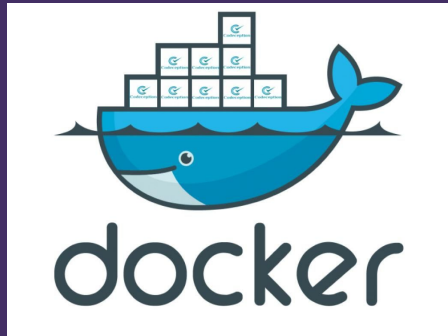


The background is a dark purple gradient. It features a variety of geometric shapes and patterns: a large dark blue circle in the center; a pink circle with diagonal stripes on the left; a blue circle with a dot pattern below it; a yellow zigzag line on the far left; a yellow triangle with a dashed outline above the center; a solid yellow triangle below it; a light blue circle with a dashed outline above the center; a solid light blue circle on the right; a pink triangle with a dashed outline above the center; a solid pink pentagon below it; a blue circle with diagonal stripes on the right; a yellow triangle with vertical stripes below the center; a solid yellow circle on the far right; and several dashed circles in light blue and pink. The word "Docker" is written in white serif font in the center of the large dark blue circle.

Docker



# Hello!

**I am Sandeep Anuragi**

I am here because I love to do docker Integration.

You can find me at [blackmagiclinux@gmail.com](mailto:blackmagiclinux@gmail.com)



1.

# Docker Network ?

Let's start with the first set of slides





“

One of the reason docker container and services are so powerful is that you can connect them together, or connect them to non-docker workloads. Docker containers and services do not event need to be aware that they are deployed on docker, or whether their peers are also docker workloads or not. Whether you docker hosts run linux, windows, or mix of the two, you can use docker to manage them in a platform-agnostic way.



2.

Network Drivers ?



“

Docker's networking subsystem is pluggable, using drivers. Several drivers exist by default, and provide core networking functionality:

# Understanding containers



- Bridge
- Host
- Macvlan
- None:
- Network Plugins:

“

## Bridge :

The default network driver. If you don't specify a driver, this is the type of network you are creating.

**Bridge networks are usually used when your applications run in standalone containers that need to communicate.**



“

## Host :

For standalone containers, remove network isolation between the container and the docker host, and use the host's networking directly.

“

## Overlay :

Overlay networks connect multiple docker daemon together and enable swarm services to communicate with each other. You can also use overlay networks to facilitate communication between a swarm service and a standalone containers, or between two standalone containers on different docker daemon.

# Who is Docker for?



## Macvlan:

Macvlan networks allow you to assign a MAC address to a container, making it appear as a physical device on your network. The Docker daemon routes traffic to containers by their MAC addresses. Using the macvlan driver is sometimes the best choice when dealing with legacy applications that expect to be directly connected to the physical network, rather than routed through the Docker host network stack. See [Macvlan networks](#).

# Who is Docker for?

## None:

For the container, disable all networking, usually used in conjunction with a custom network driver. none is not available for swarm services.

# Who is Docker for?

## Network Plugins:

You can install and use third-party network plugins with docker. These plugins are available from docker hub or from third-party vendors.

More info on how to use docker visit at <http://trainingbasket.in>



**Thanks For Join Us**



blackmagiclinux@gmail.com



Are you Ready to face  
of challenges



Comment and Survey  
fo this course