2018 Global Azure BOOTCAMP

Architecting Domain Driven Development using Azure Kubernetes Service

Speaker : Naren Sivakumar

VALUEMOMENTUM
Deliver Value. Drive Momentum.

Key Contributors
Srinivas Battula
Gurunadha Sivaprasad
Sai Chaitanya

Corporate Overview – ValueMomentum Software Services

- Software & Services Firm
- **Financial Services & Insurance focused**
- Established in 2000 with HQ in NJ, USA
- 150+ dedicated R&D team
- Executive Leadership and Practice Heads based in the US
- Offshore centers are SSAE 16 SOC 2 certified. Clean Rooms for several clients offshore

23%
Compound Annual Growth Rate since 2000

4
Analysts covering ValueMomentum Software & Services

>65
Clients Served in North America

1,850+
Global employee strength

Top 15
IT Services Vendor for North American P&C Carriers by # of customers*

14
>5 Year Customer Relationships Average ~8 years

BUSINESS FOCUS

- Banking & Lending
- Capital Markets

- Property & Casualty
- Healthcare
- Life & Annuities

2


VALUEMOMENTUM
Deliver Value. Drive Momentum
Agenda
Domain Driven Design
Micro Services
Azure Kubernetes
History & Overview
Pods
Deployment
Service
Security
Docker Swarm vs Kubernetes
Demo
Best Practices

2018 Global Azure BOOTCAMP

Trivia Questions

VALUEMOMENTUM
Deliver Value. Drive Momentum

1. Who popularized the concept of "Domain Driven Design" ? It was the author of the book - "Domain Driven Design: Tackling Complexity of the heart of software" - released in 2004.

