**1. Atualizar pacotes e instalar Docker e Docker Compose**

bash

Copiar código

sudo apt update

sudo apt install -y docker.io docker-compose

sudo systemctl start docker

sudo systemctl enable docker

**2. Iniciar Docker Swarm**

bash

Copiar código

sudo docker swarm init

**3. Criar um projeto e configurar o docker-compose**

**Criar e entrar na pasta do projeto**

bash

Copiar código

mkdir nome\_do\_projeto

cd nome\_do\_projeto

**Criar e editar o arquivo docker-compose.yml**

bash

Copiar código

nano docker-compose.yml

Digite ou copie o conteúdo abaixo no arquivo docker-compose.yml:

yaml

Copiar código

version: '3.8'

services:

wordpress:

image: wordpress:latest

ports:

- '8080:80'

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress\_password

WORDPRESS\_DB\_NAME: wordpress

volumes:

- wordpress\_data:/var/www/html

deploy:

replicas: 1

restart\_policy:

condition: on-failure

db:

image: mysql:5.7

environment:

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress\_password

MYSQL\_ROOT\_PASSWORD: root\_password

volumes:

- db\_data:/var/lib/mysql

deploy:

replicas: 1

restart\_policy:

condition: on-failure

redis:

image: redis:latest

ports:

- '6379:6379'

volumes:

- redis\_data:/data

deploy:

replicas: 1

restart\_policy:

condition: on-failure

prometheus:

image: prom/prometheus:latest

ports:

- '9090:9090'

volumes:

- ./prometheus.yml:/etc/prometheus/prometheus.yml

deploy:

replicas: 1

restart\_policy:

condition: on-failure

grafana:

image: grafana/grafana:latest

ports:

- '3000:3000'

environment:

- GF\_SECURITY\_ADMIN\_PASSWORD=admin

volumes:

- grafana\_data:/var/lib/grafana

deploy:

replicas: 1

restart\_policy:

condition: on-failure

volumes:

wordpress\_data:

db\_data:

redis\_data:

grafana\_data:

Salve e saia do editor (CTRL + O, Enter, CTRL + X).

**4. Criar e editar o arquivo prometheus.yml**

bash

Copiar código

nano prometheus.yml

Digite ou copie o conteúdo abaixo no arquivo prometheus.yml:

yaml

Copiar código

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'wordpress'

static\_configs:

- targets: ['wordpress:80']

- job\_name: 'mysql'

static\_configs:

- targets: ['db:3306']

- job\_name: 'redis'

static\_configs:

- targets: ['redis:6379']

- job\_name: 'docker'

metrics\_path: /metric

static\_configs:

- targets: ['docker:9323']

Salve e saia do editor (CTRL + O, Enter, CTRL + X).

**5. Deploy do stack no Docker Swarm**

bash

Copiar código

sudo docker stack deploy -c docker-compose.yml nome\_da\_stack

**6. Verificar status dos serviços**

bash

Copiar código

docker service ls

**7. Puxar imagens, se necessário**

Se algum serviço estiver com réplica 0/1, utilize os comandos abaixo para puxar as imagens necessárias:

bash

Copiar código

docker pull prom/prometheus

docker pull grafana/grafana

docker pull redis

docker pull mysql:5.7

docker pull wordpress

**8. Acessar a instalação do WordPress**

Acesse http://localhost:8080 e siga os passos necessários para concluir a instalação do WordPress.

**9. Instalar e configurar o plugin Redis Object Cache no WordPress**

1. No WordPress, vá até Plugins -> Adicionar novo.
2. Procure por "Redis Object Cache" e clique em "Instalar".
3. Ative o plugin e clique em "Configurar".
4. Clique em "Ativar o cache de objeto".

Isso completa a configuração do seu ambiente Docker Swarm com WordPress, MySQL, Redis, Prometheus e Grafana.