ArcGIS Enterprise on Kubernetes Introduction

Shreyas Shinde
Subrat Bora
Markus Walker
Eva Mui
Nidhi Arora
Vaibhav Singh

2021 ESRI DEVELOPER SUMMIT

In this session we will cover…

- Technical Introduction to ArcGIS Enterprise on Kubernetes
- Key Concepts and Architecture
- Preparing for Deployment and Cloud Provisioning
- Deployment and Setup
- Enterprise Manager
- Updates and Upgrades

Key Concepts and Architecture

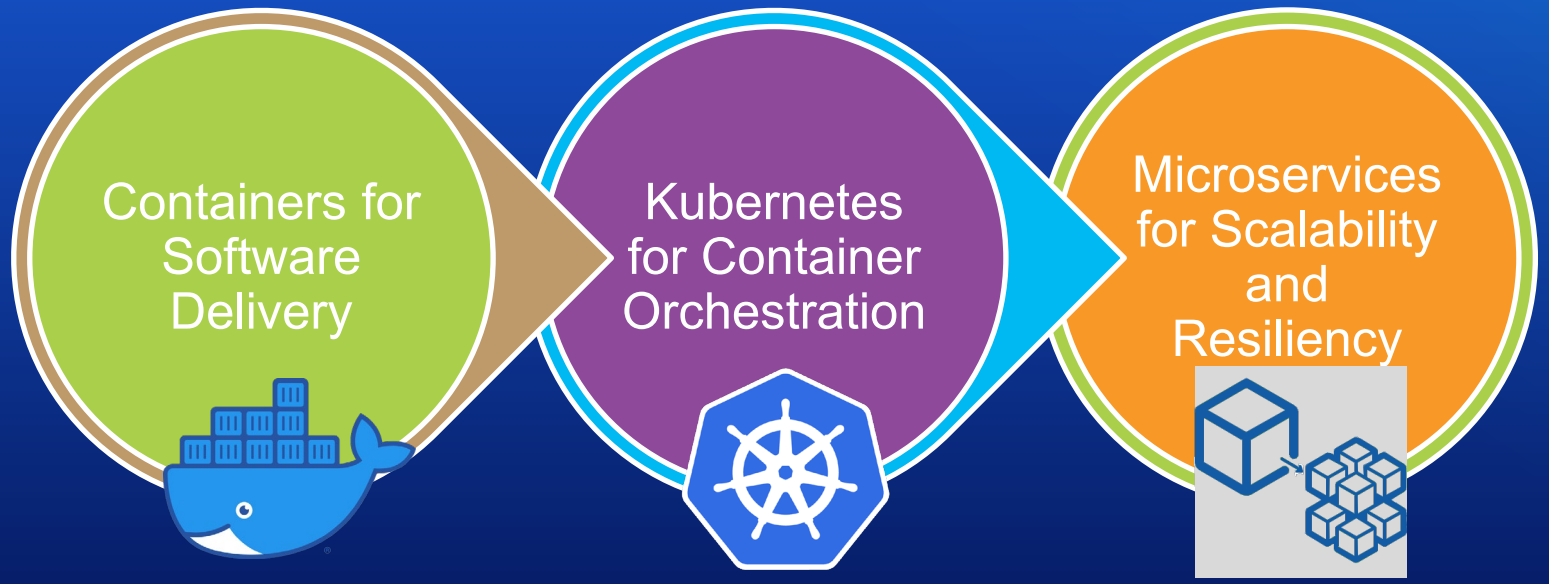
Shreyas Shinde

Vision

ArcGIS Enterprise on Kubernetes is a new cloud native deployment option for Enterprise GIS, designed to optimize scalability, resilience and maintainability.

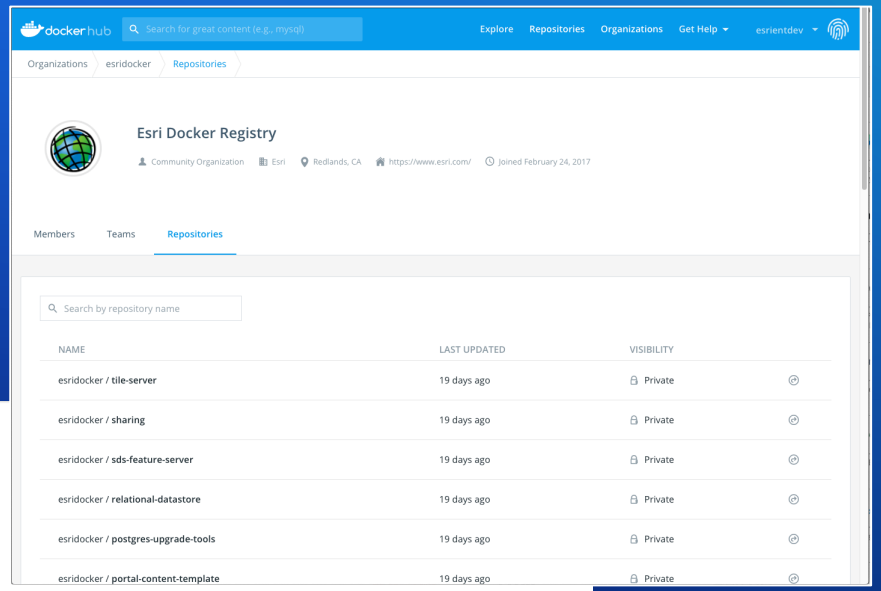
Modern microservices architecture geared for Kubernetes

