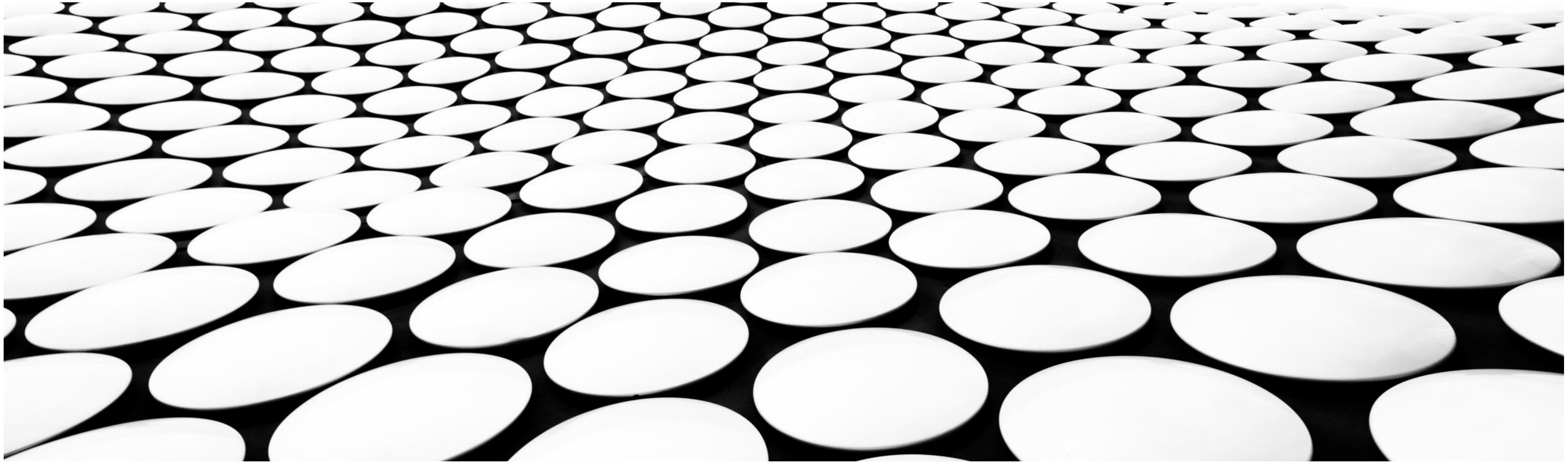
DOCKER AND KUBERNETES

DHANANJAYAN

20TH JULY – 24TH JULY 2020



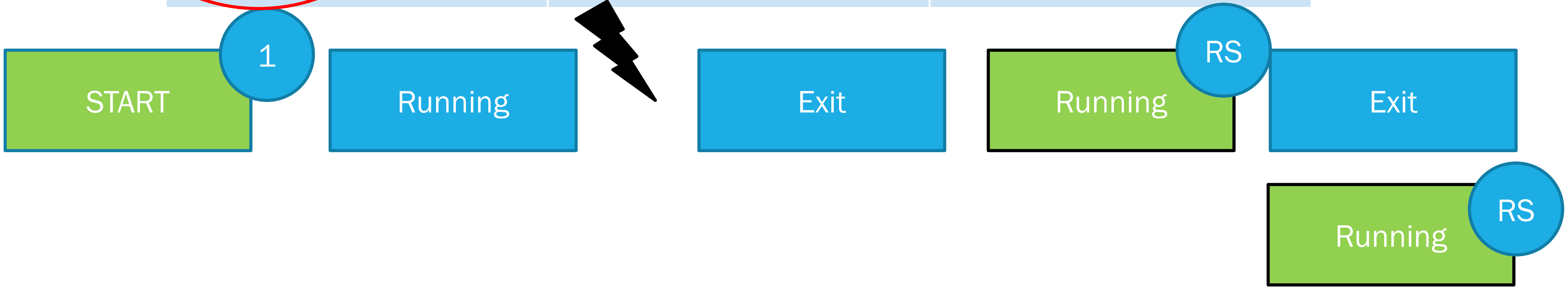
DAY 4

- Release Automation (Docker)
- Restart Policies (Docker)
- GIT Automation *
- Pod Architecture
- Architecture of Kubernetes
- Install Kubernetes
- Services Deployment
- Service management techniques

