

## Course Outline:

### CN100: Docker Containerization Essentials

- Containerization motivations and implementation
  - Usecases
  - Comparison to virtual machines
- Creating, managing and auditing containers
  - Container implementation from the Linux kernel
  - Container lifecycle details
  - Core container creation, auditing and management CLI
- Best practices in container image design
  - Layered filesystem implementation and performance implications
  - Creating images with Dockerfiles
  - Optimising image builds with multi-stage builds and image design best practices
- Single-host container networking
  - Docker native networking model
  - Software defined networks for containers
  - Docker-native single-host service discovery and routing
- Provisioning external storage
  - Docker volume creation and management
  - Best practices and usecases for container-external storage.

### CN110: Docker Swarm Application Essentials

- Setting up and configuring a Swarm
  - Operational priorities of container orchestration
  - Containerized application architecture
  - Swarm scheduling workflow & task model
  - Automatic failure mitigation
  - Swarm installation & advanced customization
- Deploying workloads on Swarm
  - Defining workloads as services
  - Scaling workloads
  - Container scheduling control
  - Rolling application updates and rollback
  - Application healthchecks
  - Application troubleshooting
  - Deploying applications as Stacks
- Networking Swarm workloads
  - Swarm service discovery and routing implementation

- Routing strategies for stateful and stateless workloads
  - Swarm ingress traffic
- Provisioning dynamic configuration
  - Application configuration design
  - Environment variable management
  - Configuration file management
  - Provisioning sensitive information
- Provisioning persistent storage
  - Storage backend architecture patterns
  - NFS backed Swarms
- Monitoring Swarm
  - What to monitor in production-grade Swarms
  - Potential Swarm failure modes & mitigations
  - Swarm workload monitoring

## **CN212: Mirantis Kubernetes Engine (MKE)**

- Mirantis Kubernetes Engine architecture
  - Production-grade deployment patterns
  - Containerized components of MKE
  - Networking & System requirements for MKE
  - Installing MKE via Launchpad for high availability
- Access control in MKE
  - MKE RBAC systems
  - PKI, client bundle and API authentication
  - Swarm and Kubernetes access control comparison
- L7 networking features
  - Interlock for Swarm
  - Istio for Kubernetes
  - Sticky sessions, canary or blue/green deployments, and cookie usage for both orchestrators
- MKE Support Dumps
  - Generating and understanding MKE support dumps
  - Finding critical information in support dumps for troubleshooting MKE
  - Enabling and exporting API audit logs for disaster post-mortem
- MKE Troubleshooting
  - Correlating MKE symptoms with components
  - Probing and reading MKE state databases
  - Recovering failed MKE managers
  - MKE backups & restore
  - Disaster recovery in event of critical MKE failure

## **CN213: Mirantis Secure Registry (MSR)**

- Mirantis Secure Registry architecture
  - Production-grade deployment patterns
  - Containerized components of MSR
  - Networking & System requirements for MSR
  - Installing MSR via Launchpad for high availability
  - Integrating external storage into MSR
- Access control in MSR
  - MSR RBAC system
- Content Trust
  - Defeating man in the middle attacks with The Update Framework & Notary
  - Content Trust usage in MSR
- Security Scanning
  - Auditing container images for known vulnerabilities
  - Setting up MSR security scanning
  - Security scan integration in continuous integration
- Repository Automation
  - Continuous integration pipeline architecture featuring MSR
  - Promoting and mirroring images through pipelines
  - Integrating MSR with external tooling via webhooks
- Image Management
  - Image pruning and garbage collection strategies and automation
  - Registry sizing strategy
  - Content caching for distributed teams
- MSR Troubleshooting
  - Correlating MSR symptoms with components
  - Probing and reading MSR state databases
  - Recovering failed MSR replicas
  - MSR backups & restore
  - Disaster recovery in event of critical MSR failure