**Technical Writing Sample #1: A Beginner’s Guide to Installing and Configuring Docker on Windows**

**Table of Contents**

1. [Introduction](#introduction)
2. [System Requirements](#system-requirements)
3. [Step-by-Step Installation](#step-by-step-installation)
   1. [Enable Virtualization](#step-1-enable-virtualization)
   2. [Download Docker Desktop Installer](#step-2-download-docker-desktop-installe)
   3. [Run the Installer](#step-3-run-the-installer)
   4. [Configure Docker Settings](#step-4-configure-docker-settings)
4. [Verifying Your Installation](#verifying-your-installation)
5. [Basic Usage](#basic-usage)
   1. [Running Your First Container](#running-your-first-container)
   2. [Pulling an Image](#pulling-an-image)
6. [Troubleshooting Tips](#troubleshooting-tips)
7. [Conclusion](#conclusion)

<a name="introduction"></a>

**1. Introduction**

Docker is a popular platform that allows developers to package applications into containers—lightweight, isolated environments that run consistently across different computing platforms. By using containers, you can ensure your application behaves the same regardless of the underlying operating system or infrastructure.

This guide is intended for **beginners** who want to install and configure Docker on a Windows machine. We will walk through essential setup steps, verification procedures, and basic usage. By the end of this tutorial, you’ll have a functional Docker environment and a basic understanding of how to run containers.

<a name="system-requirements"></a>

**2. System Requirements**

Before you begin, confirm that your system meets the following requirements:

* **Windows Version**: Docker Desktop for Windows supports:
  + Windows 10 64-bit: Pro, Enterprise, or Education (Build 1903 or later)
  + Windows 11 64-bit: All editions (recommended)
* **RAM**: A minimum of 4GB of system RAM (8GB recommended for comfortable performance).
* **Processor**: 64-bit processor with support for **hardware virtualization** (Intel VT-x or AMD-V).
* **Free Disk Space**: At least 5GB of free space to accommodate Docker’s installation and container storage.

**Note**: Home editions of Windows 10 and older builds can use Docker Toolbox or WSL2-based installations, but the recommended approach for modern systems is Docker Desktop. Check Microsoft and Docker documentation for compatibility details.

<a name="step-by-step-installation"></a>

**3. Step-by-Step Installation**

<a name="step-1-enable-virtualization"></a>

**Step 1: Enable Virtualization**

1. **Check if Virtualization is Enabled**
   * Open **Task Manager** by right-clicking the Taskbar and selecting *Task Manager*.
   * Click the **Performance** tab.
   * Look for “Virtualization” on the bottom-right corner. If it says “Enabled,” you can proceed. If it says “Disabled,” you must enable it in your system BIOS or UEFI settings.
2. **Enable Virtualization in BIOS/UEFI**
   * Restart your computer and enter BIOS/UEFI setup (commonly by pressing **F2**, **F10**, or **DEL** during boot).
   * Locate the setting for **Intel VT-x** or **AMD-V**, often found under “Advanced” or “CPU Configuration.”
   * Enable hardware virtualization, save your changes, and restart your machine.

<a name="step-2-download-docker-desktop-installer"></a>

**Step 2: Download Docker Desktop Installer**

1. **Navigate to the Docker Website**
   * Go to https://www.docker.com/products/docker-desktop/.
2. **Select Windows**
   * Download the **Docker Desktop for Windows** installer (*.exe* file).
3. **Save the Installer**
   * Choose a folder where you can easily find the downloaded installer (e.g., your *Downloads* directory).

<a name="step-3-run-the-installer"></a>

**Step 3: Run the Installer**

1. **Double-Click the Installer**
   * Locate the downloaded Docker Desktop Installer.exe file.
   * Double-click to launch the installation wizard.
2. **Accept User Account Control Prompt**
   * If prompted, click **Yes** to allow changes to your system.
3. **Follow the Wizard**
   * Check the option to enable **Hyper-V** or **WSL2** based on your preference (Hyper-V is recommended for Windows 10 Pro; WSL2 is recommended for Windows 10 Home or Windows 11).
   * Click **Install** and wait until the process completes.
4. **Restart if Required**
   * Docker Desktop may prompt you to restart your computer. Save your work and follow the prompt if needed.

