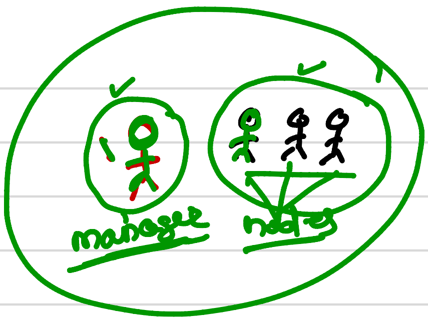
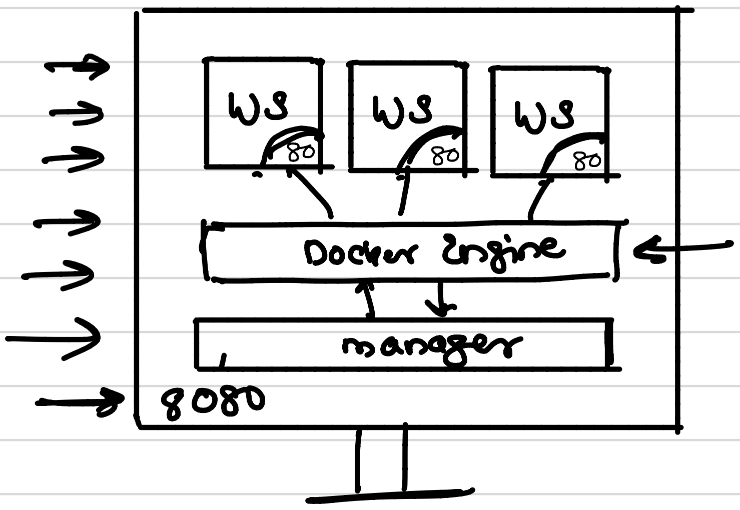
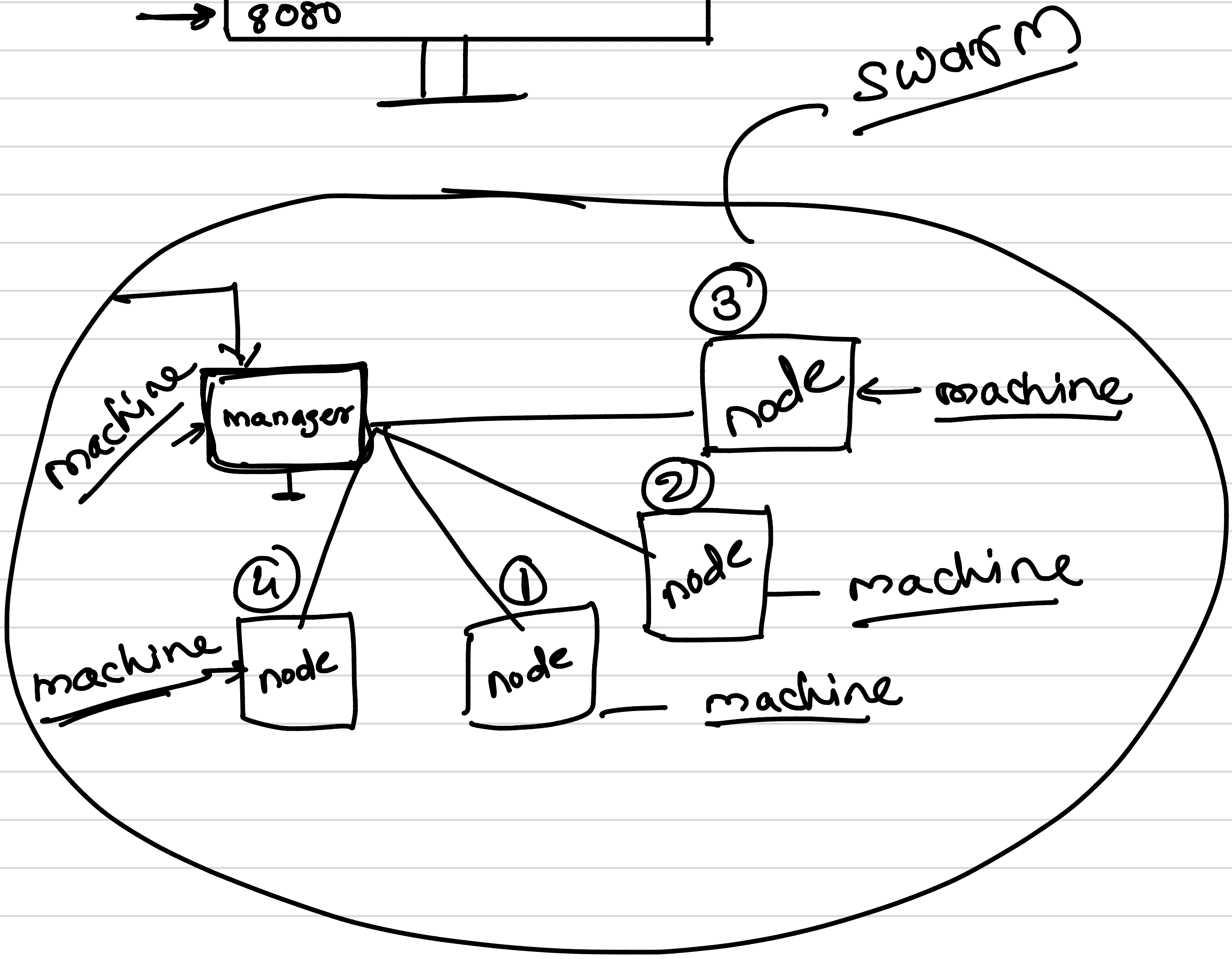
Orchestration