RESTART POLICY

EXPLCIT STOP / EXPLICIT START	IMPLICIT STOP
Error → Exit → Resolve Error	*Resource Utilization* Logical Exceptions *Attach → exit*

No	Always	On-failure
Never restarts automatically 1 (Start Actual)	Always restarts on implicit stop On Explicit Stop , The container exits.	EXIT CODE Lock Restart Count # Container exited in case if exceeded threshold restart count



RESTART ON FAILURE

1 SCRIPT with return of Exit Code

2 Container ++ Script

3 Container To Image Commit -Add Boot straper Script

4 Container from Image RESTART POLICY On-failure:3

5 Verify Logs of the Container

RELEASE AUTOMATION

Architecture

Code

Dockerfile

Docker Image

Container Network

Logs

Health ?

Release one Layer

Customer Env

Buys→ Tenancy → Host Containers
Export Images to REPOSITORY → Build Containers (CANNOT BE CLI) - → Setup / Uninstall
Environments
MSI Installer → Windows ? → Not Fit for Containers
Configuration Language – Automation File for Setting up Services (YAML) → docker-compose

YAML

YAML FILE (DOCKER) (.YAML, .YML)
UTILITY

- Docker keywords
- Key: value (all keys are values are case sensitive), “Any Data Type”
- JSON → { public, private} → “ | ”
- Collection → [ ] → “-”
- Indendation
- Version :3
- Services:

USE CASE : YAML IMPLEMENTATION

```
Version: '3'
Network:
Volume:
Services:
  database-container:
      image: mysql:5.7
      environment:
      - MYSQL_ROOT_PASSWORD=admin
      - MYSQL_DATABASE=demo
      - MYSQL_USER=scott
   webserver-container:
        image: httpd
        ports:
        - 8000:80
        - 8001: 00
```

```
#docker-compose up  -d
# docker-compose down
```

Property	CLI /Reference
Setup / Release	YAML – docker-compose/TF/Python
Test/Troubleshoot/maintain	CLI – docker
Environment/Infrastructure (CRUD)	Machine – docker-machine

CLOUD NATIVE ENVIRONMENTS

CLOUD/CUSTOM ENVIRONMENT INFRASTRUCTURE
IDENTITY , POLICY , USERMANAGEMENT, NOTIFICATION, BIG DATA

DISTRIBUTED COMPUTING ENVIRONMENT (MULTI TENANCY)
KUBERNETES – CLUSTER – SWARM - MARATHON

Monitoring, Troubleshooting and Logging → Prometheus/Kibana/ELK/Heapster

Container Networking Interface | Container Storage Interface

DEPLOY APPLICATIONS – CONTAINERS - OPEN CONTAINER RUNTIME (docker, Maesos,oci*)

DEVELOP APPLICATIONS – MICROSERVICES (recommended)

K8S

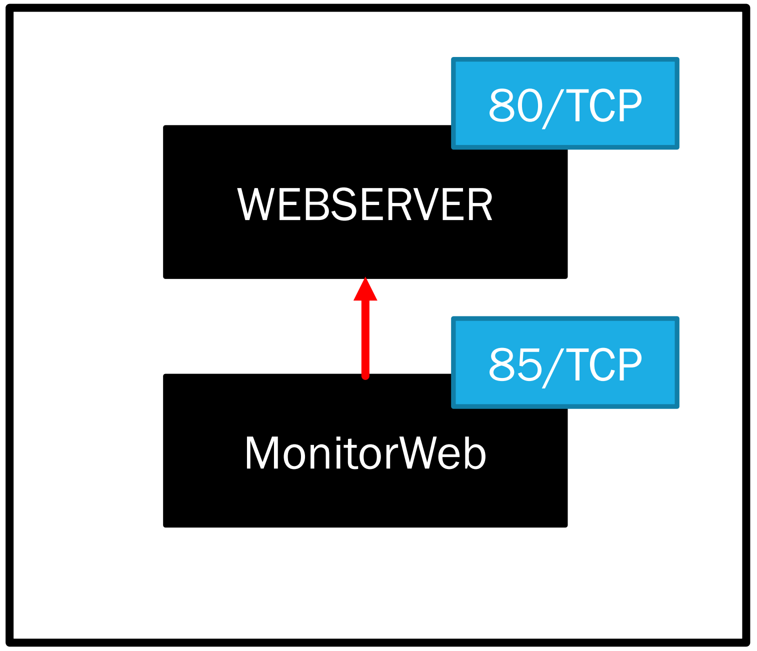
- **ERADICATING SINGLE POINT OF FAILURE**
  - HORIZONTAL SCALING (INFRA DEPENDENCIES)
  - VERTICAL SCALING (CONTAINER SCALING)
- Ha of INFRA & SERVICES
- RUNNING (GENERIC TIMES)
- **FASTTRACK ROLLOUTS → Ha of Infra/Services**
- PAY PER USAGE /DYNAMIC (Automation)
- **“SCALE” SERVICES**
  - **ORCHESTRATE** -- “WHATEVER IS DEPLOYED SHOULD BE RUNNING AS IT IS – ALL TIMES/SCALED ASA” (Ha)

