good responses.

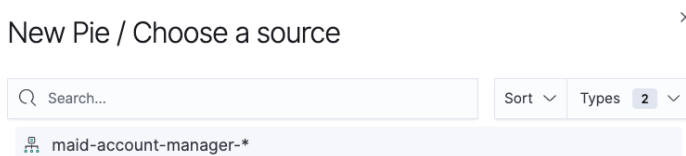
Visualize



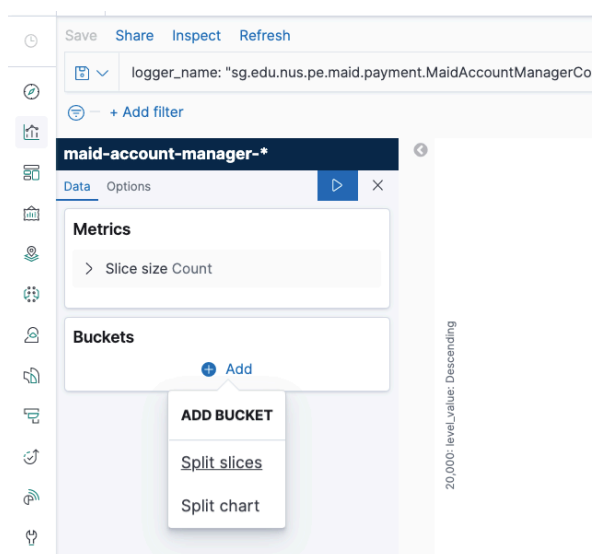
Create a pie chart



New Pie / Choose a source





Create a new Split slice



Select “Terms”

Buckets

Split slices



Aggregation

Select an aggregation

Date Range

Filters

Histogram

IPv4 Range

Range

Significant Terms

Terms

In the Field, select “level_keyword”

maid-account-manager-*

DataOptions

> Slice size Count

Buckets

Split slices

AggregationTerms

Terms

Terms help

Field

level.keyword

Order by

Metric: Count

OrderDescending

Size5

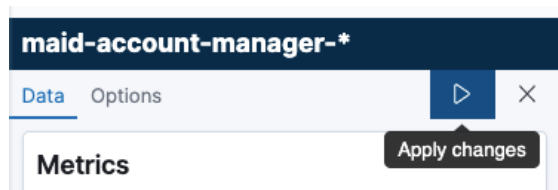
☐ Group other values in separate bucket

☐ Show missing values

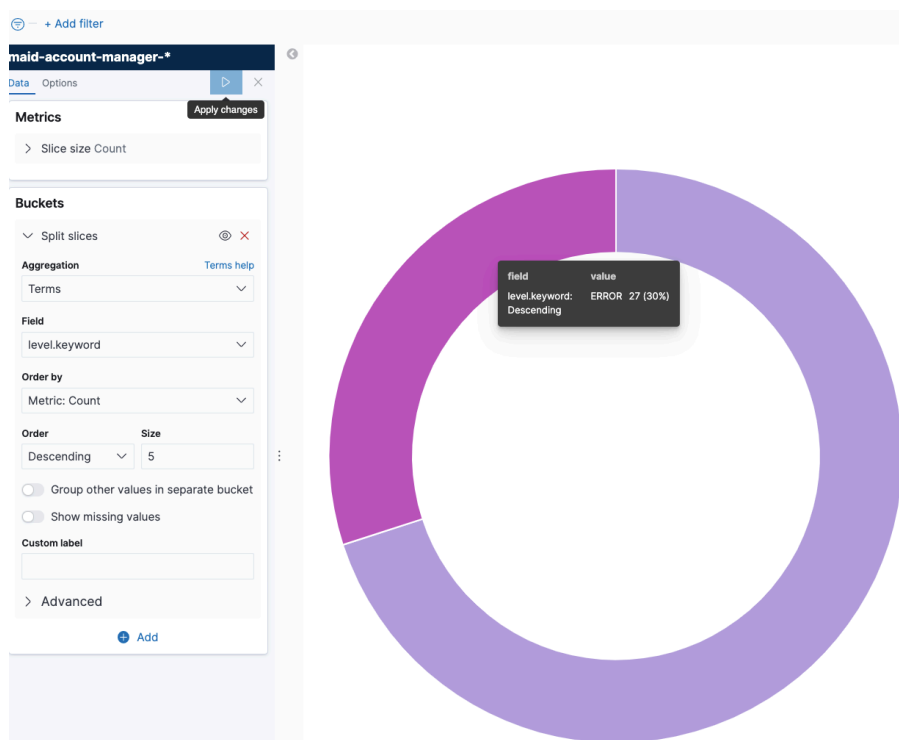
Custom label

> Advanced

Click “Apply changes”



We now have a live visualisation of the error/good responses.



