How To run Docker Containers on Windows Server 2019

By **Josphat Mutai** - May 9, 2020

You can support us by downloading this article as PDF from the Link below.

Download the guide as PDF

In this tutorial, we will look at how you can configure your Windows server 2019 to run Docker containers. Docker has been a game changer in Applications containerization and the whole microservices design and deployment patterns. Docker makes it easy to build, ship and run images containing applications with their dependencies and avoid crazy dependency issues common with the use of Virtual Machines.

Docker engine is what powers docker containers. It was originally written for Linux but a lot of work has been done to enable Windows and macOS users to run Docker containers.

One pre-requisite is the installation of a Windows server. This can be on a Virtual Machine running on-premise, a Physical server deployment or a Cloud instance running in Azure. You can refer to our installation guide below.

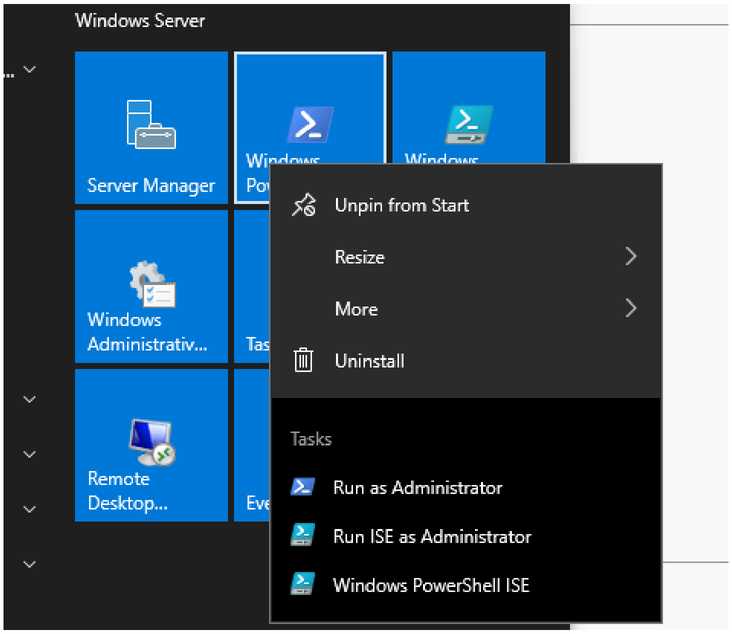
How to Install Windows Server 2019 Step by Step

How to run Docker Containers on Windows Server 2019

Before you can use the Windows Containers to run multiple isolated applications your system, you'll need to enable the containers feature and install Docker on Windows Server 2019.

Step 1: Enable the containers feature in Windows Server 2019

The first step is to enable the Windows Server 2019 containers feature. Open PowerShell as Administrator.

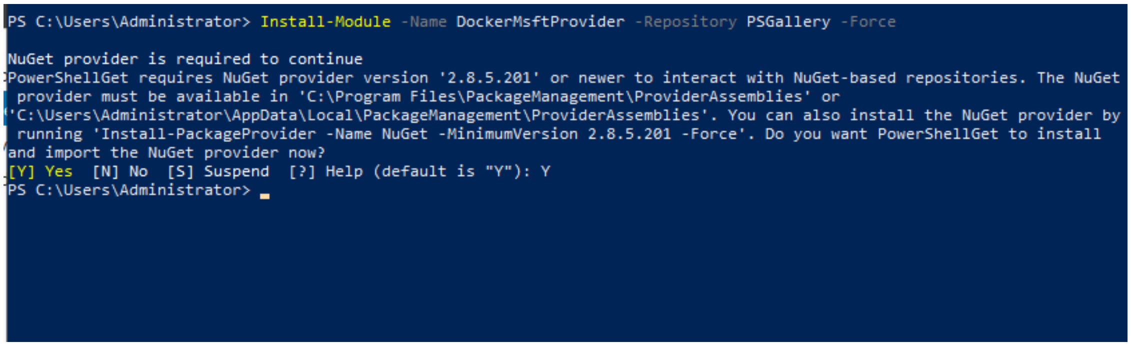


Run the following commands.

```
Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
```

This will install the Docker-Microsoft PackageManagement Provider from the PowerShell Gallery.

Sample output is as shown below:



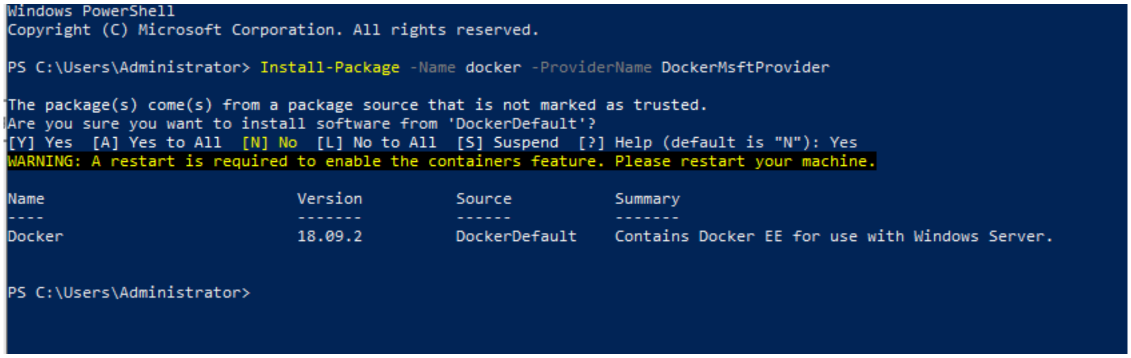
```
PS C:\Users\Administrator> Install-Module -Name DockerMsftProvider -Repository PSGallery -Force

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
 provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\Administrator\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider by
 running 'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install
and import the NuGet provider now?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
PS C:\Users\Administrator>
```

