Module Nine

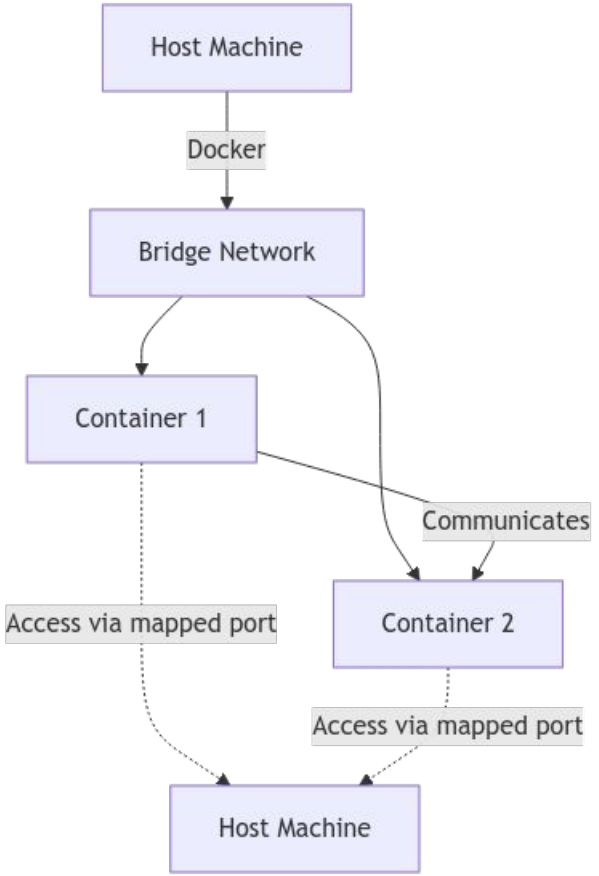
Docker Networking

Docker Network Types

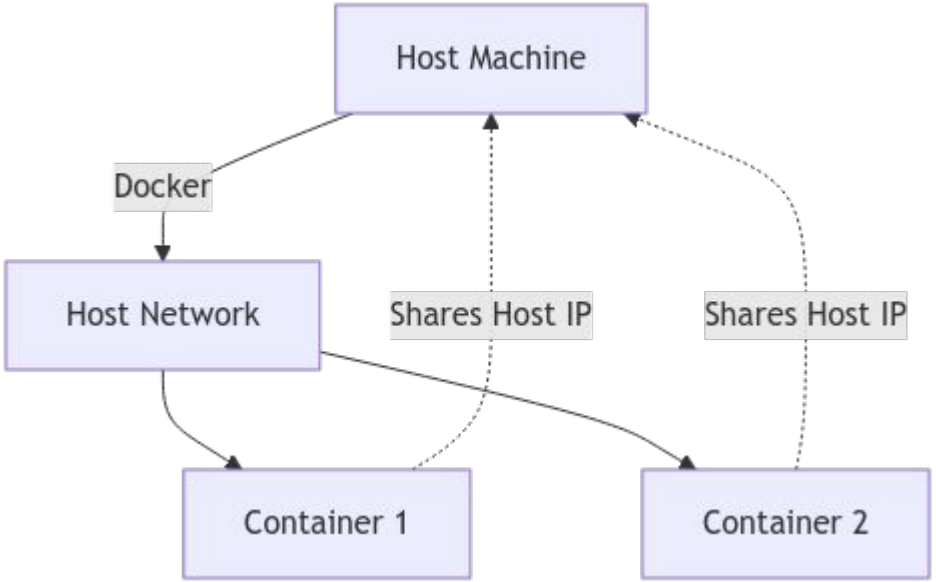
Networking Options

Type	Description	Usage
Bridge	Default network	Best for development and local testing
Host	Shares host network stack	Discards exposed ports, all ports declared appear on host
IPvlan	Each container has a unique MAC address	Containers appear as physical devices on the network, including IP addressing
MACvlan	Each container has a unique IP address	Provides control over L2 VLAN tagging and IPvlan L3 routing
None	No network connectivity	Useful for security isolation of containers
Overlay	Distributed network across multiple docker hosts	Primarily used for Swarm Mode, but can also be used with standalone containers
Airgapped	Fine grained control of ingress/egress for containers (note: not a network driver)	Useful for secure applications; part of Docker Business

