What is docker?

Docker is a container engine which basically allows you application to run in side containers. Before docker we used to run our application on VMs .

Container is nothing but a logical space .Container is logical grouping all binaries and dependencies required to run an application and finally bundling them to run on docker engine.

Used for application like micro service which are indecpendent they do not dependent on external factors they do not need any infrastructure.

Docker is free there is an paid version is present with few more features.

Docker is like mini platform as service. Container solutions like Docker provide a starting point for building PaaS(platform as service).

To put it simple, PaaS is a method used by a provider to deliver deployment and management capabilities to the end user. Containers, in turn, are one of the possible technical solutions to produce such an environment. In short, containers make PaaS possible.

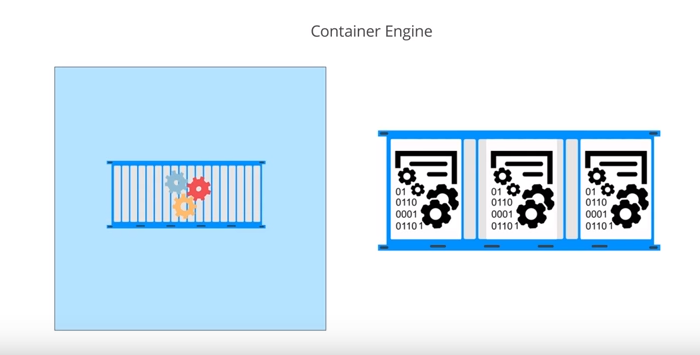
**Containers vs. PaaS**

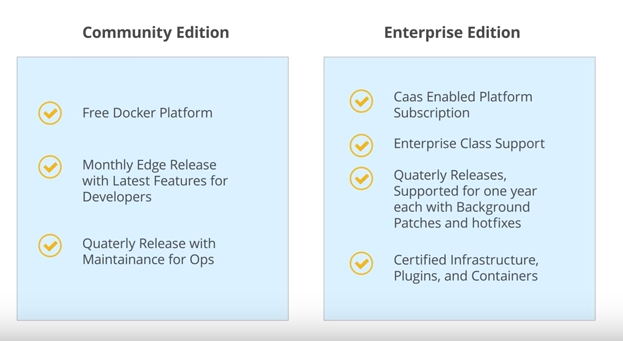
Using containers ‘as-is’ doesn’t mean that you get a PaaS solution, though. A standard container infrastructure like Docker provides essential tools to run and manage your apps from containers, but without orchestration solutions that facilitate and automate autoscaling, traffic routing, load balancing etc.

