

Docker installation on Win 11 OS, with WSL2 backend.

The screenshot displays the Docker Desktop application window. The top bar shows the Docker logo, a search bar, and system icons. The left sidebar contains navigation options: Ask Gordon (BETA), Containers (selected), Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' and includes a sub-header 'View all your running containers and applications. [Learn more](#)'. Below this, there's a graphic of three containers and the text 'Your running containers show up here' and 'A container is an isolated environment for your code'. Two cards are visible: 'What is a container?' (5 mins) and 'How do I run a container?' (6 mins). A link 'View more in the Learning center' is at the bottom. The bottom status bar shows 'Resource Saver mode', RAM usage (0.83 GB), CPU usage (0.00%), and disk usage (1.08 GB used of 1006.85 GB limit). Below the Docker Desktop window, a Windows Subsystem for Linux (WSL2) terminal is open, showing the command prompt for 'nodejs-shopping-cart@1.0.0'. The terminal output shows the directory structure and the command to start the application. A web browser window is also open, displaying 'NodeJS Shopping Cart' at localhost:3000, with a welcome message and a search bar. The bottom part of the image shows the 'Networking Integration' section of the Docker Desktop documentation, explaining how to access Linux networking apps from Windows and the mirrored mode networking capabilities.

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Your running containers show up here

A container is an isolated environment for your code

What is a container?
5 mins

How do I run a container?
6 mins

[View more in the Learning center](#)

Resource Saver mode | RAM 0.83 GB CPU 0.00% Disk: 1.08 GB used (limit 1006.85 GB) | [Update](#)

Welcome to Windows Subsystem for Linux

General

Working Across File Systems

GUI Apps

GPU Acceleration

Networking Integration

Distro Management

Docker Desktop Integration

VS Code Integration

Settings

```
nodejs-shopping-cart git:(master) x ls
LICENSE  README.md  app.js  bin  data  models  node_modules  package-loc
k.json  package.json  public  routes  views
nodejs-shopping-cart git:(master) x npm start

> nodejs-shopping-cart@1.0.0 start
> node ./bin/www

Your website has started running, connect here: http://localhost:3000
GET / 304 23.625 ms - -
GET /stylesheets/style.css 304 1.557 ms - -
```

NodeJS Shopping Cart

Welcome to the NodeJS Shopping Cart

Networking Integration

You can easily access networking apps across Windows and Linux operating systems.

