V-BOX Setup for stress testing

Part 1: Download and Install VirtualBox

Step 1: Download VirtualBox

- Go to: https://www.virtualbox.org/
- Click **"Downloads"**
- Choose the installer based on your OS (Windows, macOS, or Linux)
- Download and run the installer.

Step 2: Install VirtualBox Extension Pack (Optional but recommended)

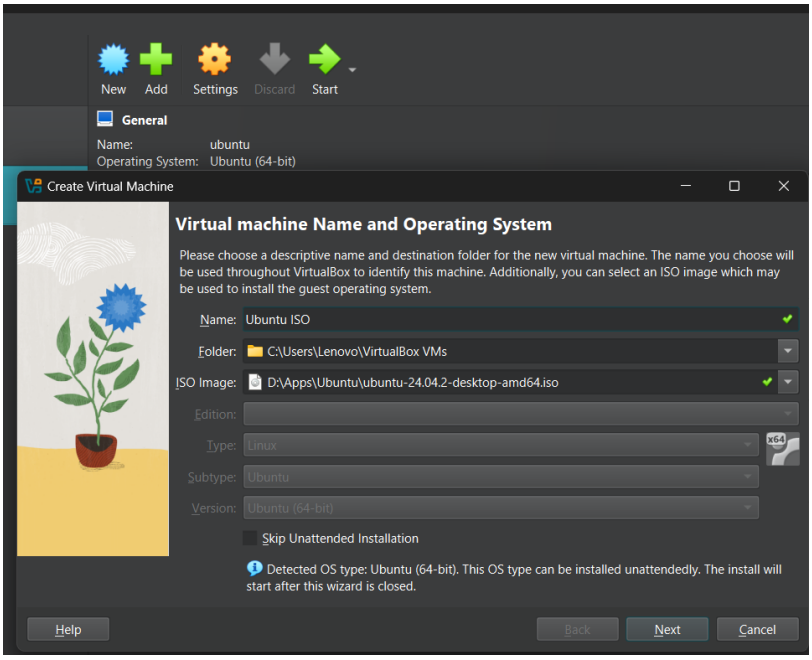
- Download from the same Downloads page (scroll down)
- Open VirtualBox → File → Preferences → Extensions → Add (+) → Select the .vbox-extpack file

Part 2: Add Ubuntu Instance to VirtualBox

Step 1: Download Ubuntu ISO

- Go to: https://ubuntu.com/download/desktop (use Ubuntu Server for headless setups)
- Full Guide : https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview
- Download the ISO (e.g., Ubuntu 22.04 LTS)

Step 2: Create VM in VirtualBox



1. Open VirtualBox
2. Click **"New"**
3. Name: Ubuntu-Kafka (or any name)
4. Give folder
5. Add the location of the ISO image you downloaded.
6. Type: Linux
7. Version: Ubuntu (64-bit)
8. Assign memory (at least 2 GB recommended)
9. Create a virtual hard disk (20 GB+ recommended)

Step 3: Start Ubuntu

1. Start the VM

Part 3: Install Kafka in Ubuntu VM

Step 1: Install Java

```
sudo apt update
sudo apt install -y openjdk-17-jdk
java -version
```

Step 2: Download and Extract Kafka

```
wget https://downloads.apache.org/kafka/3.7.0/kafka_2.13-3.7.0.tgz
tar -xvzf kafka_2.13-3.7.0.tgz
cd kafka_2.13-3.7.0
```

Step 3: Start Kafka

In **one terminal** (Zookeeper):

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

In **another terminal** (Kafka Broker):

```
bin/kafka-server-start.sh config/server.properties
```

Part 4: Install Prometheus

Step 1: Download Prometheus

```
wget https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz

tar -xvzf prometheus-2.52.0.linux-amd64.tar.gz

cd prometheus-2.52.0.linux-amd64
```

Step 2: Configure Prometheus

Edit prometheus.yml:

Add the following configuration :-

```
global:
  scrape_interval: 15s
scrape_configs:
  - job_name: 'kafka-jmx'
    static_configs:
      - targets: ['localhost:7071']


  - job_name: 'node'
    static_configs:
      - targets: ['localhost:9100']
```