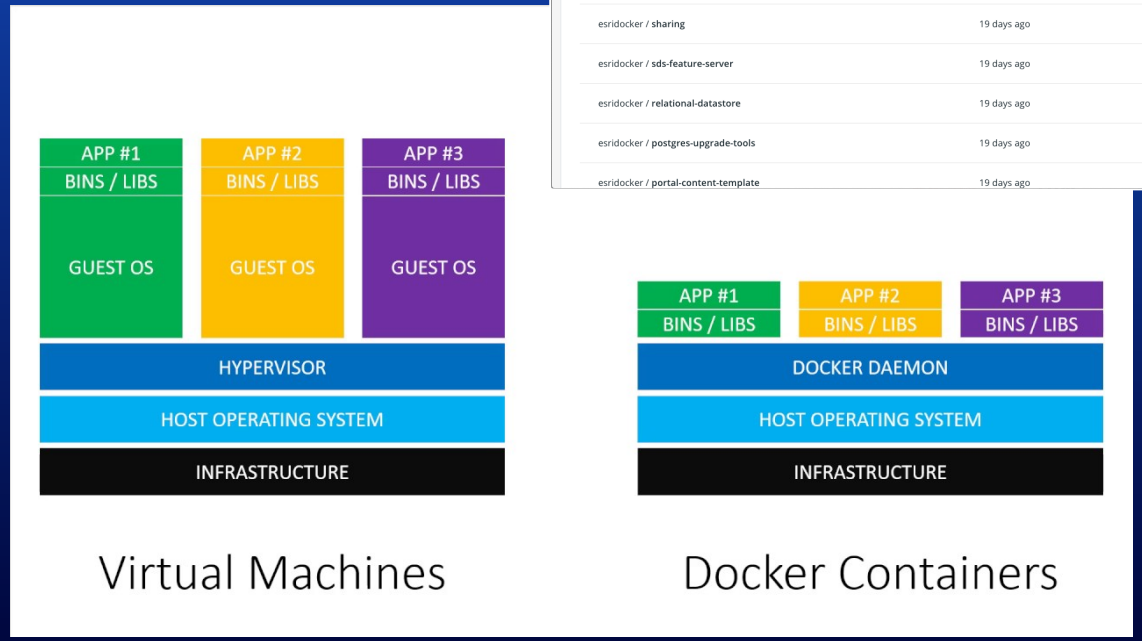
Key Concepts



Containers and Registry

- Packaging of code and configuration files as immutable images
  - Built via modern CI/CD pipelines
  - Automatable
  - Repeatable
- Distribution via Container Registry
  - Releases and patches
  - Mirrored or replicated

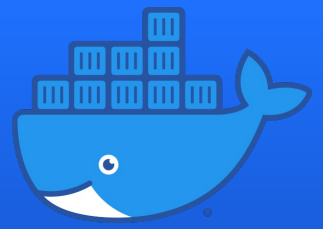




Container Orchestration: Kubernetes

- Manages containerized applications across a cluster of computers
- Google initiated, open-source
- New distributed operating system for applications
- Cloud, On-premises, Edge, …
- Strives to achieve user-defined Desired State

Kubernetes Architecture



Containers are downloaded from Container Registry

Pod is an instance of a running container

Kubernetes process called 'kubelet' runs on every node

File volumes can be attached to a pod called Persistent Volumes

Application 3 Pod

pv

Application 2 Pod

Application 1 Pod

Node

Application 2 Pod

Application 1 Pod

Node

Application 3 Pod

pv

Application 2 Pod

Node

Worker Pool

Clusters can be made up of thousands of computers, managed automatically to run containerized applications

Also called Kubernetes Master

Control Plane (Provider Managed)

