

# Docker Compose And Nagios

## Docker Compose

### 1. Writing a Basic docker-compose.yml File

#### Step 1: Create a Folder

Go to your desktop or any folder you prefer.

Right-click and select New Folder.

Name the folder my\_docker\_project.

#### Step 2: Open a Text Editor

Open Visual Studio Code, Notepad, or any other text editor.

Create a new file.

#### Step 3: Write the YAML Configuration

Here's a simple example of a docker-compose.yml file for WordPress and MySQL:

```
version: '3.1'

services:
  db:
    image: mysql:5.7
    container_name: mysql_container
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: wordpress_db
      MYSQL_USER: wordpress_user
      MYSQL_PASSWORD: wordpress_pass
    volumes:
      - db_data:/var/lib/mysql
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    container_name: wordpress_container
    ports:
      - "8000:80"
```

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress\_user

WORDPRESS\_DB\_PASSWORD: wordpress\_pass

WORDPRESS\_DB\_NAME: wordpress\_db

volumes:

- ./wordpress\_data:/var/www/html

volumes:

db\_data:

```
docker-compose.yml ×
dock ~/Documents/languages/SE/Docker/docker-compose.yml
1 version: '3.1'
2 services:
3 db:
4   image: mysql:5.7
5   container_name: mysql_container
6   environment:
7     MYSQL_ROOT_PASSWORD: rootpassword
8     MYSQL_DATABASE: wordpress_db
9     MYSQL_USER: wordpress_user
10    MYSQL_PASSWORD: wordpress_pass
11   volumes:
12     - db_data:/var/lib/mysql
13 wordpress:
14   depends_on:
15     - db
16   image: wordpress:latest
17   container_name: wordpress_container
18   ports:
19     - "8000:80"
20   environment:
21     WORDPRESS_DB_HOST: db:3306
22     WORDPRESS_DB_USER: wordpress_user
23     WORDPRESS_DB_PASSWORD: wordpress_pass
24     WORDPRESS_DB_NAME: wordpress_db
25   volumes:
26     - ./wordpress_data:/var/www/html
27   volumes:
28     db_data:
29
```

## 2. Saving the File

Save the file as docker-compose.yml.

Place it in the my\_docker\_project folder.

SE		Today at 2:18 PM	--	Folder
22bd1a1272-week-6.pdf		22 Nov 2024 at 6:50 PM	6.7 MB	PDF Document
22BD1A1272-week7.pdf		22 Nov 2024 at 6:51PM	10.9 MB	PDF Document
22BD1A1272-WEEK8.pdf		4 Dec 2024 at 2:09 PM	3.9 MB	PDF Document
22BD1A1272-WEEK9		13 Dec 2024 at 10:45 AM	7.6 MB	Pages
22BD1A1272-WEEK9.pdf		13 Dec 2024 at 11:40 AM	5.1 MB	PDF Document
calculator.js		14 Dec 2024 at 3:12 PM	515 bytes	JavaSc...t script
class.mdj		18 Oct 2024 at 12:04 PM	44 KB	mdj
Docker		Today at 2:22 PM	--	Folder
docker-compose.yml		Today at 2:22 PM	584 bytes	YAML Document
Dockerfile		14 Dec 2024 at 3:13 PM	89 bytes	Document
E-ticketing.mdj		18 Oct 2024 at 11:49 AM	367 KB	mdj
Gittest		22 Nov 2024 at 6:52 PM	--	Folder
SE ss		22 Nov 2024 at 6:44 PM	--	Folder

### 3. Running the Setup

#### Step 1: Open Command Line

Open PowerShell or Command Prompt.

Navigate to the my\_docker\_project folder:

```
cd
```

```
jcs@Jeevas-MacBook-Air ~ % cd documents
jcs@Jeevas-MacBook-Air documents % cd languages
jcs@Jeevas-MacBook-Air languages % cd se
jcs@Jeevas-MacBook-Air se % cd docker
jcs@Jeevas-MacBook-Air docker %
```

path\_to\_my\_docker\_project

#### Step 2: Start the Containers

Run:

```
docker-compose up -d
```

```
jcs@Jeevas-MacBook-Air docker % docker-compose up -d
[+] Running 35/24
  ✓ db Pulled
  ✓ wordpress Pulled
                                         48.7s
                                         28.6s

[+] Running 4/4
  ✓ Network docker_default          Created          0.0s
  ✓ Volume "docker_db_data"        Created          0.0s
  ✓ Container mysql_container      Started         0.3s
  ✓ Container wordpress_container  Started         2.4s
```