Accessing Linux Networking Apps from Windows

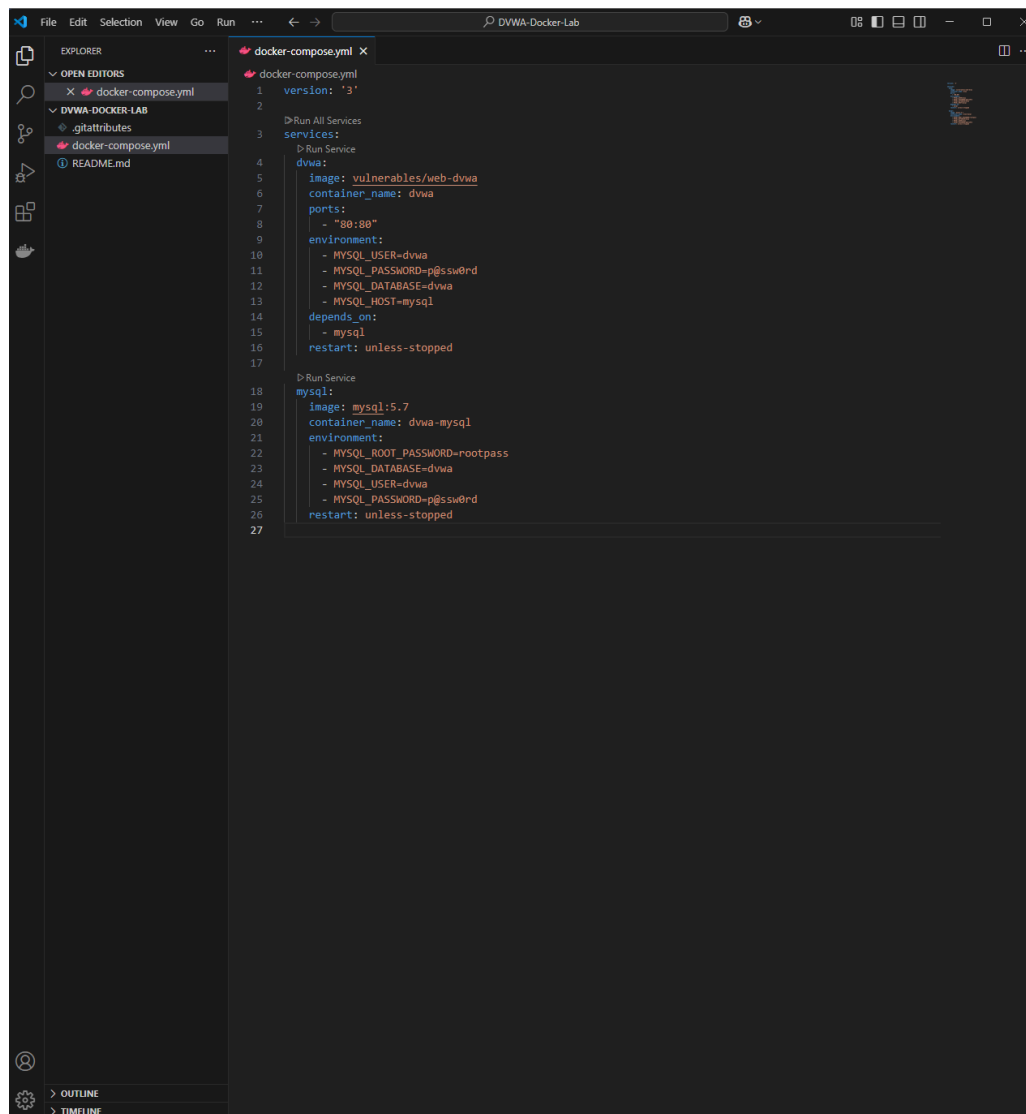
If you are building a networking app (for example an app running on a NodeJS or SQL server) in your Linux distribution, you can access it from a Windows app (like your Edge or Chrome internet browser) using localhost (just like you normally would). This means if you started a Linux server that is listening to port 3000, you can go to <http://localhost:3000> in Edge on Windows to access it.

Mirrored Mode Networking

WSL also includes a new networking mode, called mirrored mode which adds advanced capabilities like IPv6 support and the ability to access your networking applications in your local are

[Learn More About Networking Applications](#)

[Learn More About Mirrored Mode Networking](#)



The image shows a Visual Studio Code editor window titled "DVWA-Docker-Lab". The Explorer sidebar on the left shows the project structure with files: `docker-compose.yml`, `.gitattributes`, `docker-compose.yml`, and `README.md`. The main editor area displays the content of `docker-compose.yml` with the following YAML configuration:

```
1 version: '3'
2
3 services:
4   dvwa:
5     image: vulnerables/web-dvwa
6     container_name: dvwa
7     ports:
8       - "80:80"
9     environment:
10      - MYSQL_USER=dvwa
11      - MYSQL_PASSWORD=p@ssw0rd
12      - MYSQL_DATABASE=dvwa
13      - MYSQL_HOST=mysql
14     depends_on:
15       - mysql
16     restart: unless-stopped
17
18   mysql:
19     image: mysql:5.7
20     container_name: dvwa-mysql
21     environment:
22      - MYSQL_ROOT_PASSWORD=rootpass
23      - MYSQL_DATABASE=dvwa
24      - MYSQL_USER=dvwa
25      - MYSQL_PASSWORD=p@ssw0rd
26     restart: unless-stopped
27
```

