1) What is Docker container.

Use containers to Build, Share and Run your applications

Package Software into Standardised Units for Development, Shipment and Deployment

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers – images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker containers that run on Docker Engine:

- **Standard:** Docker created the industry standard for containers, so they could be portable anywhere

- **Lightweight:** Containers share the machine’s OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
- **Secure:** Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

2)Write a Benifit of Docker contained.

Speed

Because of their lightweight and modular nature, containers can enable rapid iteration of your applications. Development speed is improved by the ability to deconstruct applications into smaller units. This reduces shared resources between application components, leading to fewer compatibility issues between required libraries or packages. Operational speed is improved, because code built in a container on a developer’s local machine can be easily moved to a test server by simply moving the container. The container startup time primarily depends on the size of the container image, cache, and the time to pull the image and start the container on host. To improve the container startup time, you must keep the size of image as small as possible, using techniques like multi-stage builds and local cache when applicable. For more information, see Best practices for writing Dockerfiles.

Consistency

The consistency and fidelity of a modular development environment provide predictable results when moving code between development, test, and production systems. By verifying that the container encapsulates exact versions of necessary libraries and packages, it is possible to minimize the risk of bugs due to slightly different dependency revisions. This concept easily lends itself to a *disposable system* approach, in which patching individual containers is less preferable than building new containers in parallel, testing, and replacing the earlier. This practice helps avoid *drift* of packages across a fleet of containers, versions of your application, or development, test, and production environments; the result is more consistent, predictable, and stable applications.

Density and resource efficiency

Containers facilitate enhanced resource efficiency by allowing multiple containers to run on a single system. Resource efficiency is a natural result of the isolation and allocation techniques that containers use. Containers can easily be restricted to a certain number of CPUs and allocated specific amounts of memory. By understanding what resource a container needs and what resource is available to your VM or underlying host server, it’s possible to maximize the containers running on a single host, resulting in higher density, increased efficiency of compute resources, and less wastage on excess capacity. Amazon ECS achieves this through placement strategies. The *binpack* placement strategy tries to optimize placement of containers to be as cost-efficient as possible. Containers in Amazon ECS are part of Amazon ECS tasks placed on compute instances to leave the least amount of unused CPU or memory. This in turn minimizes the number of computed instances in use, resulting in better resource efficiency. The placement strategies can be supported by placement constraints, which lets you place tasks by constraints like the instance type or the availability zone. This further enables you to efficiently utilize resources by verifying that your tasks are running on instance types suitable for your workload, by logically separating your tasks using task groups.

Amazon EKS uses the native Kubernetes scheduling and placement strategy, which tries to place pods on nodes to match the requirements of your workloads across nodes and not to place pods on nodes where there aren't sufficient resources.

Kubernetes allows you to limit the resources like CPU and memory to Kubernetes namespaces, pods, or containers. For more information, see Scheduling.

Portability

The flexibility of Docker containers is based on their portability, ease of deployment, and smaller size compared to virtual machines. Like Git, Docker provides a simple mechanism for developers to download and install Docker containers and their subsequent applications using the command `docker pull`. Because Docker provides a standard interface, it makes containers easy to deploy wherever you like, providing portability among different versions of Linux, your laptop, or the cloud. The images Docker builds are compliant with OCI (Open Container Initiative), which was created to support fully interoperable container standards. Docker can build images by reading the instructions from a *Dockerfile*, which is a text-based manifest. You can run the same Docker container on any supported version of Linux if you have the Docker stack installed on the host. Additionally, Docker supports Windows containers which can run on supported Windows versions. Containers also provide flexibility by making a microservices architecture possible. In contrast to common infrastructure models in which a virtual machine runs multiple services, packaging services inside their own container on top of a host OS allows a service to be moved between hosts, isolated from failure of other adjacent services, and protected from errant patches or software upgrades on the host system. Because Docker provides clean, reproducible, and modular environments, it streamlines both code deployment and infrastructure management. Docker offers numerous benefits for a variety of use cases, whether in development, testing, deployment, or production.

3)write a installation step of Docker.

Step-By-Step Docker Installation on Windows

1. Go to the website https://docs.docker.com/docker-for-windows/install/ and download the docker file.

Note: A 64-bit processor and 4GB system RAM are the hardware prerequisites required to successfully run Docker on Windows 10.

2. Then, double-click on the Docker Desktop Installer.exe to run the installer.

Note: Suppose the installer (Docker Desktop Installer.exe) is not downloaded; you can get it from Docker Hub and run it whenever required.