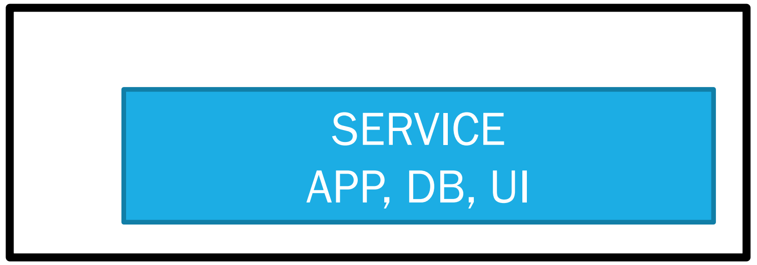
CLUSTER = MACHINES (ORCH+WORKER)

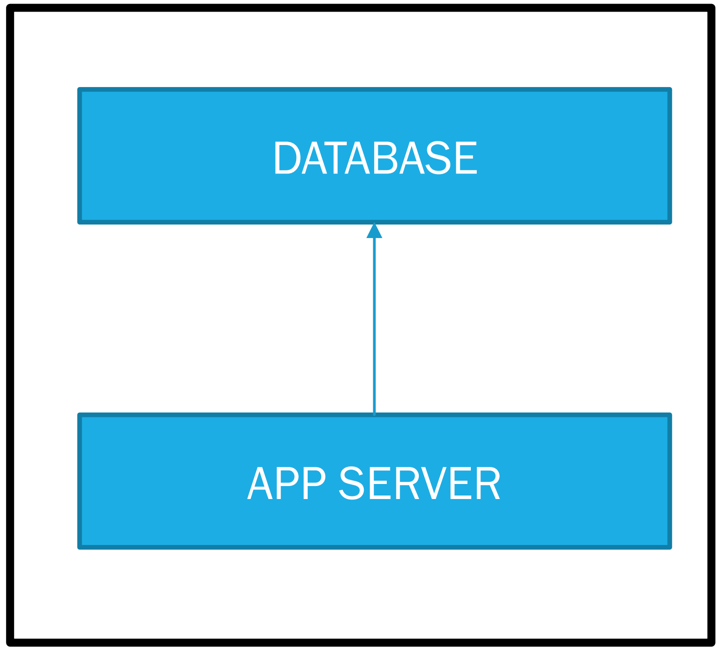
- # Machines (Network of Machines)
- Dedicated Machine (host) for Orchestration
  - Copies of Orchestration (Mirror Orchestrators)
  - Run services (orchestration services, pods) → dockerd
- Dedicated machines for Services (Worker)
  - Run Services (as PODS) → dockerd

- Services Gateway (end to end) as pod
- Services Registry (SPOT) of Services as pod
- Service Discovery (Where, What, How) as pod