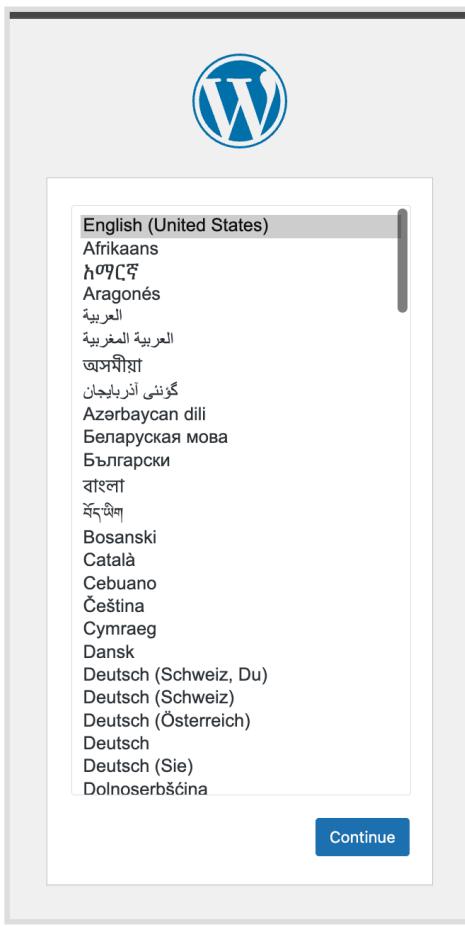
This command reads the docker-compose.yml file and creates both the WordPress and MySQL containers.

-d runs the containers in the background.

#### 4. Accessing the Application

Open your web browser.

Go to <http://localhost:8000>



Follow the WordPress setup wizard to complete the installation:

Site Name.

Admin Username and Password.

Email Address.

The image contains two screenshots of a Chrome browser window. The top screenshot shows the 'WordPress > Installation' page at `localhost:8000/wp-admin/install.php`. It displays the WordPress logo and a 'Welcome' message: 'Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' Below this is a form for 'Information needed' with fields for 'Site title', 'Username', and 'Password'. The password field contains '45hGDbgvTBYSeZ1yF!' and is marked as 'Strong'. The bottom screenshot shows the WordPress dashboard at `localhost:8000/wp-admin/`. The dashboard features a large 'Welcome to WordPress!' header and three main sections: 'Author rich content with blocks and patterns', 'Customize your entire site with block themes', and 'Switch up your site's look & feel with Styles'. On the left, a sidebar lists navigation links like Home, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings.

## 8. Managing Containers

Stop the Containers

To stop the containers without removing them:

**docker-compose stop**

Start Again

To restart the containers:

**docker-compose start**

## Remove Containers

To stop and remove everything:

**docker-compose down**

```
jcs@Jeevas-MacBook-Air docker % docker-compose stop
[+] Stopping 2/2
✓ Container wordpress_container Stopped

✓ Container mysql_container Stopped

jcs@Jeevas-MacBook-Air docker % docker-compose start
[+] Running 2/2
✓ Container mysql_container Started

✓ Container wordpress_container Started

jcs@Jeevas-MacBook-Air docker % docker-compose down
[+] Running 3/3
✓ Container wordpress_container Removed

✓ Container mysql_container Removed

✓ Network docker_default Removed
```

1.Run Docker Desktop 2.Open PowerShell as administrator and type the following command

**docker**

**version** [to check that both client and dockere daemon are running]

```
jcs@Jeevas-MacBook-Air docker % docker version
Client:
  Version:          27.3.1
  API version:     1.47
  Go version:      go1.22.7
  Git commit:      ce12230
  Built:           Fri Sep 20 11:38:18 2024
  OS/Arch:         darwin/arm64
  Context:          desktop-linux

Server: Docker Desktop 4.36.0 (175267)
Engine:
  Version:          27.3.1
  API version:     1.47 (minimum version 1.24)
  Go version:      go1.22.7
  Git commit:      41ca978
  Built:           Fri Sep 20 11:41:19 2024
  OS/Arch:         linux/arm64
  Experimental:    false
  containerd:
    Version:        1.7.21
    GitCommit:      472731909fa34bd7bc9c087e4c27943f9835f111
  runc:
    Version:        1.1.13
    GitCommit:      v1.1.13-0-g58aa920
  docker-init:
    Version:        0.19.0
    GitCommit:      de40ad0
jcs@Jeevas-MacBook-Air docker %
```