Controller

etcd

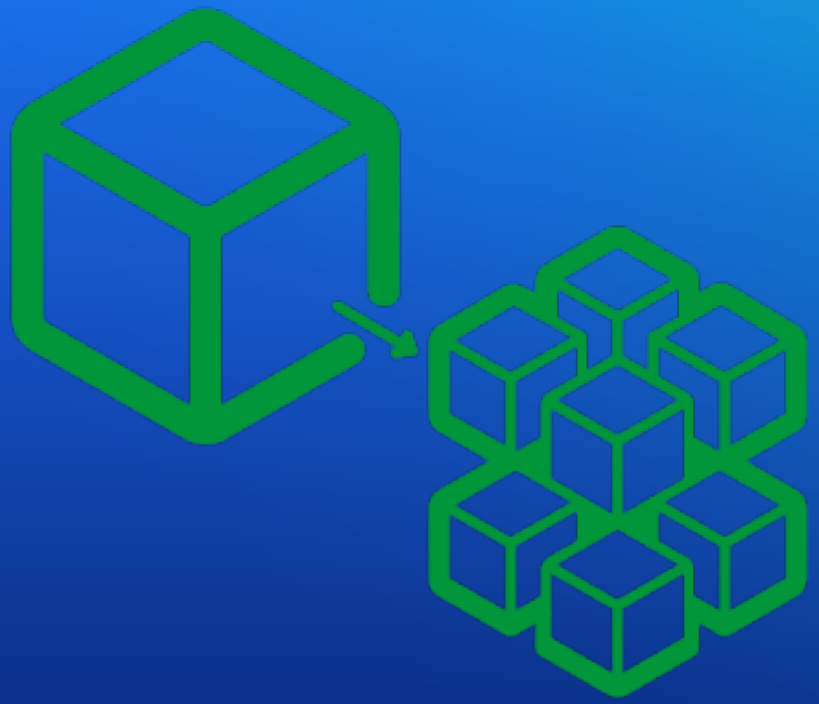
API Server

Kubectl/APIs/Dashboard

Kubernetes Cluster

Microservices

- Architectural style that's
  - Highly maintainable and testable
  - Loosely coupled
  - Independently deployable
  - Organized around business capabilities
  - Owned by a small team
- Enables
  - Rapid, frequent and reliable delivery of large, complex applications



While ArcGIS was always made up of REST APIs for GeoServices and Administration, with Enterprise on Kubernetes, we are now truly deploying them as loosely coupled processes that are individually manageable and scalable!

Cloud Native Architecture

Cloud native computing is an approach in software development that utilizes cloud computing to "build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds".[1] Technologies such as containers, microservices, serverless functions and immutable infrastructure, deployed via declarative code are common elements of this architectural style.[2][3]

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

Frequently, cloud-native applications are built as a set of microservices that run in Docker containers, and may be orchestrated in Kubernetes and managed and deployed using DevOps and Git CI workflows[4] (although there is a large amount of competing open source that supports cloud-native development). The advantage of using Docker containers is the ability to package all software needed to execute into one executable package. The container runs in a virtualized environment, which isolates the contained application from its environment.[2]