POD = UNIT OF ABSTRACTION → SCALE /EXPOSING SERVICES







IMPLEMENTATION DIFFERENCES

DOCKER	K8S
Independent objects are Containers	Independent objects are pods
Container expose services – static/dynamic	Preferred Port – Dynamic port
Properties are in JSON	Key Value Pair Format
Automation is in YAML	YAML/Python/CNI /Perl /TF
Restart Policy = No (Never)	Self Heal Application (Pods/Container – Restart Policy → ALWAYS)
Choice of exposure for Static Ports	No Choice of Exposure → Dynamic Ports only
Container runtime → dockerd	Any Container runtime.
Docker, docker-machine, docker-compose	Kubectl,kubeadm

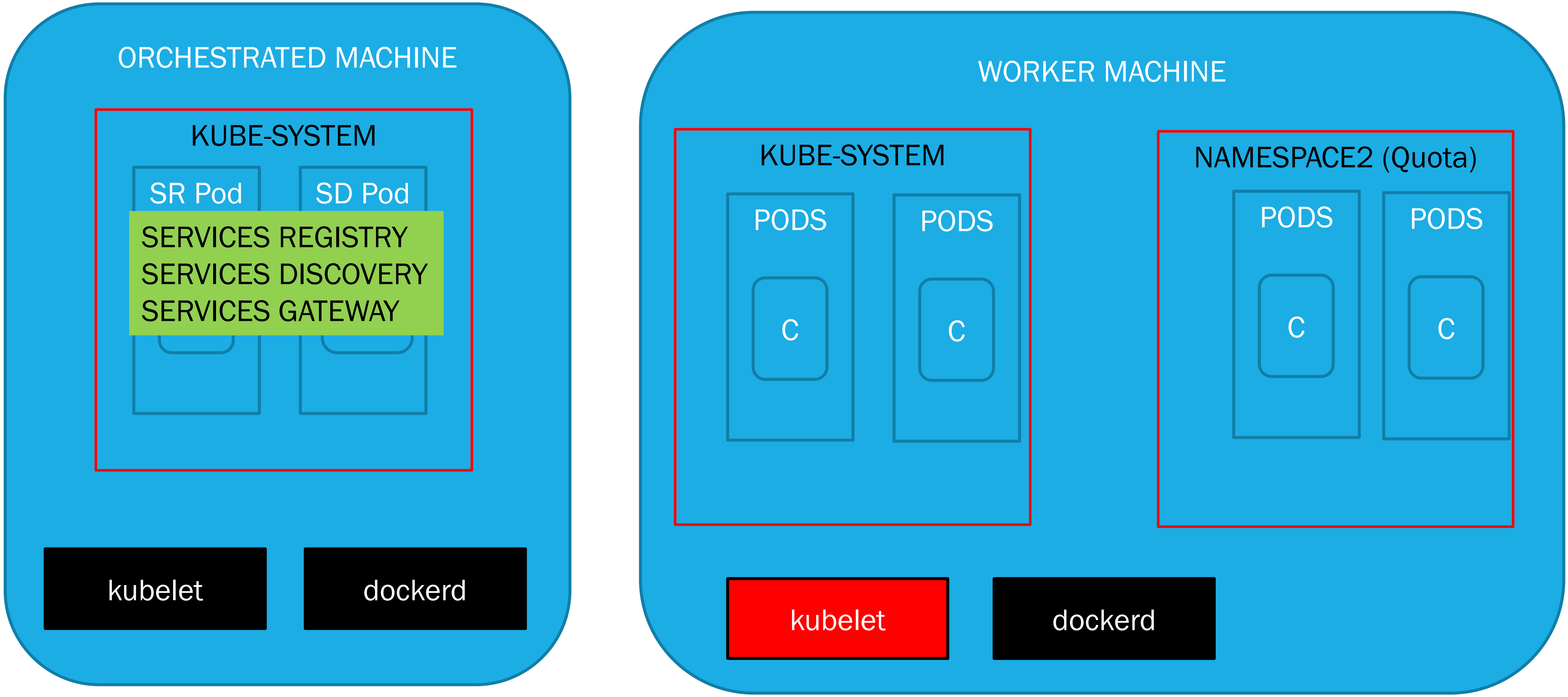
MOSAIC ORCHESTRATIONS

Service Registry	Service Discovery	Service Gateway
Etcd – No SQL , key value pair	API Server – Trace where service API	Core DNS (Kube DNS)
Zoo Keeper	Stack – Where service is running	Jetty, Vertex
Kong	Zookeeper Discovery/Hue	Nginx plus
Consul IO	Consul IO	**Oracle Traffic Director**
Eureka	Ribbon	Nginx , voyager