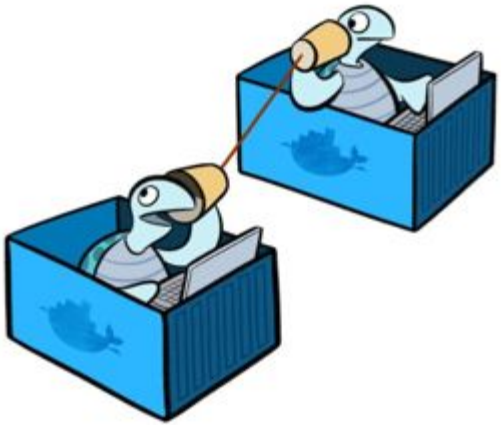
Bridge Networking

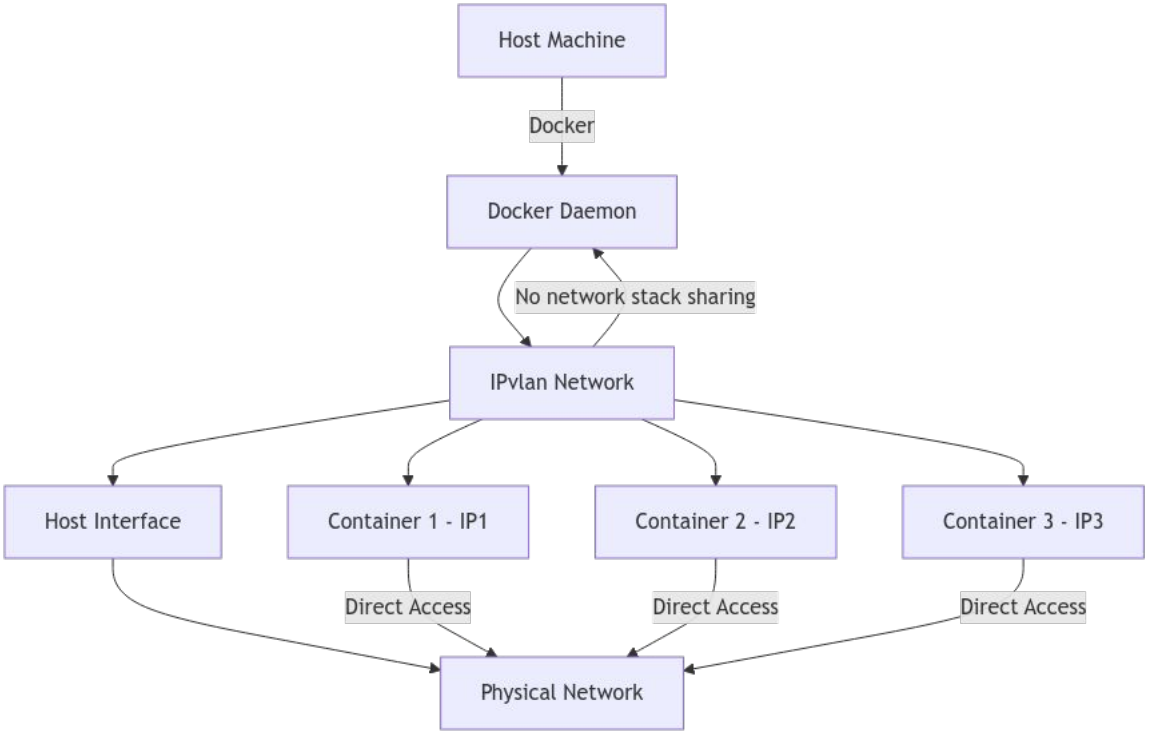


Host Networking

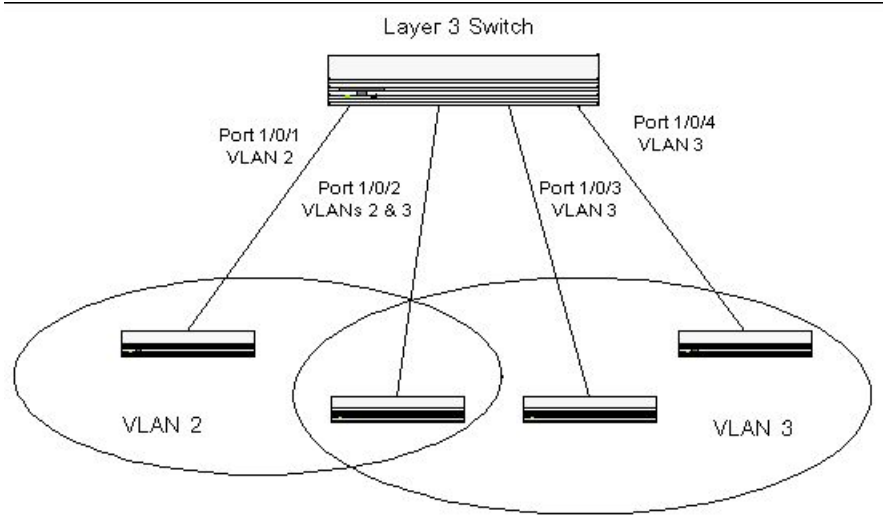


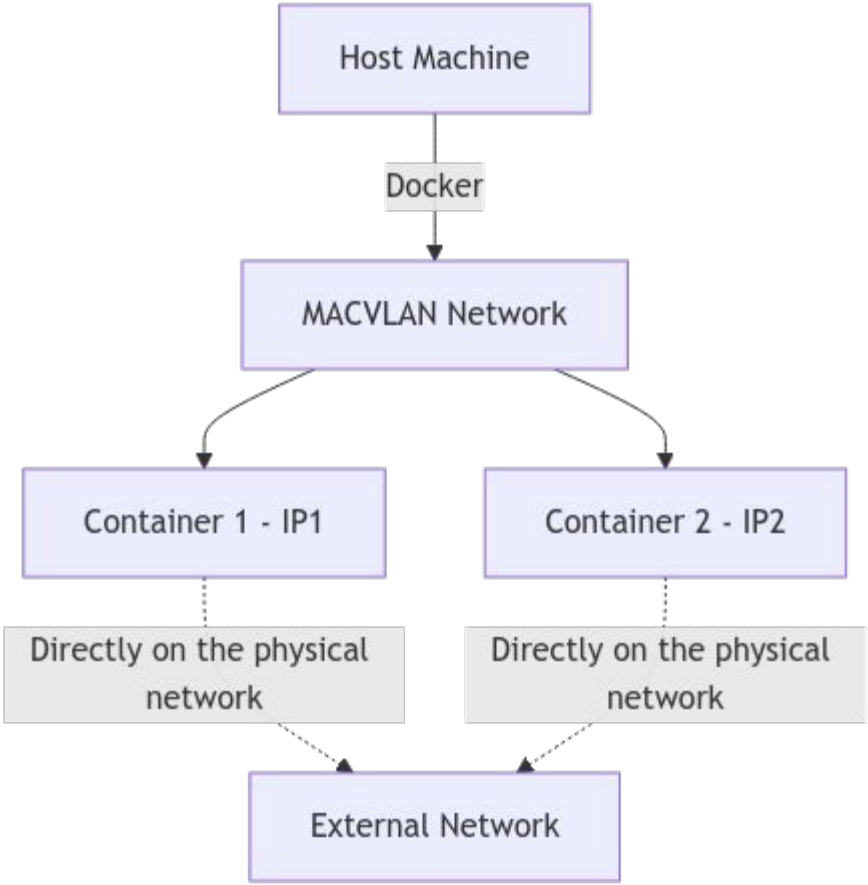
IPVLAN Networking





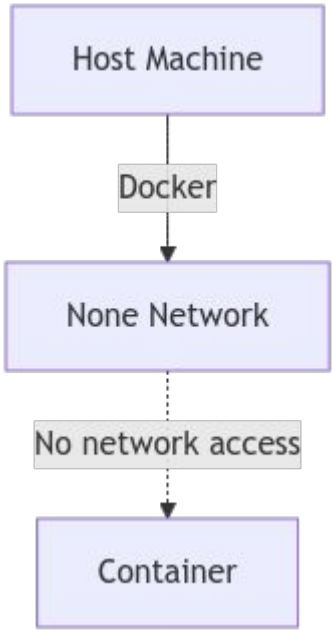
MACVLAN Networking



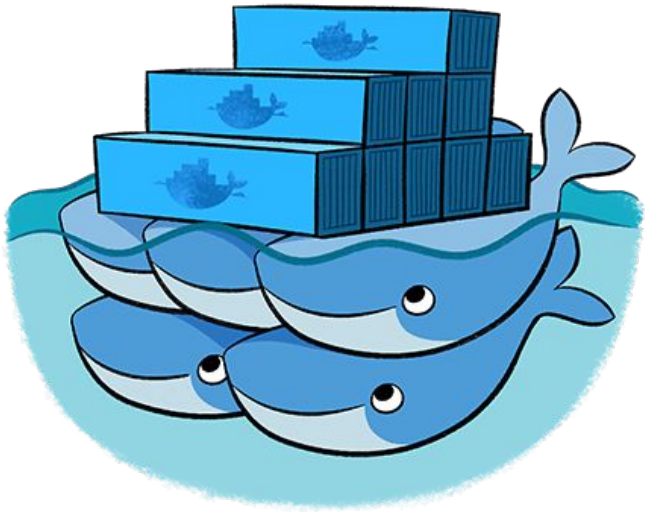


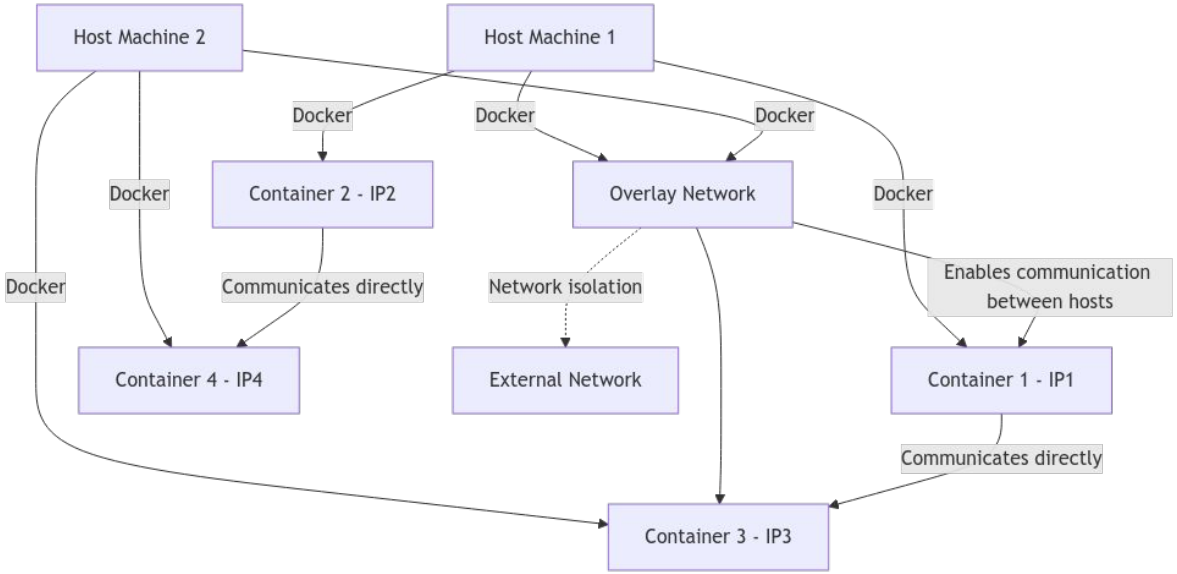
None Networking

NONE

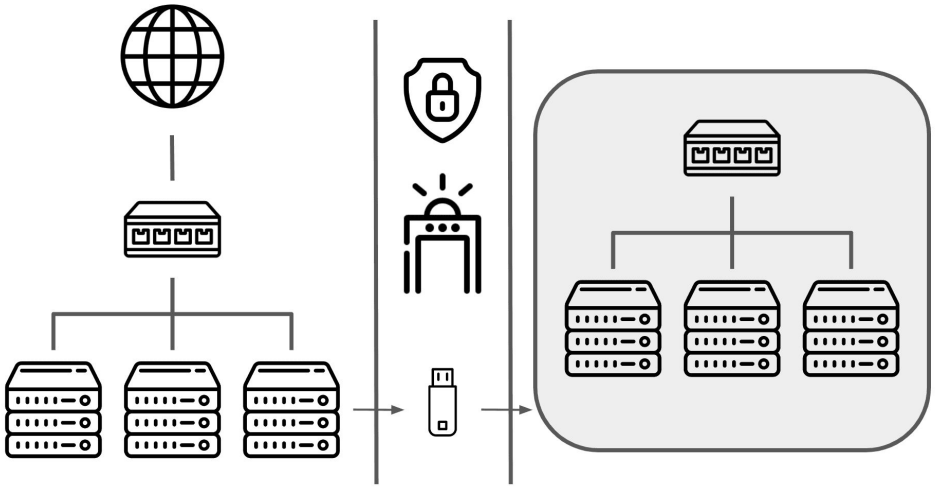


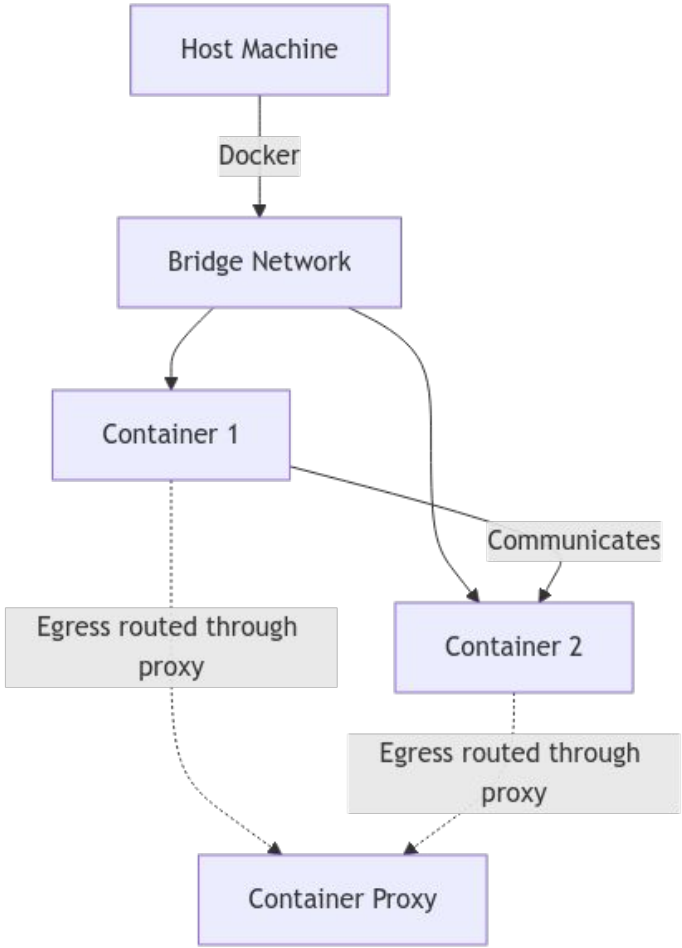
Overlay Network





