Toward Cloud Native Digital Repositories

Yinlin Chen, Lee Hunter, Zhiwu Xie

{ylchen, whunter, zhiwuxie}@vt.edu

Research and Informatics
Virginia Tech Libraries

Agenda

- Virginia Tech Libraries direction and goals
- Monolithic and Cloud native application
- Cloud native approaches
- Architecture design strategies
- Challenges, design pattern and best practice
- Cloud platforms

VT Libraries Direction and Goals

- Improve development and deployment process
  - Continuous Integration (CI) / Continuous Delivery (CD)
  - Increase the frequency of new service/version release
  - Local and cloud environment
- Cloud native applications for core repository infrastructure
  - Cloud native digital repositories

Monolithic Architecture

- Develop and deploy as a single unit
- Long-term commitment to a technology stack or even version
- Hard to scale development
- Difficult to scale the application
- Lots of human intervention

Why Cloud Native

- Limited resource:
  - Developers, Devops, Infrastructure, Time
- Facilitate the development and delivery process
- Provide better services: fault-tolerant, auto-scale, update/rollback without downtime, etc
- Optimize resource usage
- Use services that can help delivering the project, not build everything by ourselves

Toward Cloud Native

It is not just putting applications in the cloud

Toward Cloud Native

It is not just putting applications in the cloud

It is about applications in the cloud that utilize the **advantages** provided by the cloud

AS MUCH AS POSSIBLE

Toward Cloud Native

It is not just putting applications in the cloud

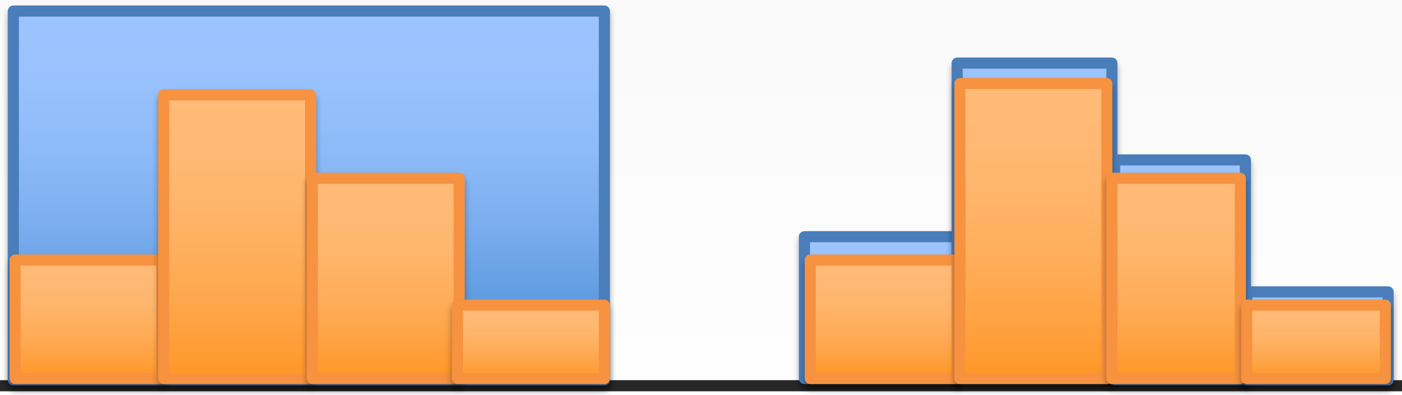
It is about applications in the cloud that utilize the **advantages** provided by the cloud

AS MUCH AS POSSIBLE

Do things that matters

Resource Usage Optimization and Automation

- Consume only the required resources for the applications
- Scale up and down automatically
- Services and functions oriented, not server oriented
- Utilize cloud services to help understanding your applications (CloudWatch, Auto Scaling, Trusted Advisor, etc)