Step 3: Run Prometheus

```
./prometheus --config.file=prometheus.yml
```

Access: http://<VM-IP>:9090

Part 5: Install Node Exporter

Step 1: Download and Run

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz

tar -xvzf node_exporter-1.8.1.linux-amd64.tar.gz

cd node_exporter-1.8.1.linux-amd64

./node_exporter
```

Access: http://<VM-IP>:9100/metrics

Part 6: Add JMX Exporter for Kafka

Step 1: Download JMX Exporter Java Agent

```
wget https://repo1.maven.org/maven2/io/prometheus/jmx/jmx_prometheus_javaagent/1.0.1/jmx_prometheus_javaagent-1.0.1.jar
```

Step 2: Create JMX Exporter Config File

Create kafka-jmx-config.yaml file

```
startDelaySeconds: 0

ssl: false

lowercaseOutputName: true

lowercaseOutputLabelNames: true

pattern: 'kafka.server<type=(.+), name=(.+)><>Count' name: kafka_server_$1_$2_total type: COUNTER

pattern: 'kafka.server<type=(.+), name=(.+)><>OneMinuteRate' name: kafka_server_$1_$2_1m_rate type: GAUGE

pattern: 'kafka.server<type=(.+), name=(.+)><>MeanRate' name: kafka_server_$1_$2_mean_rate type: GAUGE

pattern: 'kafka.server<type=(.+), name=(.+)><>Value' name: kafka_server_$1_$2 type: GAUGE labels: kafka_server: "$1"

pattern: 'kafka.log<type=Log, name=(.+), topic=(.+), partition=(.+)><>Value' name: kafka_log_$1 type: GAUGE labels: topic: "$2" partition: "$3"
```

Step 3: Modify Kafka Startup to Include JMX Exporter

Edit bin/kafka-server-start.sh and add:

```
export KAFKA_HEAP_OPTS="-Xmx256M -Xms128M"

export KAFKA_OPTS="$KAFKA_OPTS -javaagent:/opt/jmx-exporter/jmx_prometheus_javaagent-1.0.1.jar=7071:/opt/jmx-exporter/kafka-jmx-config.yaml"
```

Step 4: Restart Kafka

```
bin/kafka-server-stop.sh

bin/kafka-server-start.sh config/server.properties
```

Part 7: Install and Set Up Ngrok

Step 1: Download and Install Ngrok

```
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-stable-linux-amd64.zip

unzip ngrok-stable-linux-amd64.zip
```

```
sudo mv ngrok /usr/local/bin
```

Step 2: Authenticate Ngrok

Get your auth token from: https://dashboard.ngrok.com/get-started/setup

Then run:

```
ngrok config add-authtoken <YOUR_AUTH_TOKEN>
```

Expose Kafka Port (9092) using TCP

Kafka clients require a **TCP** tunnel (not HTTP). Ngrok's free tier supports TCP tunnels on a specific domain/port.

Step 3: Start TCP Tunnel for Kafka

```
ngrok tcp 9092
```

You'll see output like:

Forwarding tcp://0.tcp.ngrok.io:13579 -> localhost:9092

Use this host and port in your Kafka producer/consumer.

⚠️ **Important: You must set the Kafka advertised.listeners to this address for external clients to connect:**

advertised.listeners=PLAINTEXT://0.tcp.ngrok.io:13579

To apply:

1. Edit config/server.properties
2. Restart Kafka

Expose Prometheus Port (9090) using HTTP

Prometheus uses HTTP, so you can use a standard HTTP tunnel.

Step 4: Start HTTP Tunnel for Prometheus

```
ngrok http 9090
```

You'll see:

Forwarding http://abc123.ngrok.io -> http://localhost:9090

Access Prometheus metrics dashboard at:

http://abc123.ngrok.io

Summary of Ports:

Service	Port
Kafka Broker	9092
Zookeeper	2181
Prometheus	9090
Node Exporter	9100
Kafka JMX Exporter	7071