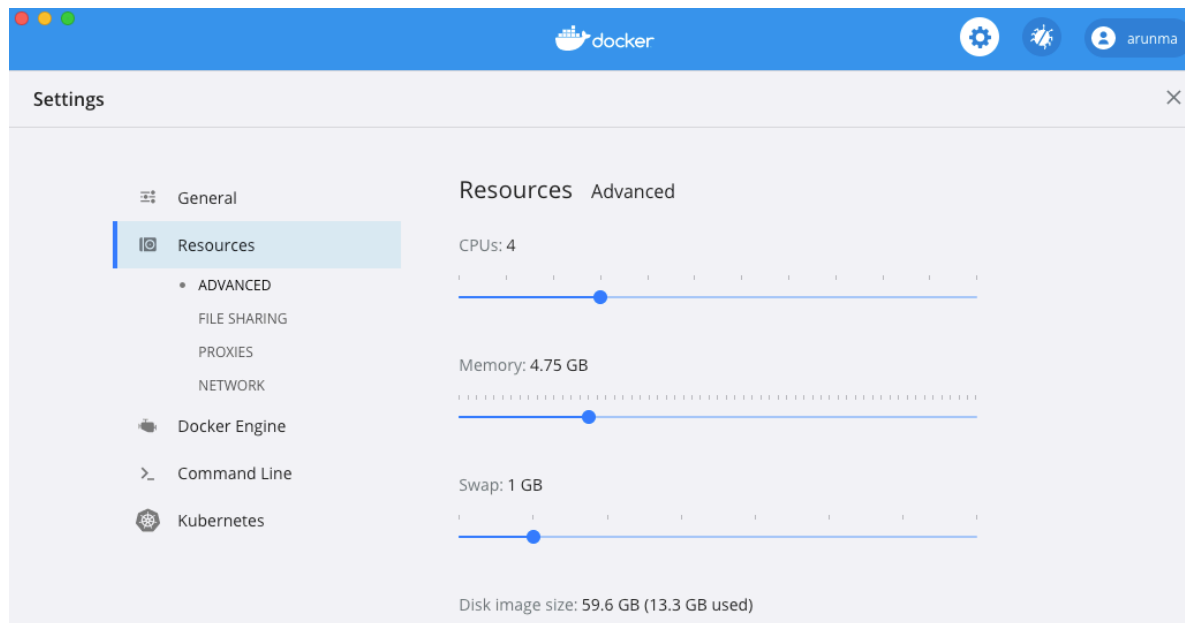
End of workshop

OS Specific instructions

Mac & Windows Pro Special Instructions:

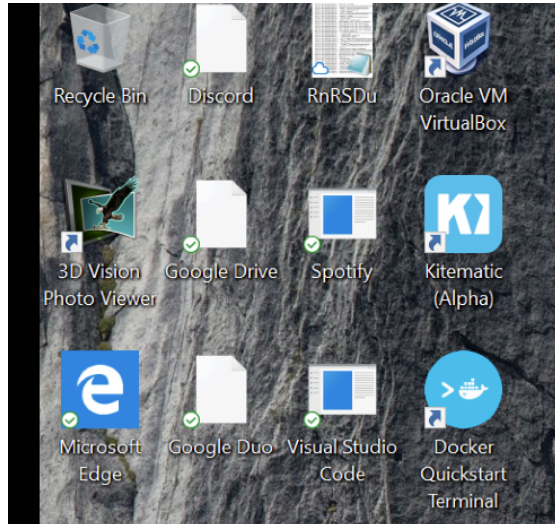
Extra Adjust your docker settings

Give at least 4 cores and 3 GB memory



Windows Home Special instructions (for windows >7)

1. Download Docker toolbox from <https://github.com/docker/toolbox/releases>
2. Install Linux VM
 - a. You must see the Docker Quickstart Terminal on your desktop



