OZAN ARIF YAVUZ

ozanarifyavuz@gmail.com

23rd May 2023

TEAMVIEWER DEVOPS TEST TASK

After having researched about the Vagrant and NAST tools and understood their basic functionalities, I have installed the pre-required software and tools such as Oracle VM VirtualBox on my local computer

1. The Vagrantfile, Virtual Machines (VMs), and the NATS Server

Vagrant tool uses hypervisors to create VMs and, it uses Vagrantfile to manage its lightweight, reproducible, and portable development environments. I created three VMs on my computer using Vagrantfile. You can see the content of the required Vagrant file in the “DevOps” GitHub repository or in the zipped file. I have chosen the Linux OS for the VMs because I have seen that it is widely used by the other developers according to information on the internet. As it is stated in the task, in order to define a shared network between three VMs, I have assigned private IP addresses to the VMs. The names of the VMs are **vm1**, **vm2**, and **vm3**, and their IP addresses are **192.168.33.10**, **192.168.33.11**, and **192.168.33.12**, respectively.

I designed vm1 as a NATS server, vm2 as a NATS publisher, and vm3 as a NATS subscriber. I tried to install a NATS server on vm1 using the Vagrantfile, but I got so many errors that I couldn’t understand in the creation and configuration phase of the vm1. That’s why, I manually installed a NATS server on the vm1. In addition, the required configuration settings such as installing pip3, the required Python libraries to execute Python scripts and Docker have been manually installed after spinning up the VMs. For the configuration of the VMs, I have prepared a shell script file, **configuration.sh**, and written all related commands having been executed one by one on every VMs before the simulation of the whole system.