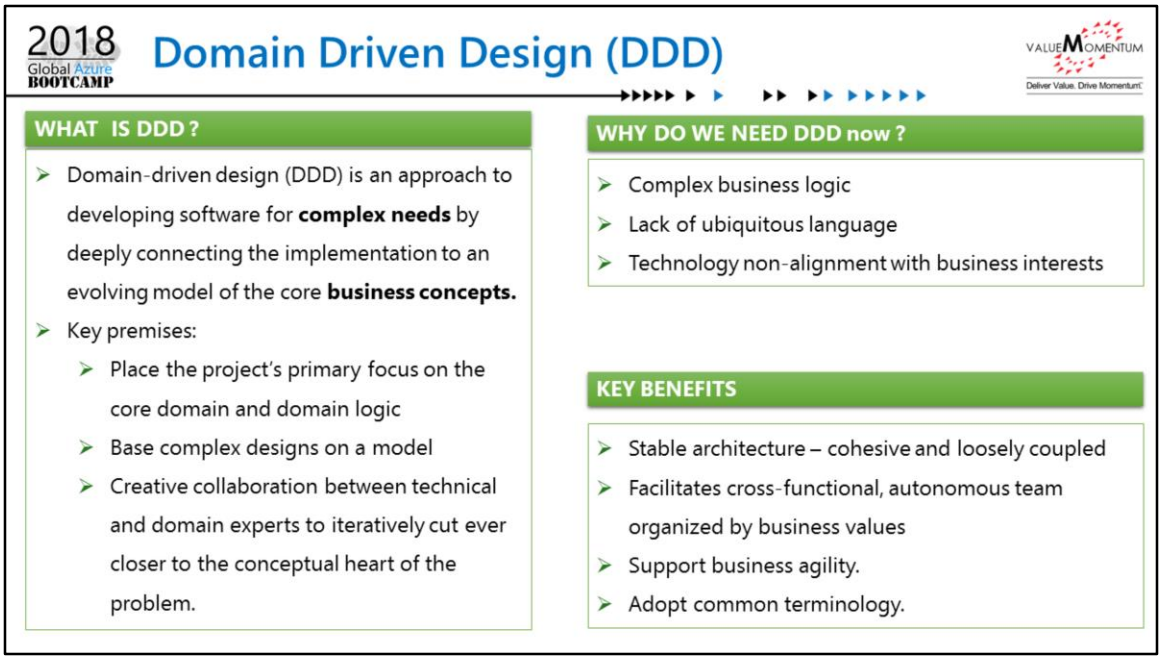
ANSWER : Eric Evans

2. Which company initially designed Kubernetes (or) K8s ?

ANSWER : Google

3. What does AKS stand for ?

ANSWER : Azure Container Service



Strategic DDD

The strategic aspect of DDD [aligns software development teams' efforts with the interests of the business]

Tactical, technical DDD

The tactical, technical aspect of DDD guides the implementation process with the fundamental purpose of protecting the model from corruption.

Why it's hard

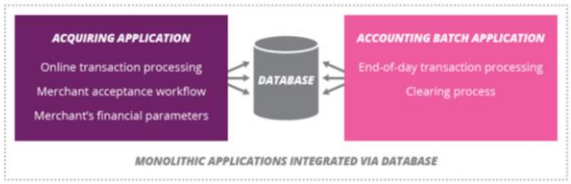
In many ways this is why DDD is considered hard to do right: it takes a certain amount of self-discipline to adhere to the philosophy and requires another level of restraint to resist designing when you should be modeling. Finally, it requires patience to keep refining, refactoring, iterating, and accepting feedback until the model, the code, and the business coalesce into a cooperative synergy.

2018 Global Azure BOOTCAMP

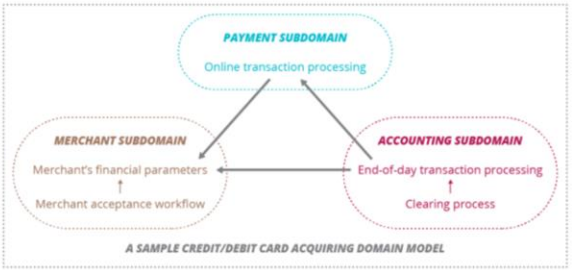
Domain Driven Design (DDD)

VALUE MOMENTUM
Deliver Value. Drive Momentum

BEFORE DDD



AFTER DDD



2018 Global Azure BOOTCAMP

MicroServices

VALUEMOMENTUM
Deliver Value. Drive Momentum

Key characteristics of MicroServices

- Service enabled, autonomous components classified around some business capabilities.
- Easy to scale as individual components
- Product mentality over project.
- Decentralize standards.
- Decentralized data management.
- Automated infrastructure management.
- Application design considering failure in mind.
- Better fault isolation.
- Easily replaceable and upgradable.

Popularity of MicroServices

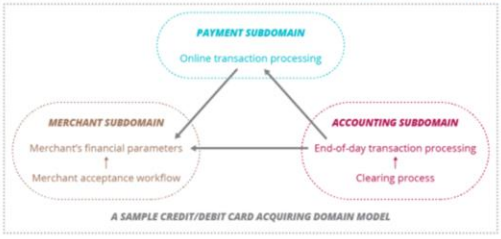
- Frustration at not getting the desired output expected from an architecture like monolith.
- Availability of tools and technologies to develop and deploy microservices applications with ease.
- Wide adaptation of Infrastructure as a Service (IaaS), have opened the door for easy DevOps operations.
- Big technology product company adaptation for microservices architecture.
- Support business agility and technology independence.