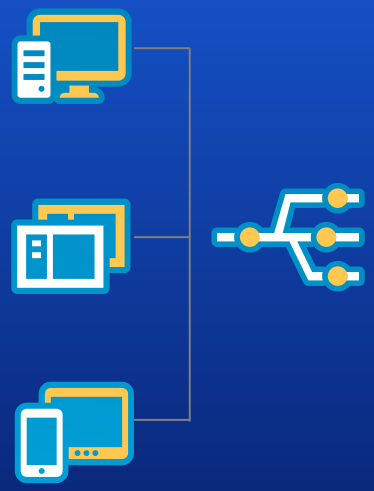
Sources: CNCF, VMWare

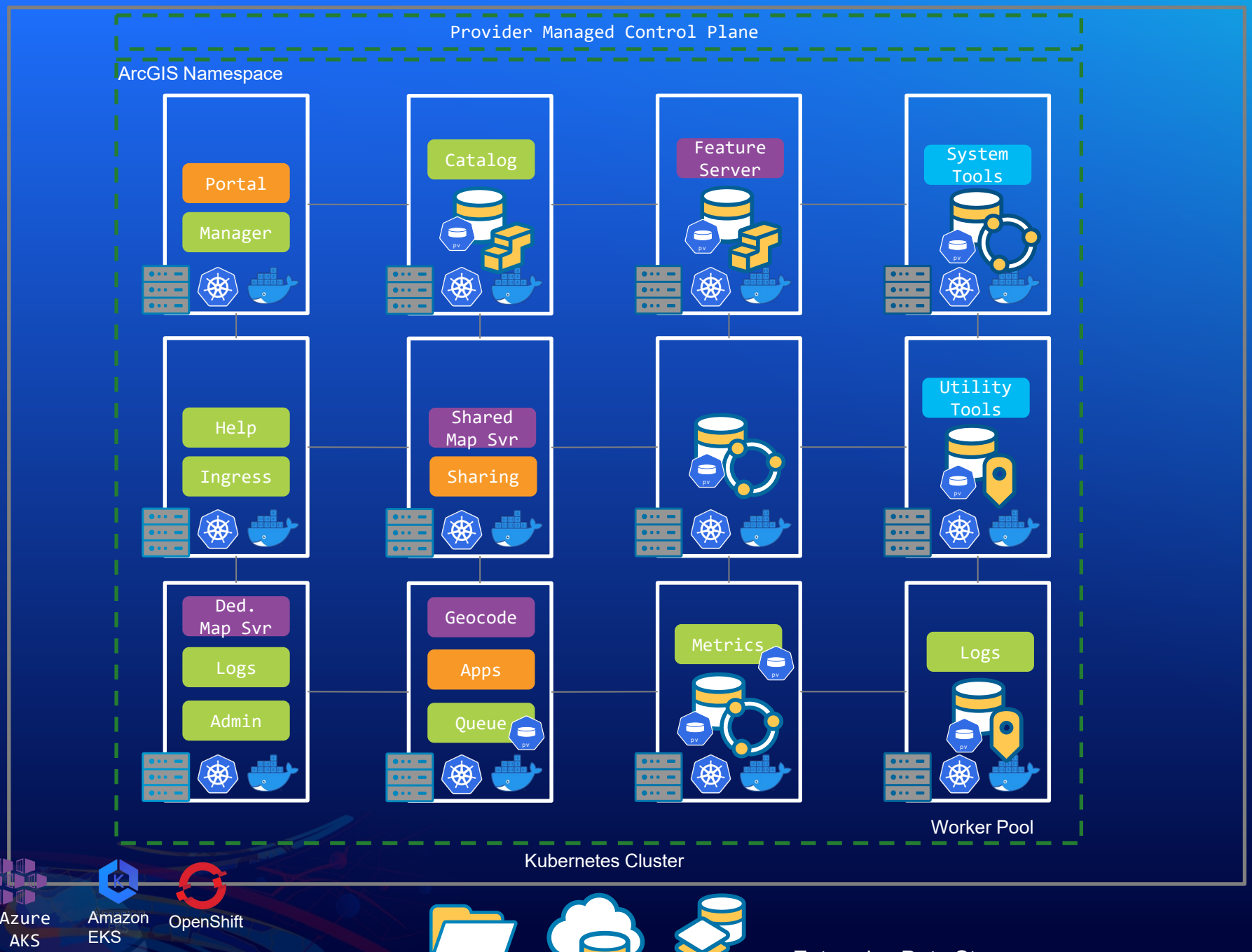
Achieving Cloud Native

- Cloud Native is a pattern with many ways to implement
- Applications taking advantage of cloud services/capabilities
  - Elasticity
  - Various managed service like Databases, Queues, Compute Engines, etc.
- Enterprise on Kubernetes aims to be cloud native by using cloud-managed services wherever possible

Our cloud native journey continues to improve with subsequent releases!

Architecture





Container Registry

Federated Servers

Hosted Data Stores

Azure AKS

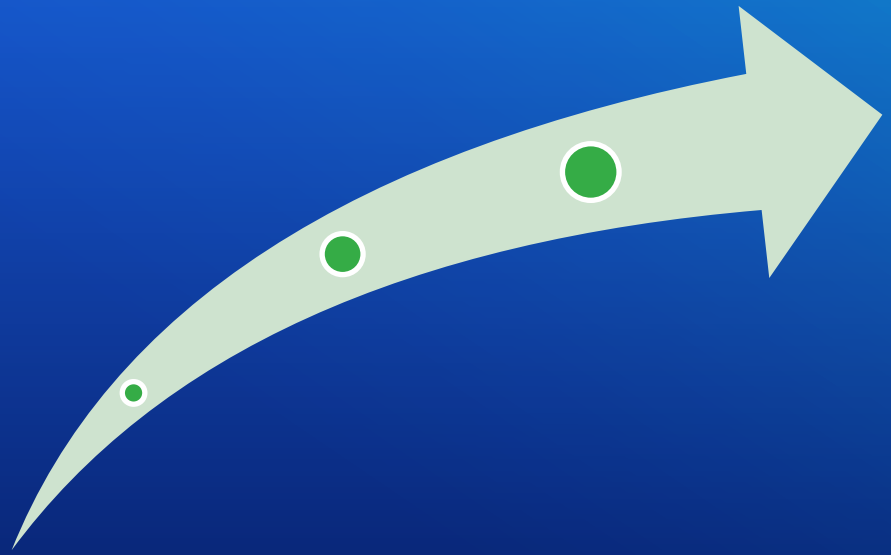
Amazon EKS

OpenShift

Enterprise Data Stores

Availability and Scale

- No single point of failure by default
- Replicas
  - Adjust pod numbers for better availability and scale
- Fine-grained resource management
  - Adjust CPU and Memory associated with a pod



