MesosCon
EUROPE

What Building Multiple Scalable DC/OS Deployments Taught Me about Running Stateful Services on DC/OS

Nathan Shimek - VP of Client Solutions at New Context
Dinesh Israin – Senior Software Engineer at Portworx

THE LINUX FOUNDATION

Challenge Domains

THE LINUX FOUNDATION

Challenge Domains

- Platform Availability
  - Build in Resiliency
  - Monitoring and Metrics
  - Testing
  - Limit the Blast Radius
- Within the Cluster
  - Platform Security
  - Isolation
  - Maintenance

Challenge Domains

- Outside the Cluster
  - Routing
  - Load Balancing
  - Service Discovery
- Organizational
  - Adoption and User Experience
  - Rules and Controls
  - Training
  - Fostering the Right Skillsets

Platform Availability

THE LINUX FOUNDATION

Platform Availability, or Lack Thereof

- There are lots of ways to negatively impact availability
  (this isn't comprehensive by any means)
  - Host failure
  - Zone outage
  - Loss of subnet connectivity / Network segmentation
  - Failed volume
  - Loss of storage connectivity
  - Storage driver failure
  - Runaway application
  - Unresponsive tasks
  - Orphaned tasks
  - Runaway job/app launching
  - Unresponsive app/job scheduler
  - Escaped bugs

Addressing Platform Availability

- Build in Resiliency
  - Go for HA right off the bat
    - Multi-Master, Multi-Zone, separate subnets
  - Scalable architecture informs other automation and tool choices
    - Platform is more resilient and availability leads to adoption and happy users
  - Automated cluster build / re-build

Addressing Platform Availability

- Resiliency continued
  - Ability to add and remove masters and workers independently and easily
    - Operators should be safe in terminating at least one node at a time
  - Execute a single command to recreate any missing nodes
    - Newly created workers and masters should initialize and join the cluster with no additional human intervention

Addressing Platform Availability

- Test fault recovery features
  - Cause **real world** outages
  - In a **production like** environment
- Monitor for failure scenarios
  - Aligned with failure scenarios
- Infrastructure is multi-disciplinary, DC/OS is no exception

Addressing Platform Availability

- Limit the blast radius
  - Isolation
    - User applications from each other
    - Platform services from users
    - Platform services from each other
  - Effective controls to enforce isolation
    - Be especially careful with inter-service dependencies

Within the Cluster

THE LINUX FOUNDATION

Within the Cluster

- Platform Security
  - All of this is new so attacks are evolving rapidly
  - Engage the security team
  - Areas to Review:
    - Marathon app and metronome job config
    - Privilege escalations
    - Sandbox escapes
    - Docker file

Within the Cluster

- Isolation
  - Limit damage users can cause to each other
  - Also limit damage users can cause to platform services
  - Isolate platform services from each other to avoid cascading failures

Within the Cluster

- Maintenance
  - Remember how you started with an HA deployment? If you didn't, now is the time to frown.
  - Metrics and monitoring should detect these situations
  - Alerts and pre-built workplans are necessary
  - Automated clean up jobs are ideal

Outside the Cluster

Outside the Cluster

- Routing
  - Cross-app reverse proxies
    - Public agents pre-populated with shared proxies
    - Public agents auto-scaled to maintain space for spikes
    - Some high-traffic apps may still need their own
  - External Port Management
    - In addition to IPAM, some means of managing ports on external load balancers may be necessary

Outside the Cluster

- Service Discovery and Load Balancing
  - Synchronization with app events
  - Tune DNS cache times / service discovering polling interval
- Several common tools available for this including:
  - mesosphere/marathon-lb
  - containous/traefik
  - gliderlabs/registrator
  - AVI Vantage

Organizational

THE LINUX FOUNDATION

Organizational

- Adoption and User Experience
  - Treat dev like prod
  - Devs are first users
  - Integrations make or break the experience

Organizational

- Training
  - Formal training
  - Find experienced advocates
  - Developer training is a must

Organizational

- Fostering the Right Skillsets
  - Pairing with experienced engineers
  - Operations playground
  - Fail fast and fail often
  - Hack days

Organizational

- Rules and Controls
  - Fundamentally a shared resource
  - Beta user program
  - Organizational controls and ground rules


MesosCon
EUROPE
THE LINUX FOUNDATION