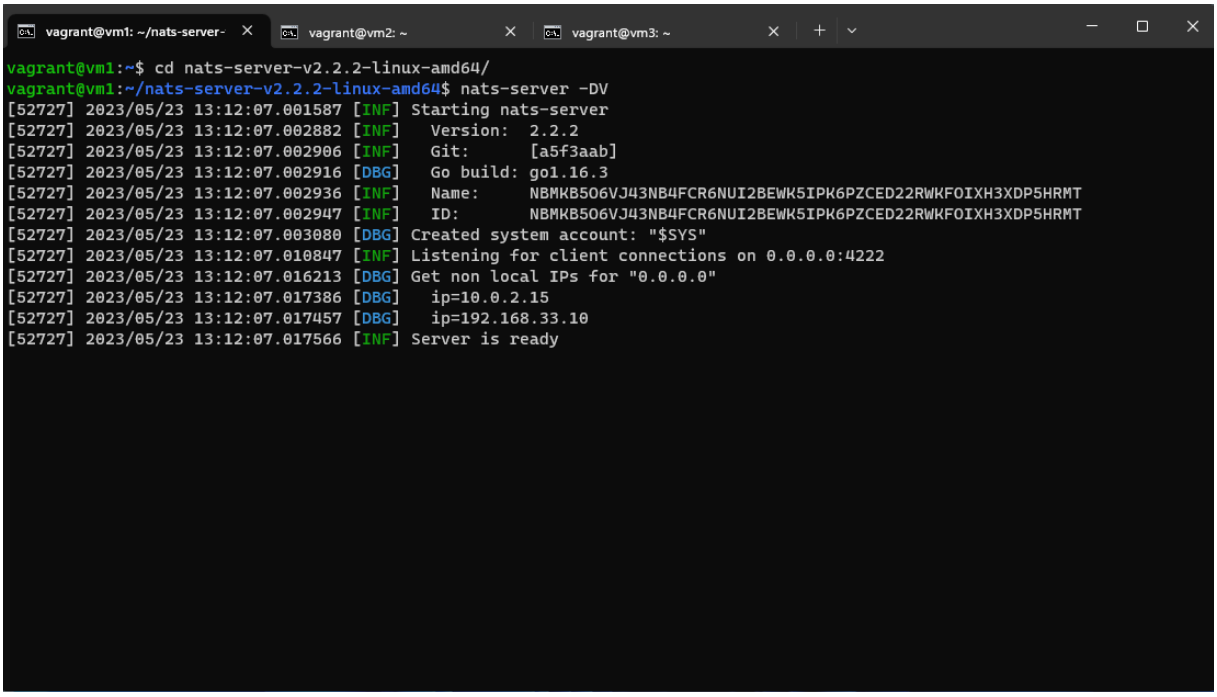
2. Publisher and Subscriber Scripts

As mentioned in the first section, the vm2 and vm3 have been designed as the NATS publisher and NATS subscriber, respectively. Moreover, the subscriber script has been installed in a container during the simulation phase. Since the NATS server has been installed on the vm1, both vm2 and vm3 have been connected to the vm1’s NATS server to listen to each other via the NATS server. You can see a try-catch-finally code block in the publisher.py script, and it has been designed to send consecutive messages to the NATS server. When a user enters “exit” as a message, the connection of the publisher to the NATS server is killed. It has been successfully managed to send messages to the NATS server running on the vm1, and the subscriber script running in a container could get the publisher’s messages and print them on the screen. To dockerize the subscriber script, a Dockerfile has been created and Docker has been installed on the vm3. To clearly see the outputs, which has been taken from the publisher via the NATS server, I have used “–u” flag in the last row of the Dockerfile. (**CMD ["python3", "-u", "subscriber.py"]**) You can see the Dockerfile in the shared document, as well.

3. The Simulation Phase and Results

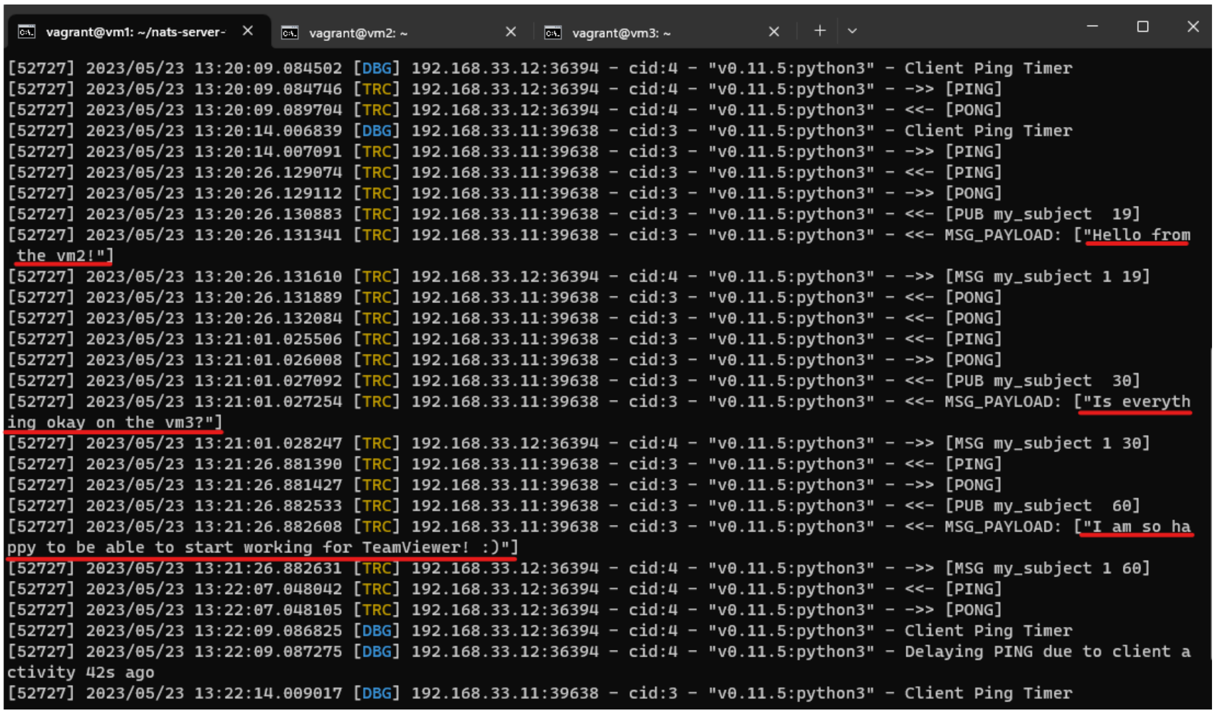
After setting up all required configurations, I have produced a directory and executed **“vagrant up”** command in it to produce three VMs. I have tried to connect to the VMs from my local by running **“vagrant ssh <virtualMachineName>”** command, but it hasn’t worked out. Therefore, I have found an alternative solution on the internet, and the command **“vagrant ssh vm1 -- -i .vagrant/machines/vm1/virtualbox/private_key”** has helped me connect to the VMs.