Summary

- Enterprise on Kubernetes is delivered as a set of containers and installation scripts
  - We have done the work to break down capabilities as manageable and scalable microservices
- You can choose capabilities via licensing and what you publish
  - Your data continues to resides in your data stores (folders, EGDBs, cloud)
  - You cannot choose Ala carte containers and deploy them to your Kubernetes cluster
- You can federate GIS Servers (on Windows and Linux) with Enterprise on Kubernetes to compliment GIS capabilities

Preparing for Deployment

Subrat Bora

Demonstration details

- Creating a Kubernetes Cluster
  - Azure Kubernetes Service (AKS)
  - Amazon Elastic Kubernetes Service (EKS)
  - Red Hat OpenShift (on-premise data center)

- Preparing Client Workstation
  - Bash Shell
  - Provider’s Command Line Interface (CLI)
  - Kubectl
  - OpenSSL

Deployment and Setup

Markus Walker

How To Get Started?

Deployment Script: deploy.sh

- Provided to you is a *.tar.*gz file. Inside of it contains the deployment script named *deploy.sh*.
  - Additionally, this *.tar.gz* file comes with the following scripts:
    - *undeploy.sh* - > Undeploys ArcGIS Enterprise on Kubernetes
    - *configure.sh* -> Creating an Enterprise Organization.

- Minimum Pre-requisites:
  - Client machine with a bash shell
  - Kubernetes cluster that meets system requirements

One Deployment Script, One Deployment Experience

- *deploy.sh* gives the same experience across all supported Kubernetes providers.

- This includes the following:
  - Red Hat OpenShift
  - Amazon EKS
  - Azure AKS

Deployment Is Finished, What's Next?

- After a successful deployment, it is time to configure an Enterprise Organization.
- The new ArcGIS Enterprise Manager Setup Wizard allows us to do just this.
  - Similar experience to how ArcGIS Enterprise on Windows/Linux.

ArcGIS Enterprise

Reverse Proxy

Ingress

Map Server
Utilities Tools
Help
Kubernetes
Docker

Object Store
Kubernetes
Docker

System Tools
Utility Tools
Kubernetes
Docker

Relational
Kubernetes
Docker

Metrics
Catalog
Kubernetes
Docker

Feature Server
Apps
Logs
Kubernetes
Docker

Geocode
Shared Map
Kubernetes
Docker

Object Store
Kubernetes
Docker

Spatio Temporal
Kubernetes
Docker

Object Store
Kubernetes
Docker

Portal
Admin
Kubernetes
Docker

Sharing
Admin
System Tools
Kubernetes
Docker

Queue
Manager
Kubernetes
Docker

Relational
Kubernetes
Docker

Spatio Temporal
Kubernetes
Docker

Persistent Volume Controller

Volumes

Summary

- Remember: Deploy, Configure, Use!
- One deployment script, one deployment experience.
- Configure an Enterprise Organization using the Setup Wizard in Enterprise Manager.
- Silent deployment and configuration is supported.