Kubectl
CLI

MINIKUBE
LINUX
+DOCKER
+KUBERNETES

ARCHITECTURE



KUBERNETES ROLES

Define Configuration of cluster – Node Pools
Define configuration of Node (machine)
Decide SR, SD, SG (Tools and Softwares)
Define Policy, Quota

Administration
(Cloud Admin – OCI)
Defining Infrastructure

Node (Machine)
Namespaces (Quota)
Pods (Services)
Scale Services
Expose Services
Policy for Services
Monitor Services

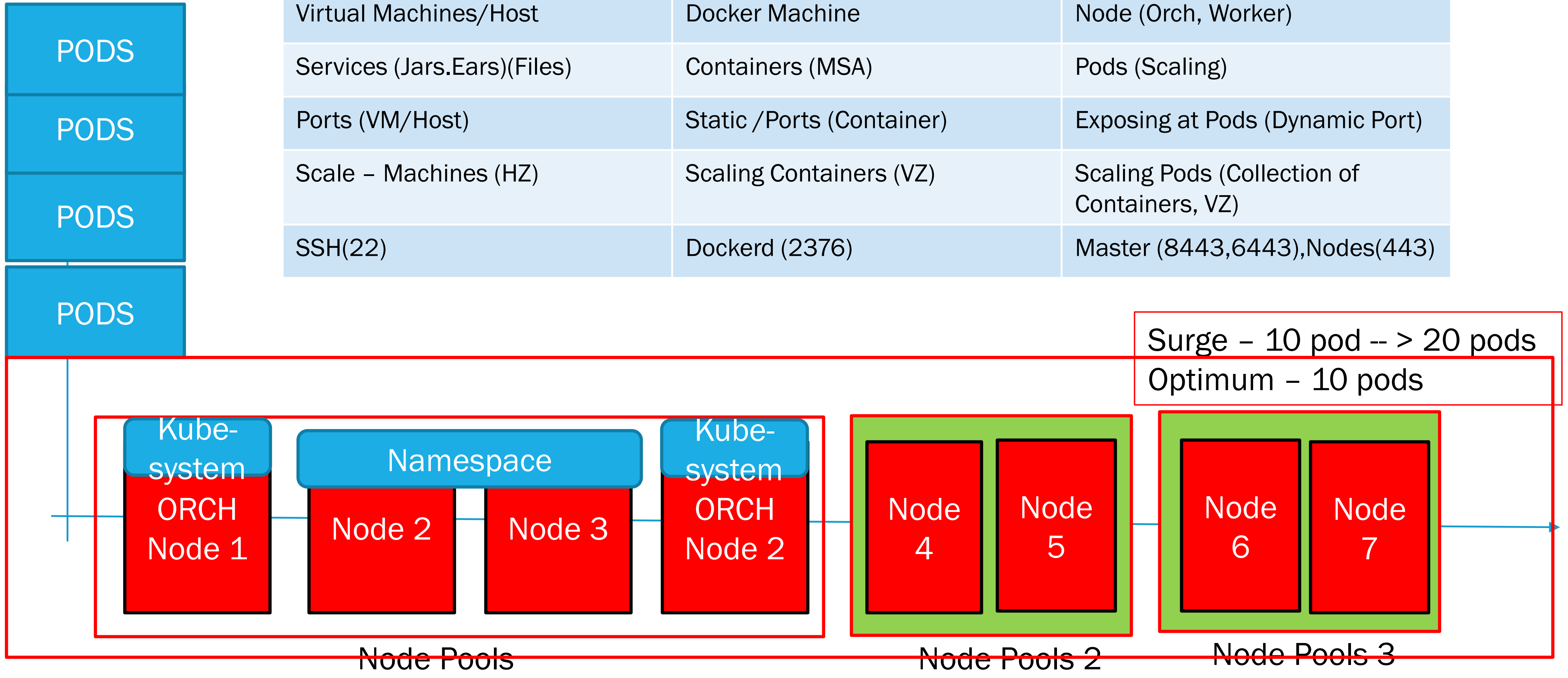
Operations (BRM)
Deploying Services

Code Arch
Dockerfile
Docker Images
Containers
Health , Logs
GIT Repository
Docker Image Repository

