# Build and run an application with or without .NET Core 2.0 or PowerShell Core 6

The Nano Server base OS Container image in this release has removed .NET Core and PowerShell, though both .NET Core and PowerShell are supported as an add-on layered container on top of the base Nano Server container.

If your container is to run native code or open frameworks such as Node.js, Python, Ruby, etc, the base Nano Server container is sufficient. One nuance is that certain native code may not run as a result of [footprint savings](https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel) in this release compared to WS2016 release. If there are any regression issues you notice, let us know in the [forums](https://social.msdn.microsoft.com/Forums/en-US/home?forum=windowscontainers).

To build your container from a Dockerfile, use docker build and to run it, docker run. The following command will download the Nano Server Container base OS image, which may take a few minutes, and print a “Hello World!” message at the host console.

docker run microsoft/nanoserver-insider cmd /c echo Hello World!

You can build more complicated applications using [Dockerfiles on Windows](https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-docker/manage-windows-dockerfile), with Dockerfile syntax such as FROM, RUN, COPY, ADD, CMD, etc. However, you cannot use syntax like RUN powershell, or RUN dism.exe with this base image, due to the removal of PowerShell and Servicing Stack.

As a result of both .NET Core and PowerShell not being available in the base Nano Server container OS image, one challenge is on how to build a container with content in compressed zip format. For example, a Dockerfile containing PowerShell unzip cmdlet will not work anymore from this base image:

FROM microsoft/nanoserver-insider

RUN powershell -command Expand-Archive -Path c:\apache.zip -DestinationPath c:\

With the [multi-stage build](https://docs.docker.com/engine/userguide/eng-image/multistage-build/) feature available in Docker 17.05, you can leverage PowerShell in another container to unzip the content and copy into the Nano container. This approach can be used to create a .NET Core container and a PowerShell container.

## Deploy apps based on .NET Core 2.0

You can leverage the .NET Core 2.0 container image in the Insider release to run your .NET Core apps, where your .NET Core application is built elsewhere and you want to run it in the container. You can find more information on how to run a .NET Core application with the .NET Core container images at [.NET Core GitHub](https://github.com/dotnet/dotnet-docker-nightly). If you are developing an application inside the container, the .NET Core SDK should be used instead. For advanced users, you can build your own .NET Core 2.0 container with the .NET Core 2.0 version, Dockerfile, and URL specified in the [dotnet-docker-nightly](https://github.com/dotnet/dotnet-docker-nightly/tree/master/2.0). To do that, a Windows Server Core container can be used to accomplish the download and unzip function. The Dockerfile sample is as below.

# escape=`

FROM microsoft/windowsservercore-insider AS installer-env

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

ENV DOTNET\_VERSION 2.0.0-preview2-25407-01

ENV DOTNET\_DOWNLOAD\_URL https://dotnetcli.blob.core.windows.net/dotnet/Runtime/$DOTNET\_VERSION/dotnet-runtime-$DOTNET\_VERSION-win-x64.zip

ENV DOTNET\_DOWNLOAD\_SHA 234F702025C292C6A432F2F93439294EE164F294485343027B952CEA2085B871064A92E6AED335DA7E05FCEA99AC303E81B750021D7782ACECA867F6D789F3DC

RUN Invoke-WebRequest $Env:DOTNET\_DOWNLOAD\_URL -OutFile dotnet.zip; `

if ((Get-FileHash dotnet.zip -Algorithm sha512).Hash -ne $Env:DOTNET\_DOWNLOAD\_SHA) { `

Write-Host 'CHECKSUM VERIFICATION FAILED!'; `

exit 1; `

}; `

Expand-Archive dotnet.zip -DestinationPath \dotnet; `

Remove-Item -Force dotnet.zip

FROM microsoft/nanoserver-insider

ENV ProgramFiles C:\Program Files

COPY --from=installer-env ["\\dotnet\\", "$ProgramFiles\\dotnet"]

RUN setx /M PATH "%PATH%;%ProgramFiles%\dotnet"

With this Dockerfile, a .NET Core 2.0 container can be built using the following command.

docker build -t nanoserverdnc -f Dockerfile-dotnetRuntime .

## Run PowerShell Core 6 in a container

Using the same [multi-stage build](https://docs.docker.com/engine/userguide/eng-image/multistage-build/) method, a PowerShell Core 6 container can be built with the following Dockerfile.

# escape=`

FROM microsoft/windowsservercore-insider AS installer-env

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

ARG PS\_VERSION=6.0.0-beta.3

ARG PS\_DOWNLOAD\_SHA=FF45FE7D184B50297CA706EFAEA4E37C59807D33577521A715C147BCD5098B01

ENV PS\_DOWNLOAD\_URL https://github.com/PowerShell/PowerShell/releases/download/v$PS\_VERSION/PowerShell-$PS\_VERSION-win10-win2016-x64.zip

RUN Invoke-WebRequest $Env:PS\_DOWNLOAD\_URL -OutFile powershell.zip; `

if ((Get-FileHash powershell.zip -Algorithm sha256).Hash -ne $Env:PS\_DOWNLOAD\_SHA) { `

Write-Host 'CHECKSUM VERIFICATION FAILED!'; `

exit 1; `

}; `

Expand-Archive powershell.zip -DestinationPath \PowerShell; `

Remove-Item -Force powershell.zip

FROM microsoft/nanoserver-insider

ENV ProgramFiles C:\Program Files

COPY --from=installer-env ["\\PowerShell\\", "$ProgramFiles\\PowerShell"]

ENV PSCORE="$ProgramFiles\PowerShell\PowerShell.exe"

RUN setx /M PATH "%PATH%;%ProgramFiles%\PowerShell"

Then issue docker build to create the PowerShell container image.

docker build -t nanoserverPowerShell6 -f Dockerfile-PowerShell6 .

You can find more information at [PowerShell GitHub](https://github.com/PowerShell/PowerShell/tree/master/docker/release). It is worth mentioning that the PowerShell zip contains a subset of .NET Core 2.0 that is required to build PowerShell Core 6. If your PowerShell modules depend on .NET Core 2.0, it is safe to build the PowerShell container on top of the Nano .NET Core container, instead of base Nano container, i.e. using FROM microsoft/nanoserver-insider-dotnet in the Dockerfile.

## Next steps

* Use one of the new container images based on Nano Server, available in Docker Hub, i.e. base Nano Server image, Nano with .NET Core 2.0, and Nano with PowerShell Core 6
* Build your own container image based on the new Nano Server Container base OS image, using the Dockerfile sample content in this guide