Container Orchestration

Container orchestration is all about managing the lifecycles of containers, especially in large, complex and dynamic environments. Dev/Ops teams use container orchestration to manage and automate many tasks like:

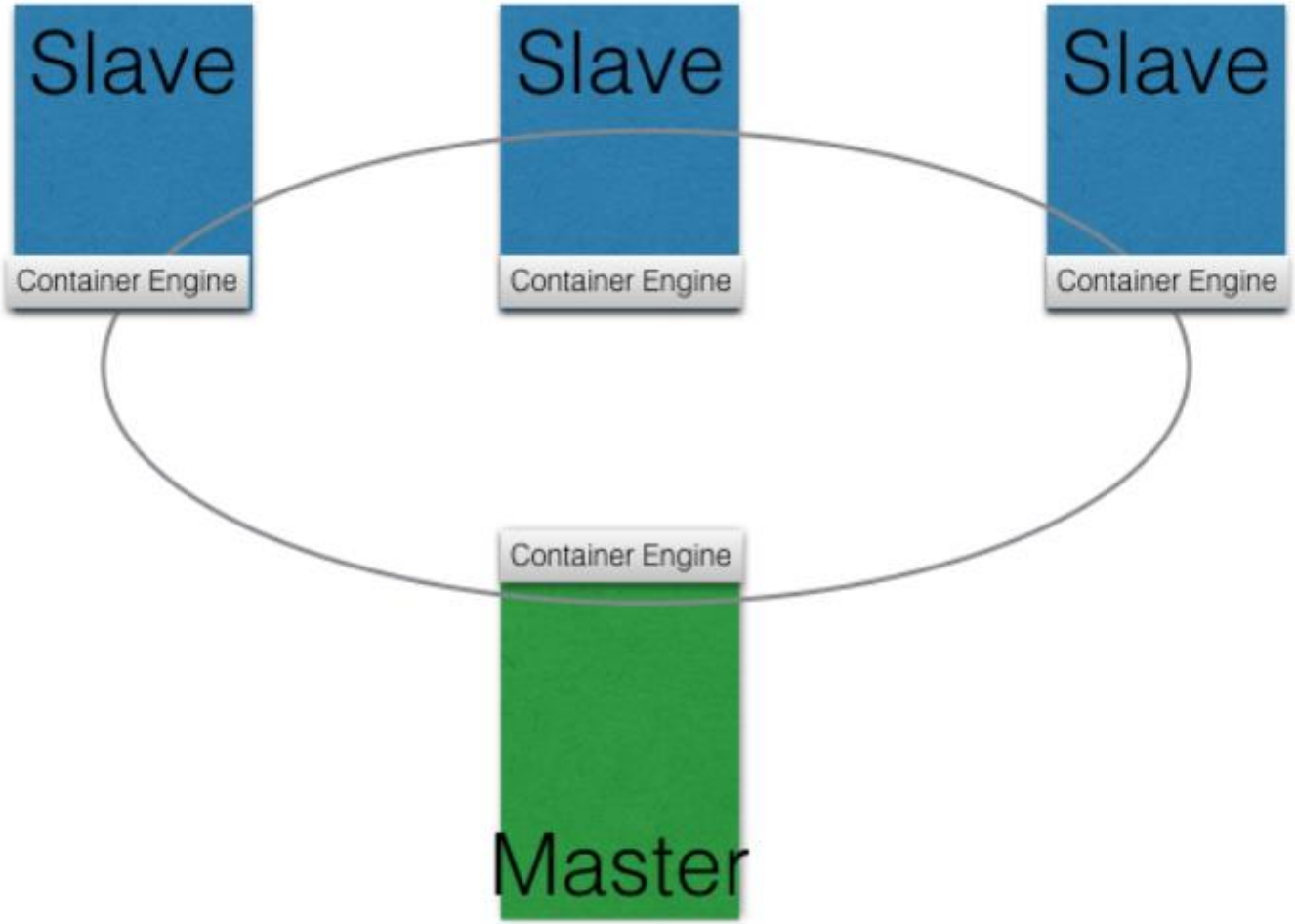
- Provisioning and deployment of containers
- Redundancy and availability of containers
- Scaling up or Scaling down containers.
- Shifting containers from one host to another if host goes down
- Resources allocation to containers
- Expose container with the outside world
- Load balancing and service discovery of containers
- Health check and monitoring of containers and hosts

Container Orchestrators

- Container orchestrators
- Kubernetes
- Docker Swarm
- Mesos Marathon
- Amazon ECS . . . etc