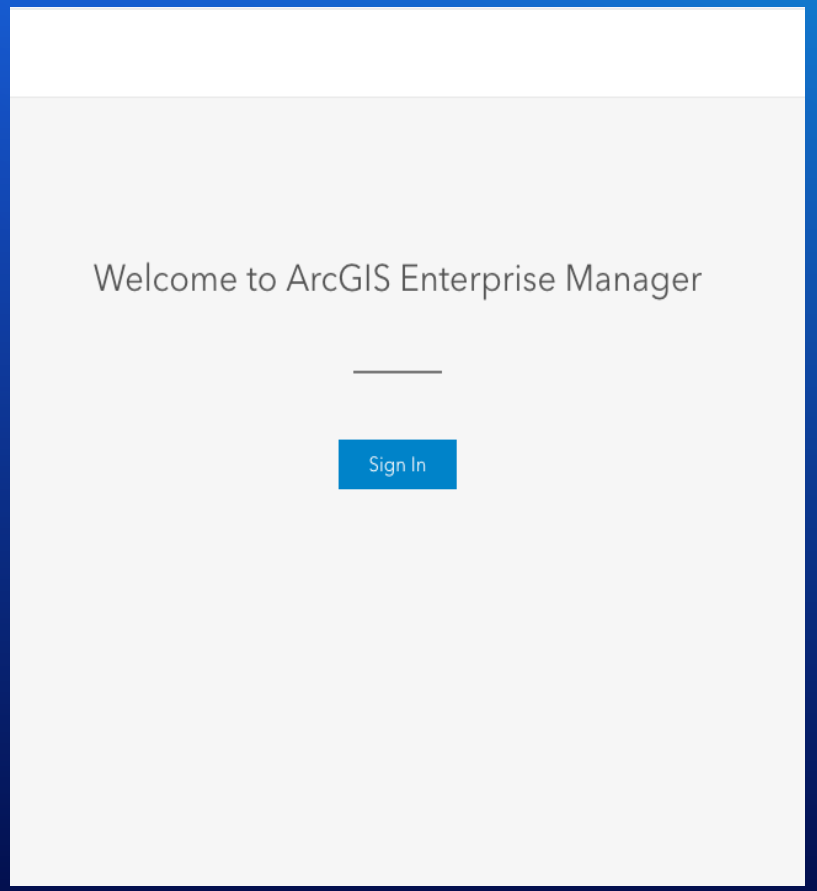
Enterprise Manager

Eva Mui

Working With ArcGIS Enterprise on Kubernetes

ArcGIS Enterprise Manager

ArcGIS Enterprise system management tool



Initial Deployment

Ingress

Kubernetes
Docker

Admin

Manager

Help

Persistent Volume Controller

Volumes

Who We Have in Mind?

Azure AKS

Amazon EKS

OpenShift

ArcGIS Enterprise Manager

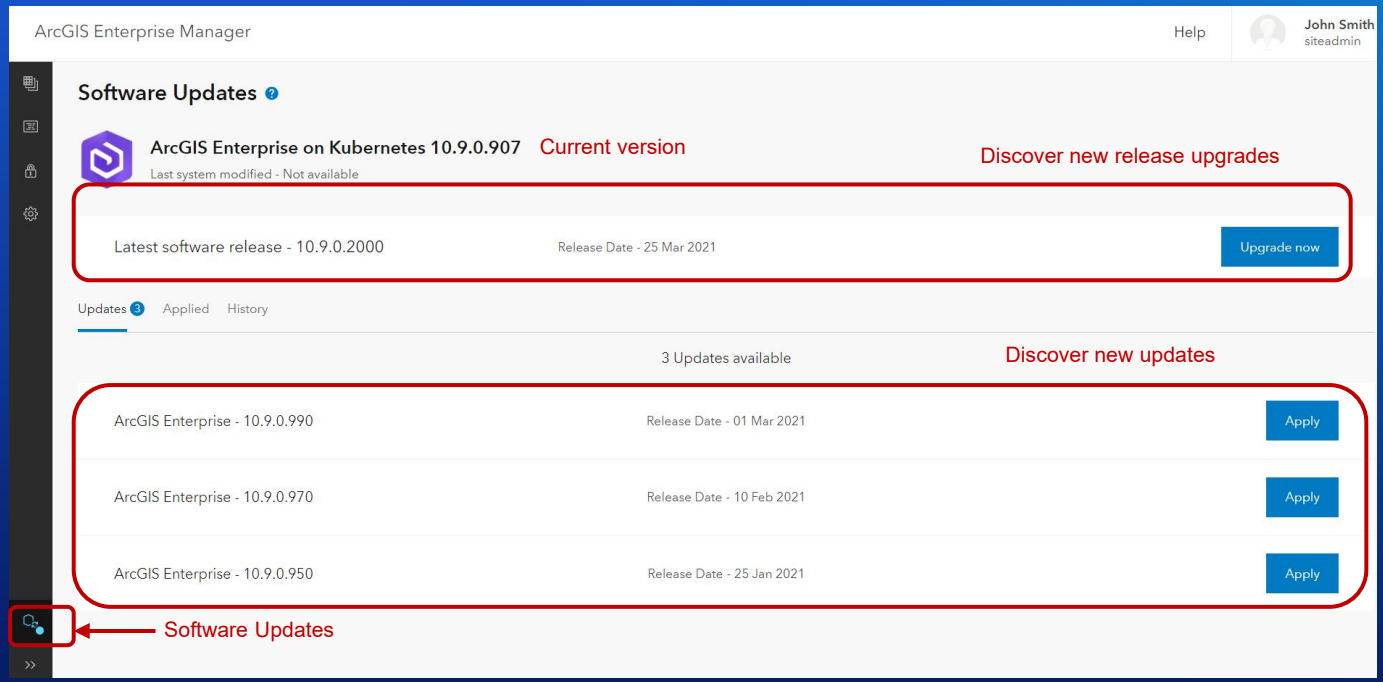
- Offer unified administrative experience to manage enterprise system
- Provide intuitive user experience around common administrative workflows
- Enhance observability to the health of the system
- Identify and rectify system problems in a timely manner
- Discover and apply updates and upgrade to the system seamlessly
- Localize ArcGIS Enterprise Manager to support international languages