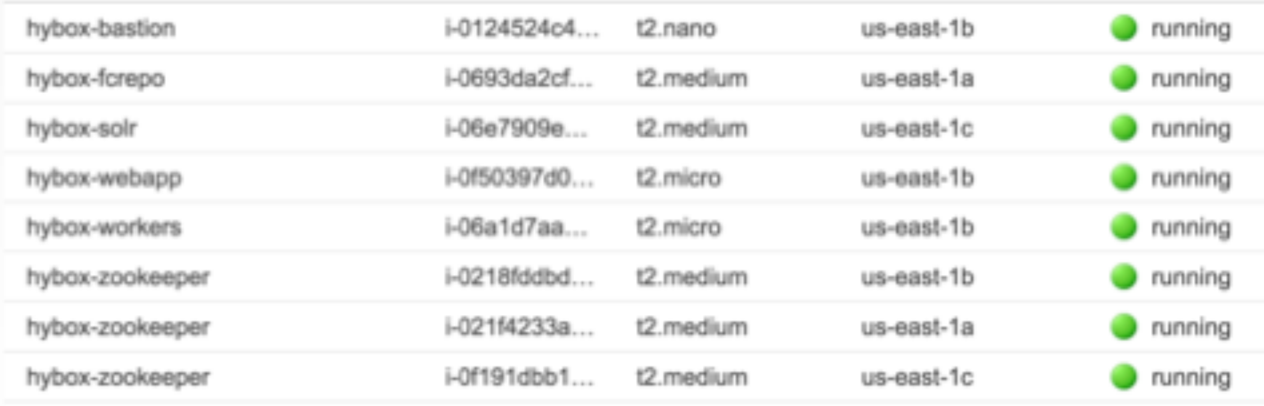


Example: Hydra-in-a-box

- Hyku using the configuration defined in AWS cloudformation templates is roughly $800-$900 per month (https://github.com/hybox/aws)

- $300/month setting

- Reserved instance (50 - 75% off) $75/month



hybox-bastion	i-0124524c4...	t2.nano	us-east-1b	● running
hybox-fcrepo	i-0693da2cf...	t2.medium	us-east-1a	● running
hybox-solr	i-06e7909e...	t2.medium	us-east-1c	● running
hybox-webapp	i-0f50397d0...	t2.micro	us-east-1b	● running
hybox-workers	i-06a1d7aa...	t2.micro	us-east-1b	● running
hybox-zookeeper	i-0218fddbd...	t2.medium	us-east-1b	● running
hybox-zookeeper	i-021f4233a...	t2.medium	us-east-1a	● running
hybox-zookeeper	i-0f191dbb1...	t2.medium	us-east-1c	● running

VIRGINIA TECH.

Cloud Native

- Cloud Native Computing Foundation (CNCF)
  – An open source software foundation dedicated to making cloud native computing universal and sustainable.
- Microservices oriented
- Containerized
- Dynamically orchestrated

Microservices oriented

Microservice

- Small software piece
- Decentralized
  - Autonomously developed
  - Independently deployable
  - Change independently of each service
  - Scale individually by load
- Messaging enabled – communicate with messages
- Build and released with automated processes
- More complex architecture