Firstly, I have created the NATS server on the vm1, and you can see the created NATS server example on the below screenshot:

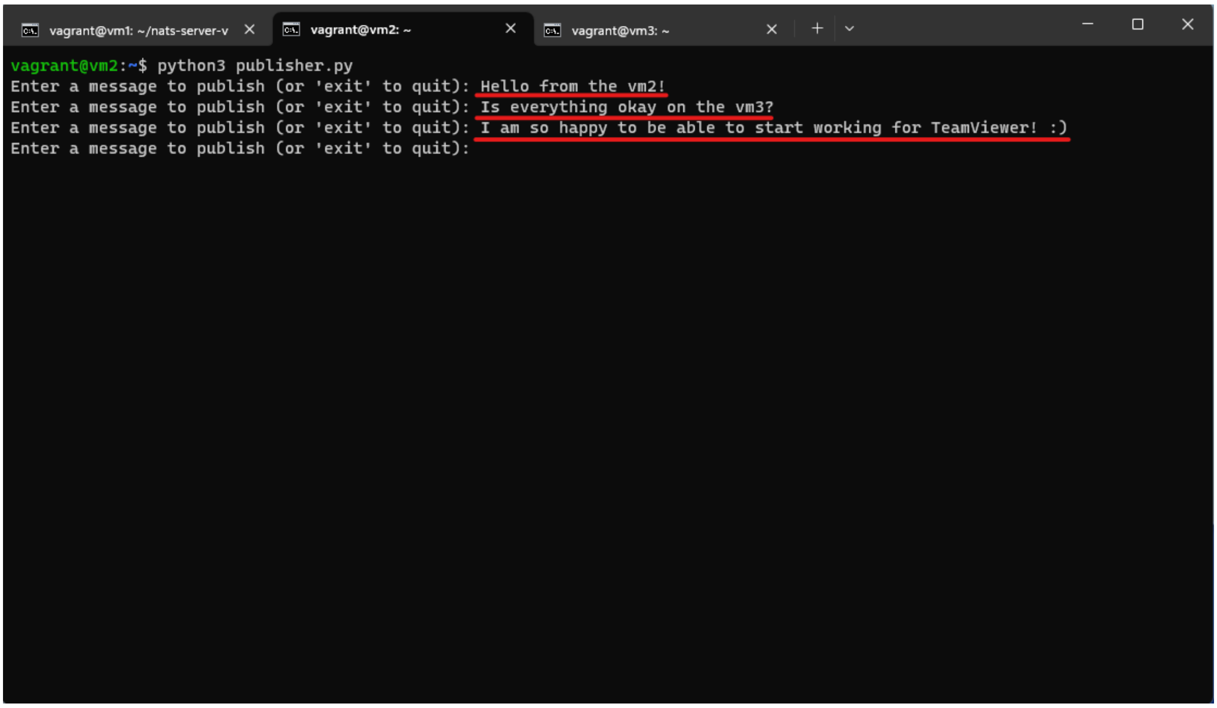


```
vagrant@vm1:~$ cd nats-server-v2.2.2-linux-amd64/
vagrant@vm1:~/nats-server-v2.2.2-linux-amd64$ nats-server -DV
[52727] 2023/05/23 13:12:07.001587 [INF] Starting nats-server
[52727] 2023/05/23 13:12:07.002882 [INF]   Version:  2.2.2
[52727] 2023/05/23 13:12:07.002906 [INF]   Git:      [a5f3aab]
[52727] 2023/05/23 13:12:07.002916 [DBG]   Go build: go1.16.3
[52727] 2023/05/23 13:12:07.002936 [INF]   Name:     NBMKB5O6VJ43NB4FCR6NUI2BEWK5IPK6PZCED22RWKFOIXH3XDP5HRMT
[52727] 2023/05/23 13:12:07.002947 [INF]   ID:       NBMKB5O6VJ43NB4FCR6NUI2BEWK5IPK6PZCED22RWKFOIXH3XDP5HRMT
[52727] 2023/05/23 13:12:07.003080 [DBG] Created system account: "$SYS"
[52727] 2023/05/23 13:12:07.010847 [INF] Listening for client connections on 0.0.0.0:4222
[52727] 2023/05/23 13:12:07.016213 [DBG] Get non local IPs for "0.0.0.0"
[52727] 2023/05/23 13:12:07.017386 [DBG]   ip=10.0.2.15
[52727] 2023/05/23 13:12:07.017457 [DBG]   ip=192.168.33.10
[52727] 2023/05/23 13:12:07.017566 [INF] Server is ready
```