- b. Upon clicking, it would create a new VM inside your Oracle Virtual Box (which was installed as part of your toolbox installation process)

```
Docker Quickstart Terminal
(default) Starting the VM...
(default) Check network to re-create if needed...
(default) Windows might ask for the permission to configure a dhcp server. Sometimes
zed in the taskbar.
(default) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...

This machine has been allocated an IP address, but Docker Machine could not
reach it successfully.

SSH for the machine should still work, but connecting to exposed ports, such as
the Docker daemon port (usually <ip>:2376), may not work properly.

You may need to add the route manually, or use another related workaround.

This could be due to a VPN, proxy, or host file configuration issue.

You also might want to clear any VirtualBox host only interfaces you are not using.
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtua
ker Toolbox\docker-machine.exe env default
```

- c. You must see the Docker ASCII art on your screen

MINGW64:/c/Program Files/Docker Toolbox

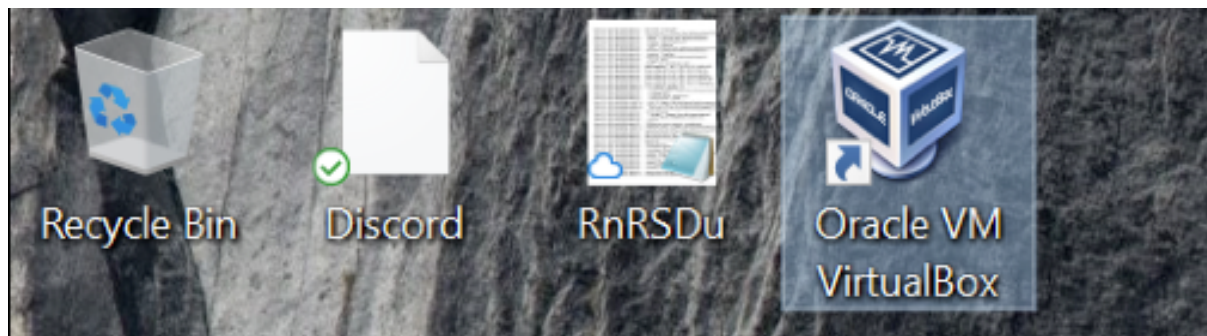
```
      ##          .  
    ## ## ##     ==  
   ## ## ## ## ===  
 / ..... \      ===  
{ ~~~~~ }       ===- ~~~~  
 |         o  
 \_____/
```

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

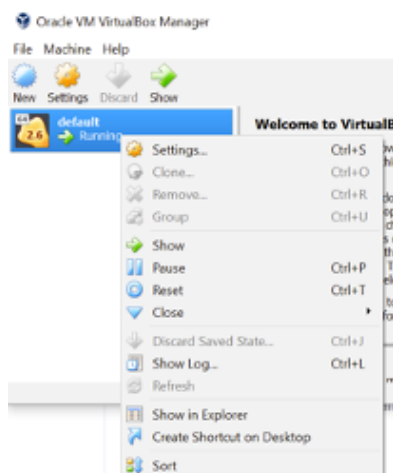
Start interactive shell

arun@DESKTOP-D5NIRR9 MINGW64 /c/Program Files/Docker Toolbox

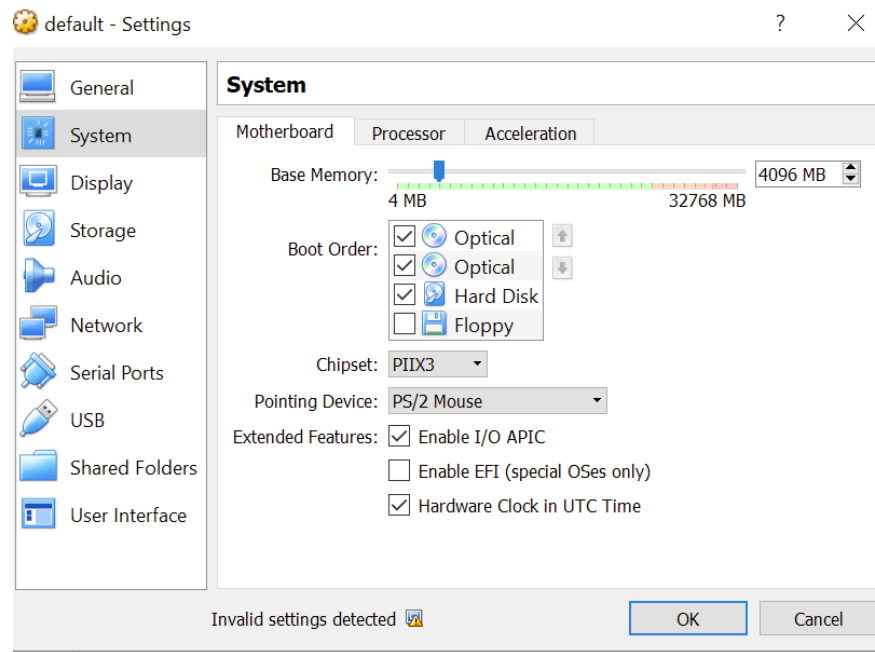
3. Adjust your memory and processor settings:
 - a. Open your virtual box