Step 2: Install Docker on Windows Server 2019

once the Containers feature is enabled on Windows Server 2019, install the latest Docker Engine and Client by running the command below in your PowerShell session.

```
Install-Package -Name docker -ProviderName DockerMsftProvider
```

Agree to the installation using “Yes” or “Y” or “A” to Agree to all.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Install-Package -Name docker -ProviderName DockerMsftProvider

The package(s) come(s) from a package source that is not marked as trusted.
Are you sure you want to install software from 'DockerDefault'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Yes
WARNING: A restart is required to enable the containers feature. Please restart your machine.

Name                           Version          Source           Summary
----                           -------          ------           -------
Docker                         18.09.2          DockerDefault    Contains Docker EE for use with Windows Server.


PS C:\Users\Administrator>
```

When the installation is complete, reboot the computer.

```
Restart-Computer -Force
```

Installed Docker version can be checked with:

```
Administrator> Get-Package -Name Docker -ProviderName DockerMsftProvider
 Name                           Version          Source
ProviderName
 ----                           -------          ------
------------
 docker                         18.09.2          DockerDefault
DockerMsftProvider
```

The same can be achieved with the `docker --version` command.

```
PS C:\Users\Administrator> docker version
 Client:
  Version:           18.09.2
  API version:       1.39
  Go version:        go1.10.6
  Git commit:        1ac774dfdd
  Built:             unknown-buildtime
  OS/Arch:           windows/amd64
  Experimental:      false
 error during connect: Get
http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.39/version: open
//./pipe/docker_engine: The system cannot find the file specified. In the
default daemon configuration on Windows, the docker client must be run
elevated to connect. This error may also indicate that the docker daemon is
not running.
```

Upgrade can be done anytime by running the following commands on PowerShell.

```
Install-Package -Name Docker -ProviderName DockerMsftProvider -Update -Force
Start-Service Docker
```

Step 3: Run Docker Container

Start Docker Daemon

```
Start-Service Docker
```

After starting Docker Engine service, Download the pre-created **.NET** sample image from the Docker Hub registry:

```
docker pull microsoft/dotnet-samples:dotnetapp-nanoserver-1809
```

Then deploy a simple container running a .Net Hello World application.

```
docker run microsoft/dotnet-samples:dotnetapp-nanoserver-1809
```

The container will start, print the hello world message, and then exits.

Running Linux Containers on Windows Server 2019

Out of the box, Docker on Windows only run Windows container. To use Linux containers on Windows Server, you need to use the Docker Enterprise Edition Preview which includes a full LinuxKit system for running Docker Linux containers.

Uninstall your current Docker CE.

```
Uninstall-Package -Name docker -ProviderName DockerMSFTProvider
```

Enable Nested Virtualization if you're running Docker Containers using Linux Virtual Machine running on Hyper-V.

```
Get-VM WinContainerHost | Set-VMProcessor -ExposeVirtualizationExtensions
$true
```

Then install the current preview build of Docker EE.

```
Install-Module DockerProvider
Install-Package Docker -ProviderName DockerProvider -RequiredVersion preview
```

Enable LinuxKit system for running Linux containers

```
[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", "1", "Machine")
```

Restart Docker Service after the change.

```
Restart-Service docker
```

Pull a test docker image.

```
> docker run -it --rm ubuntu /bin/bash
 root@1440a7fef7e0:/# cat /etc/os-release
 NAME="Ubuntu"
 VERSION="18.04.1 LTS (Bionic Beaver)"
 ID=ubuntu
 ID_LIKE=debian
 PRETTY_NAME="Ubuntu 18.04.1 LTS"
 VERSION_ID="18.04"
 HOME_URL="https://www.ubuntu.com/"
 SUPPORT_URL="https://help.ubuntu.com/"
 BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
 PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-
policies/privacy-policy"
 VERSION_CODENAME=bionic
 UBUNTU_CODENAME=bionic

 root@1440a7fef7e0:/# exit
 exit
```

To Switch back to running Windows containers, run:

```
[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", "$null", "Machine")
```

Enjoy running Linux and Windows containers on Windows Server 2019. Drop us a comment in case of any issues.

Also check:

How to install Applications from Windows command line

How to Allow ICMP Echo Reply on Windows Server 2019

How to enable Remote Desktop Protocol (RDP) on Windows Server 2019

How to run Linux on Windows Server 2019 with WSL

You can support us by downloading this article as PDF from the Link below.

Download the guide as PDF

Josphat Mutai

https://computingforgeeks.com/

Founder of Computingforgeeks. Expertise in Virtualization, Cloud, Linux/UNIX Administration, Automation,Storage Systems, Containers, Server Clustering e.t.c.