



CINCINNATI CLOUD NATIVE

Shad Beard • Feb 2018

Twitter: @CloudCincy, @dahs81

CNCF

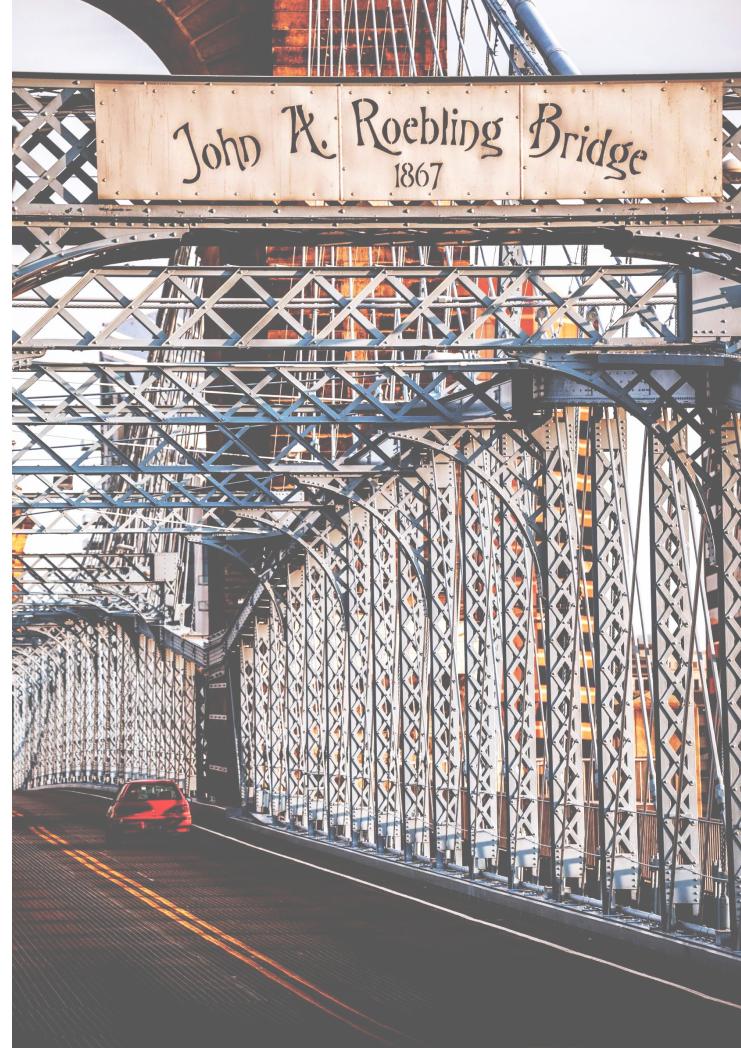
A Linux Foundation organization
created in 2015 to host projects in the
microservices and cloud space.



CNCF Projects...

Cloud Native Computing Foundation (CNCF) (29)