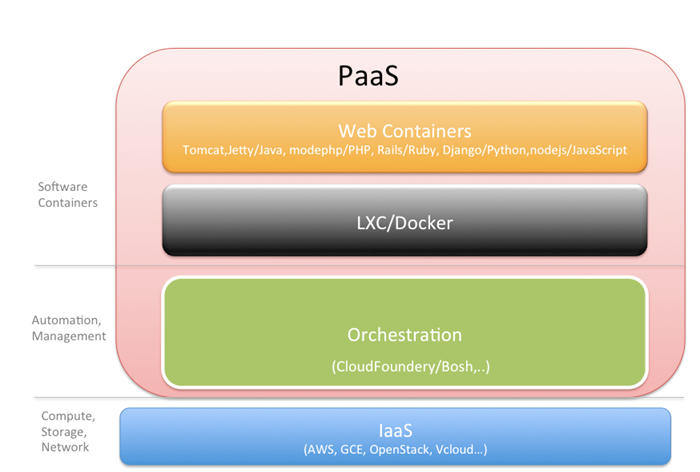
**Orchestration**

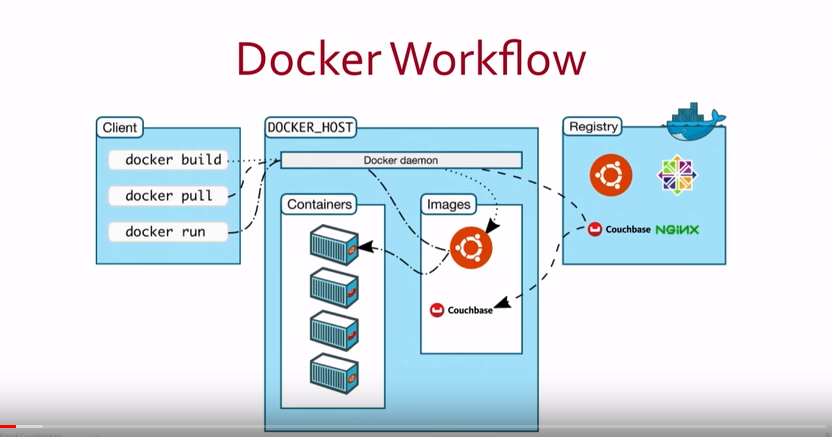
To make a platform providers leverage PaaS-enabling software like Kubernetes. It leverages containers to provide deployment and runtime environment for client’s apps. Due to powerful orchestration tools that interact with Docker containers, such solutions allow controlling your app behaviour, allocating additional RAM and CPU when needed, simplifying build, test, deploy, and release management and many more. Orchestrator solutions make container management more flexible and allow developing teams to concentrate on developing and fine-tuning their app instead of fretting with container infrastructure.

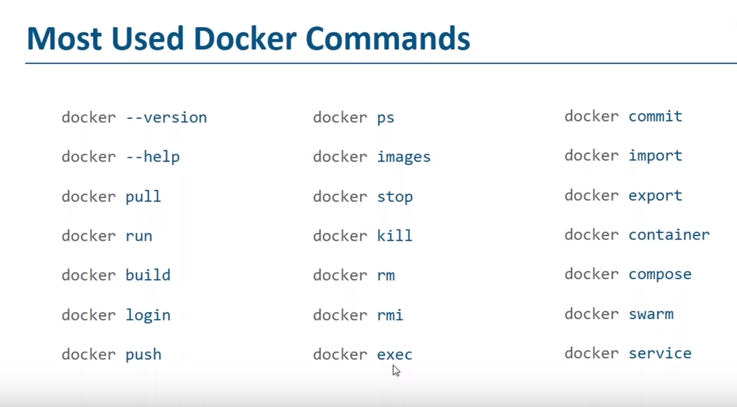
https://dzone.com/articles/orchestration-tool











If the features described are enabled the problem is with Hyper-V that is disabled or Hypervisor agent not running

**SOLUTION A** (If Hyper-V is totally disabled or not installed)

1. Open PowerShell as administrator and
2. Enable Hyper-V with

dism.exe /Online /Enable-Feature:Microsoft-Hyper-V /All

**SOLUTION B** (If Hyper-V feature is already enabled but doesn't work)

Enable Hypervisor with

bcdedit /set hypervisorlaunchtype auto

Now restart the system and try again.

**SOLUTION C**

If the problem persists probably Hyper-V on your system is corrupted, so

1. Go in Control Panel -> [Programs] -> [Windows Features] and completely uncheck all Hyper-V related components. Restart the system.
2. Enable Hyper-V again. Restart.

***NOTE 1*:**

Hyper-V needs hardware virtualization as prerequisite. Make sure your PC support it, if yes and still won't work there is the possibility your BIOS is not configured correctly and this feature is disabled. In this case, check, enable it and try again.

***NOTE 2:***

Hyper-V **can** be installed only with some version e.g.:

Windows 10 Enterprise; Windows 10 Professional; Windows 10 Education.

Hyper-V **cannot** be installed on cheaper or mobile Windows versions e.g.:

Windows 10 Home; Windows 10 Mobile; Windows 10 Mobile Enterprise.

Task manager -> perfermance -> virtualisation

Enable the the same from boot loader

**Docker file**

FROM openjdk:8 🡪 download the open jdk with version

ADD taget/spring-boot-docker.jar spring-boot-docker.jar 🡪 move the jar file to root dir of the docker container

EXPOSE 8085 🡪 run the docker in 8085 port

ENTRYPOINT ["java", "-jar", "spring-boot-docker.jar"] 🡪 command to execute the jar file

<https://hub.docker.com>

from this site search for openjdk for java projects to create docker image

docker build -f Dockerfile -t docker-spring-boot .

docker-spring-boot is image name here

. represents the current directory of project where i am executing the command

**CMD**

F:\springboot\spring-boot-docker>docker build -f Dockerfile -t docker-spring-boot .

Sending build context to Docker daemon 16.24MB

Step 1/4 : FROM openjdk:8

---> 891c9734d5ab

Step 2/4 : ADD ./target/spring-boot-docker.jar spring-boot-docker.jar

---> c16fb7647950

Step 3/4 : EXPOSE 8085

---> Running in c50b49267a61

Removing intermediate container c50b49267a61

---> de37e8dde60e

Step 4/4 : ENTRYPOINT ["java", "-jar", "spring-boot-docker.jar"]

---> Running in b2b496d5c5ec

Removing intermediate container b2b496d5c5ec

---> 7ade3e1d8bd8

Successfully built 7ade3e1d8bd8

Successfully tagged docker-spring-boot:latest

SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.