2018 Global Azure BOOTCAMP

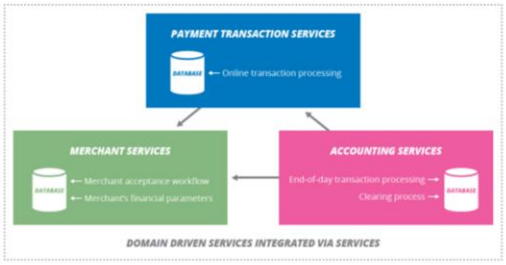
MicroServices

VALUE MOMENTUM
Deliver Value. Drive Momentum

WITH DDD



WITH MicroServices

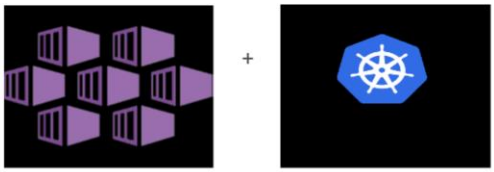


2018 Global Azure BOOTCAMP

Azure Managed Kubernetes

VALUE MOMENTUM
Deliver Value. Drive Momentum

- Kubernetes is a portable, extensible platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
- It can be thought of as:
  - a container platform
  - a microservices platform
  - a portable cloud platform and a lot more.
- **Container-centric** management environment – orchestrates computing, network & storage for users
- Provides simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS),



AKS (managed Kubernetes)

Why do you need Kubernetes?

Real production apps span multiple containers. Those containers must be deployed across multiple server hosts. Kubernetes gives you the orchestration and management capabilities required to deploy containers, at scale, for these workloads. Kubernetes orchestration allows you to build application services that span multiple containers, schedule those containers across a cluster, scale those containers, and manage the health of those containers over time.

<u>What Kubernetes is not</u>

Kubernetes is not a traditional, all-inclusive PaaS (Platform as a Service) system. Since Kubernetes operates at the container level rather than at the hardware level, it provides some generally applicable features common to PaaS offerings, such as deployment, scaling, load balancing, logging, and monitoring. However, Kubernetes is not monolithic, and these default solutions are optional and pluggable. Kubernetes provides the building blocks for building developer platforms, but preserves user choice and flexibility where it is important.

Does not limit the types of applications supported.

Does not deploy source code and does not build your application.
Does not provide application-level services, such as middleware (e.g., message buses), data-processing frameworks (for example, Spark), databases (e.g., mysql), caches, nor cluster storage systems (e.g., Ceph) as built-in services
Does not dictate logging, monitoring, or alerting solutions.

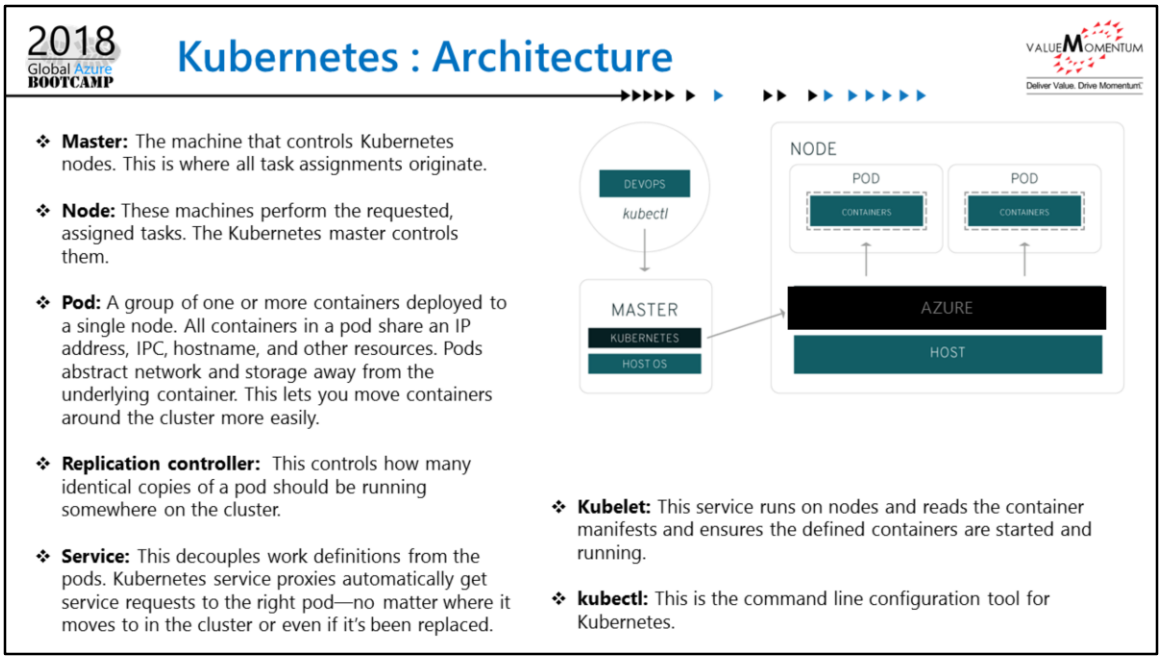
Kubernetes is comprised of a set of independent, composable control processes that continuously drive the current state towards the provided desired state.