<a name="step-4-configure-docker-settings"></a>

**Step 4: Configure Docker Settings**

1. **Launch Docker Desktop**
   * After installation, click the Docker icon on your desktop or in your Start menu to open Docker Desktop.
2. **Initial Setup**
   * If prompted, sign in with your Docker account or create a new one (optional).
   * Docker Desktop will start setting up the environment.
3. **Configure Resources** (Optional)
   * Open the **Settings** panel.
   * Under **Resources** > **Advanced**, you can allocate CPU cores and memory for Docker.
   * Adjust these settings if you have specific performance needs.
4. **Switch Between Hyper-V and WSL2** (Optional)
   * In **Settings** > **General**, you can opt to use WSL2 as your backend if your system meets the prerequisites.

**Tip**: Keep your Docker Desktop updated to the latest version for security patches and improved stability.

<a name="verifying-your-installation"></a>

**4. Verifying Your Installation**

1. **Check Docker Version**
   * Open **Command Prompt** or **PowerShell**.
   * Type docker --version and press Enter.
   * The output should display the current Docker version installed.
2. **Run Hello World Container**
   * In the terminal, type docker run hello-world.
   * If everything is configured correctly, Docker pulls the *hello-world* image from Docker Hub, runs the container, and prints a message confirming your installation.

<a name="basic-usage"></a>

**5. Basic Usage**

<a name="running-your-first-container"></a>

**5.1 Running Your First Container**

1. **Pull a Simple Image**
   * Example: docker pull nginx
   * This command downloads the official Nginx web server container image from Docker Hub.
2. **Run the Container**
   * Example:

bash

CopyEdit

docker run -d -p 80:80 --name mynginx nginx

* + **-d** runs the container in detached mode.
  + **-p 80:80** publishes the container’s port 80 to your local machine’s port 80.
  + **--name mynginx** names the container *mynginx*.

1. **Open Your Browser**
   * Go to http://localhost.
   * You should see the default Nginx welcome page.

<a name="pulling-an-image"></a>

**5.2 Pulling an Image**

* **Basic Command**: docker pull [image\_name]
  + For instance, docker pull python:3.9 downloads the Python 3.9 image.
* **Listing Images**: docker images shows all images on your system.
* **Removing Unused Images**: docker image rm [image\_id] helps keep your system clean.

<a name="troubleshooting-tips"></a>

**6. Troubleshooting Tips**

* **Virtualization Disabled**: If Docker fails to start, verify that hardware virtualization remains enabled in your BIOS/UEFI.
* **Port Conflicts**: Containers may fail to run if a host port is already occupied. Try stopping any service using the same port or mapping the container to a different port.
* **Low System Resources**: Docker Desktop can slow down if you have insufficient RAM or CPU resources. Increase your available memory in the Docker Settings.
* **WSL2 Configuration (Windows 10/11 Home)**: Ensure WSL2 is installed and updated via Windows features. If Docker fails to initialize, run wsl --update in PowerShell and restart Docker Desktop.
* **Permissions Issues**: If you’re unable to pull or run images, verify that Docker has permission to write to its default storage locations or adjust path settings in Docker Desktop.

<a name="conclusion"></a>

**7. Conclusion**

Congratulations! You’ve successfully installed Docker Desktop on Windows, verified the setup, and run your first container. By following these steps, you now have a fully functional Docker environment ready to support a wide range of containerized applications—from simple web servers to complex multi-service solutions.

For more advanced usage, consider exploring Docker Compose for orchestrating multi-container applications or Docker Swarm/Kubernetes for scaling workloads across multiple hosts. Always consult the official Docker documentation (https://docs.docker.com) for the most up-to-date information on new features, best practices, and troubleshooting procedures.

**Author**:  
*Frederick Walter James*

**Target Audience**:

* Beginners who have little or no prior experience with Docker.
* Windows users looking for a straightforward, step-by-step installation guide.