Development (+QA)
Defining Services

SCALE CUBE OF K8S

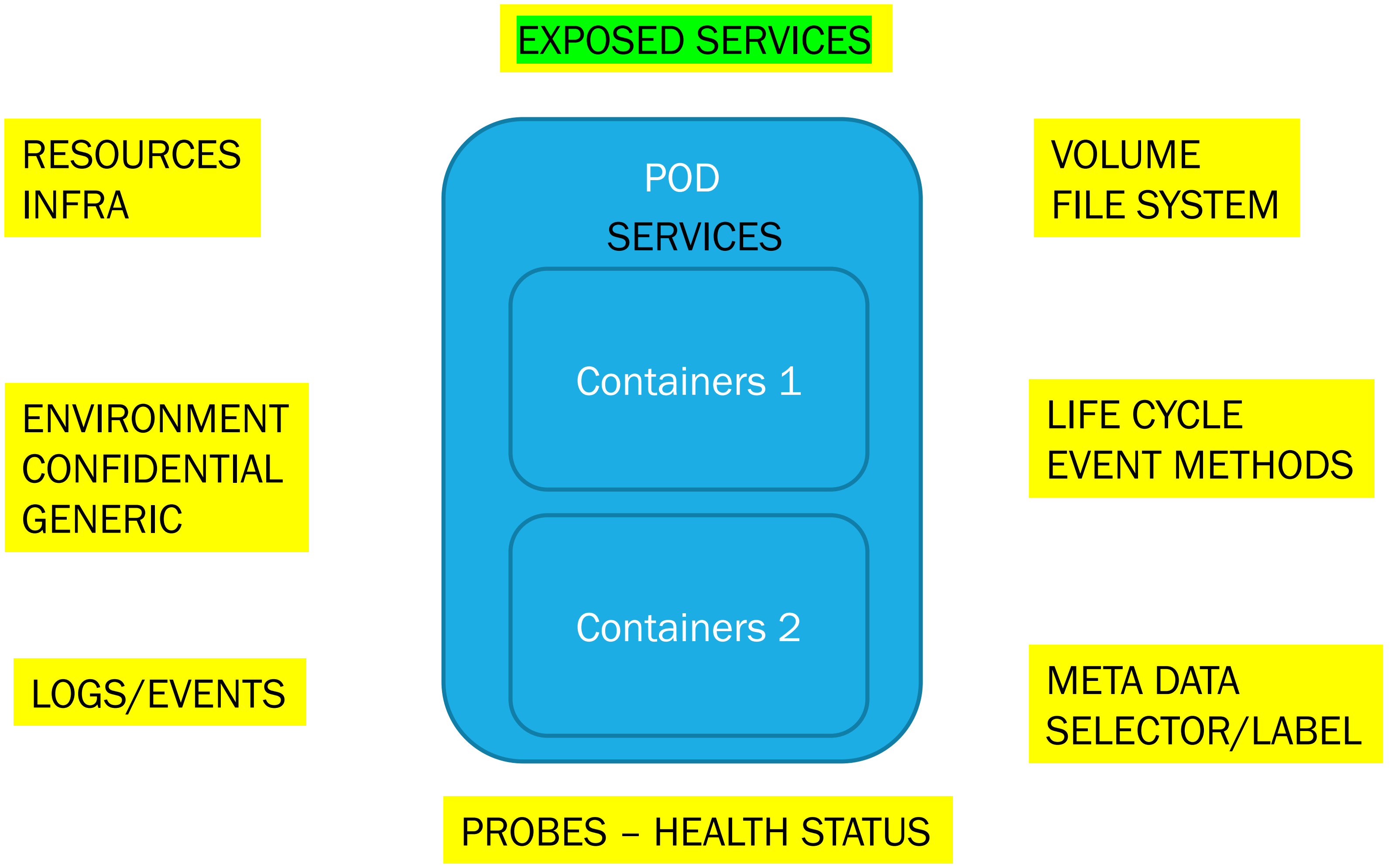
VM Deployment	Container Architecture	Kubernetes Architecture
Virtual Machines/Host	Docker Machine	Node (Orch, Worker)
Services (Jars.Ears)(Files)	Containers (MSA)	Pods (Scaling)
Ports (VM/Host)	Static /Ports (Container)	Exposing at Pods (Dynamic Port)
Scale – Machines (HZ)	Scaling Containers (VZ)	Scaling Pods (Collection of Containers, VZ)
SSH(22)	Dockerd (2376)	Master (8443,6443),Nodes(443)