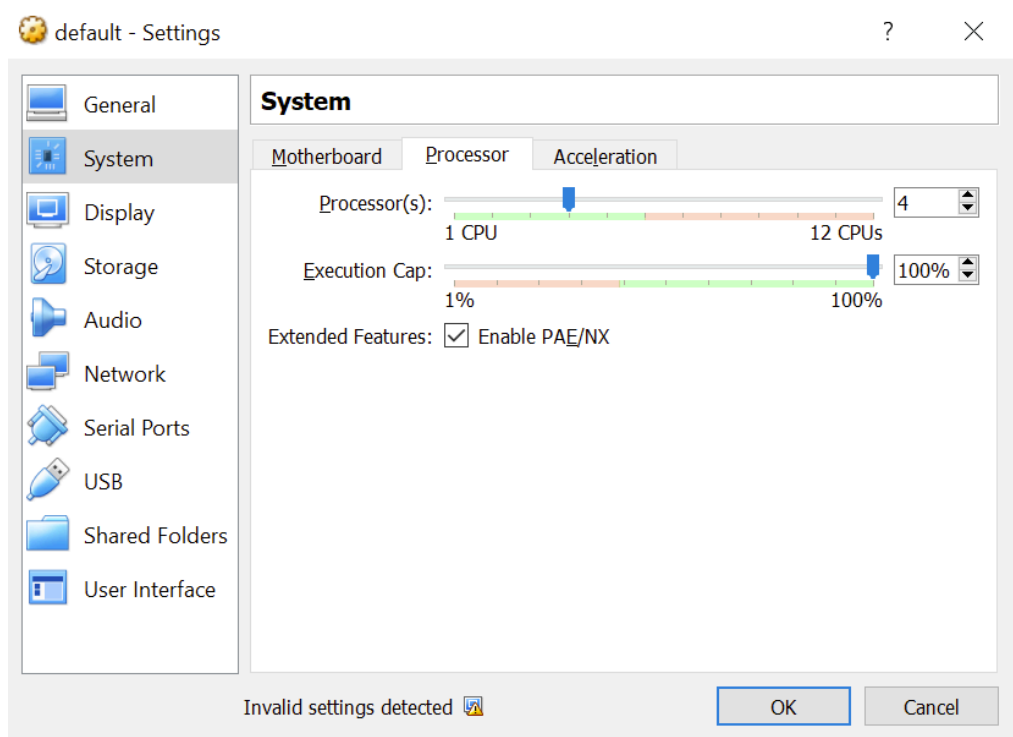
- b. Close and shut down your linux VM



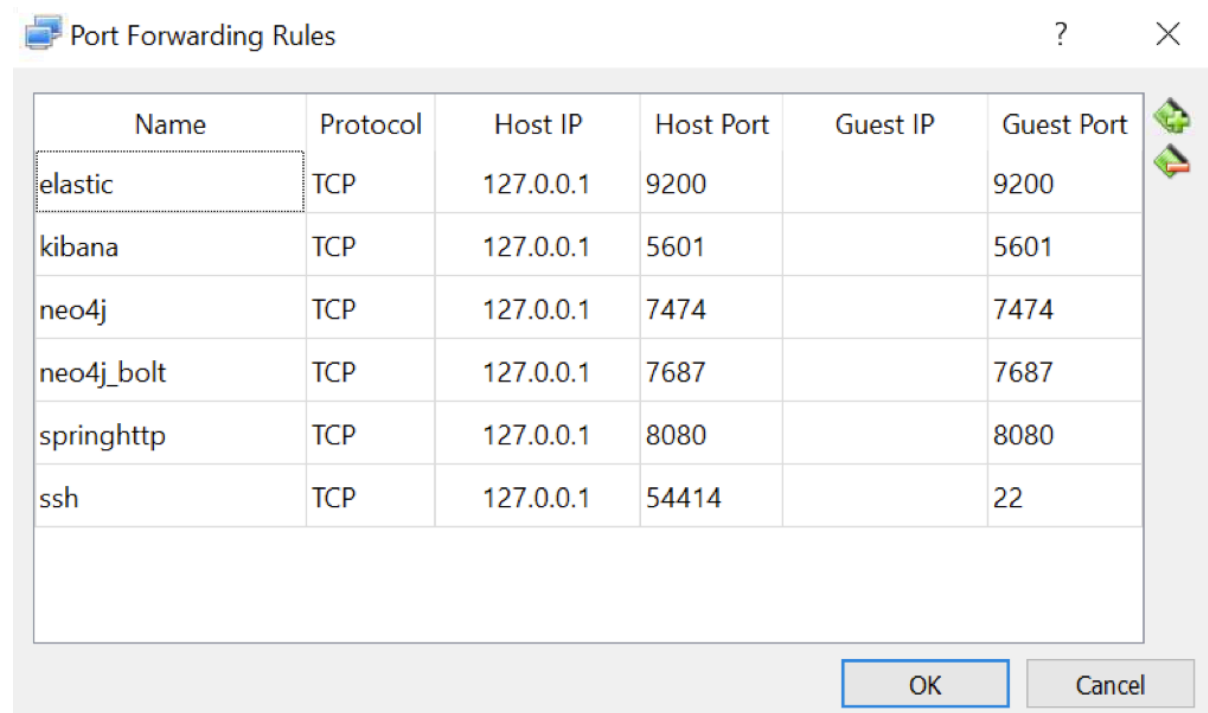
