Kubernetes Assoc Prof Panche Ribarski, PhD | Assoc Prof Milos Jovanovik, PhD Continuous Integration and Delivery 2023 Kubernetes Continuous Integration and Delivery 2023 Kubernetes Background - Kubernetes is an application orchestrator - Deploy your application - Scale it up and down dynamically based on demand - Self-heal it when things break - Perform zero-downtime rolling updates and rollbacks - Lots more… 2 Kubernetes Continuous Integration and Delivery 2023 What is a containerised app A containerized application is an app that runs in a container But you know this already 3 Kubernetes Continuous Integration and Delivery 2023 What is a cloud-native/microservice app Cloud-native app demands: - auto-scaling - self-healing - rolling updates - rollbacks Microservices app demands: - lots of small, specialised, independent parts that work together to form a meaningful application - front-end, back-end, database, tons of services in a mesh 4 Kubernetes Continuous Integration and Delivery 2023 Where did Kubernetes come from - At the beginning, there was AWS, bringing us modern cloud computing - Google played catch up in cloud services, already having Search and Gmail as containerised apps - To orchestrate and manage these containerised apps, Google had a couple of in-house proprietary technologies called Borg and Omega - Google took the lessons learned from these in-house systems, and created a new platform called Kubernetes that it donated to the newly formed Cloud Native Computing Foundation (CNCF) in 2014 as an open-source project 5 Kubernetes Continuous Integration and Delivery 2023 Kubernetes and Docker At first: - Docker build tools are used to package applications as containers - Kubernetes makes scheduling and other orchestration decisions - Have Docker runtime installed on each worker Now: - Container runtime layer (in 2016) (CRI) 6 Kubernetes Continuous Integration and Delivery 2023 The Orchestrator Wars In 2016/17 we had: - Docker Swarm - Mesosphere - Kettle - Kubernetes Long story short: Kubernetes won 7 Kubernetes Continuous Integration and Delivery 2023 What’s in the name - “Kubernetes” (koo-ber-net-eez) - comes from the Greek word meaning Helmsman - the person who steers a ship - Some of the people involved in the creation of Kubernetes wanted to call it Seven of Nine - copyright laws prevented it from being called Seven of Nine - the creators gave the logo seven spokes as a subtle reference to Seven of Nine - often see it shortened to “K8s” - (pronounced “kates”) - number 8 replaces the 8 characters in between 8 Kubernetes Continuous Integration and Delivery 2023 Kubernetes from 40K feet - A cluster to run applications on - Nodes - Control plane nodes - Worker nodes - An orchestrator of cloud-native microservices apps 9 Kubernetes Continuous Integration and Delivery 2023 The Control plane - Nodes that run the Control plane - A collection of services that control and run everything - HA (3, 5,+ nodes) - Spread across availability zones depending on your SLA - Services: - API Server - The Cluster store (etcd, HA) - Controller manager - The Controller of Controllers (later on Controllers) - Ensures the declarative statement - Desired state + Current state -> differences -> actions - The Scheduler 10 Kubernetes Continuous Integration and Delivery 2023 Control Plane Node 11 Kubernetes Continuous Integration and Delivery 2023 Worker nodes - The place where user applications run - Work logic: - Watch the API server for new work assignments - Execute work assignments - Report back to API server 12 Kubernetes Continuous Integration and Delivery 2023 Worker node components - Kubelet - The main Kubernetes agent, runs on every worker node - Watches API server for new work tasks - Executes tasks and maintains a reporting channel - Doesn’t schedule, doesn’t think - just executes and reports - Container runtime - Pulls images, starts/stops containers - Plugin based - Container Runtime Interface (CRI) - containerd - stripped down docker - Donated to cncf by Docker Inc - Kube proxy - Local cluster networking: ip addresses, iptables, routing, load balancing 13 Kubernetes Continuous Integration and Delivery 2023 Dumbed down Kubernetes deployment - Write your app in any programming language - Containerize it (dockerize for the sake of the argument) - Upload it to a registry - Define a Kubernetes Pod to run the container - A Pod is just a wrapper to a container - Deploy the Pod to Kubernetes - This is called a static Pod deployment, but it’s never used in real life - You’d wrap the Pod in a higher level controller - like a Deployment 14 Kubernetes Continuous Integration and Delivery 2023 Logical architecture of a Controller (Deployment) 15 Kubernetes Continuous Integration and Delivery 2023 Declarative, declarative everywhere - Remember imperative vs declarative in Jenkins - Declarative way of thinking is the new kid on the block - In Kubernetes it’s like this: - Declare the desired state in a manifest (yaml) - Post it to the API server - Kubernetes stores it as app’s desired state - Kubernetes implements the desired state on cluster - Controller makes sure the observed state doesn’t differ from the desired state 16 Kubernetes Continuous Integration and Delivery 2023 Declarative example - You have an app with desired state of 10 replicas - 2 replicas fail - Controller sees this and schedules 2 new replicas - You have an app with 3 replicas - You change the desired number of replicas to 6 - Controller sees this and schedules 3 new replicas - You have an app with 6 replicas, version 1