2018 Global Azure BOOTCAMP

Kubernetes : Overview & History

VALUEMOMENTUM
Deliver Value. Drive Momentum

- Also known as **K8s** or **Kube**
- Originally developed by **Google**.
- Maintained by **Cloud Computing Foundation Native**.
- **Open source**
- Cluster machine with **Master** and **nodes**
- Creating objects through files in the format **YAML**
- Kubernetes: Manage containers better
  - Orchestration
  - Auto recovery
  - restart
  - replication
  - stagger



Master Components

Master components provide the cluster's control plane. Master components make global decisions about the cluster (for example, scheduling), and detecting and responding to cluster events (starting up a new pod when a replication controller's 'replicas' field is unsatisfied).

1. kube-apiserver - Component on the master that exposes the Kubernetes API.
2. Etcd - Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data
3. kube-scheduler - Component on the master that watches newly created pods that have no node assigned, and selects a node for them to run on
4. kube-controller-manager - Component on the master that runs controllers.
5. cloud-controller-manager runs controllers that interact with the underlying cloud providers

Node Components

1. Kubelet - An agent that runs on each node in the cluster. It makes sure that containers are running in a pod

2. kube-proxy enables the Kubernetes service abstraction by maintaining network rules on the host and performing connection forwarding
3. Container Runtime - The container runtime is the software that is responsible for running containers

2018 Global Azure BOOTCAMP

Kubernetes : Pods

VALUEMOMENTUM
Deliver Value. Drive Momentum

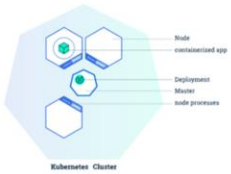
- Group of one or more containers in a deployed Node (Node)
- Share the same IP address, IPC, hostname, and other resources
- Pods are the atomic unit on the Kubernetes platform. When we create a Deployment on Kubernetes, that Deployment creates Pods with containers inside them (as opposed to creating containers directly).

