|  |  |
| --- | --- |
| Use Case Description | Kubernetes Multi-node Deployment (interactive mode application launch) |
| Actors | Fabric Manager, Administrator, Master Node, Worker nodes, Ethernet switches |
| Description | Create template and deploy multiple K8s Pods on multiple nodes |
| Initial State | * Master Node   + Running K8s Services   + Running etcd data base mgr * Worker Nodes   + Running Kubelet service   + Running container runtime     - Docker, rkt, runc * FM   + Running Redfish services   + Running CNI daemon   + Clusters configured, CIDR blocks (IP address pools) assigned * Ethernet Network   + ***Master Node and Worker Nodes connected (cluster already created)*** |
| Normal Flow | * Admin: Create YAML Deployment and Services files describing micro-service containers, Pods, replications, and connections (internal and external) * Admin: invoke ‘Kubectl apply ‘on Master node with YAML files as arg’s * MN: Parse YAML, update etcd data base, select worker nodes * MN: launch appropriate numbers of Pods on target worker nodes via kubelet * WN: extract Pod descriptions from etcd data base * MN: assign each Pod an Ethernet namespace, construct port connections according to YAML template * WN: invoke CNI plugin, which will contact FM CNI daemon and obtain IP address for Pod * WN: update etcd data base with Pod’s IP address, query IP address for other Pods in this deployment * MN: parse YAML Services file and establish external IP address through which the deployment converses with clients * MN: update etcd data base to include Services IP * ??: Set up event monitoring for Deployment & the Service * MN: monitor etcd data base status of deployment, adjust resource allocations as necessary * Admin: invoke ‘Kubectl destroy‘ on Master node with YAML files as arg’s * All: tear down connections, shut down processes on worker nodes, update etcd data base   Missing:   * Security key management * Authentication and authorization steps |
| Alternate Flow 1 | * Admin: Create YAML Deployment and Services files describing micro-service containers, Pods, replications, and connections * Admin: invoke ‘Kubectl apply ‘on Master node with YAML files as arg’s * MN: Parse YAML, update etcd data base, select worker nodes * MN: launch appropriate numbers of Pods on target worker nodes via kubelet * WN: extract Pod descriptions from etcd data base * MN: assign each Pod an Ethernet namespace, construct port connections according to YAML template * WN: invoke CNI plugin, which will contact FM CNI daemon and obtain IP address for Pod * WN: update etcd data base with Pod’s IP address, query IP address for other Pods in this deployment * MN: parse YAML Services file and establish external IP address through which the deployment converses with clients * MN: update etcd data base to include Services IP * ??: Set up event monitoring for Deployment & the Service * MN: monitor etcd data base status of deployment, adjust resource allocations as necessary * MN: Error – K8s controller cannot match Status of executing Pods to desired deployment because containing cluster has insufficient resources * MN: ??? -need some mechanism to increase cluster resources, still researching how K8s might already handle this |