AirGapped Containers*





* Not a networking driver; proxy feature in docker business

Best Practices

Network Ports

- **Minimize network exposure**
  - **Only expose necessary ports with** -p
- **Specify TCP or UDP as needed**
  - docker run -p 8080:8080 <image_name>
  - docker run -p 8080:8080/udp <image_name>
  - docker run -p 8080:8080 -p 8080:8080/udp <image_name>
- **Avoid** --publish-all (-P)
  - **This exposes all published ports to random host ports**
- **Use Specific IP Binding**
  - -p [IP]:[HostPort]:[ContainerPort]
  - **Default is 0.0.0.0**

Container Communications

- Docker provides internal name resolution
  - Prefer inter-container communication versus exposed ports
- Use user-created bridge network
  - Creates isolation from default network
  - Can use for microsegmentation
- Use MACVLAN for network level isolation
  - Treats containers as separate network devices
- Use Airgapped containers to limit container egress
  - Can use separate proxy from system
  - Can limit to defined subnets
  - Can entirely block egress

Security Considerations

- Restrict network privileges
  - Containers should run with minimal network access
  - Leverage both Docker and OS/Infrastructure tools
- Enable TLS for remote contexts
  - Do not allow access to Docker socket
- Use Docker enhanced container isolation (ECI)
  - Provides VM and container hardening
- Isolate container networks
  - Projects should have separate networks

Questions and Answers