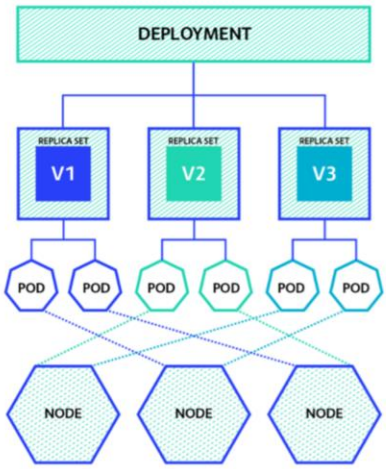
2018 Global Azure BOOTCAMP

Kubernetes : Deployment

VALUE MOMENTUM
Deliver Value. Drive Momentum

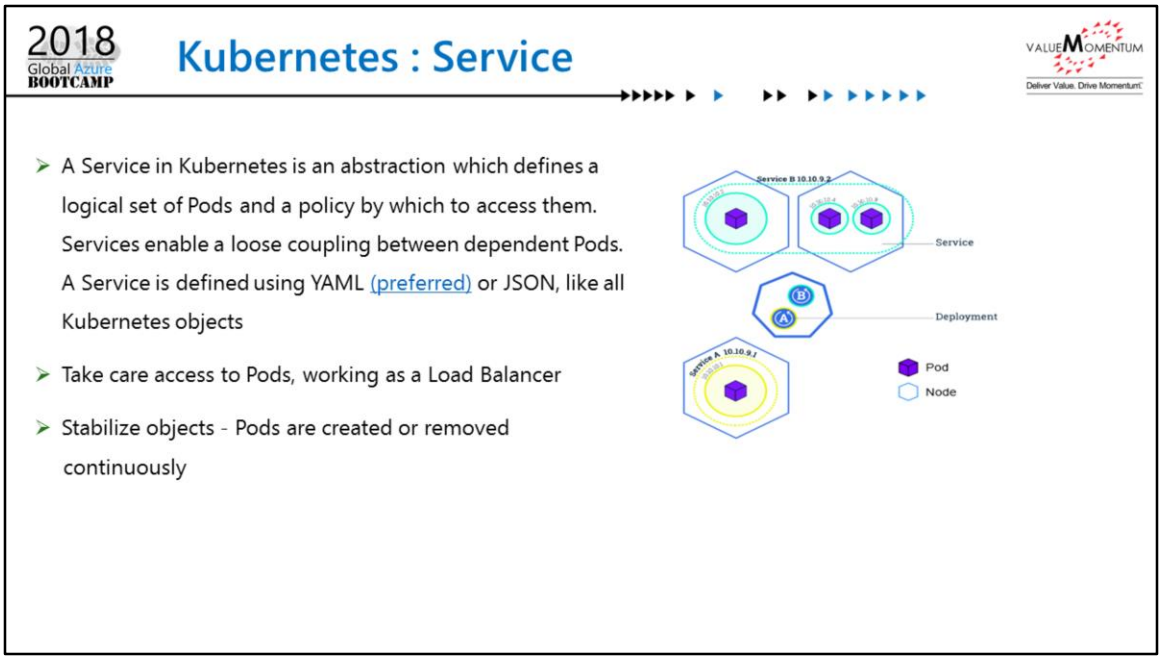
- With Kubernetes Deployments, you "describe a desired state in a Deployment object, and the Deployment controller changes the actual state to the desired state at a controlled rate".
- Once you've created a Deployment, the Kubernetes master schedules mentioned application instances onto individual Nodes in the cluster.





When to use a ReplicaSet

A ReplicaSet ensures that a specified number of pod replicas are running at any given time. However, a Deployment is a higher-level concept that manages ReplicaSets and provides declarative updates to pods along with a lot of other useful features. Therefore, we recommend using Deployments instead of directly using ReplicaSets, unless you require custom update orchestration or don't require updates at all.



Kubernetes Pods are mortal. They are born and when they die, they are not resurrected. ReplicationControllers in particular create and destroy Pods dynamically (e.g. when scaling up or down or when doing rolling updates). While each Podgets its own IP address, even those IP addresses cannot be relied upon to be stable over time. This leads to a problem: if some set of Pods (let's call them backends) provides functionality to other Pods (let's call them frontends) inside the Kubernetes cluster, how do those frontends find out and keep track of which backends are in that set?

A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service. The set of Pods targeted by a Service is (usually) determined by a Label Selector (see below for why you might want a Service without a selector).

As an example, consider an image-processing backend which is running with 3 replicas. Those replicas are fungible - frontends do not care which backend they use. While the actual Pods that compose the backend set may change, the frontend clients should not need to be aware of that or keep track of the list of backends themselves. The Service abstraction enables this decoupling.
For Kubernetes-native applications, Kubernetes offers a simple Endpoints API that is

updated whenever the set of Pods in a Service changes. For non-native applications, Kubernetes offers a virtual-IP-based bridge to Services which redirects to the backend Pods.