Update and Upgrade

Nidhi Arora and Vaibhav Singh

Discover and Apply Updates and Upgrades

ArcGIS Enterprise Manager provides unified, intuitive and easy experience to discover and apply updates and upgrades

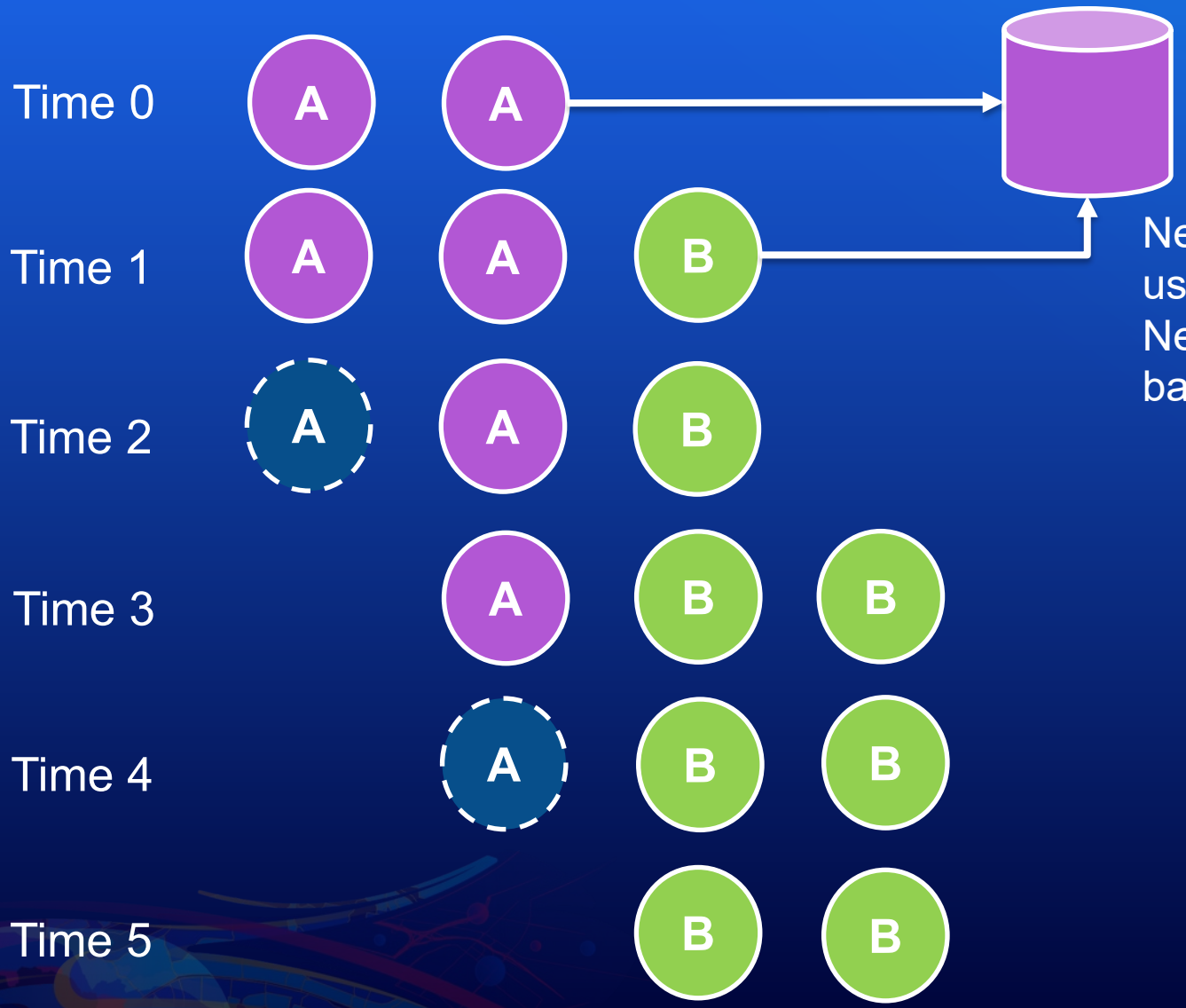


What is our Upgrade Strategy?

ArcGIS Enterprise upgrade on Kubernetes employs industry accepted software deployment strategies for containerized applications to provide seamless upgrades and patching at scale.

