DIAMANTI

Hack The Stack: Fast-Tracking Enterprise-grade Kubernetes Infrastructure

Enterprises Are Going Cloud-Native

What defines ‘Cloud-Native’?

- Packaged as lightweight containers
- Designed as loosely coupled microservices
- Architected with a clean separation of stateless and stateful services
- Isolated from server and operating system dependencies
- Deployed on self-service, elastic, cloud infrastructure
- Defined, policy-driven resource allocation

THENEWSTACK
10 Key Attributes of Cloud-Native Applications

Infrastructure must offer:

- Compatibility with CNCF ecosystem
- Scalability, elasticity
- Persistent container storage, high availability
- No technology lock-in
- Automation and multi-cloud management
- Multi-tenancy, guaranteed SLAs

Kubernetes Infrastructure Challenges

"Enterprise interest in Kubernetes to build and deploy new applications is off the charts. Security, storage, networking and monitoring are the top challenges that our user community have highlighted on the Kubernetes adoption path."

--Dan Kohn, Executive Director

Major Container Adoption Challenges

Day 1 (Hard)

Deploy infrastructure

- Build the whole container stack
- Configure network, storage
- Install container runtime and orchestration

Day 2 (Harder)

Manage containers in production

- Guarantee real-time SLAs
- Infrastructure services
- 24x7 support

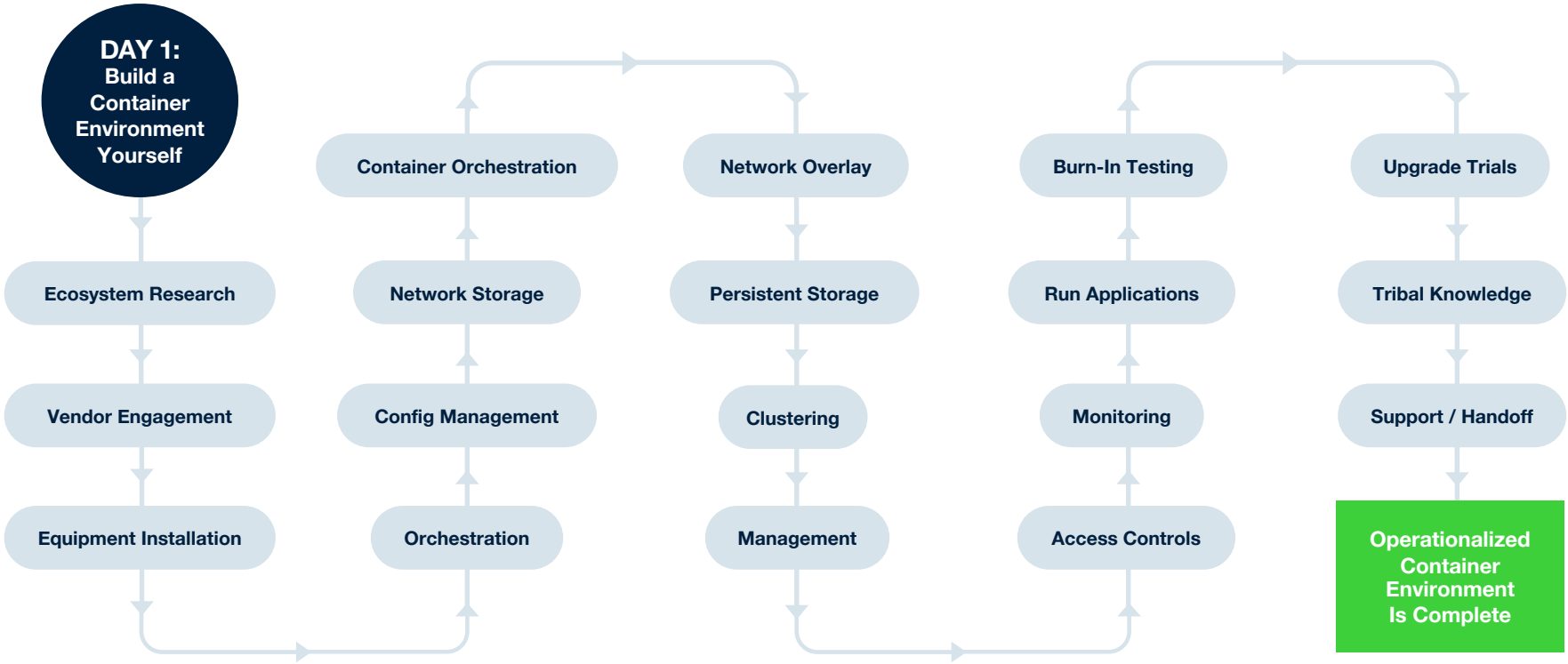
Day 3 (Hardest)