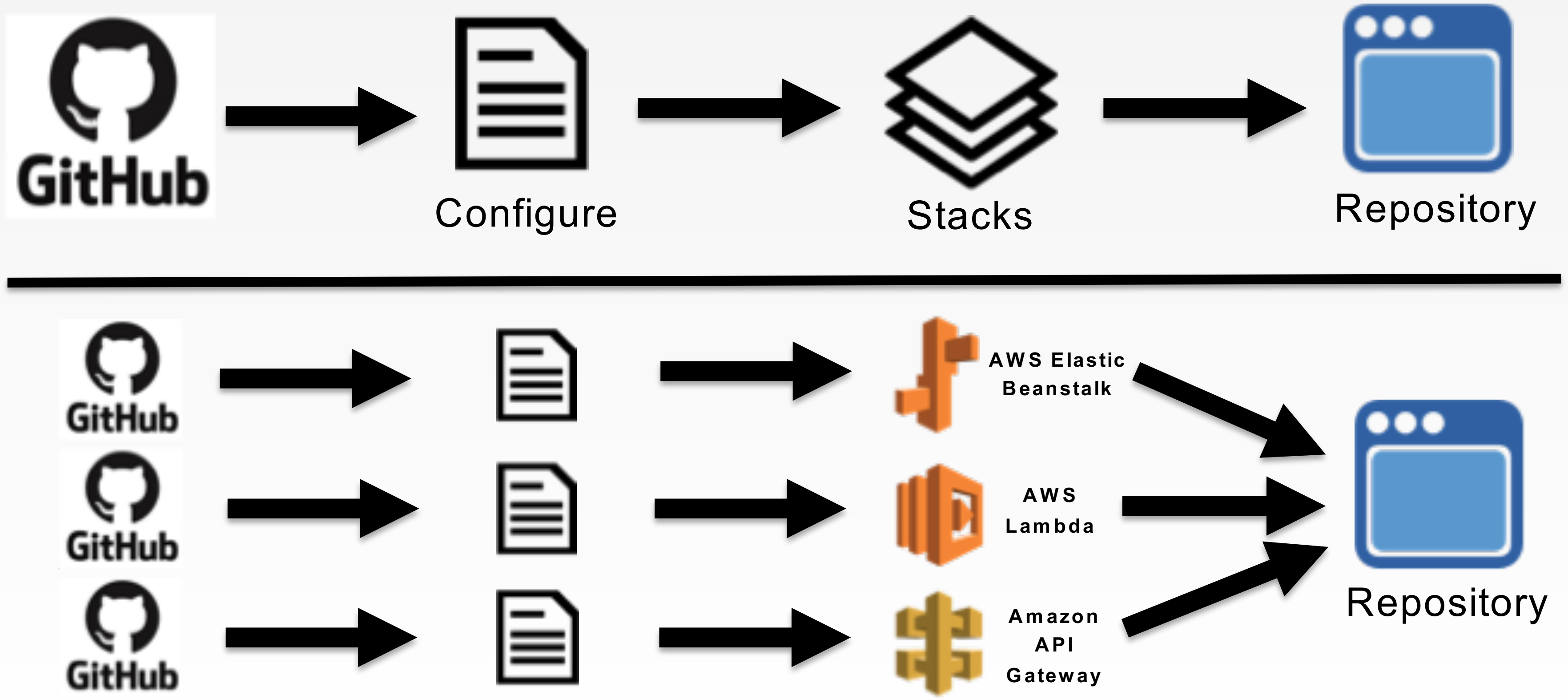
Serverless

Do not mean ”There are no servers at all”

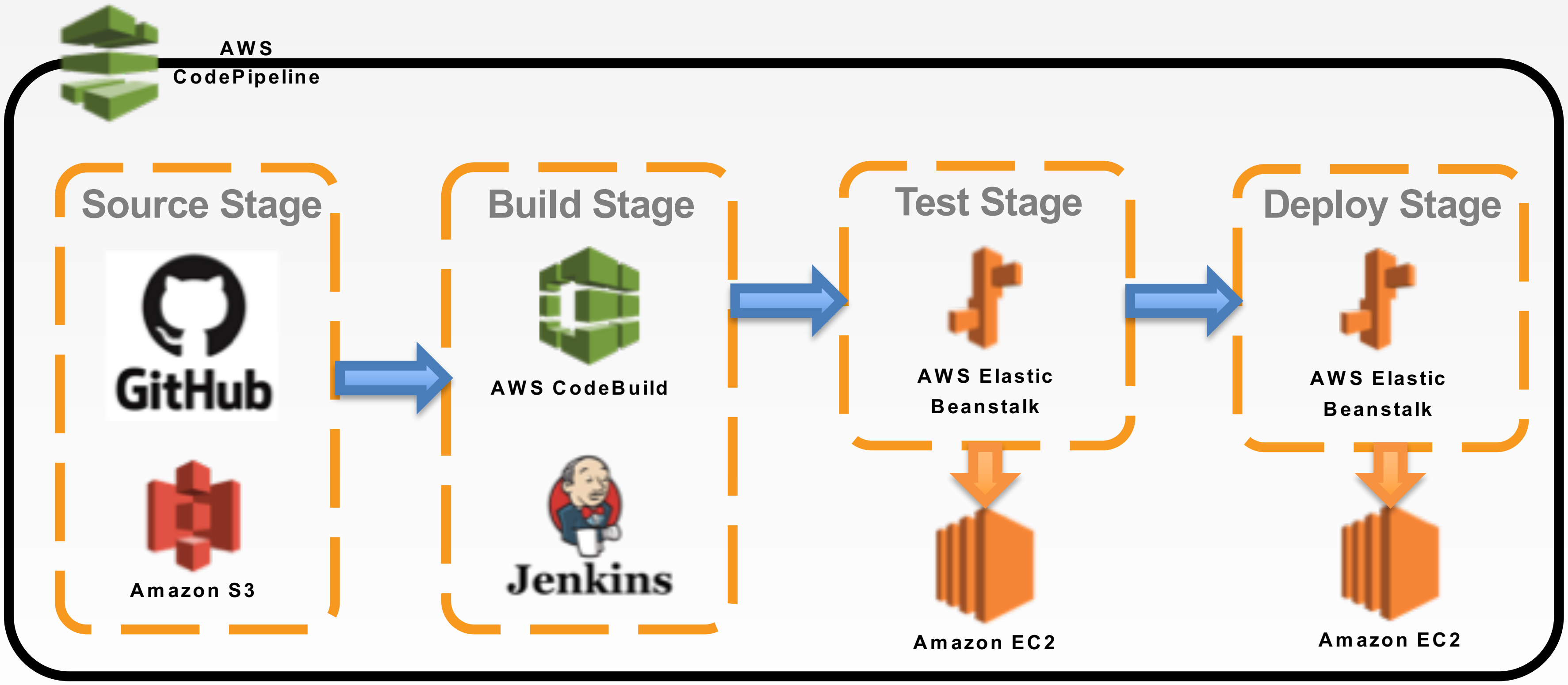
Do mean “Use fully managed services”