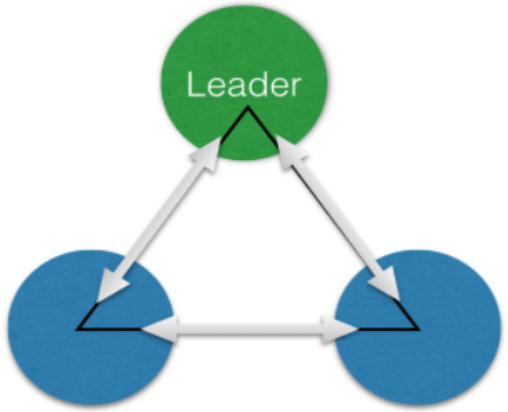
Container Orchestrator Features

- Container Deployment / Provisioning
- Fault Tolerance
- Autoscaling
- Service Discovery


Slave
Slave
Slave
Container Engine
Container Engine
Container Engine
Container Engine
Master

Key/Value Store

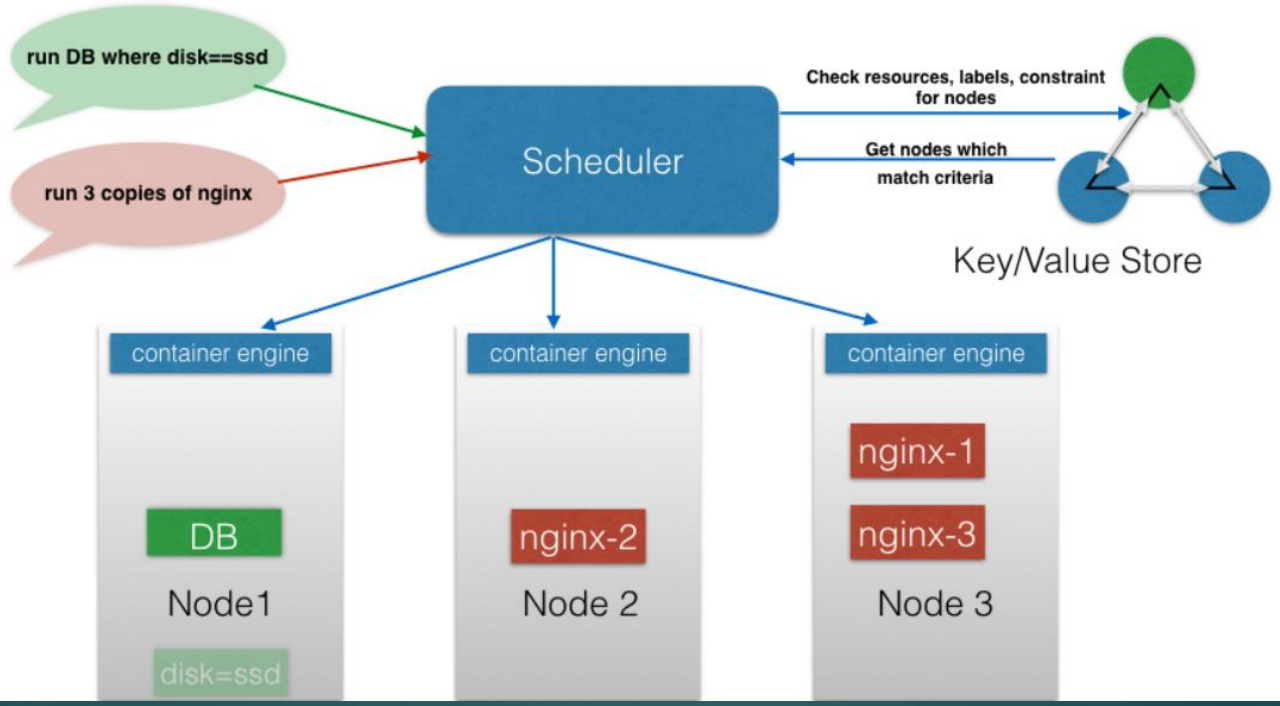
Single source of truth



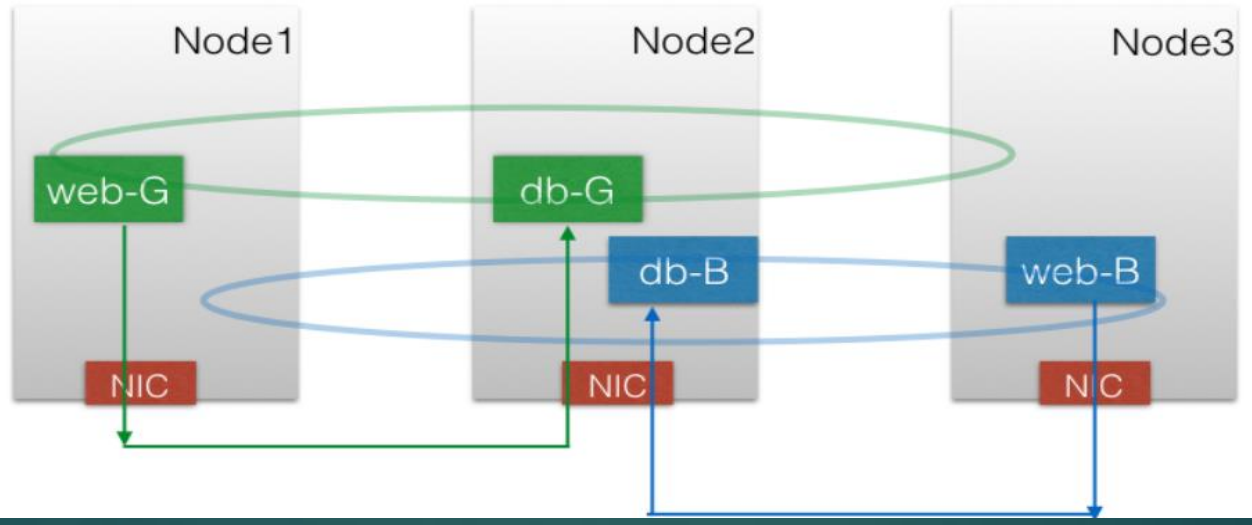
Key/Value Store

- Accessed over HTTP
- Can store any information in key/value format, like :-
  - configuration
  - subnet details
  - Node specific information.
- One can watch on key changes and trigger action based on changes
- Examples
  - etcd