F:\springboot\spring-boot-docker>

F:\springboot\spring-boot-docker>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker-spring-boot latest 7ade3e1d8bd8 6 minutes ago 742MB

openjdk 8 891c9734d5ab 5 weeks ago 726MB

F:\springboot\spring-boot-docker>

F:\springboot\spring-boot-docker>docker run -p 8085:8085 docker-spring-boot

F:\springboot\spring-boot-docker>docker build -f Dockerfile -t docker-spring-boot .

Sending build context to Docker daemon 16.24MB

Step 1/4 : FROM openjdk:8

8: Pulling from library/openjdk

c73ab1c6897b: Pull complete

1ab373b3deae: Pull complete

b542772b4177: Pull complete

57c8de432dbe: Pull complete

da44f64ae999: Pull complete

0bbc7b377a91: Pull complete

1b6c70b3786f: Pull complete

d9bbcf733166: Pull complete

b1d3e8de8ec6: Pull complete

Digest: sha256:a2a5aff7337dbc24d0343262fc6fc8eac18e123c77d7101dbd326680bbcc03c2

Status: Downloaded newer image for openjdk:8

---> 891c9734d5ab

Step 2/4 : ADD ./target/spring-boot-docker.jar spring-boot-docker.jar

---> 609fad21a640

Step 3/4 : EXPOSE 8085

---> Running in 9b84c0bcffcd

Removing intermediate container 9b84c0bcffcd

---> 63bf1da233ca

Step 4/4 : ENTRYPOINT ["java", "-jar", "spring-boot-docker.jar"]

---> Running in bb004530c304

Removing intermediate container bb004530c304

---> 8b6fbfc13562

Successfully built 8b6fbfc13562

Successfully tagged docker-spring-boot:latest

SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.

F:\springboot\spring-boot-docker>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker-spring-boot latest 8b6fbfc13562 10 minutes ago 742MB

openjdk 8 891c9734d5ab 6 weeks ago 726MB

F:\springboot\spring-boot-docker>docker run -p 8085:8085 docker-spring-boot

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v2.0.1.RELEASE)

2018-05-01 15:15:13.967 INFO 1 --- [ main] c.e.demo.SpringBootDockerApplication : Starting SpringBootDockerApplication v0.0.1-SNAPSHOT on 0d873cfc39b3 with PID 1 (/spring-boot-docker.jar started by root in /)

2018-05-01 15:15:14.007 INFO 1 --- [ main] c.e.demo.SpringBootDockerApplication : No active profile set, falling back to default profiles: default

2018-05-01 15:15:14.386 INFO 1 --- [ main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@27abe2cd: startup date [Tue May 01 15:15:14 UTC 2018]; root of context hierarchy

2018-05-01 15:15:16.804 INFO 1 --- [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)

2018-05-01 15:15:16.865 INFO 1 --- [ main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2018-05-01 15:15:16.866 INFO 1 --- [ main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.29

2018-05-01 15:15:16.884 INFO 1 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: [/usr/java/packages/lib/amd64:/usr/lib/x86\_64-linux-gnu/jni:/lib/x86\_64-linux-gnu:/usr/lib/x86\_64-linux-gnu:/usr/lib/jni:/lib:/usr/lib]

2018-05-01 15:15:16.987 INFO 1 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

2018-05-01 15:15:16.988 INFO 1 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2618 ms

2018-05-01 15:15:17.220 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]

2018-05-01 15:15:17.230 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/\*]

2018-05-01 15:15:17.231 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/\*]

2018-05-01 15:15:17.232 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/\*]

2018-05-01 15:15:17.233 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/\*]

2018-05-01 15:15:17.432 INFO 1 --- [ main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/\*\*/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]

2018-05-01 15:15:17.779 INFO 1 --- [ main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@27abe2cd: startup date [Tue May 01 15:15:14 UTC 2018]; root of context hierarchy

2018-05-01 15:15:17.923 INFO 1 --- [ main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/rest/docker/hello],methods=[GET]}" onto public java.lang.String com.example.demo.resource.HelloResource.hello()