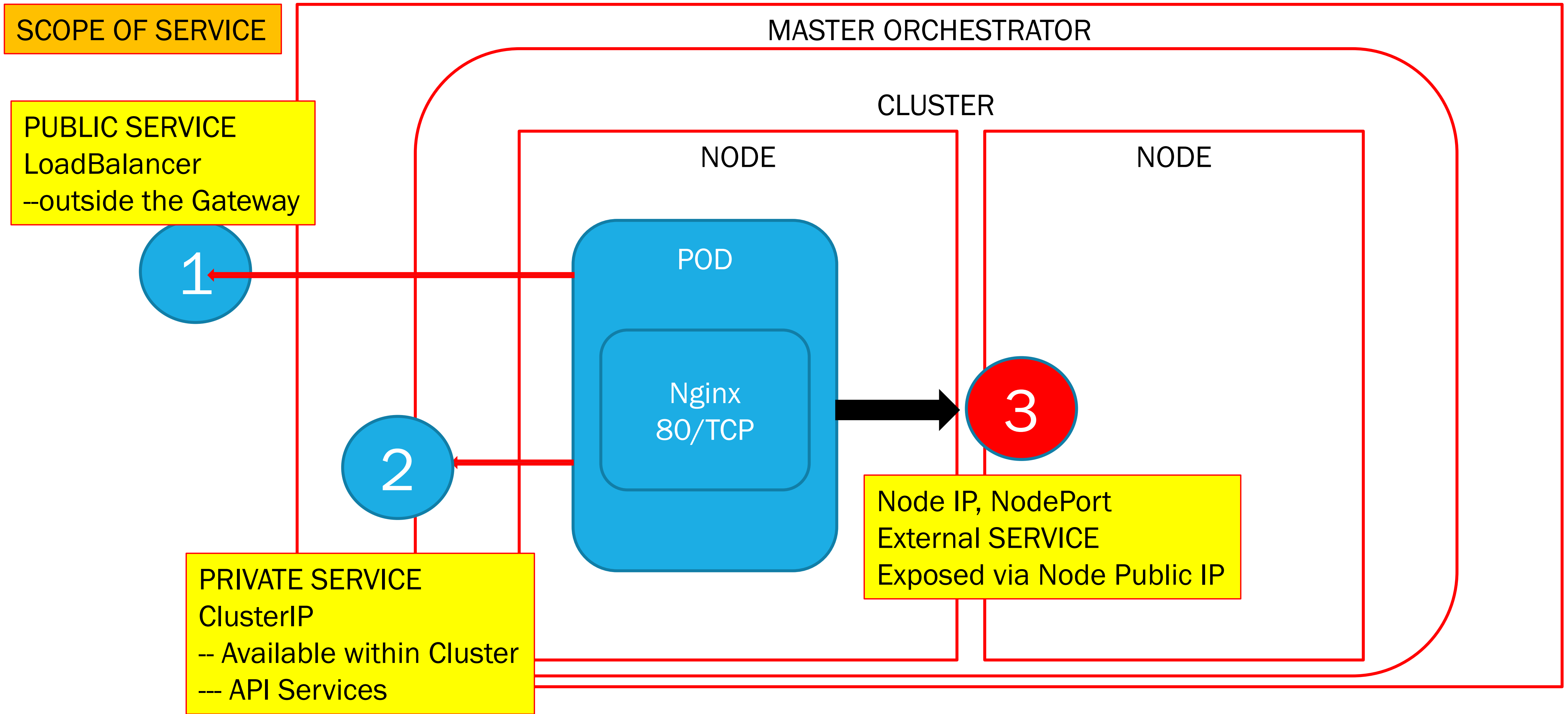
BIBLE OF K8S (API-RESOURCES)