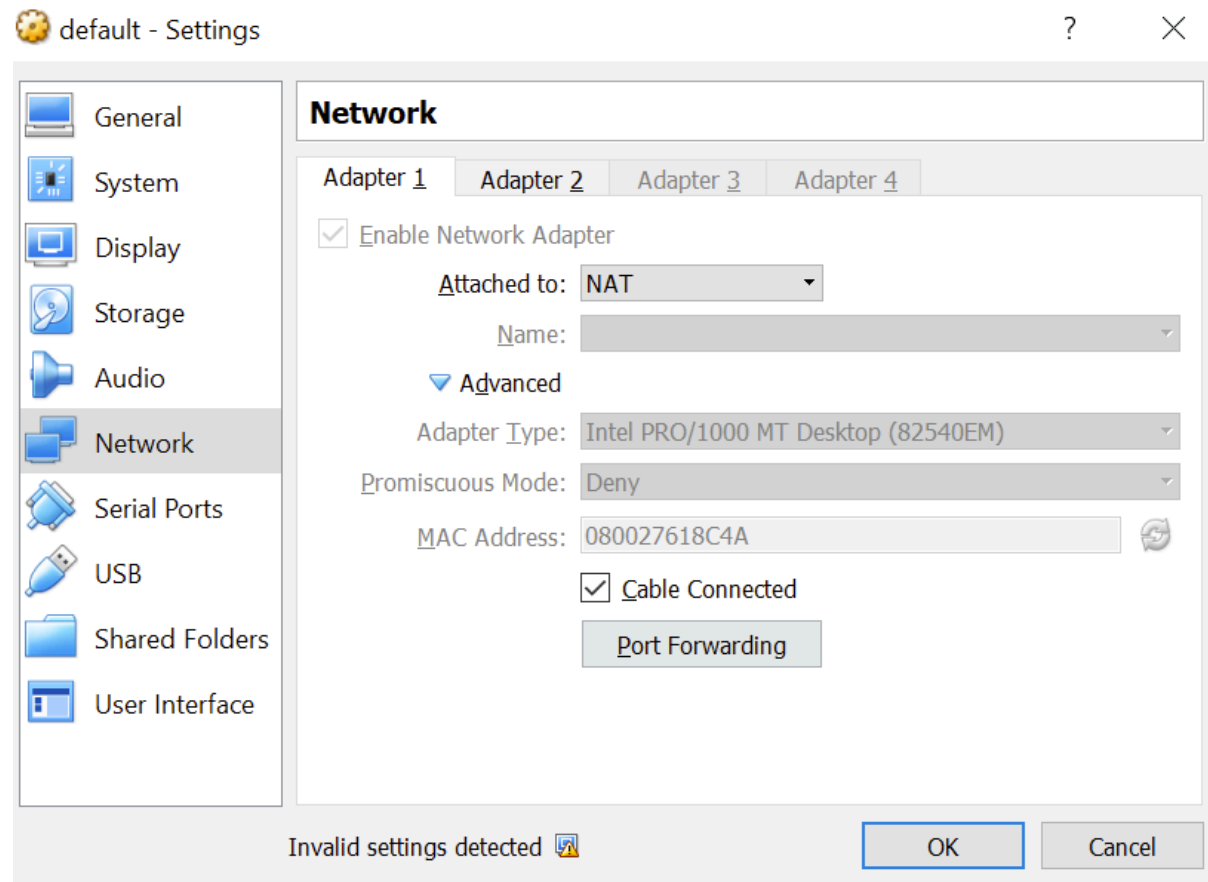
- c. Update memory to be at least 4 GB