Configure initial yaml file.

```

time="2025-03-31T16:25:50-04:00" level=warning msg="C:\\Users\\Juan\\Documents\\GithubProjects\\DVWA-Docker-Lab\\docker-
compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 21/21
  ✓ dvwa Pulled 10.0s
    ✓ b3d64a33242d Pull complete 0.1s
    ✓ e9968e5981d2 Pull complete 7.8s
    ✓ 6cff5f35147f Pull complete 7.9s
    ✓ 0c57df616dbf Pull complete 7.7s
    ✓ 098cffd43466 Pull complete 8.2s
    ✓ 2cd72dba8257 Pull complete 0.4s
    ✓ eb05d18be401 Pull complete 0.5s
    ✓ 3e17c6eae66c Pull complete 4.3s
  ✓ mysql Pulled 9.5s
    ✓ df9a4d85569b Pull complete 0.2s
    ✓ 43d05e938198 Pull complete 7.7s
    ✓ ae71319cb779 Pull complete 6.0s
    ✓ 064b2d298fba Pull complete 0.2s
    ✓ ffc89e9dfd88 Pull complete 0.4s
    ✓ 6b95a940e7b6 Pull complete 0.4s
    ✓ 90986bb8de6e Pull complete 0.4s
    ✓ 68c3898c2015 Pull complete 5.7s
    ✓ 20e4dcae4c69 Pull complete 5.5s
    ✓ 1c56c3d4ce74 Pull complete 0.4s
    ✓ e9f03a1c24ce Pull complete 5.6s
[+] Running 3/3
  ✓ Network dvwa-docker-lab_default Created 0.0s
  ✓ Container dvwa-mysql Started 0.7s
  ✓ Container dvwa Started 0.5s
PS C:\Users\Juan\Documents\GithubProjects\DVWA-Docker-Lab>

```

Run docker compose up -d, enable connection through firewall.

```

  ✓ 1c56c3d4ce74 Pull complete 0.4s
  ✓ e9f03a1c24ce Pull complete 5.6s
[+] Running 3/3
  ✓ Network dvwa-docker-lab_default Created 0.0s
  ✓ Container dvwa-mysql Started 0.7s
  ✓ Container dvwa Started 0.5s
PS C:\Users\Juan\Documents\GithubProjects\DVWA-Docker-Lab> docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                    NAMES
42339a571aaa   vulnerables/web-dvwa "/main.sh"              9 minutes ago Up 9 minutes   0.0.0.0:80->80/tcp      dvwa
7f56c6d47a22   mysql:5.7           "docker-entrypoint.s..." 9 minutes ago Up 9 minutes   3306/tcp, 33060/tcp     dvwa-mysql
PS C:\Users\Juan\Documents\GithubProjects\DVWA-Docker-Lab> |

```

Ensure dvwa is running.

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

DVWA

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- hWAPP
- NOWASP (formerly known as Mutillidae)
- OWASP Broken Web Applications Project

You have logged in as 'admin'

Username: admin

Security Level: low

PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.10 "Development"

Visit localhost/index.php – DVWA is now live.

New Inbound Rule Wizard

Name

Specify the name and description of this rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name**

Name:
Allow DVWA from Kali Only

Description (optional):

< Back Finish Cancel

Configure firewall to allow connection from internal network only on port 80.

```
(kali㉿kali)-[~]  
$ curl -I http://[REDACTED]  
  
HTTP/1.1 302 Found  
Date: Mon, 31 Mar 2025 20:41:52 GMT  
Server: Apache/2.4.25 (Debian)  
Set-Cookie: PHPSESSID=5alb9pmgq6svl3cmljpdhvmkn7; path=/  
Expires: Thu, 19 Nov 1981 08:52:00 GMT  
Cache-Control: no-store, no-cache, must-revalidate  
Pragma: no-cache  
Set-Cookie: PHPSESSID=5alb9pmgq6svl3cmljpdhvmkn7; path=/  
Set-Cookie: security=low  
Location: login.php  
Content-Type: text/html; charset=UTF-8  
  
(kali㉿kali)-[~]  
$
```

Run `curl -I http://<your-ip-address>`