- # kubectl api-resources
- Name of Object (for CLI)
- Short name of Object (for CLI)
- API Group (Library) → YAML
- Namespaced (True/False)- Scope of Object (Yaml and CLI)
- Kind (Type of Object in Yaml) – Proper Case

POD ARCHITECTURE



EXPOSING PODS



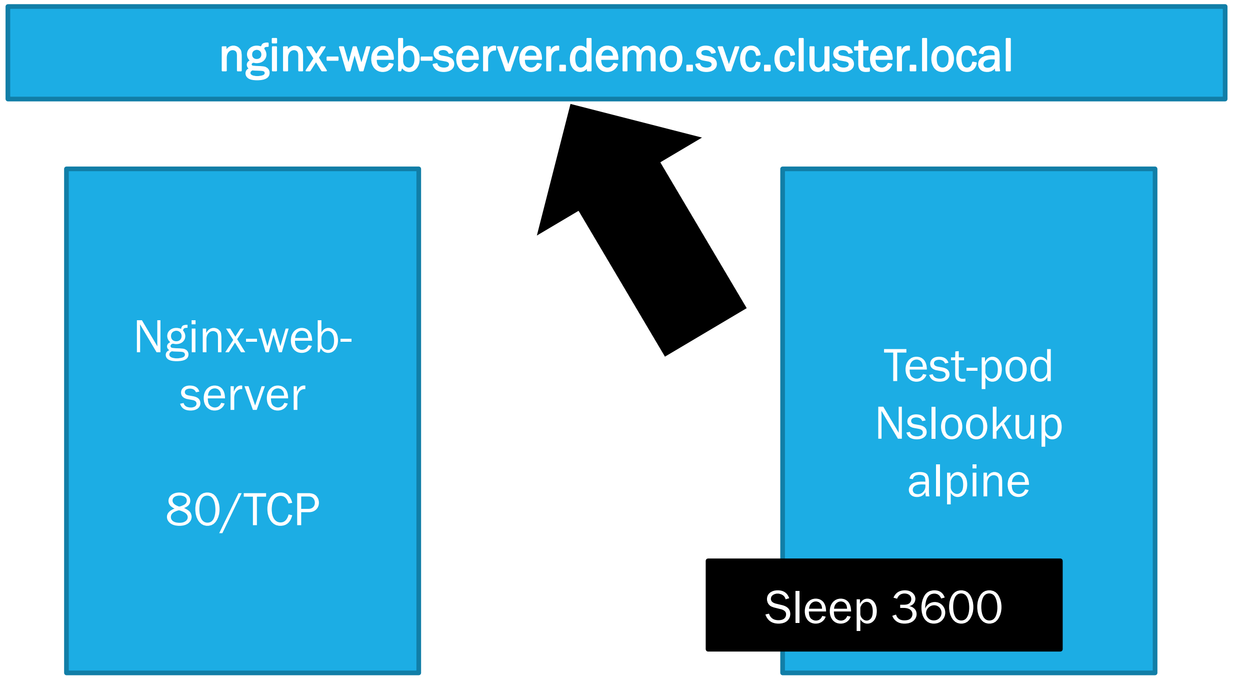
PUBLIC SERVICE
LoadBalancer
--outside the Gateway

PRIVATE SERVICE
ClusterIP
-- Available within Cluster
--- API Services

Node IP, NodePort
External SERVICE
Exposed via Node Public IP

SERVICE ACCESS ?

- SERVICE NAME , AS PER SERVICE REGISTRY (SERVICE DNS ENTRY) **WITHIN API** – SERVICE DNS ENTRY
  - **SERVICE-NAME.NAMESPACE-NAME.SVC.CLUSTER.LOCAL (SERVER SIDE DISCOVERY)**
  - /etc/resolv.conf (DNS SERVICE ENTRIES)
- SERVICE-NAME WITH NAMESPACE
- EXTERNAL IP
- SERVICE IP (NOT RECOMMENDED)
- PORT FORWARDED NUMBER (PRIMITIVE API)


nginx-web-server.demo.svc.cluster.local
Nginx-web-server
80/TCP
Test-pod
Nslookup
alpine
Sleep 3600