

Checking the Docker-compose:

```
root@Bryan: /home/bryan
root@Bryan:/home/bryan# docker-compose --help
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE             Specify an alternate compose file (default: docker-compose.yml)
  -p, --project-name NAME     Specify an alternate project name (default: directory name)
  --verbose                   Show more output
  --no-ansi                   Do not print ANSI control characters
  -v, --version               Print version and exit
  -H, --host HOST             Daemon socket to connect to

  --tls                       Use TLS; implied by --tlsverify
  --tlscacert CA_PATH        Trust certs signed only by this CA
  --tlscert CLIENT_CERT_PATH Path to TLS certificate file
  --tlskey TLS_KEY_PATH      Path to TLS key file
  --tlsverify                 Use TLS and verify the remote
  --skip-hostname-check       Don't check the daemon's hostname against the name specified
                              in the client certificate (for example if your docker host
                              is an IP address)
  --project-directory PATH    Specify an alternate working directory
                              (default: the path of the Compose file)

Commands:
  build                       Build or rebuild services
  bundle                      Generate a Docker bundle from the Compose file
  config                      Validate and view the Compose file
```

Create the compose file:

```
root@Bryan:/home/bryan/docker-homework# cat docker-compose.yml
version: '3.8'

networks:
  monitoring:
    driver: bridge

volumes:
  prometheus_data: {}

services:
  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    restart: unless-stopped
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    command:
      - '--path.procfs=/host/proc'
      - '--path.rootfs=/rootfs'
      - '--path.sysfs=/host/sys'
      - '--collector.filesystem.mount-points-exclude=^/(sys|proc|dev|host|etc)($$|/)'
    expose:
      - 9100
    networks:
      - monitoring

  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    restart: unless-stopped
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus_data:/prometheus
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--storage.tsdb.path=/prometheus'
      - '--web.console.libraries=/etc/prometheus/console_libraries'
      - '--web.console.templates=/etc/prometheus/consoles'
      - '--web.enable-lifecycle'
    expose:
      - 9090
    networks:
      - monitoring
```

The next thing to do is create the prometheus configuration file

```
root@Bryan:/home/bryan/docker-homework# cat prometheus.yml
global:
  scrape_interval:     15s

scrape_configs:
- job_name: "prometheus"
  scrape_interval: 5s
  static_configs:
    - targets: ["localhost:9090"]

- job_name: "node"
  static_configs:
    - targets: ["node-exporter:9100"]

remote_write:
- url: "<Your Prometheus remote_write endpoint>"
  basic_auth:
    username: "<Your Grafana Username>"
    password: "<Your Grafana API key>"
root@Bryan:/home/bryan/docker-homework#
```

In this fields you need top ut your grafana usenarname, and your grafana password.