Focus on application development,
not server maintenance

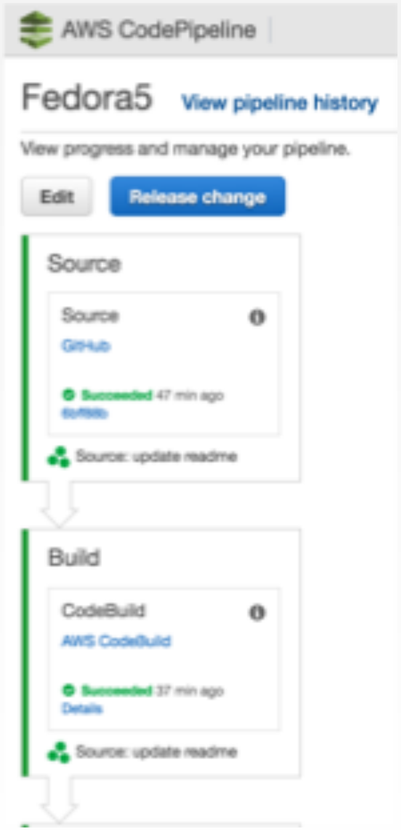
Parallel Development and Deployment



Continuous Integration and Delivery (CI / CD)



Example: Fedora 5 CI/CD



AWS CodePipeline

Fedora5 View pipeline history

View progress and manage your pipeline.

Edit | Release change

Source

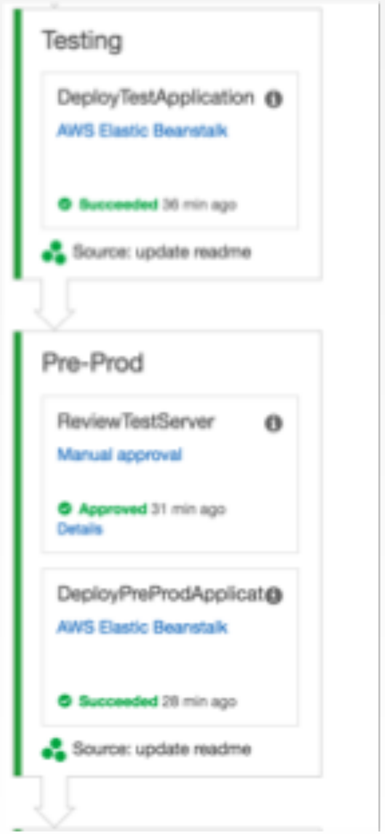
Source
GitHub
Succeeded 47 min ago
6off56b

Source: update readme

Build

CodeBuild
AWS CodeBuild
Succeeded 37 min ago
Details

Source: update readme



Testing

DeployTestApplication
AWS Elastic Beanstalk
Succeeded 36 min ago

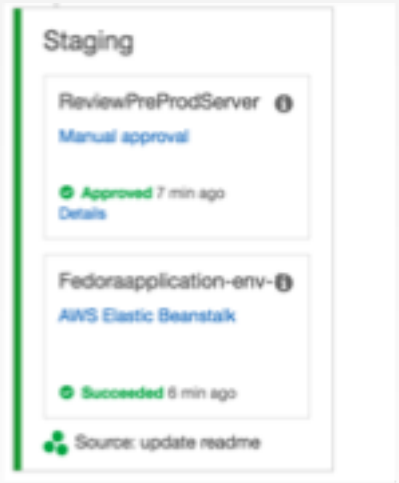
Source: update readme

Pre-Prod

ReviewTestServer
Manual approval
Approved 31 min ago
Details

DeployPreProdApplicat
AWS Elastic Beanstalk
Succeeded 28 min ago

Source: update readme



Staging

ReviewPreProdServer
Manual approval
Approved 7 min ago
Details

Fedoraapplication-env-
AWS Elastic Beanstalk
Succeeded 6 min ago

Source: update readme

Containerized

VIRGINIA TECH.