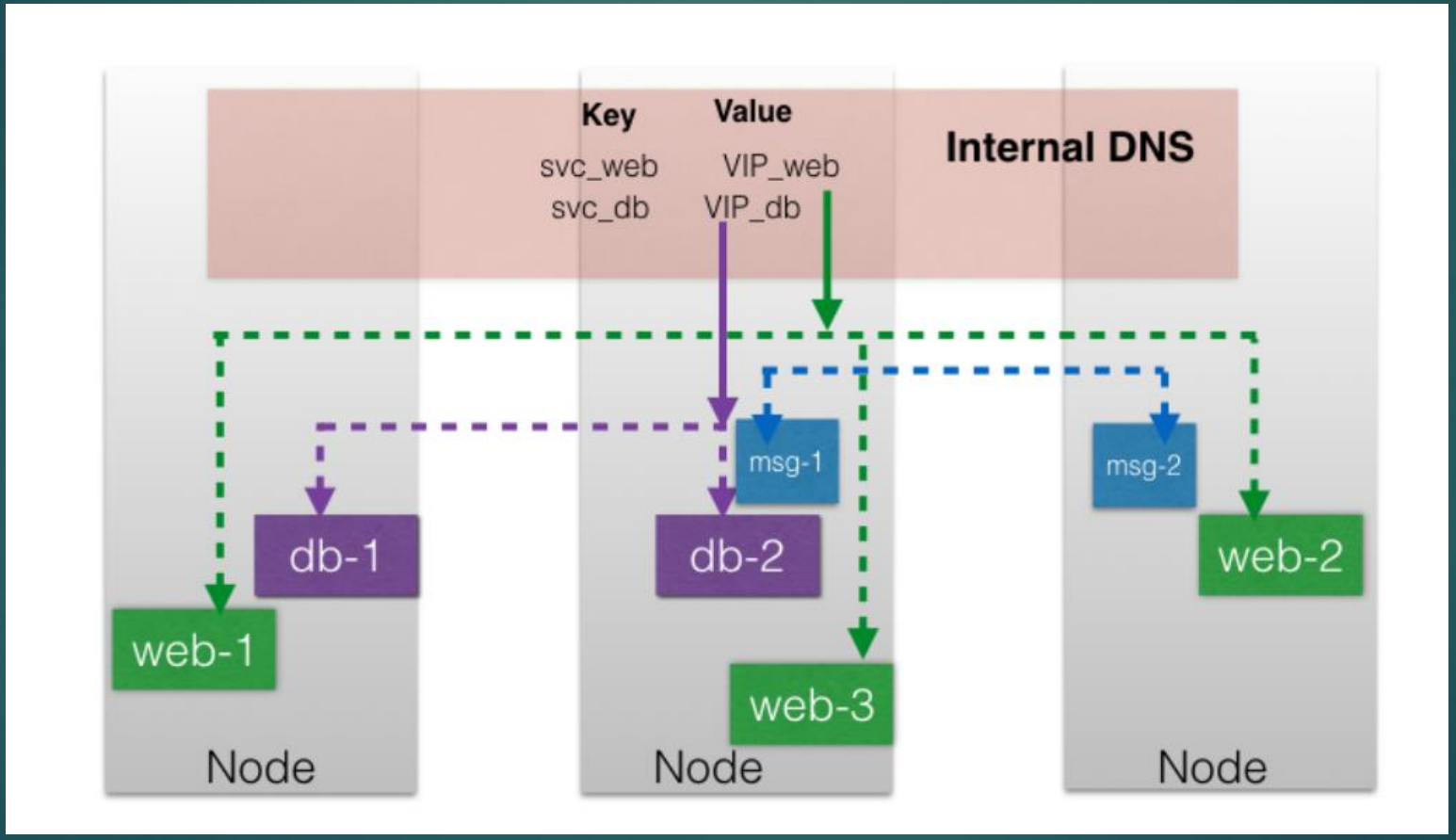
Scheduler


run DB where disk==ssd
run 3 copies of nginx
Scheduler
Check resources, labels, constraint for nodes
Get nodes which match criteria
Key/Value Store