Expand with multi-cloud

- Quick movement of containers across cloud environments
- Seamless scalability
- Policy-driven

Choose your cloud-native infrastructure wisely; it matters at every step.

Do-It-Yourself Approach to Container Infrastructure



How Kubernetes Uses Compute Resources

Kubernetes podspecs offers a declarative model for allocating CPU and memory resources on a per-container basis

- Limits = maximums
- Requests = guaranteed minimum
- K8S prevents oversubscription of requests

```
apiVersion: v1
kind: Pod
metadata:
  name: qos-demo
  namespace: qos-example
spec:
  containers:
  - name: qos-demo-ctr
    image: nginx
    resources:
      limits:
        memory: "200Mi”
        cpu: "700m”
      requests:
        memory: "200Mi”
        cpu: "700m"
```

Storage For Stateful Containers

Local Storage

- Host paths can be mounted to containers in Docker and Kubernetes
- Red flags:
  - High Availability
  - Who is managing it?

Kubernetes Native Storage Drivers

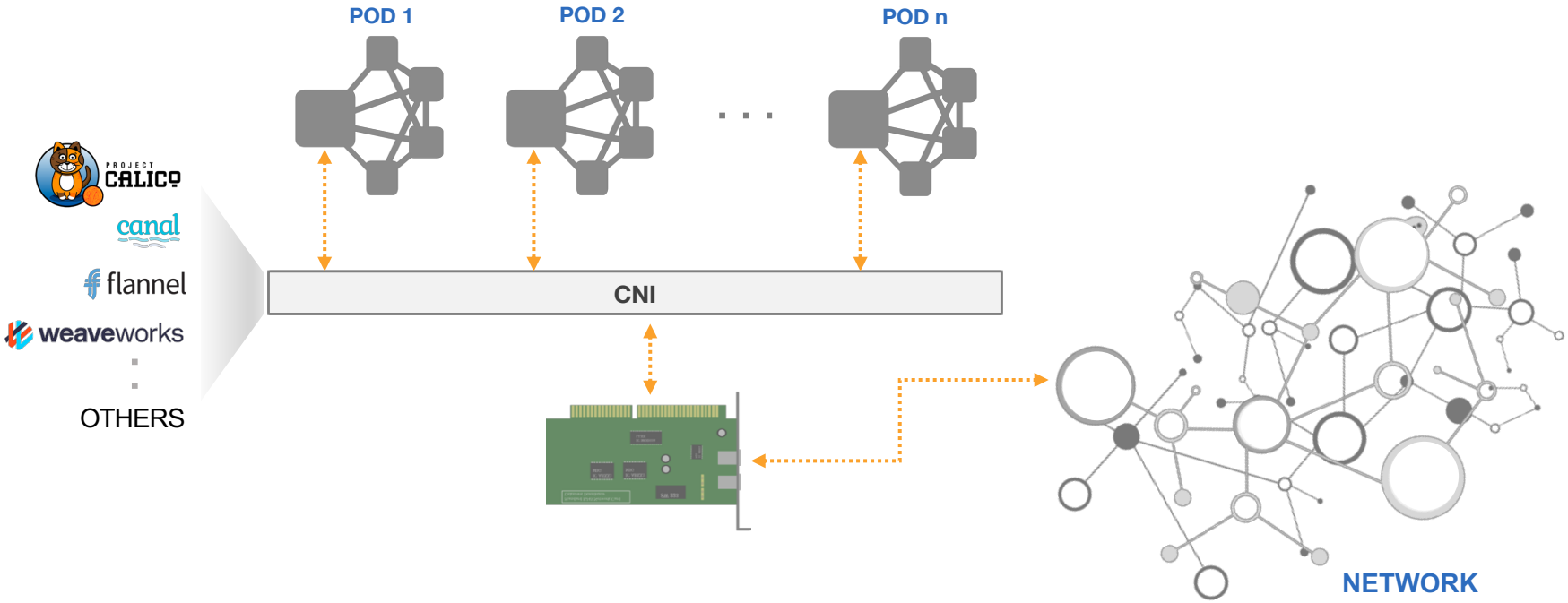
- K8S community integrated several popular filesystems and storage drivers:
- iSCSI, NFS, GlusterFS, CEPH, GCEpersistent, AzureFile, etc.