cluster of containers
(Group)



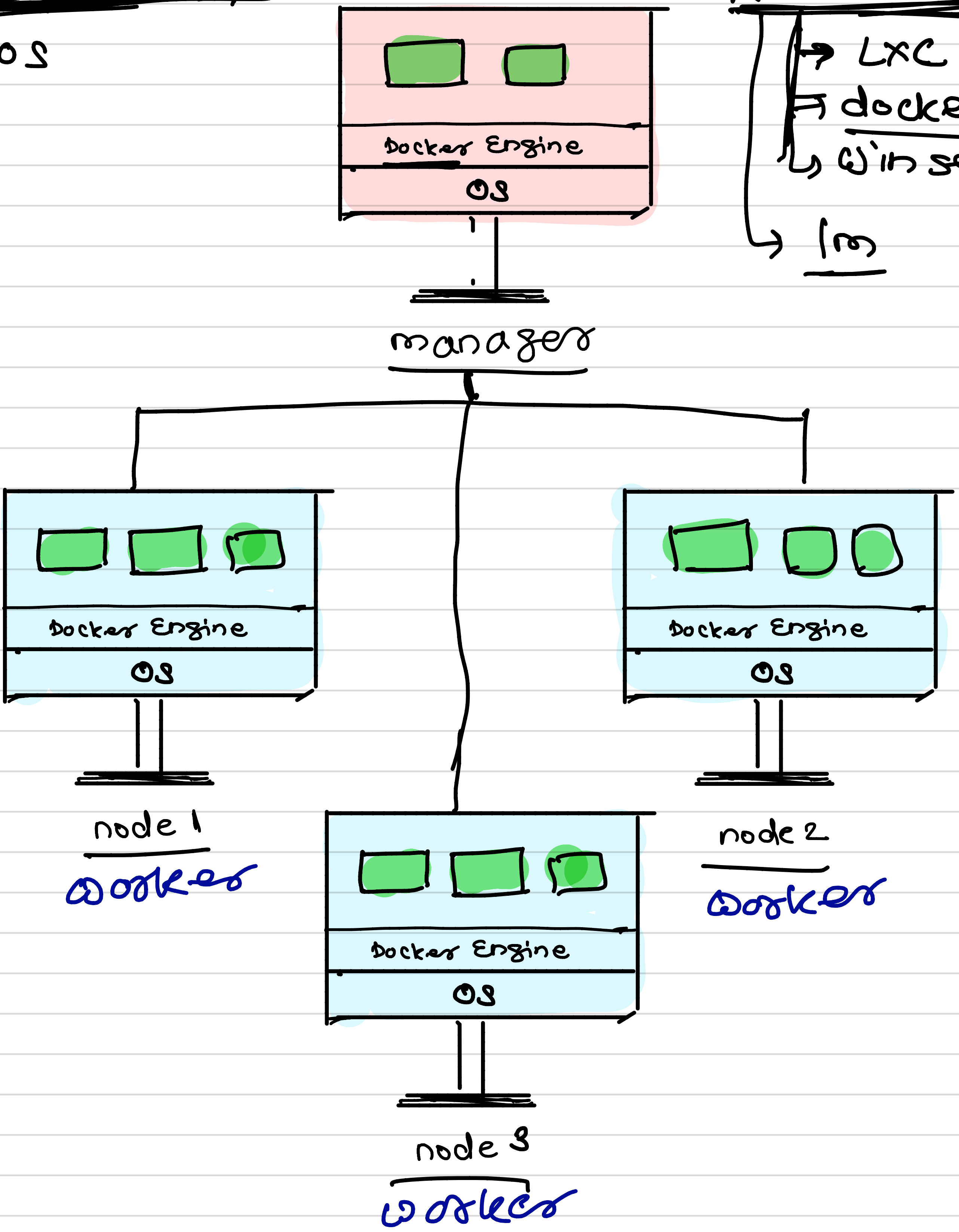




Docker swarm -

1000s

(Google)
Kubernetes
- → LXC
- → docker
- → win server
- → lm -



Containers

- Containers are all about portable software
- They are a technology that allows you to run software on a variety of systems, including a developer's laptop, all the way to a production system
- This speeds up deployment, simplifies automation, and ensures your code can run consistently in production, as well as everywhere else
- Like virtual machines, containers wrap your software in a standardized environment that allows it to run consistently on varied machines
- But containers are smaller, use fewer resources, and are easier to automate than virtual machines

What are Containers Used For ?

- Containers are great at accomplishing things like:
  - Software Portability – Running software consistently on different machines
  - Isolation – Keeping individual pieces of software separate from one another
  - Scaling – Increasing or decreasing resources allocated to software as needed
  - Automation – Automating processes to save time and money
  - Efficient Resource Usage – Containers use resources efficiently (which saves money)

Advantages of Containers

- The isolation and portability of VMs
- More lightweight then VMs - Less resource usage
- Faster than VMs – Containers can start up in seconds, not minutes
- Smaller than VMs – Container images can be measured in megabytes, not gigabytes
- All of these add up to faster and simpler automation

Limitations of Containers

- Less flexibility than VMs – You can’t run a Windows container on a Linux machine (yet)
- Introduces new challenges around orchestration and automation

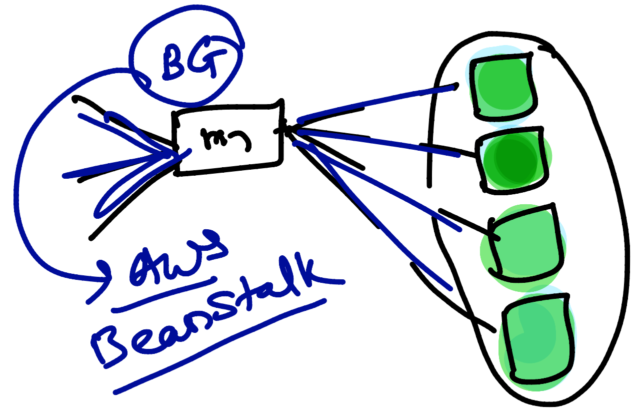
Orchestration

- Container Orchestration simply refers to processes used to manage containers and to automate the management of containers
- For example:
  - I want to start up a set of five containers in production.
  - I could spin up each container manually.
  - Or, I could tell an orchestration tool like Kubernetes that I want five containers, and let the tool do it

$\downarrow$

docker swarm

Zero-Downtime Deployments



- In the old days, deployments went like this:
  - Take the server down for maintenance (it is unavailable to customers).
  - Perform the deployment.
  - Bring the server back up.
- A zero-downtime deployment (with containers) goes like this:
  - Spin up containers running the new code.
  - When they are fully up, direct user traffic to the new containers.
  - Remove the old containers running the old code. No downtime for users

Container Orchestration Solutions

- ✓ Docker Swarm
- Marathon
- Nomad
- ✓ Kubernetes

What are Microservices?

- Microservices are a type of application architecture that involves splitting the application into a series of small, independent services
- Microservices can be built, modified, and scaled separately, with relatively little impact on one another

Containers and Microservices

- Containers excel when it comes to managing a large number of small, independent workloads
- Containers and orchestration make it easier to manage and automate the process of deploying, scaling, and connecting lots of microservice instances
- For example, I may have one microservice that needs additional resources. With containers, all I need to do is create more containers for that service to handle the load. With orchestration, that can even be done automatically and in real time

What is Automated Scaling?

- Automated Scaling refers to automatically provisioning resources in response to real-time data metrics
- Without automated scaling, you must provision enough resources to cover your peak resource needs at all times
- With automated scaling, you can automatically detect (or even predict) increase in usage. The automated system creates new servers to handle the peak usage time, then removes those servers when usage returns to normal levels

Containers and Automated Scaling

- Automated scaling depends on the ability to spin up new instances quickly and efficiently
- Since containers are small and can start up quickly, they are ideal for this purpose. This means that if the system detects an increase in usage, it can spin up new containers in a few seconds
- This increases stability and reduces cost! Your users see less downtime due to high loads, and you don’t use (and pay for) resources unnecessarily