



NEBULAWORKS

Kubernetes: Zero to Hero Deployments and Management

Anthony Ramirez

June 2020



Kubernetes: Zero to Hero Deployments & Management!



Featured Speaker:

ANTHONY RAMIREZ
Director of Consulting

WHO IS THIS TALK FOR?

FULL STACK DEVELOPERS,
DEVOPS ENGINEERS,
SOFTWARE ENGINEERS,
OPERATIONS TEAMS,
INFRASTRUCTURE TEAMS

WHAT IS THIS ABOUT?

A CLOUD NATIVE APPROACH TO K8S
DEVELOPER ENABLEMENT
TERRAFORM (IaC)
APPLICATION MANAGEMENT

WHY?

SHIFT IN APPLICATION DEPLOYMENT
MONOLITHS TO MICROSERVICE
DEVELOPER PRODUCTIVITY
PLETHORA OF TOOLING

Docker

WHY CONTAINERS?

Benefits of containers

- lightweight
- portable
- packaging is streamlined
- consistent environments for applications
- microservice based patterns

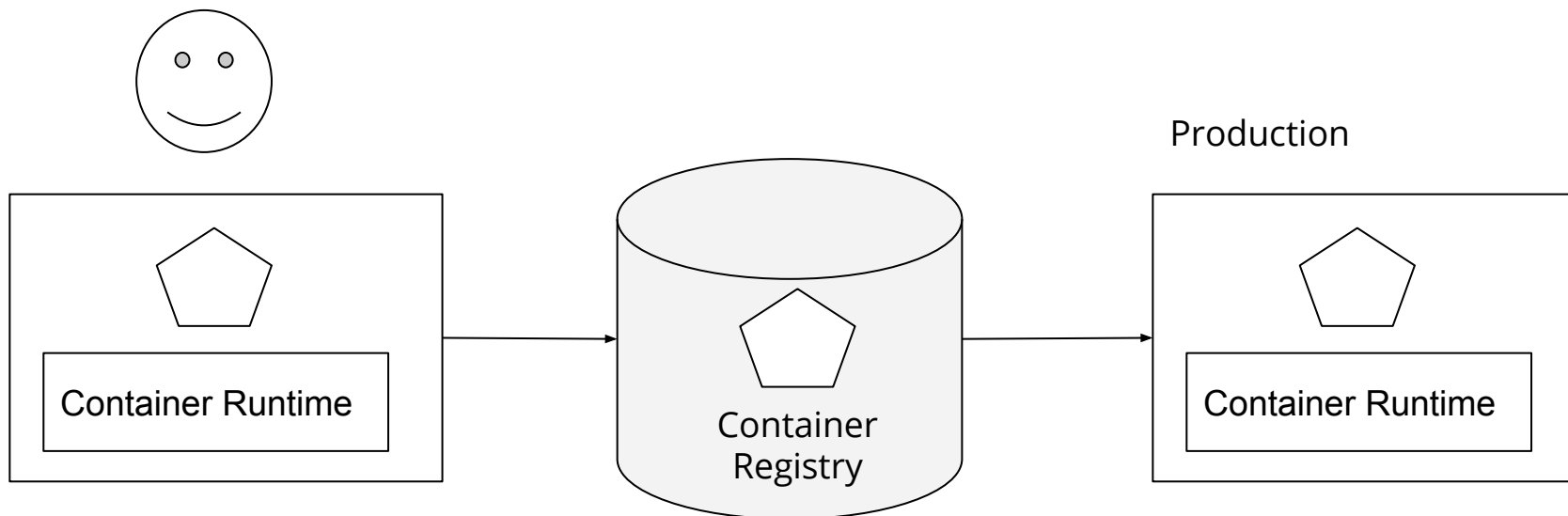
Docker Development Workflow



1. Dev builds and pushes an image using Docker

2. Container Registry holds production ready image

3. Dev deploys production container image to desired environment



Container Adoption Model

Holistic view of the evolution of container adoption



TEAM CAPABILITIES

Initial Container Adoption

Initial deployment of CaaS, no standards around development, no requirements of apps or teams