Kubernetes FlexVolume / CSI

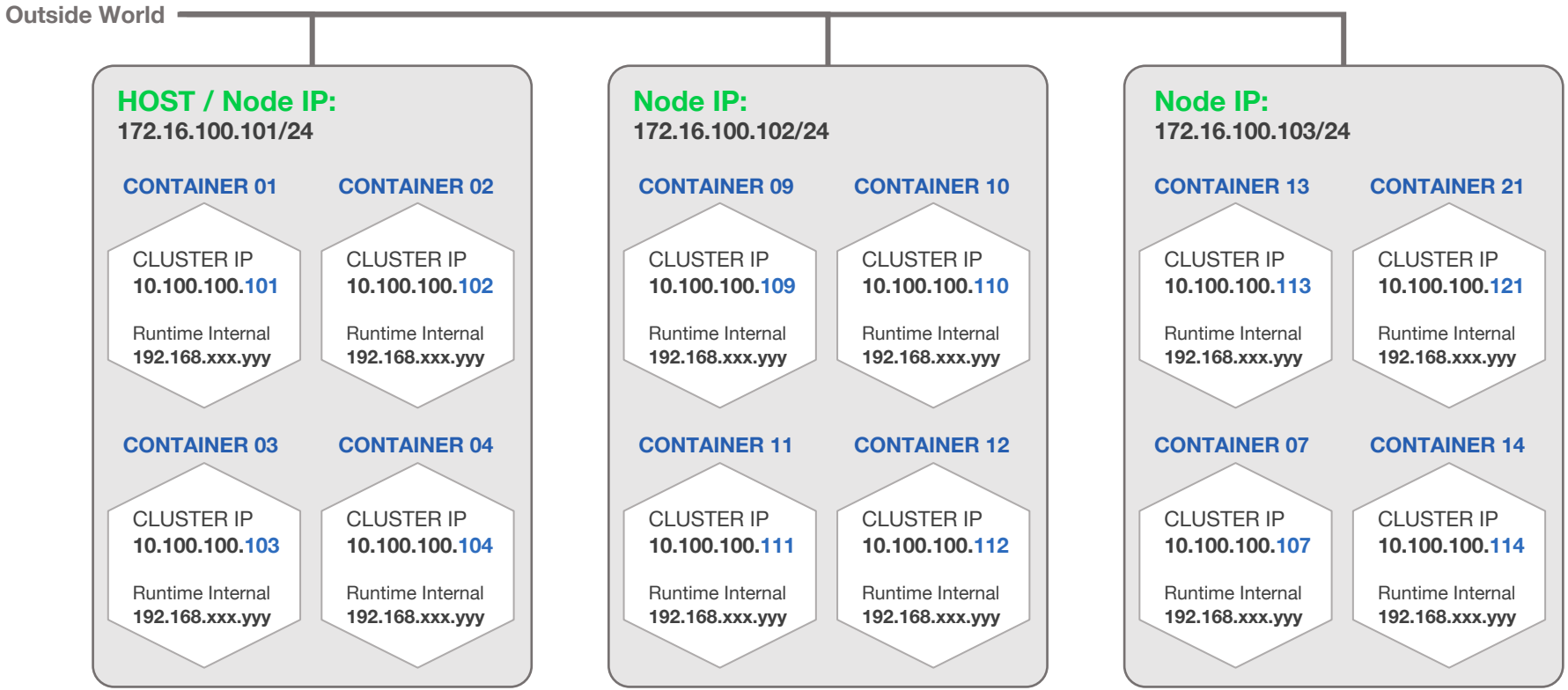
- Plug-in model
  - single way to integrate 3rd party storage
- replaces the need to maintain several different storage drivers as part of the mainline Kubernetes code