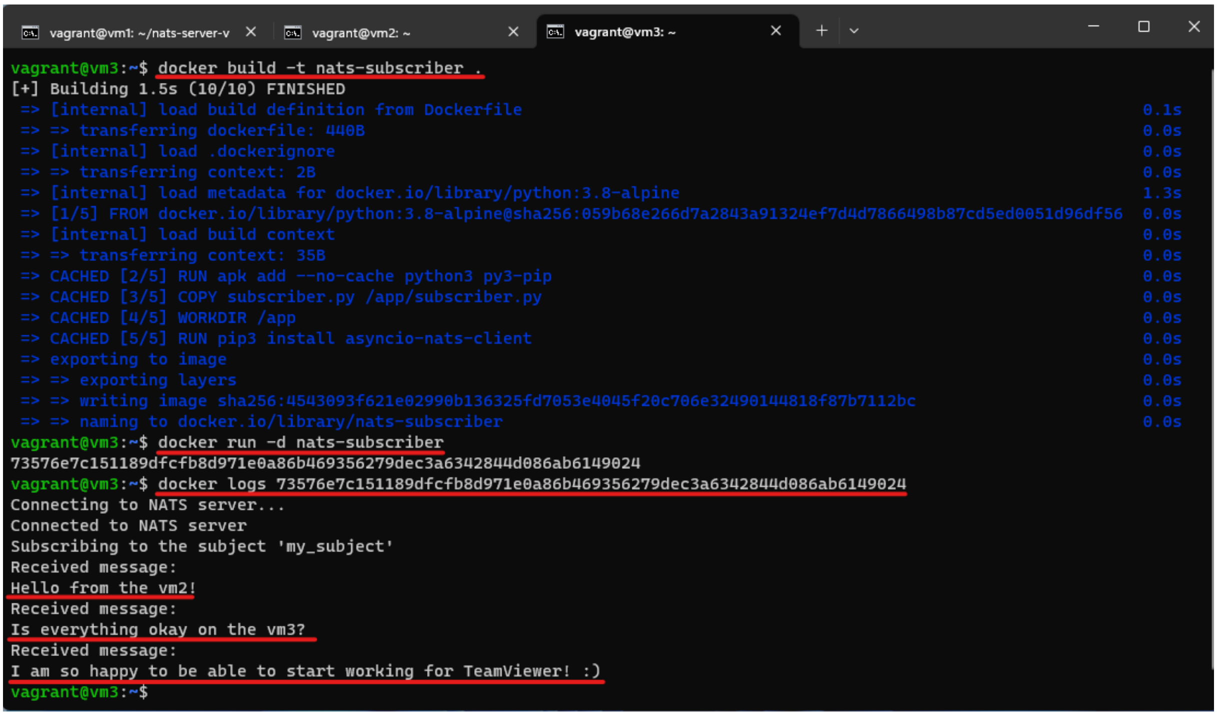
After starting the NATS server, I have executed the publisher.py script on the vm2, and I have created a container, nats-subscriber, on the wm3 to run the subscriber in a container. Then, I have consecutively entered the messages “Hello from the vm2!”, “Is everything okay on the vm3?”, and “I am so happy to be able to start working for TeamViewer! 😊”. On the following two screenshots, you can see the messages appearing in the NATS server on the vm1, the sent messages from the vm2, and the received messages, which have taken from the publisher, and the container information.



```
[52727] 2023/05/23 13:20:09.084502 [DBG] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - Client Ping Timer
[52727] 2023/05/23 13:20:09.084746 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - ->> [PING]
[52727] 2023/05/23 13:20:09.089704 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - <<- [PONG]
[52727] 2023/05/23 13:20:14.006839 [DBG] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - Client Ping Timer
[52727] 2023/05/23 13:20:14.007091 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - ->> [PING]
[52727] 2023/05/23 13:20:26.129074 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PING]
[52727] 2023/05/23 13:20:26.129112 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - ->> [PONG]
[52727] 2023/05/23 13:20:26.130883 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PUB my_subject  19]
[52727] 2023/05/23 13:20:26.131341 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- MSG_PAYLOAD: ["Hello from
 the vm2!"]
[52727] 2023/05/23 13:20:26.131610 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - ->> [MSG my_subject 1 19]
[52727] 2023/05/23 13:20:26.131889 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PONG]
[52727] 2023/05/23 13:20:26.132084 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PONG]
[52727] 2023/05/23 13:21:01.025506 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PING]
[52727] 2023/05/23 13:21:01.026008 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - ->> [PONG]
[52727] 2023/05/23 13:21:01.027092 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PUB my_subject  30]
[52727] 2023/05/23 13:21:01.027254 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- MSG_PAYLOAD: ["Is everyth
ing okay on the vm3?"]
[52727] 2023/05/23 13:21:01.028247 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - ->> [MSG my_subject 1 30]
[52727] 2023/05/23 13:21:26.881390 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PING]
[52727] 2023/05/23 13:21:26.881427 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - ->> [PONG]
[52727] 2023/05/23 13:21:26.882533 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- [PUB my_subject  60]
[52727] 2023/05/23 13:21:26.882608 [TRC] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - <<- MSG_PAYLOAD: ["I am so ha
ppy to be able to start working for TeamViewer! :)"]
[52727] 2023/05/23 13:21:26.882631 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - ->> [MSG my_subject 1 60]
[52727] 2023/05/23 13:22:07.048042 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - <<- [PING]
[52727] 2023/05/23 13:22:07.048105 [TRC] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - ->> [PONG]
