Agenda

. What resources are required to run an application ?

. Traditional way of hosting an applications ?

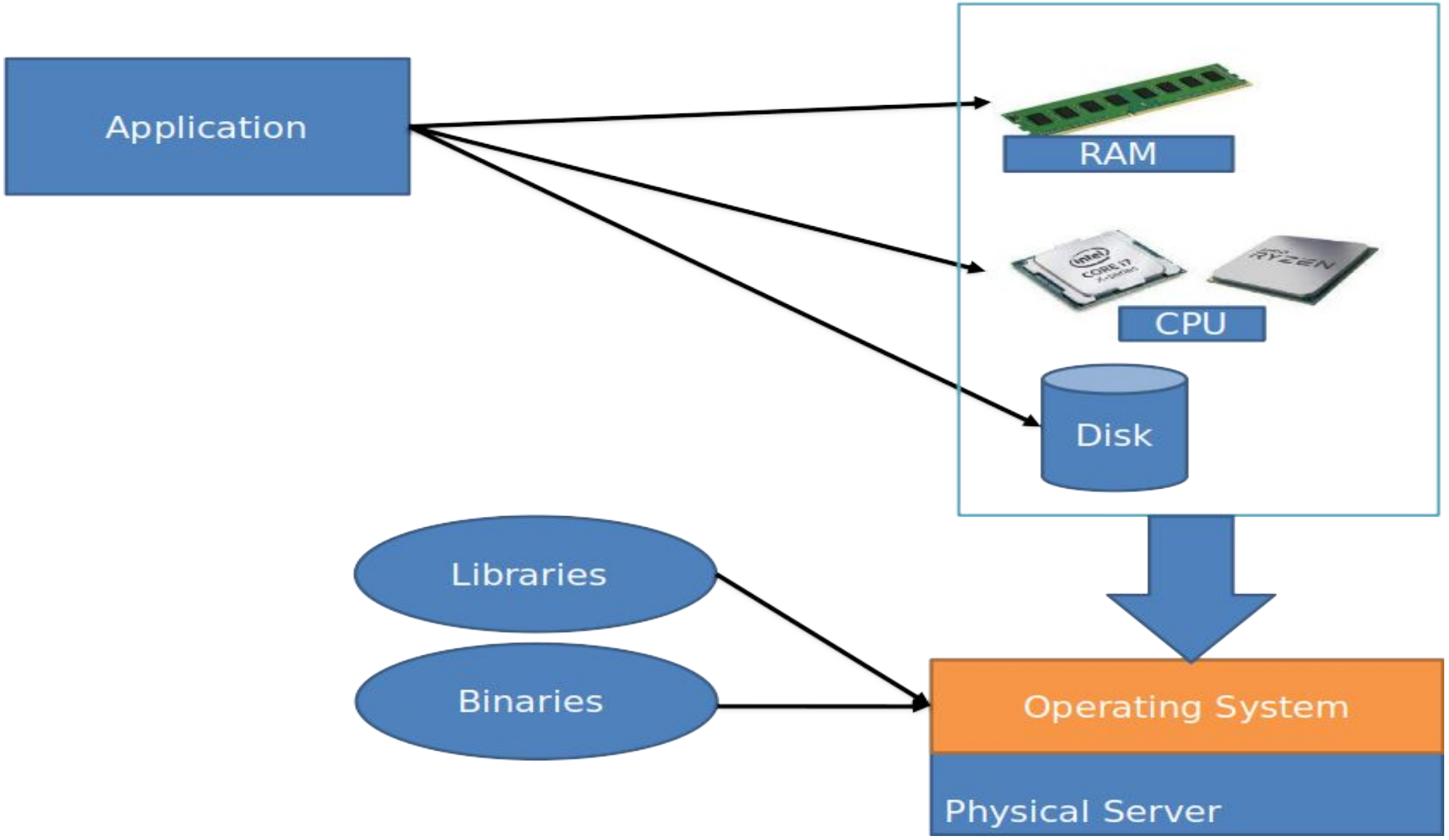
. Why we need virtualization?

. Why we need Container?

. Difference between VM and Container ?

. What is container ?

. What is Docker?


Application
RAM
CPU
Disk
Libraries
Binaries
Operating System
Physical Server

Step 1: Remove already installed container engine.

sudo yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman buildah -y

Execute below commands in order to install the Docker packages

yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman buildah -y

Step 2: yum install -y yum-utils

Why we need to install the yum-utils ?

yum-utils: is a collection of tools and programs for managing yum repositories, installing debug packages, source packages, extended information from repositories and administration.

Step 3: Need to add the repo file for Docker packages

yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

yum install containerd.io docker-ce docker-ce-cli -y

cat /etc/yum.repos.d/docker-ce.repo

systemctl enable docker.service

systemctl start docker.service

systemctl status docker.service

docker –version

containerd is available as a daemon for Linux and Windows. It manages the complete container lifecycle of its host system, from image transfer from one place to another and storage to container execution (fetch the file from the storage if container required) and supervision to low-level storage to network attachments (data will flow from where?) and beyond.

Containerd uses the OS kernel feature such as Namespace and Cgroups.

cgroups (abbreviated from control groups) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.

A namespace wraps a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource.

Changes to the global resource are visible to other processes that are members of the namespace but are invisible to other processes.

Cgroups = It limits how much you can use the memory, CPU, block I/O, network.

Namespaces = It limits what you can see (and therefore you can use).