Container as a Service (CaaS)

EVERYTHING at Google runs in a container

2 Billion containers per week in 2014

4 Billion containers per week in 2018

Fedora 4 Containerization

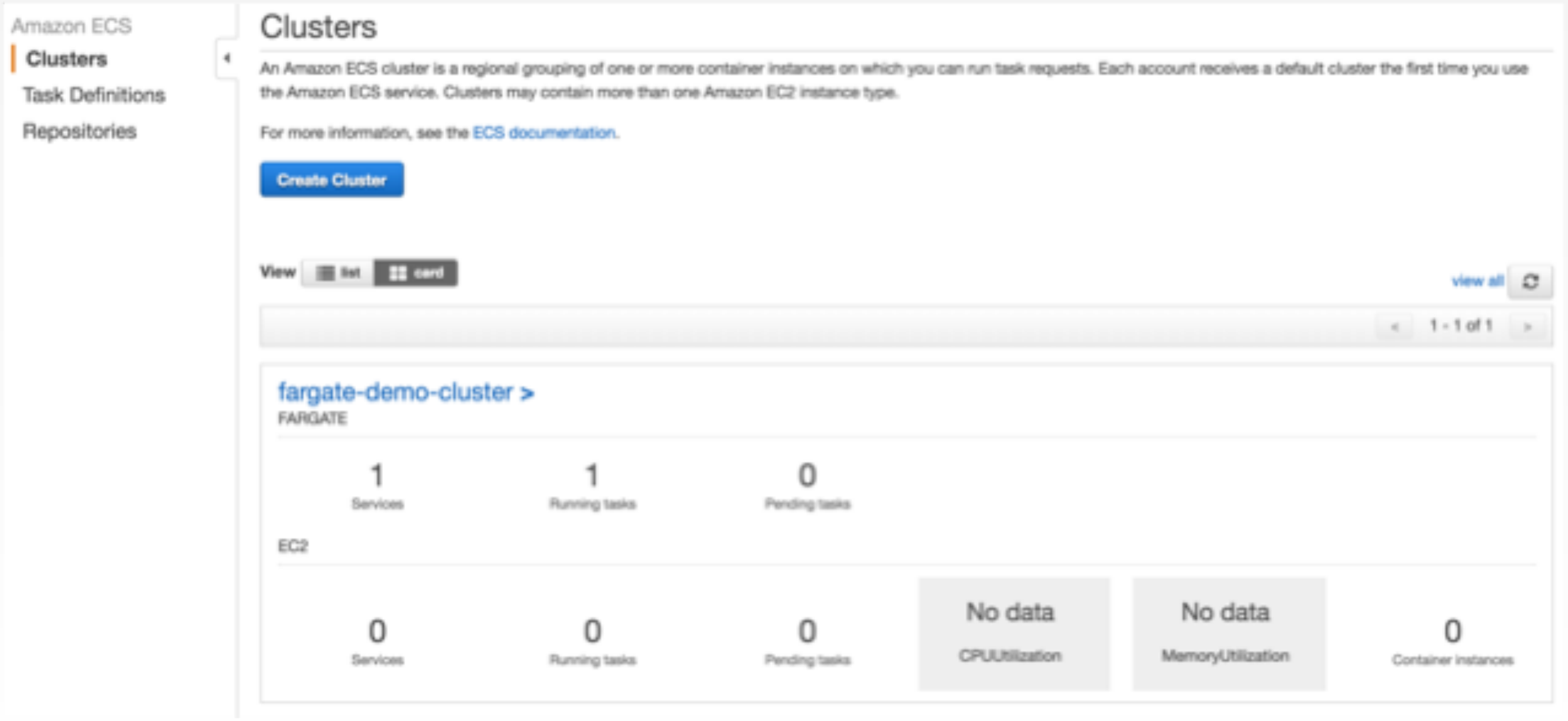
- Create a Fedora 4 Docker image
- Push to Amazon Elastic Container Registry (ECR)
- Run containerized application in Amazon Elastic Container Service (ECS)
- Run containerized application in AWS Fargate

Amazon ECR

Amazon ECS

AWS Fargate

Example: Fedora 4 Docker in AWS Fargate



Amazon ECS

Clusters

Task Definitions

Repositories

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the ECS documentation.