Kubernetes Networking Model

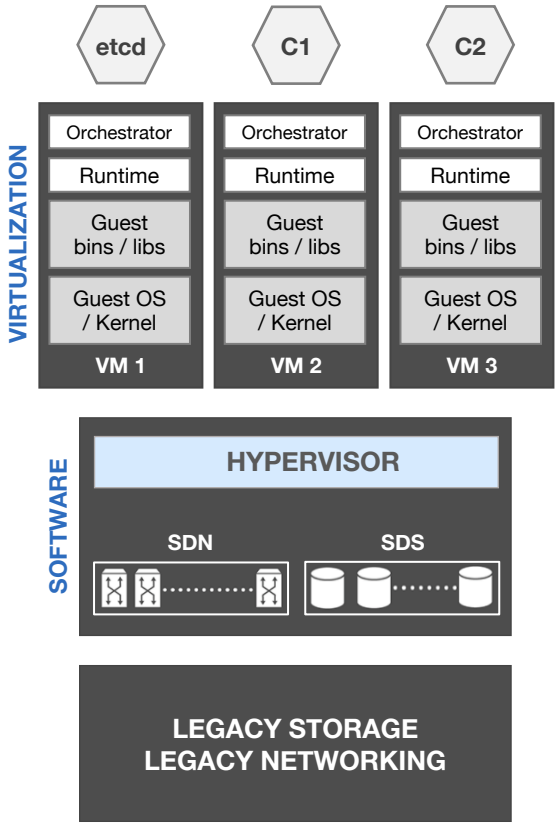
Kubernetes accepts only one container network interface (CNI)



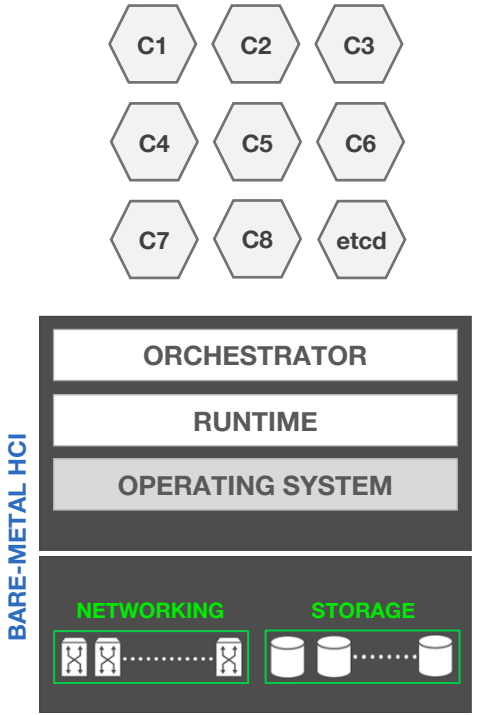
Container Networking Model: IP Layers and Port Mapping



Infrastructure: VMs VS. Bare Metal Hyperconverged



- Complex management
- Inefficient resource utilization
- Low container density
- Limited performance
- High TCO



- Simple management
- >95% resource utilization
- High container density
- Optimal performance
- Lowest TCO

Diamanti Enterprise Kubernetes Platform

Complete turnkey Kubernetes stack

- Hyperconverged 1U appliance built on x86 architecture
- Features container-optimized networking and storage models
- 24x7 full-stack support by Diamanti

Built for cloud

- Manage multiple on-prem clusters and hybrid-cloud deployments through a single UI
- Enterprise DP/DR features: mirroring/synchronous replication, snapshots/asynchronous replication
- Burst production workloads to the cloud

Benefits:

- High performance
- Efficient
- Secure
- Installs in minutes
- Low TCO

Thank You