Docker-ce-cli: It provide command line interface for docker engine, community edition the open-source application container engine.

Docker containers are both hardware-agnostic and platform-agnostic.

In an IT context, agnosticism refers to anything that is designed to be compatible across most common systems.

This means they can run anywhere, from your laptop to the largest cloud compute instance and everything in between - and they don't require you to use a particular language, framework or packaging system.

Docker is a product for you to build, ship and run any application as a lightweight container.

docker run …. This command comes with docker-ce-cli package.

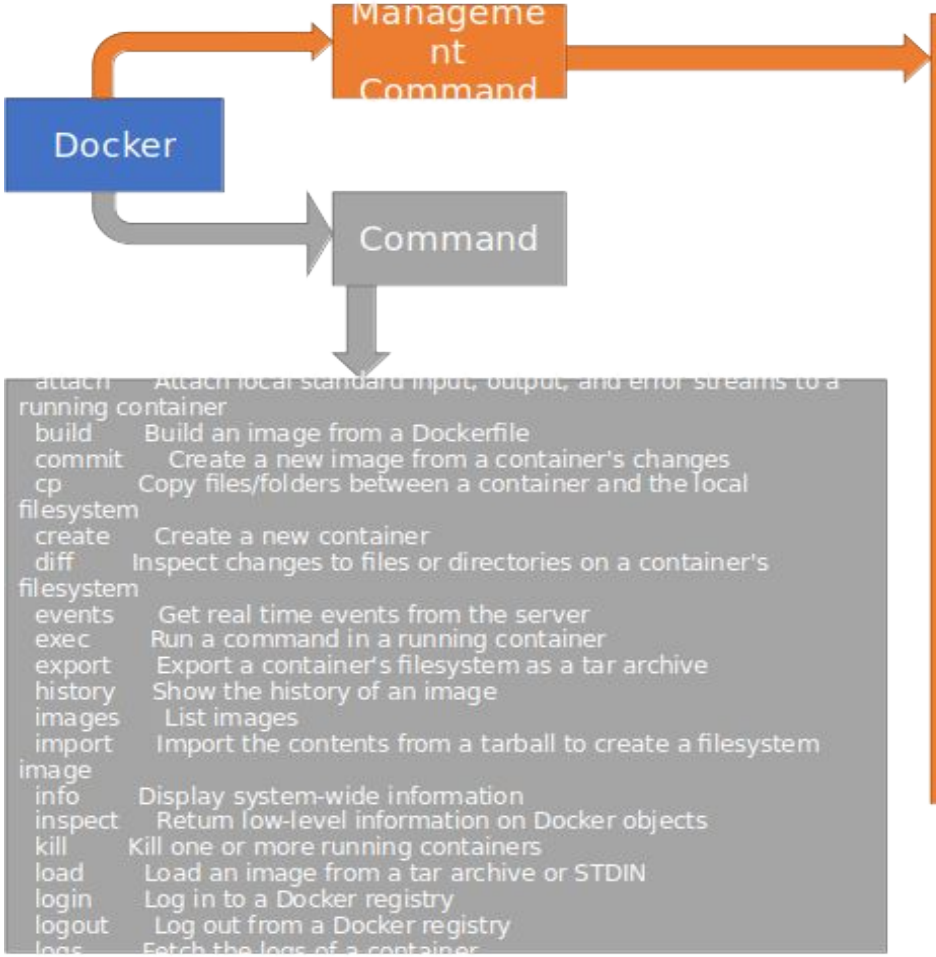
When we execute any command, docker-ce will call containerd process.

Containerd will call the runc.

Command Line:

Click to add text

docker –help

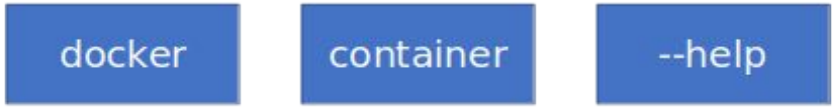

Docker
Management Command
Command
attach Attach local standard input, output, and error streams to a running container
build Build an image from a Dockerfile
commit Create a new image from a container's changes
cp Copy files/folders between a container and the local filesystem
create Create a new container
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
app* Docker App (Docker Inc., v0.9.1-beta3)
builder Manage builds
buildx* Build with BuildKit (Docker Inc., v0.6.1-docker)
config Manage Docker configs
container Manage containers
context Manage contexts
image Manage images
manifest Manage Docker image manifests and manifest lists
network Manage networks
node Manage Swarm nodes
plugin Manage plugins
scan* Docker Scan (Docker Inc., v0.8.0)
secret Manage Docker secrets
service Manage services
stack Manage Docker stacks
swarm Manage Swarm
system Manage Docker
trust Manage trust on Docker images

Command Line

```
[root@ ~]# docker container --help

Usage:  docker container COMMAND

Manage containers
Commands:
  attach      Attach local standard input, output, and error streams to a running container
  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
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  inspect     Display detailed information on one or more containers
  kill        Kill one or more running containers
  logs        Fetch the logs of a container
  ls          List containers
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  prune       Remove all stopped containers
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  run         Run a command in a new container.
```



Create a new container

create: Create a new container. (Only Create).

run: Run a command in a new container. (Create + Start).

start: Start one or more stopped containers.

stop: Stop one or more running containers.

pause: Pause all processes within one or more containers.

unpause: Unpause all processes within one or more containers.