2018-05-01 15:15:17.935 INFO 1 --- [ main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2018-05-01 15:15:17.938 INFO 1 --- [ main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String, java.lang.Object>> org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.error(javax.servlet.http.HttpServletRequest)

2018-05-01 15:15:18.003 INFO 1 --- [ main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/\*\*] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]

2018-05-01 15:15:18.003 INFO 1 --- [ main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/\*\*] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]

2018-05-01 15:15:18.254 INFO 1 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup

2018-05-01 15:15:18.353 INFO 1 --- [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''

2018-05-01 15:15:18.361 INFO 1 --- [ main] c.e.demo.SpringBootDockerApplication : Started SpringBootDockerApplication in 5.983 seconds (JVM running for 7.204)

Ctrl + c 🡪 for stopping the container

**docker** ps -s 🡪 to list the containers in docker

F:\springboot\spring-boot-docker>docker ps -s

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES SIZE

0d873cfc39b3 8b6fbfc13562 "java -jar spring-bo…" About an hour ago Up About an hour 0.0.0.0:8085->8085/tcp zen\_wiles 32.8kB (virtual 742MB)

F:\springboot\spring-boot-docker>docker stop container

Error response from daemon: No such container: container

F:\springboot\spring-boot-docker>docker stop 0d873cfc39b3

0d873cfc39b3

<https://docs.docker.com/toolbox/toolbox_install_windows/>

<https://www.youtube.com/watch?v=nqeqmVVOlrY>

<https://www.youtube.com/watch?v=vkHMDXX6QxM>

<https://www.youtube.com/watch?v=1dgUXNVQS5o>

<https://www.youtube.com/watch?v=h0NCZbHjIpY>

docker swarm init –listen-addr localhost:2377

C:\Users\niran>docker swarm init

Swarm initialized: current node (mg721e3qgcblgeizrgvwbd3cg) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-65cw7mkh4fhulge878mamz5stg37jog8wsrt5z1uzmnv082386-5cpttm6z24uzcppr1d0yv4itg 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

F:\springboot\spring-boot-docker>docker swarm init

Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm and join another one.

F:\springboot\spring-boot-docker>docker swarm leave

Error response from daemon: You are attempting to leave the swarm on a node that is participating as a manager. Removing the last manager erases all current state of the swarm. Use `--force` to ignore this message.

F:\springboot\spring-boot-docker>docker swarm leave --force

Node left the swarm.

**Docker commands**

Options:

--config string Location of client config files (default

"C:\\Users\\niran\\.docker")

-D, --debug Enable debug mode

-H, --host list Daemon socket(s) to connect to

-l, --log-level string Set the logging level

("debug"|"info"|"warn"|"error"|"fatal")

(default "info")

--tls Use TLS; implied by --tlsverify

--tlscacert string Trust certs signed only by this CA (default

"C:\\Users\\niran\\.docker\\ca.pem")

--tlscert string Path to TLS certificate file (default

"C:\\Users\\niran\\.docker\\cert.pem")

--tlskey string Path to TLS key file (default

"C:\\Users\\niran\\.docker\\key.pem")

--tlsverify Use TLS and verify the remote

-v, --version Print version information and quit

Management Commands:

config Manage Docker configs

container Manage containers

image Manage images

network Manage networks

node Manage Swarm nodes

plugin Manage plugins

secret Manage Docker secrets

service Manage services

swarm Manage Swarm

system Manage Docker

trust Manage trust on Docker images

volume Manage volumes

Commands:

attach Attach local standard input, output, and error streams to a running container

build Build an image from a Dockerfile

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes to files or directories on a container's filesystem

events Get real time events from the server

exec Run a command in a running container

export Export a container's filesystem as a tar archive

history Show the history of an image

images List images

import Import the contents from a tarball to create a filesystem image

info Display system-wide information

inspect Return low-level information on Docker objects

kill Kill one or more running containers

load Load an image from a tar archive or STDIN

login Log in to a Docker registry

logout Log out from a Docker registry

logs Fetch the logs of a container

pause Pause all processes within one or more containers

port List port mappings or a specific mapping for the container

ps List containers

pull Pull an image or a repository from a registry

push Push an image or a repository to a registry

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

save Save one or more images to a tar archive (streamed to STDOUT by default)

search Search the Docker Hub for images

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop one or more running containers

tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

top Display the running processes of a container

unpause Unpause all processes within one or more containers

update Update configuration of one or more containers

version Show the Docker version information

wait Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.