Create Cluster

View list card

view all

1 - 1 of 1

fargate-demo-cluster >

FARGATE

1	1	0
Services	Running tasks	Pending tasks

EC2

0	0	0	No data	No data	0
Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances

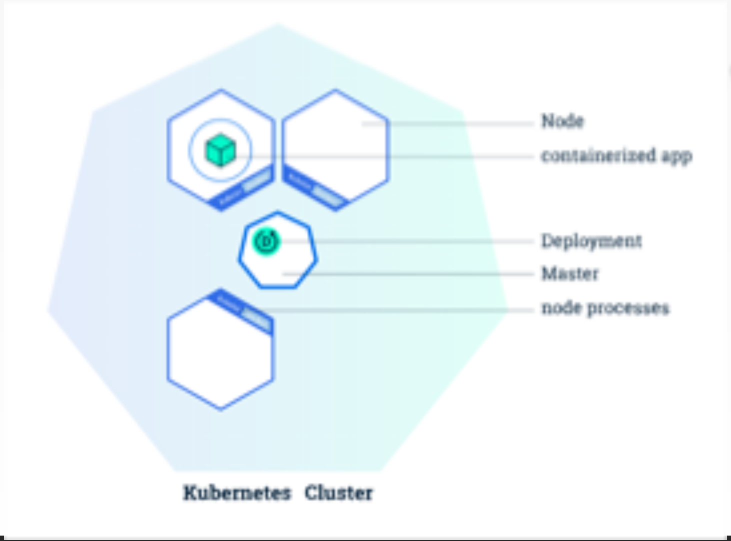
Dynamically orchestrated

Orchestration Platforms

- Apache Mesos
  - http://mesos.apache.org/
- Docker Swarm
  - https://docs.docker.com/engine/swarm/
- Kubernetes
  - https://kubernetes.io/
- Nomad
  - https://www.nomadproject.io/
- Rancher
  - https://rancher.com/

Kubernetes (a.k.a k8s)

- An open-source system for automating deployment, scaling, and management of containerized applications
- Manage containers at scale