container engine
container engine
container engine
DB
Node1
disk=ssd
nginx-2
Node 2
nginx-1
nginx-3
Node 3

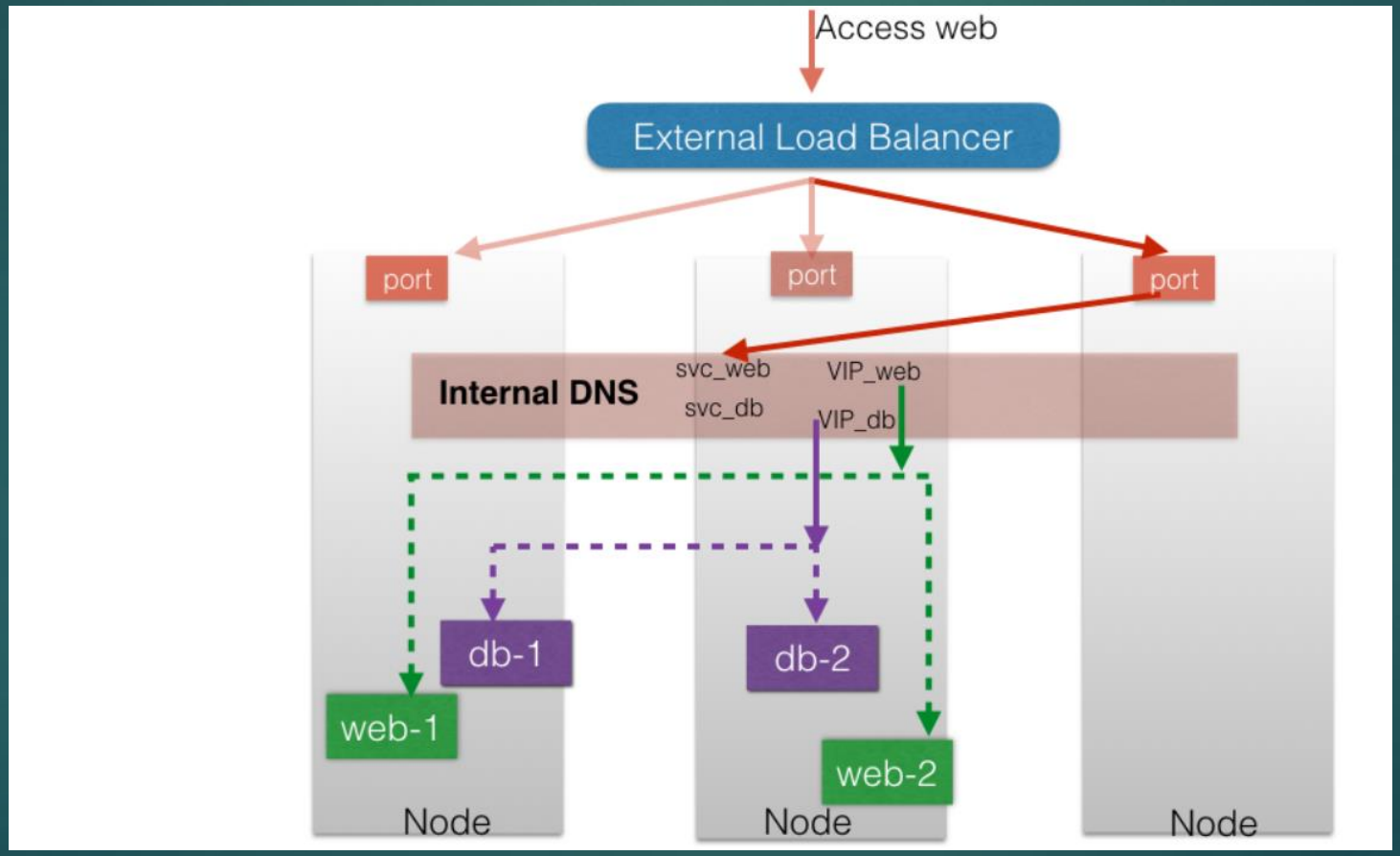
Network for containers of different nodes to talk to each other