2018 Global Azure BOOTCAMP

Kubernetes : Security

VALUEMOMENTUM
Deliver Value. Drive Momentum

- **Implement Continuous Security Vulnerability Scanning –** Containers might include outdated packages with known vulnerabilities (CVEs). This cannot be a 'one off' process, as new vulnerabilities are published every day. An ongoing process, where images are continuously assessed, is crucial to insure a required security posture.

- **Regularly Apply Security Updates to Your Environment –** Once vulnerabilities are found in running containers, you should always update the source image and redeploy the containers. Try to avoid direct updates (e.g. 'apt-update') to the running containers, as this can break the image-container relationship. Upgrading containers is extremely easy with the Kubernetes rolling updates feature - this allows gradually updating a running application by upgrading its images to the latest version.

2018 Global Azure BOOTCAMP

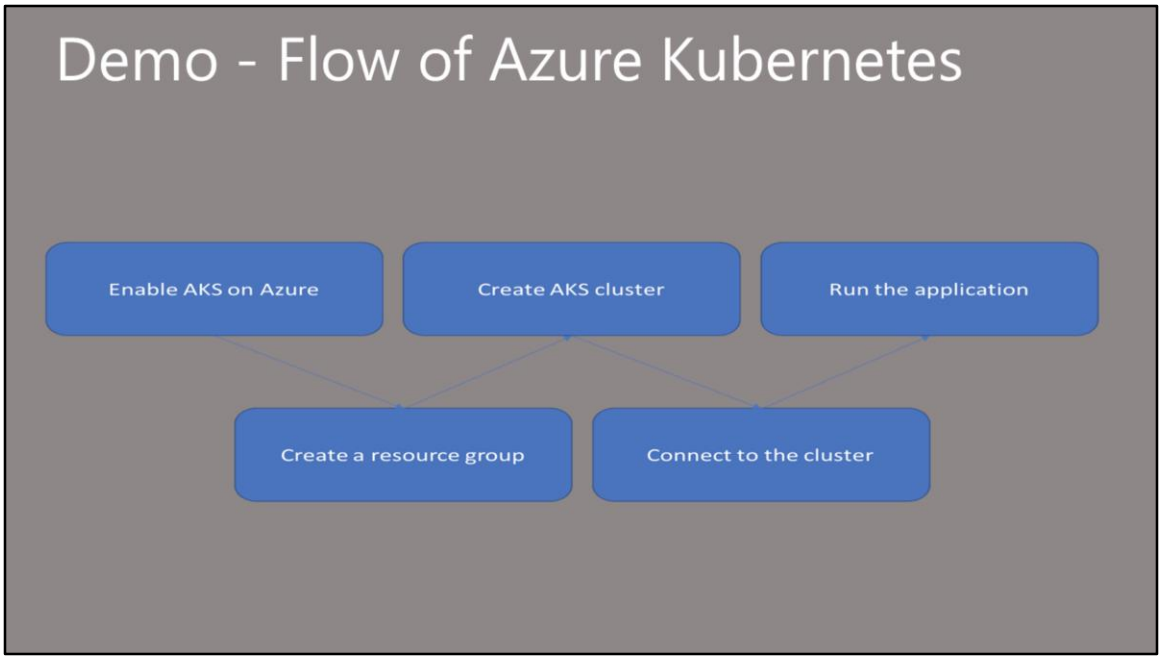
Docker Swarm vs Kubernetes

VALUE MOMENTUM
Deliver Value. Drive Momentum

Kubernetes	Docker Swarm
Most mature solution in the market.	Docker offers good features, but limited by its API.
Kubernetes is also the most popular solution in the market.	Docker's market is relatively weaker compared to Kubernetes.
Kubernetes is hard to setup and configure.	Docker's setup and installation is easy.
Kubernetes offers inbuilt logging and monitoring tools.	Docker only supports 3rd party monitoring and logging tools.
CPU utilization is a big factor in auto scaling.	It is possible to scale services manually.