3. Once you start the installation process, always enable Hyper-V Windows Feature on the Configuration page.

4. Then, follow the installation process to allow the installer and wait till the process is done.

5. After completion of the installation process, click Close and restart.

Start Docker Desktop Tool

After the installation process is complete, the tool does not start automatically. To start the Docker tool, search for the tool, and select Docker Desktop in your desktop search results.

Before starting the application, Docker offers an onboarding tutorial. The tutorial explains how to build a Docker image and run a container.

You are now successfully running Docker Desktop on Windows.

Next, follow the instruction below to install the Docker engine on your system.

Go to Docker CLI and run the Docker version to verify the version of Docker installation on the system.

Congratulations, Docker Installation on Windows is now done, and now, you are ready to build and run Docker images and containers on the Docker ecosystem.

How To Install Docker on Windows 10?

1. To start docker desktop, first you need to download the docker file on windows.
2. Docker Downloaded file can be found in the download folder.
3. Once the setup starts to run, the configuration page will appear where you need to choose options: Hyper V feature and WSL 2 feature to add particular components on windows.
4. Then click OK and simply follow the instructions on wizard and continue for the authorized installation process.
5. Click on the close button once the installation process is finished.

6. Ensure that your docker user account and administrator account should be the same, otherwise you need to add your user account to the docker user group.
7. Initially run the admin as computer management.
8. Then go to local users and groups. From there you can choose groups where you can find docker user groups. Then you can add new users into the docker group.
9. After following all these steps, restart your computer to update and start docker desktop on windows 10.
10. Once you click on docker desktop at the start, it will ask you for a subscription service agreement for docker desktop. As soon as you accept it and agree to the terms and conditions, the docker desktop windows will appear and you're ready to work on it. Don't forget to use a quick start guide by docker desktop to get the better experience.

How to Install Docker on Windows 11?

To install docker desktop on windows 11, your computer must have windows 11, 64 bit processor supported by Microsoft and rest of the steps will be followed the same as installing docker on Windows 10.

How to Install from the command line

Following command lines can be used to install docker desktop on windows 10, 11 or higher versions

1. To run in terminal,
   ```
   "Docker Desktop Installer.exe" install
   ```

2. For Powershell,
   Start-Process '.\win\build\Docker Desktop Installer.exe' -Wait install
3. For windows command prompt,
   start /w "" "Docker Desktop Installer.exe" install
4. To add user accounts into docker user groups especially when your admin account and user account are not the same.
   net localgroup docker-users <users>/add

Verification

To verify the docker installation and versions, you can use following command lines

1. With windows terminal via command prompt,
   docker-run
2. From Powershell, verify the docker version and installation
3. You can check the latest version of docker desktop from the docker option, once the installation process is complete.
4. You need to enter a token number from the Beta invitation email, when you start docker for the first time.
5. On the other hand, you can check for a web server by docker desktop.
6. You can also verify the setup from docker download with nginx from docker hubs.

Updates Guide

1. Usually a new version of docker desktop is available whenever there is an update for docker desktop.

2. One can click for an update option and updates will be downloaded with newly added features. To know more about updates and features, you can check and read release notes.
3. Similarly, Mac also suggests Updates for docker desktop where you need to go to settings for software updates.
4. If you don't want to download automatic updates, you can use docker desktop 4.2.0 to turn off the automatic download update option. Nevertheless, this option is available for all docker desktop subscribed users.
5. Once you download the updates successfully, click on update and restart option on docker start menu.
6. This will allow changes to your previous docker desktop version.
7. When all updates are done, it will again ask you to accept terms and subscription for the latest docker version. Once you accept those terms, you are ready to use the new version of docker desktop.

Uninstall Docker Desktop Tool

Suppose you want to uninstall Docker Desktop from your Windows Home machine, follow the steps mentioned below:

1. Go to the Windows Start menu, choose Settings > Apps > Apps & features.
2. Now, choose Docker Desktop from the Apps & features list and click on Uninstall.

Learn core Docker technologies such as Docker Hub, Docker Compose, Docker Swarn with the Docker Certified Associate Certification Training Course.

Conclusion

That was all the basics about Docker installation on windows.

If you are interested in gaining in-depth knowledge of the Docker tool, Simplilearn can definitely help you. This Docker In-Depth Training Course helps every individual learn all the basic and advanced concepts of Docker. With this certification, you will gain hands-on experience on how to create flexible application environments with Docker.

Do you have any doubts about this topic? Please feel free to place your questions in the comments section of this Docker Installation on Windows article. Our subject matter experts will get back to you at the earliest!