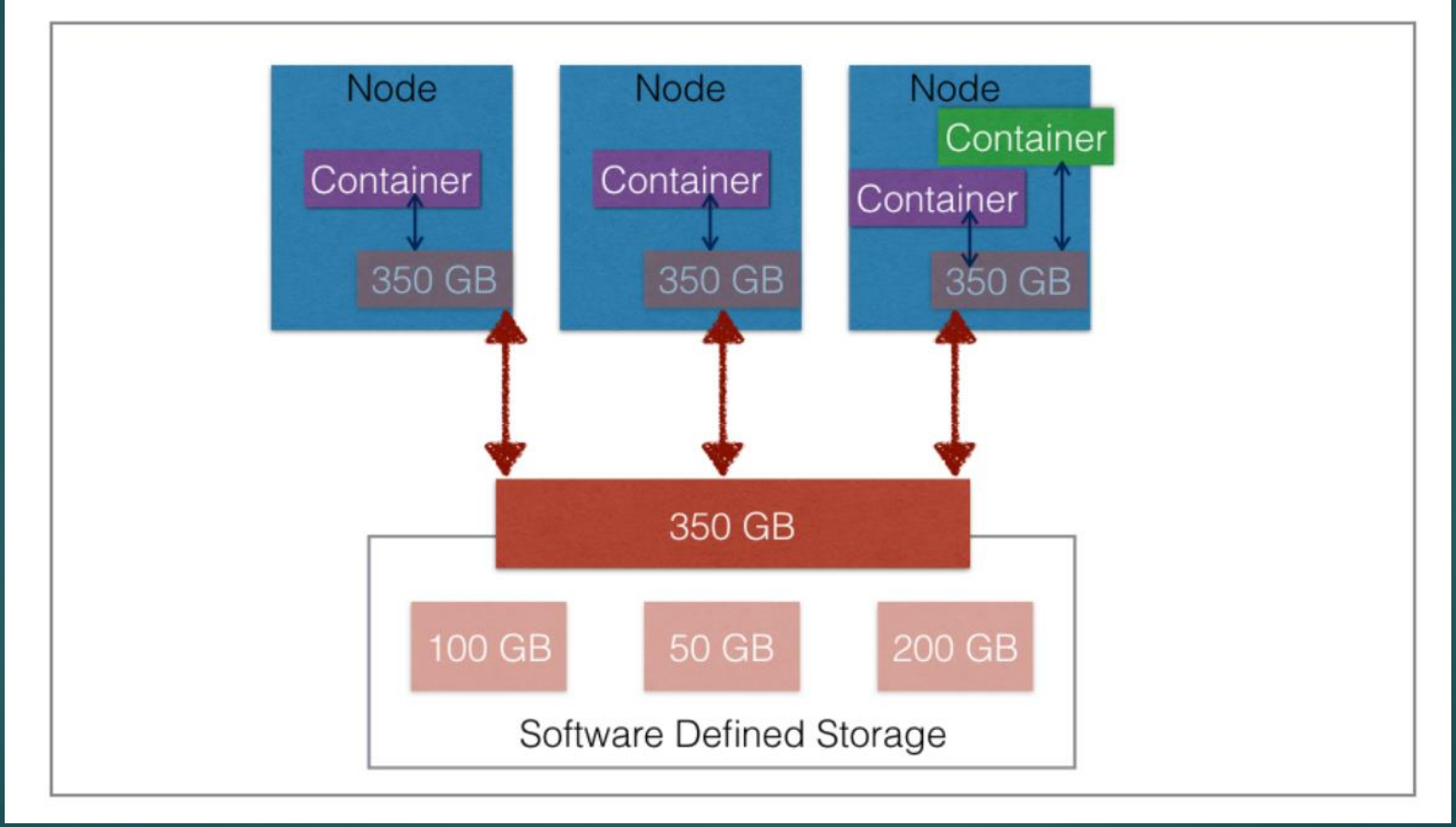
Service Discovery



Access from External World



Access to External Storage



THANK YOU