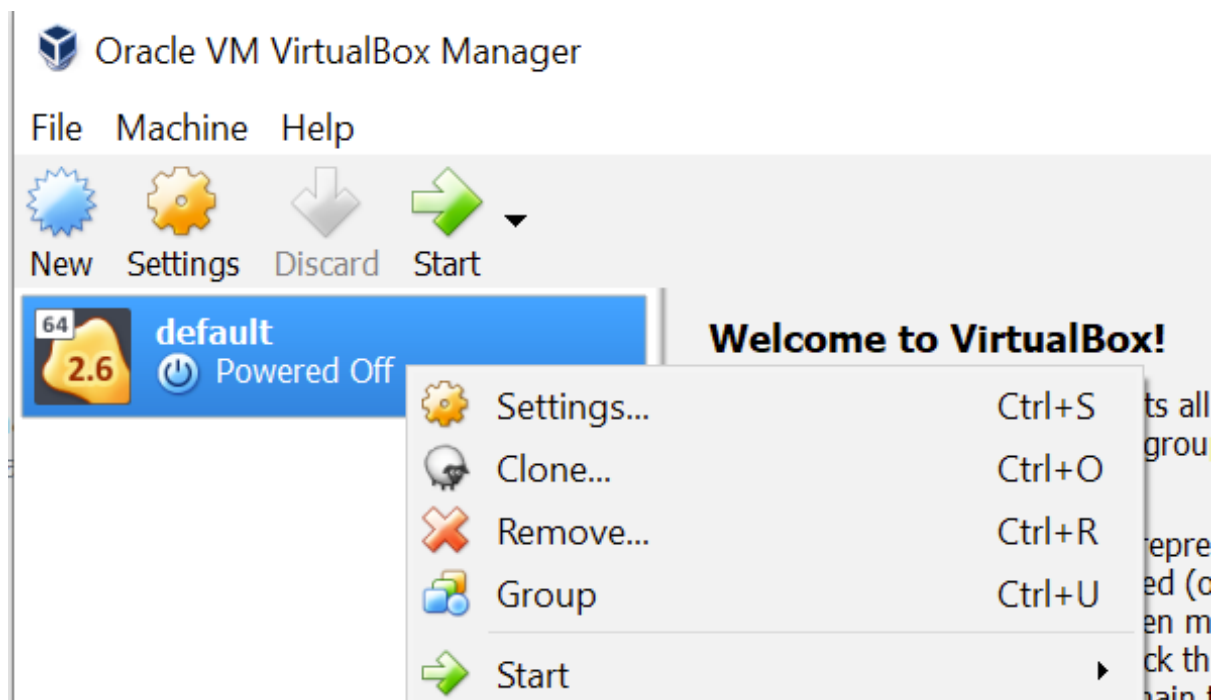
d. Update the number of processors to be 4



e. Set "Port Forwarding rules" in your Network



Restart your VM (Headless start)



Common Errors and Solutions

1. Problem

"Forbidden" while creating an index in Kibana

Solution

```
curl -XPUT -H "Content-Type: application/json" http://localhost:9200/_all/_settings -d '{"index.blocks.read_only_allow_delete": null}'
```

2. Problem

Error 137 or Kibana not coming up

Solution

Check if you have increased the Docker memory to at least 4 GB and your CPU to be 4.