CI/CD1

All groups are aligned with adoption, standard branching strategy enforced, leverage cloud operating model, CI in place for apps, identification of security domains

Standardization & Modernization

Enforce security best practices for container/image creation, execute securing all security domains, parallel app and infra pipelines with common tools, process for updating CaaS

End-to-End telemetry, security

High degree of monitoring and logging capabilities, service discovery/registration, end-to-end security policy and governance, tooling and systems are flexible and can accommodate change

BUSINESS VELOCITY

Continuous Workflow Maturity



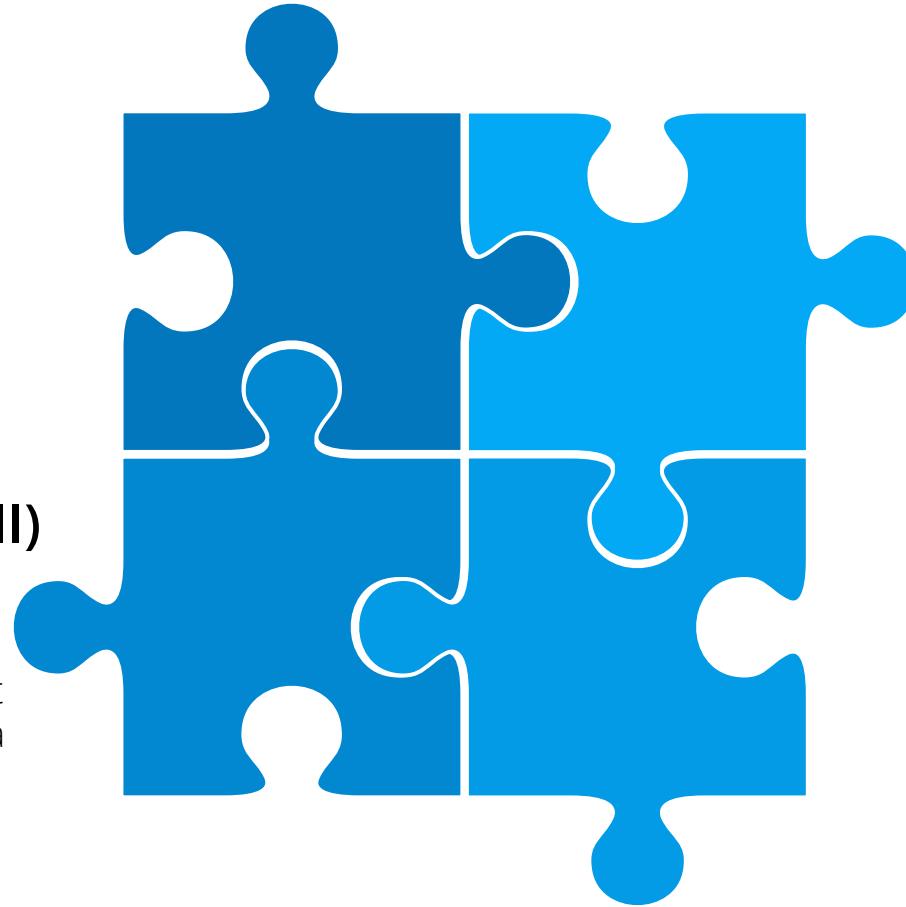
From scripts to CD-II

Script-based Testing

Individual or team-level scripted assembly and testing of applications. Entry-level configuration management. Little to no automation

Continuous Deployment (CD-II)

Parallel application and infrastructure pipelines supporting modern deployment methods (blue/green, canary) leveraging a common set of tools supporting the infrastructure, platform, and application services



Continuous Integration

Application and Infrastructure pipelines having initial repo-level automation for assembly and testing.