Stateless Pods(Home App, JSAPI etc.)	Stateful Pods(Portal, Relational data stores etc.)
• Rolling Update	• Blue-Green Update

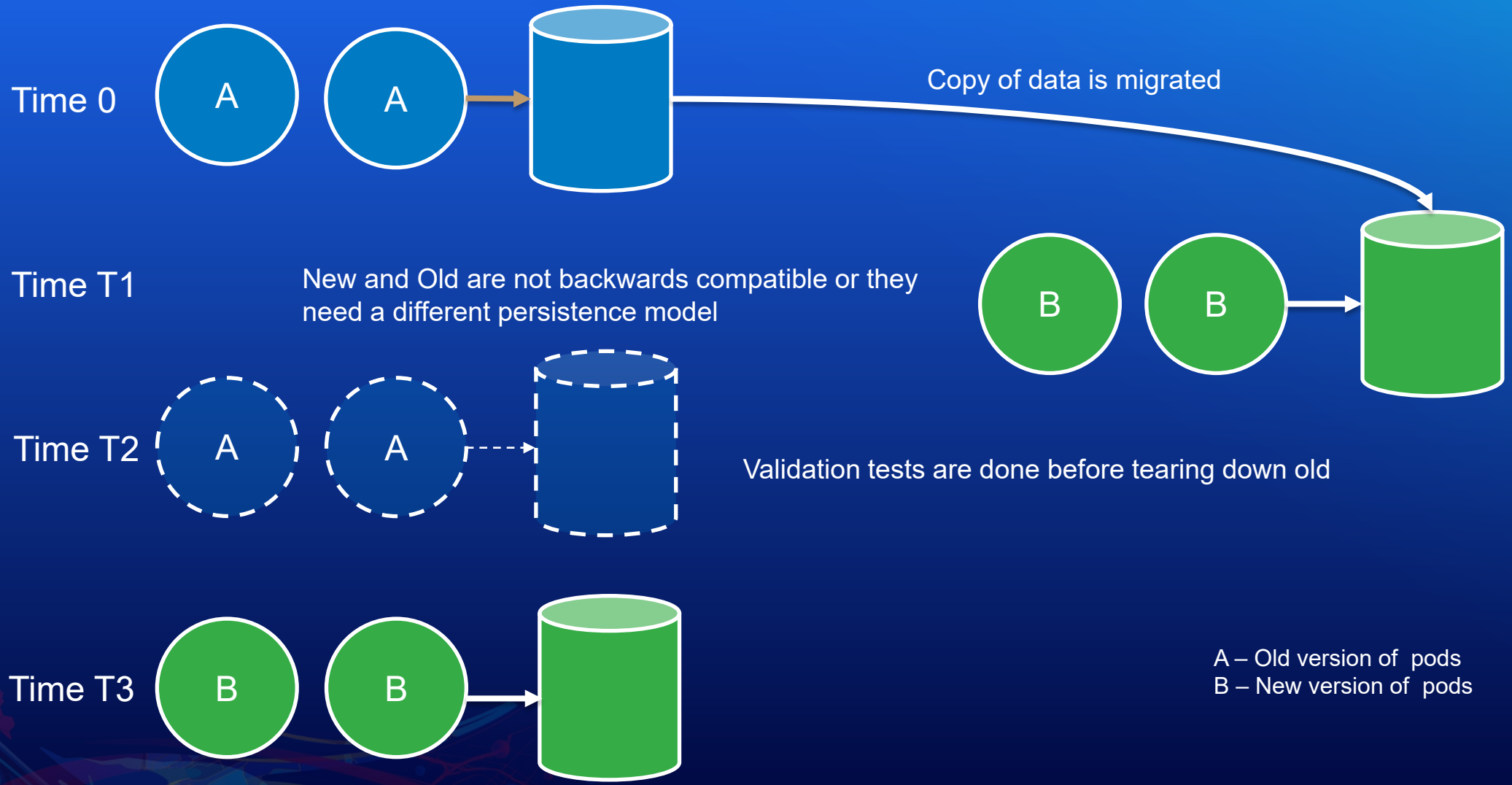
Rolling Update for Stateless Pods



New and Old **must** continue to use the same persistence model. New code must be fully backwards compatible with old.

A – Old version of pods
B – New version of pods

Blue-Green Update for Stateful Pods



Key Takeaways

- ✓ Apply upgrade and updates is based upon rolling and blue green deployment practices
- ✓ Discover and apply upgrade and updates using Enterprise Manager
- ✓ Put the system into read only mode during upgrade and updates
- ✓ Support one release upgrade at a time
- ✓ Apply updates is cumulative and can be rolled back
- ✓ Rolling back upgrades is not supported. Take a backup!

In this session we covered…

- Technical Introduction to ArcGIS Enterprise on Kubernetes
- Key Concepts and Architecture
- Preparing for Deployment and Cloud Provisioning
- Deployment and Setup
- Enterprise Manager
- Updates and Upgrades

Please provide your feedback for this session by clicking on the session survey link directly below the video.