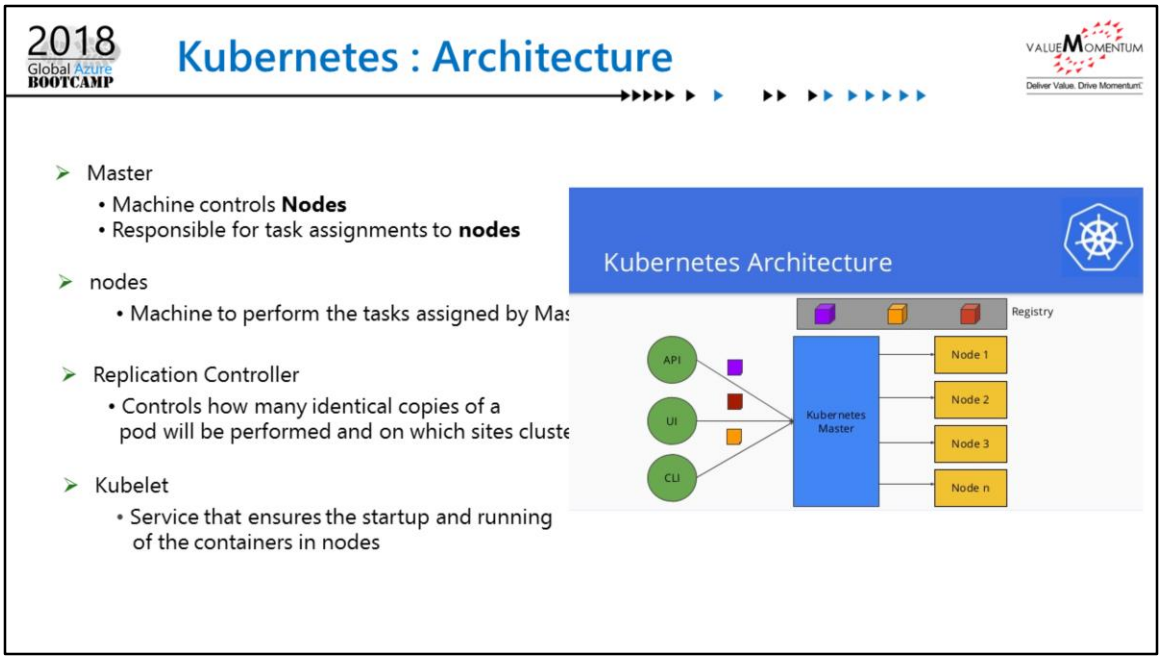
Demo 1.2 - 10-15 min

Let’s see what the remote monitoring solutions look like in reality.



Demo 1.2 - 10-15 min

Let’s see what the remote monitoring solutions look like in reality.



Master: The machine that controls Kubernetes nodes. This is where all task assignments originate.

Node: These machines perform the requested, assigned tasks. The Kubernetes master controls them.

Pod: A group of one or more containers deployed to a single node. All containers in a pod share an IP address, IPC, hostname, and other resources. Pods abstract network and storage away from the underlying container. This lets you move containers around the cluster more easily.

Replication controller: This controls how many identical copies of a pod should be running somewhere on the cluster.

Service: This decouples work definitions from the pods. Kubernetes service proxies automatically get service requests to the right pod—no matter where it moves to in the cluster or even if it's been replaced.

Kubelet: This service runs on nodes and reads the container manifests and ensures

the defined containers are started and running.

kubectl: This is the command line configuration tool for Kubernetes.

2018 Global Azure BOOTCAMP

Little bit about Docker

VALUEMOMENTUM
Deliver Value. Drive Momentum

Advantages

- Isolation, more rational use of resources, speed in deployment, less dependence on environment, micro services gaining momentum are few advantages of using Dockers.

Difficulties

- In general server databases, web applications, services need to be installed and there are difficulties in scaling in containers, ensure coordinated work between containers and container faults and fixing