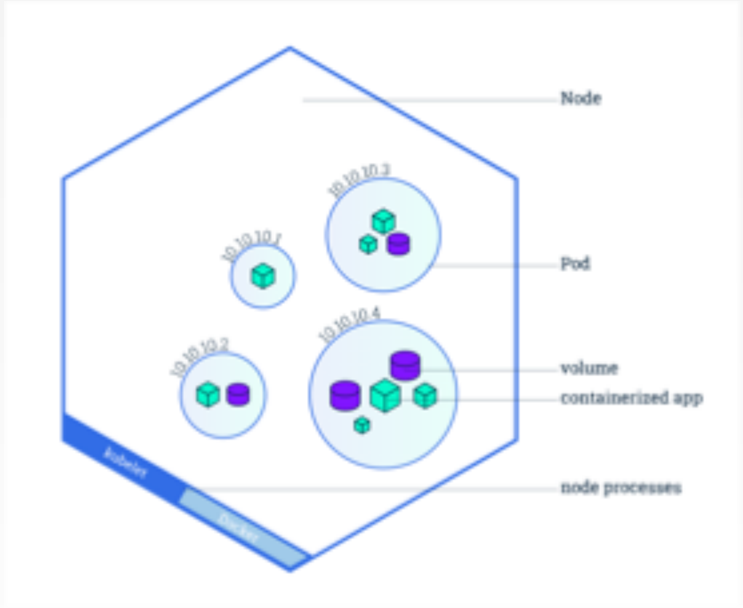


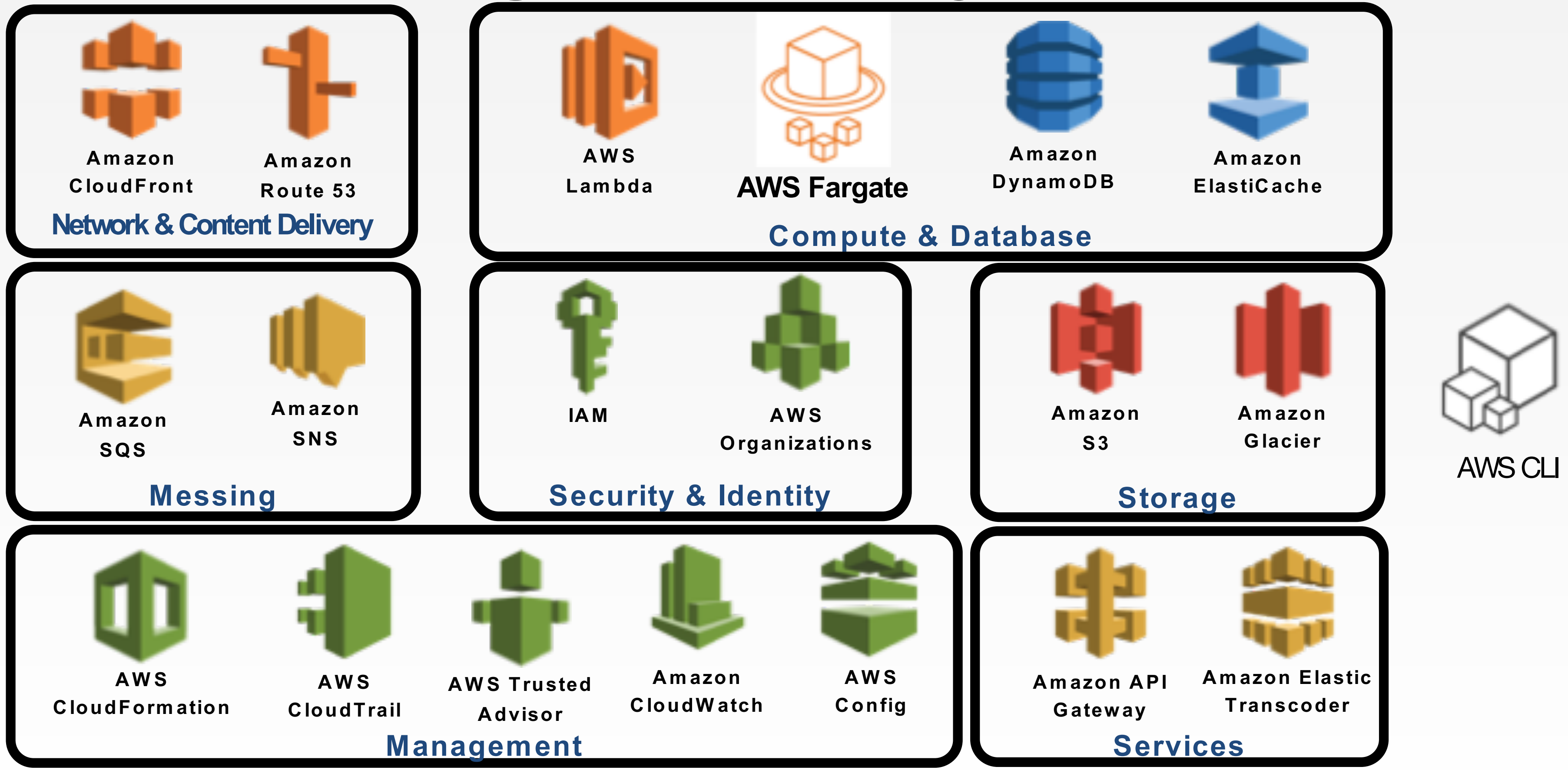
Image credits: https://kubernetes.io/

Kubernetes in the Cloud

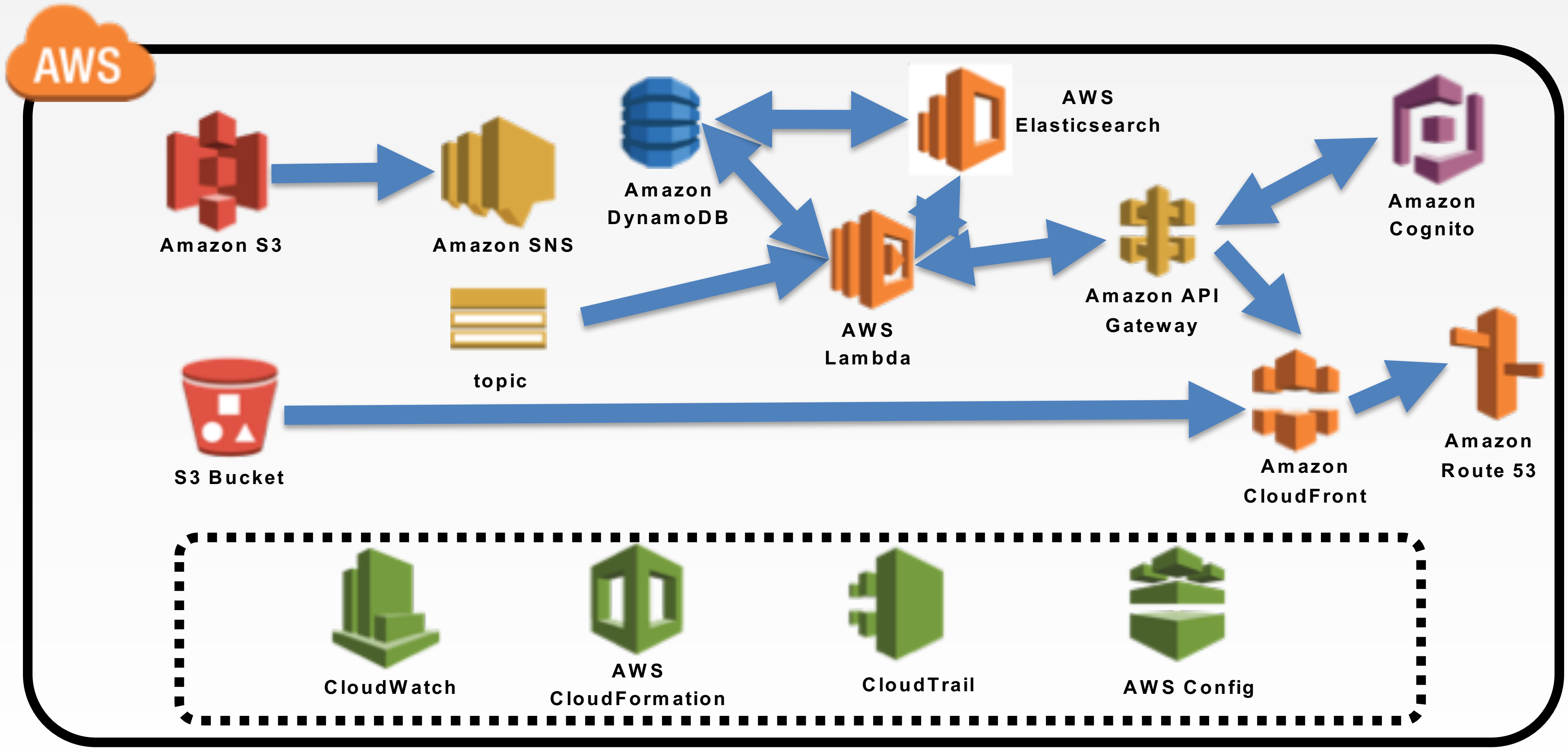
- Kubernetes on AWS using CloudFormation
  - Weaveworks
  - Heptio
- Google Kubernetes Engine (GKE)
- Amazon Elastic Container Service for Kubernetes (EKS)
- Microsoft Azure Kubernetes Service (AKS)

kubernetes weaveworks heptio GKE Amazon EKS AKS

Cloud Native Digital Repository in AWS



Example: Multimedia Digital Repository



Architecture Design Strategies

- Decouple digital repository into multiple services
- Chose the right tools (Service, Instance, Storage, etc)
- Go Microservice and Serverless:
  - Containerize the service
  - Use managed service
- Develop orchestration

Challenges

- Service granularity
- More complex architecture
- More things need to learn
- Learning curve varies
- Practical cloud experience
- Cloud investment

Design Pattern and Best Practice

- The Twelve-Factor App (http://12factor.net)
- Applying the Twelve-Factor App Methodology to Serverless Applications (https://goo.gl/TBLbhG)

Codebase	**Dependencies**	**Config**
Backing services	**Build, release, run**	**Processes**
Port binding	**Concurrency**	**Disposability**
Dev/prod parity	**Logs**	**Admin processes**

Other Cloud Platforms

- Cloud platforms
  - Amazon Web Services (AWS)
  - Google Cloud Platform (GCP)
  - Microsoft Azure, and etc.

AWS	GCP	Azure
Elastic Compute Cloud	Compute Engine	Virtual Machines
Elastic Beanstalk	Google App Engine	Cloud Services
EC2 Container Service Kubernetes (EKS)	Kubernetes Engine	Container Service (AKS)
Lambda	Cloud Functions	Functions
Simple Storage Services	Cloud Storage	Storage
Virtual Private Cloud	Virtual Private Cloud	Virtual Network

VIRGINIA TECH.

Q & A

Supported by Virginia Tech Libraries – Beyond Boundaries project and AWS Cloud Credits for Research program

Yinlin Chen
ylchen@vt.edu