Pull the Nagios image from the Docker Hub with the following command

```
docker pull jasonrivers/nagios:latest
```

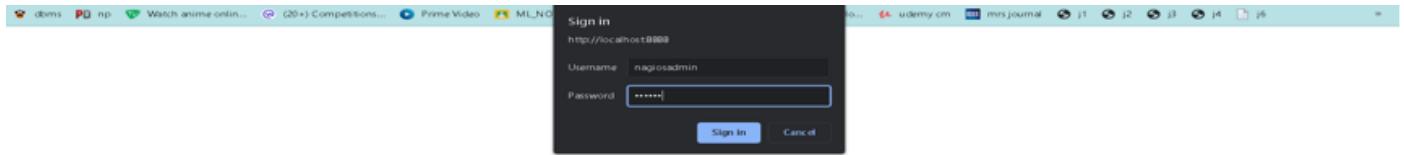
```
jcs@Jeevas-MacBook-Air docker % docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
ff65ddf9395b: Pull complete
785b9873bdf4: Pull complete
0ef9446ba5cc: Pull complete
53aff88babcb4: Pull complete
d72f92e29533: Pull complete
786ed7d4ce0a: Pull complete
d3245570f968: Pull complete
e58e184b986a: Pull complete
eab77e6dde3e: Pull complete
0bd0f5795eab: Pull complete
71bfb306f8cb: Pull complete
738fc7520889: Pull complete
fe8a6b2cf4e3: Pull complete
e6f8fab512d1: Pull complete
15f36d0b0439: Pull complete
a2fc4187e3b4: Pull complete
3d5785144815: Pull complete
566cdc02555d: Pull complete
c790be87d617: Pull complete
4f4fb700ef54: Pull complete
b69c76bd2b6b: Pull complete
d5aa2a3a6539: Pull complete
8fb30af17153: Pull complete
9ffe54c5c139: Pull complete
279b28aefaf10: Pull complete
a900dfcceb38: Pull complete
9a90645e352c: Pull complete
8e911c59da28: Pull complete
c219d58cc3f9: Pull complete
b0e280e9aa8c: Pull complete
8c389e58e867: Pull complete
Digest: sha256:2a7c2b20d118baf92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
jcs@Jeevas-MacBook-Air docker %
```

```
docker run --name nagios4 -p 0.0.0.0:8888:80 jasonrivers/nagios:latest
```

```
Successfully launched command file worker with pid 56
nagios: Successfully launched command file worker with pid 56
postfix/master[21]: daemon started -- version 3.8.6, configuration /etc/postfix
nagios: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;1;HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.018 second response time
nagios: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;2;HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.017 second response time
nagios: SERVICE ALERT: localhost;HTTP;WARNING;SOFT;3;HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.013 second response time
nagios: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.019 second response time
```

Nagios is successfully launched

We can access Nagios at the browser at port **8888** [localhost:8888] The default credentials for the web interface is nagiosadmin / nagios



Nagios dashboard is the interface to monitor the services

A new version of Nagios Core is available! Visit [nagios.org](http://nagios.org) to download Nagios 4.5.9.

## Meet Nagios Core Services Platform

The next generation of Open Source powered monitoring with advanced dashboards, monitoring wizards, and much more!

[Learn More](#) [Newsletter Sign-Up](#)

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

[https://www.nagios.org/launch?utm\\_campaign=csp&utm\\_source=nagioscore&utm\\_medium=splash\\_thumbnail&utm\\_content=4.5.7](https://www.nagios.org/launch?utm_campaign=csp&utm_source=nagioscore&utm_medium=splash_thumbnail&utm_content=4.5.7)

Host services [ left side pane can be used for selection.