Continuous Delivery (CD-I)

High degree of automation, leveraging automated delivery and configuration of applications and of Infrastructure as Code. Ability to leverage pipelines in parameterized builds in any environment.

Kubernetes

WHY CONTAINER ORCHESTRATION?

Evolution of Orchestration

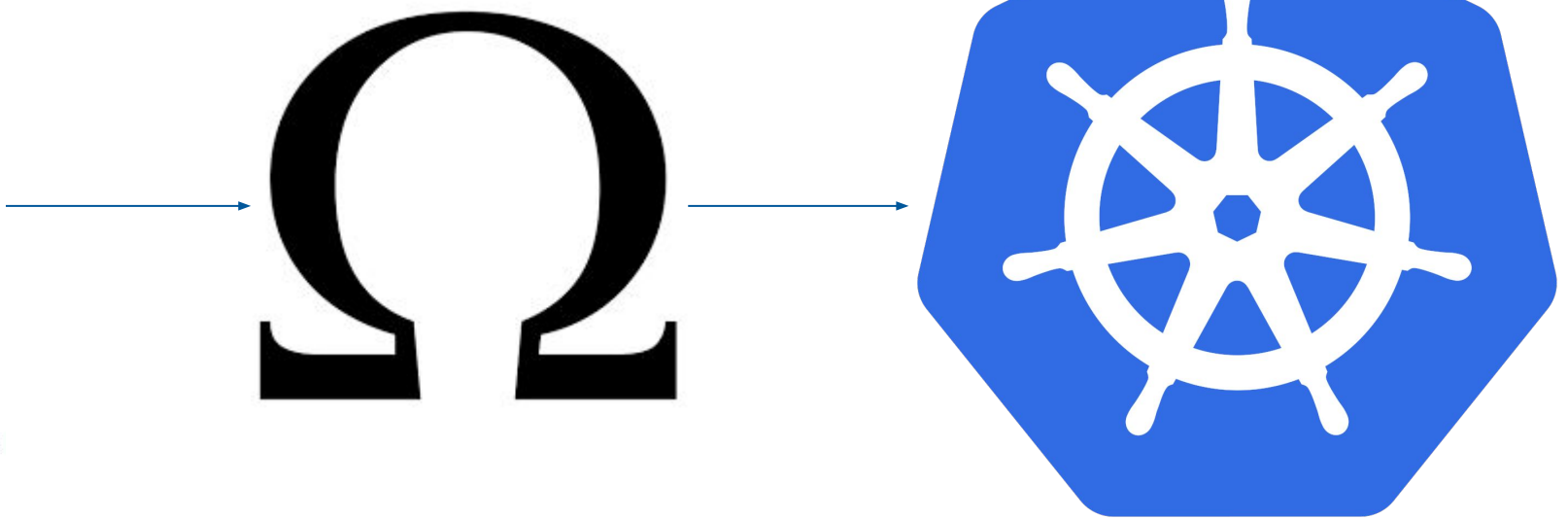
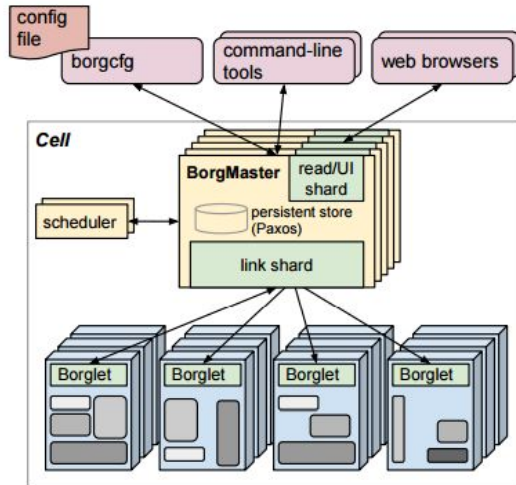
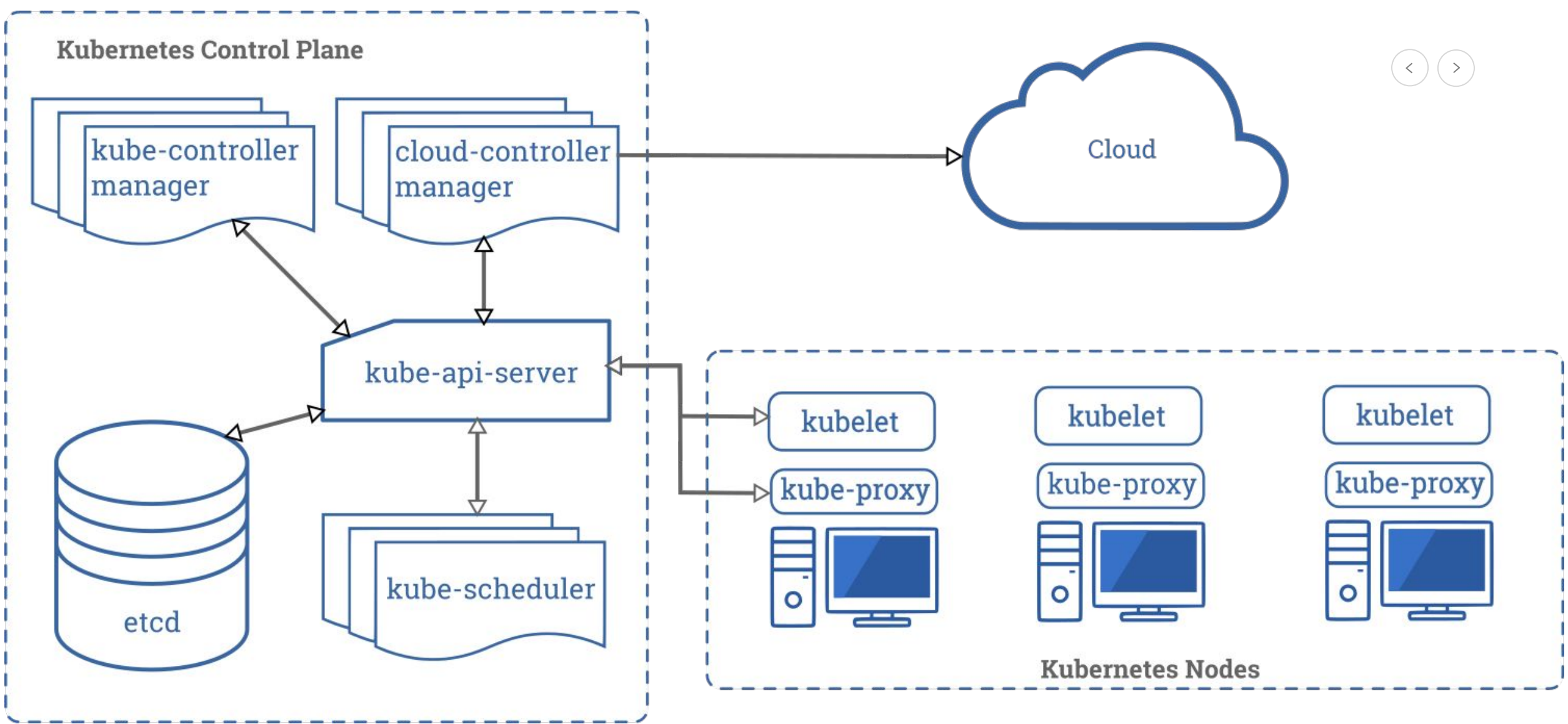
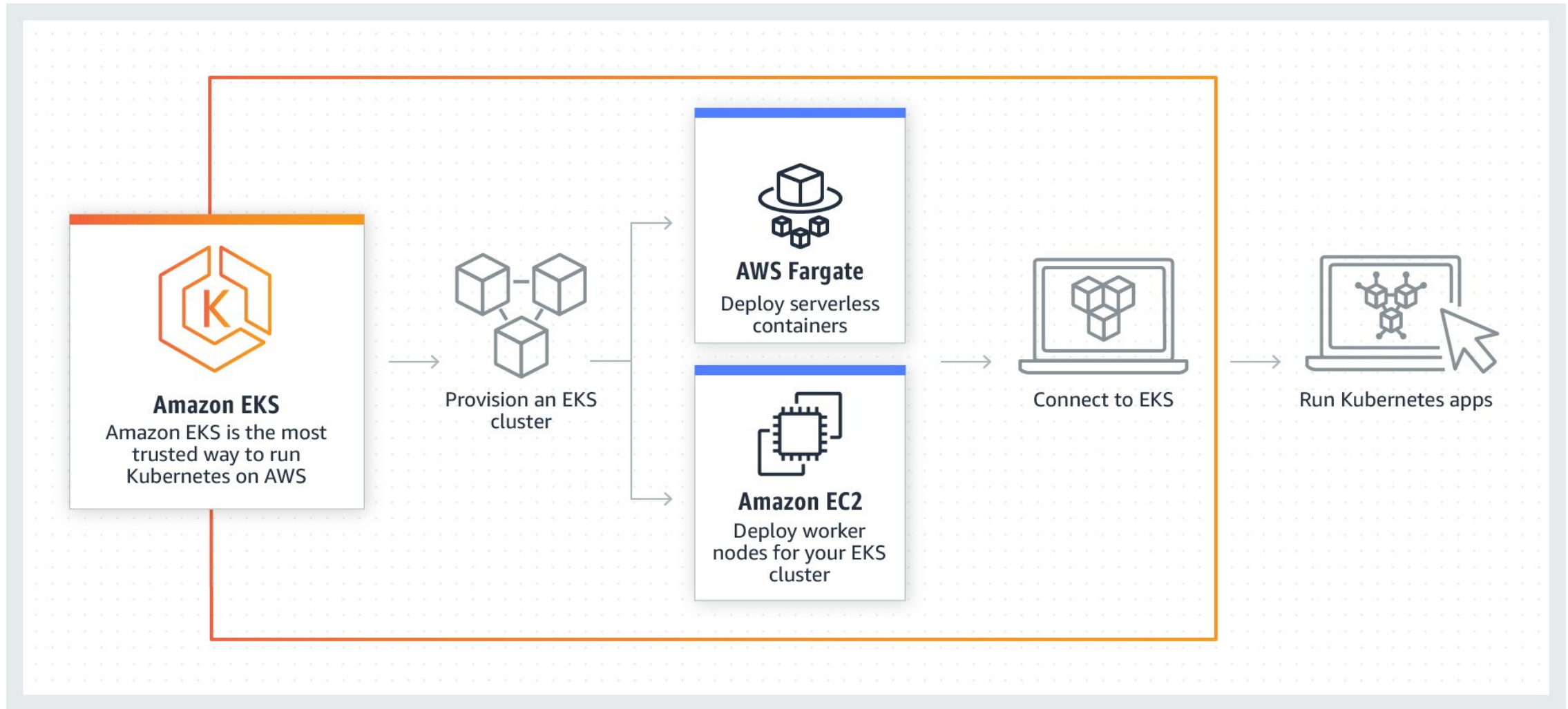