[52727] 2023/05/23 13:22:09.086825 [DBG] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - Client Ping Timer
[52727] 2023/05/23 13:22:09.087275 [DBG] 192.168.33.12:36394 - cid:4 - "v0.11.5:python3" - Delaying PING due to client a
ctivity 42s ago
[52727] 2023/05/23 13:22:14.009017 [DBG] 192.168.33.11:39638 - cid:3 - "v0.11.5:python3" - Client Ping Timer
```



```
vagrant@vm2:~$ python3 publisher.py
Enter a message to publish (or 'exit' to quit): Hello from the vm2!
Enter a message to publish (or 'exit' to quit): Is everything okay on the vm3?
Enter a message to publish (or 'exit' to quit): I am so happy to be able to start working for TeamViewer! :)
Enter a message to publish (or 'exit' to quit):
```



```
vagrant@vm3:~$ docker build -t nats-subscriber .
[+] Building 1.5s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                   0.1s
 => => transferring dockerfile: 440B                                                                   0.0s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [internal] load metadata for docker.io/library/python:3.8-alpine                                   1.3s
 => [1/5] FROM docker.io/library/python:3.8-alpine@sha256:059b68e266d7a2843a91324ef7d4d7866498b87cd5ed0051d96df56  0.0s
 => [internal] load build context                                                                      0.0s
 => => transferring context: 35B                                                                       0.0s
 => CACHED [2/5] RUN apk add --no-cache python3 py3-pip                                                0.0s
 => CACHED [3/5] COPY subscriber.py /app/subscriber.py                                                 0.0s
 => CACHED [4/5] WORKDIR /app                                                                          0.0s
 => CACHED [5/5] RUN pip3 install asyncio-nats-client                                                  0.0s
 => exporting to image                                                                                 0.0s
 => => exporting layers                                                                                0.0s
 => => writing image sha256:4543093f621e02990b136325fd7053e4045f20c706e32490144818f87b7112bc           0.0s
 => => naming to docker.io/library/nats-subscriber                                                     0.0s
vagrant@vm3:~$ docker run -d nats-subscriber
73576e7c151189dfcfb8d971e0a86b469356279dec3a6342844d086ab6149024
vagrant@vm3:~$ docker logs 73576e7c151189dfcfb8d971e0a86b469356279dec3a6342844d086ab6149024
Connecting to NATS server...
Connected to NATS server
Subscribing to the subject 'my_subject'
Received message:
Hello from the vm2!
Received message:
Is everything okay on the vm3?
Received message:
I am so happy to be able to start working for TeamViewer! :)
vagrant@vm3:~$
```

To sum up, I have detailly investigated the given project’s tools, and successfully simulated the whole system. Firstly, I have spined up three VMs on my local machine. After that, I have installed all required tools on the VMs, and made a simulation to observe the system’s behaviour. Everything has worked out properly, and it was so fun for me to learn new tools and apply what I have learned for the demonstration.

Yours sincerely,

Ozan Arif Yavuz