Click on localhost to check the Host State information

Click on localhost to check the Host State information

Click on services to check the various services running **CurrentLoad**, **Current User**, **HTTP**, **PING** etc

**STEPS TO INSTALL THE MINIKUBS :-**

**Current Network Status**

Last Updated: Tue Dec 24 09:41:09 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.7 - www.nagios.org  
Logged in as nagiosadmin

**Host Status Totals**

Up	Down	Unreachable	Pending
1	0	0	0

**Service Status Totals**

Ok	Warning	Unknown	Critical	Pending
6	1	0	0	0

**Service Status Details For Host 'localhost'**

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	12-24-2024 09:39:17	0d 0h 1m 52s	1/4	OK - load average: 1.58, 1.97, 1.79
localhost	Current Users	OK	12-24-2024 09:40:00	0d 0h 1m 9s	1/4	USERS OK - 0 users currently logged in
localhost	HTTP	WARNING	12-24-2024 09:38:43	0d 0h 7m 26s	4/4	HTTP WARNING: HTTP/1 401 Unauthorized - 695 bytes in 0.006 second response time
localhost	PING	OK	12-24-2024 09:36:26	0d 0h 9m 43s	1/4	PING OK - Packet loss = 0%, RTA = 0.14 ms
localhost	Root Partition	OK	12-24-2024 09:37:09	0d 0h 9m 0s	1/4	DISK OK - free space: / 52730 MB (92.99% inode=98%)
localhost	Swap Usage	OK	12-24-2024 09:37:52	0d 0h 8m 17s	1/4	SWAP OK - 100% free (1023 MB out of 1023 MB)
localhost	Total Processes	OK	12-24-2024 09:38:35	0d 0h 7m 34s	1/4	PROCS OK: 29 processes with STATE = RSZDT

Results 1 - 7 of 7 Matching Services

## Steps to install the minikube

Search for minikube install in browser

Google

minikube install

All Videos Images Books News More Tools SafeSearch

**Minikube**  
<https://minikube.sigs.k8s.io> › docs › start

**minikube start**

23 Oct 2023 — **minikube** is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes. All you need is Docker (or similarly compatible).

Already we have installed the docker than proceed for next steps

Copy the commands in powershell

Once done with commands start the minikube

Mini start

```
jcs@Jeevas-MacBook-Air ~ % minikube start
minikube v1.34.0 on Darwin 15.2 (arm64)
Using the docker driver
Starting "minikube" ...
Pulling base image v0
Restarting existing docker container
Preparing Kubernetes
Unable to load cached kubeconfig
> kubeadm.sha256: 64
> kubectl.sha256: 64
> kubeadm: 54.25 MiB
> kubectl: 52.44 MiB
■ Generating certificates
■ Booting up control plane
■ Configuring RBAC rules...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Installer type

[Binary download](#)

[Homebrew](#)

To install the latest minikube **stable** release on **ARM64 macOS** using **binary download**:

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-arm64
sudo install minikube-darwin-arm64 /usr/local/bin/minikube
```

## Minikube status

## Kubectl get nodes

NAME	STATUS	ROLES	AGE	VERSION
minikube	Ready	control-plane	3m48s	v1.31.0

## Creating the nginx deployment

1. To create the deployment for nginx server

Kubectl create deployment <deployment name> --image=nginx

```
jcs@Jeevas-MacBook-Air ~ % Kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
```

2. To check whether deployment is created or not

Kubectl get deployment

```
jcs@Jeevas-MacBook-Air ~ % Kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	0/1	1	0	28s

3. To check the pod status

Kubectl get pod

```
jcs@Jeevas-MacBook-Air ~ % Kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-5bbdf5d75f-9qn95	0/1	ImagePullBackOff	0	103s

4. To check service

Kubectl get service

```
jcs@Jeevas-MacBook-Air ~ % Kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	7m19s

5. To get pod information in more detail

Kubectl get pod -o wide

```
jcs@Jeevas-MacBook-Air ~ % kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS	GATES
nginx-deployment-5bbdf5d75f-9qn95	0/1	ImagePullBackOff	0	3m57s	10.244.0.4	minikube	<none>	<none>	

6. To access the nginx page

Kubectl expose deployment nginx-deployment - --type=NodePort - --port=80

```
jcs@Jeevas-MacBook-Air ~ % Kubectl expose deployment nginx-deployment nginx-deployment --type=NodePort --port=80  
service/nginx-deployment exposed
```

7. To get URL to access the nginx home page

minikube service nginx-deployment --url

8. Copy the above http path and paste it in browser to access it