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*



Elastic Kubernetes Service (K8s Managed Service)



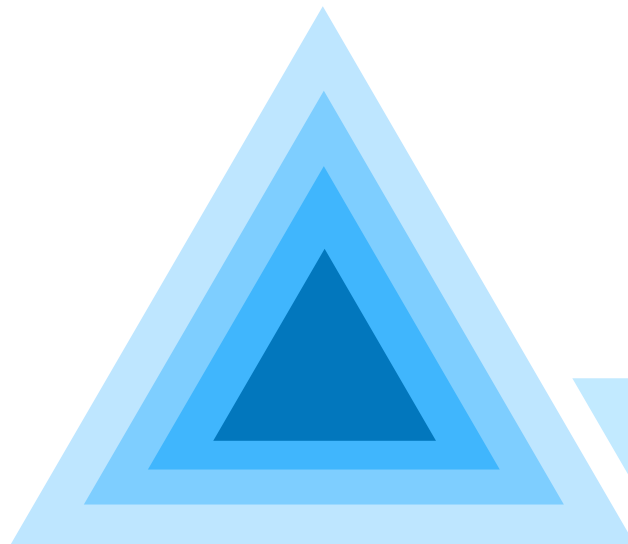
K8s Managed Service Options

Managed Service	Offering	Multi-AZ	Persistent Volumes	Version	RBAC
GKE	GA	Yes	Block	1.15.8	Yes
EKS	GA	Yes	Block	1.15.10	Yes
AKS	GA	Yes	Block and CIFS	1.16	Yes

laC and K8s

Infrastructure as Code (IaC)

Manage and provision resources in cloud platforms with source code



Terraform

Cloud agnostic deployment tool to provision resources using a declarative language

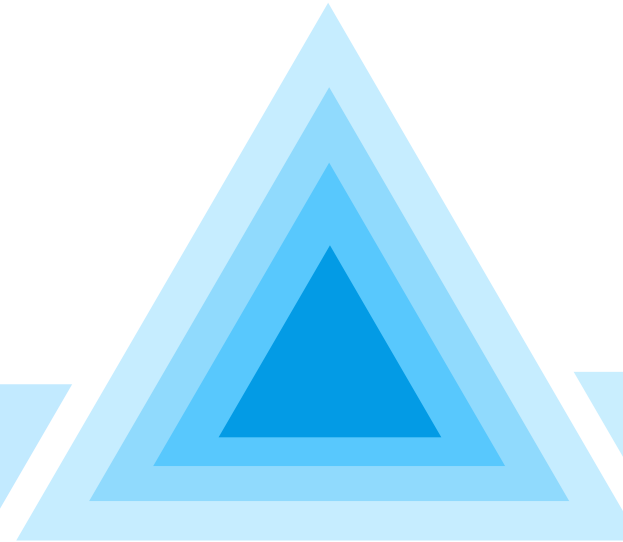
Declarative

Define what the desired resources and dependencies that should exist



Source code as Source of Truth

Leverage software development practices to manage and test IaC



Desired State Management

Use IaC to increase replicability and consistency in platform deployments



```
resource "aws_eks_cluster" "ajr" {

  name      = var.cluster_name
  role_arn = aws_iam_role.example.arn

  vpc_config {
    subnet_ids = ["subnet-054652e98390cd684", "subnet-070d6f83aeb610f33", "subnet-0943b631f710ea063"]
  }

  depends_on = [
    "aws_iam_role_policy_attachment.example-AmazonEKSClusterPolicy",
    "aws_iam_role_policy_attachment.example-AmazonEKSServicePolicy",
  ]
}

resource "aws_eks_node_group" "ajr" {
  cluster_name      = aws_eks_cluster.ajr.name
  node_group_name   = "ajr"
  node_role_arn     = aws_iam_role.node_group.arn
  subnet_ids        = ["subnet-054652e98390cd684", "subnet-070d6f83aeb610f33", "subnet-0943b631f710ea063"]
  disk_size         = "50"
  instance_types     = ["m4.xlarge"]

  scaling_config {
    desired_size = 1
    max_size     = 1
    min_size     = 1
  }
}
```

DEMO

Helm 3

“Helm is the best way to find, share, and use software built for **Kubernetes**.”

Summary

- containers enable developer productivity and high levels of portability
- Kubernetes provides users the ability to easily deploy and manage container based applications
- Cloud platforms can help teams bootstrap clusters faster than ever
- Infrastructure as Code (IAC) provides a repeatable and transparent way to provision infrastructure
- Helm is a great tool to deploy applications on to Kubernetes

Thank You!

Q & A