| | | | | | | | |
|---|---|---|---|--|---|--|---|
|  cloudevents CloudEvents Cloud Native Computing Foundation (CNCF) ★ 635 |  CNI Container Network Interface (CNI) Cloud Native Computing Foundation (CNCF) ★ 1,543 |  CSI Container Storage Interface (CSI) Cloud Native Computing Foundation (CNCF) ★ 261 |  containerd Cloud Native Computing Foundation (CNCF) ★ 2,034 |  CoreDNS CoreDNS Cloud Native Computing Foundation (CNCF) ★ 2,585 |  cri-o CRI-O Cloud Native Computing Foundation (CNCF) ★ 1,280 |  envoy Envoy Cloud Native Computing Foundation (CNCF) ★ 6,557 |  fluentd Fluentd Cloud Native Computing Foundation (CNCF) ★ 6,766 |
|  gRPC gRPC Cloud Native Computing Foundation (CNCF) ★ 17,148 |  HARBOR Harbor Cloud Native Computing Foundation (CNCF) ★ 5,377 |  HELM Helm Cloud Native Computing Foundation (CNCF) ★ 7,610 |  JAEGER Jaeger Cloud Native Computing Foundation (CNCF) ★ 5,817 |  kubernetes Kubernetes Cloud Native Computing Foundation (CNCF) ★ 41,500 |  LINKERD Linkerd Cloud Native Computing Foundation (CNCF) ★ 4,671 |  minikube Minikube Cloud Native Computing Foundation (CNCF) ★ 10,801 |  NATS NATS Cloud Native Computing Foundation (CNCF) ★ 4,598 |
|  Notary Notary Cloud Native Computing Foundation (CNCF) ★ 1,516 |  Open Policy Agent Open Policy Agent (OPA) Cloud Native Computing Foundation (CNCF) ★ 1,426 |  OPENMETRICS OpenMetrics Cloud Native Computing Foundation (CNCF) ★ 328 |  OPENTRACING OpenTracing Cloud Native Computing Foundation (CNCF) ★ 1,350 |  Prometheus Prometheus Cloud Native Computing Foundation (CNCF) ★ 18,927 |  rkt rkt Cloud Native Computing Foundation (CNCF) ★ 8,204 |  ROOK Rook Cloud Native Computing Foundation (CNCF) ★ 3,431 |  spiffe SPiffe Cloud Native Computing Foundation (CNCF) ★ 291 |
|  SPIRE Spire Cloud Native Computing Foundation (CNCF) ★ 251 |  Telepresence Telepresence Cloud Native Computing Foundation (CNCF) ★ 1,175 |  The Update Framework (TUF) The Update Framework (TUF) Cloud Native Computing Foundation (CNCF) ★ 785 |  TiKV TiKV Cloud Native Computing Foundation (CNCF) ★ 3,713 |  Vitess Vitess Cloud Native Computing Foundation (CNCF) ★ 6,539 | | | |

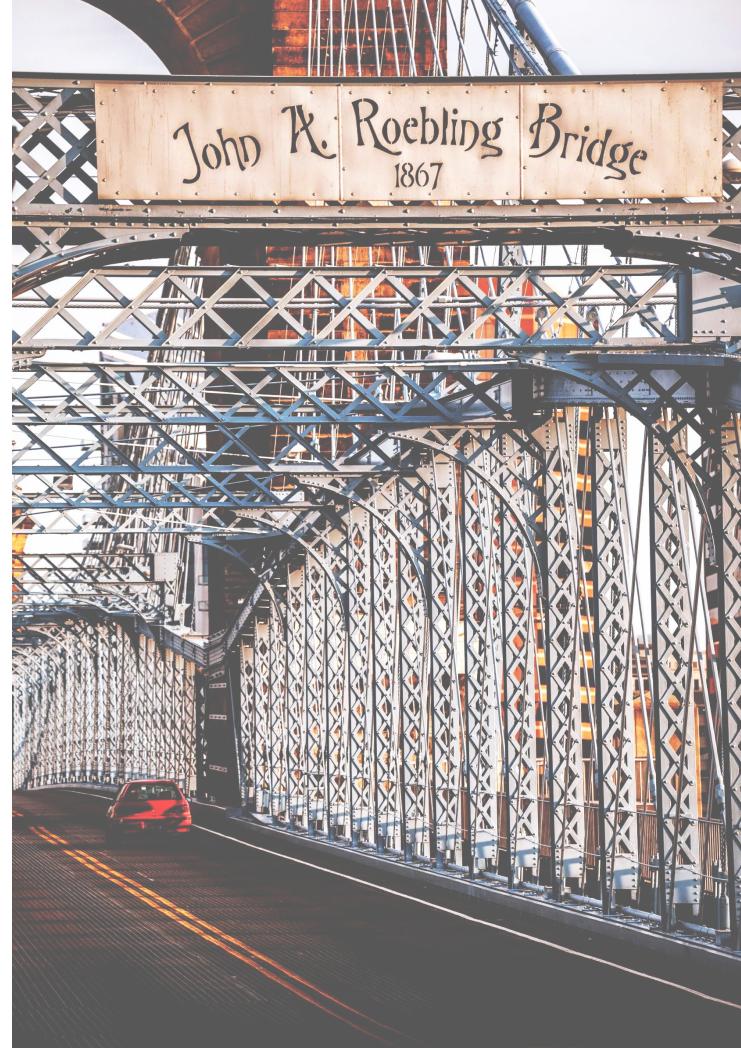


What is Cloud Native? - CNCF Definition

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

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.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.



What is Cloud Native? Continued...

How I define it for this meetup:

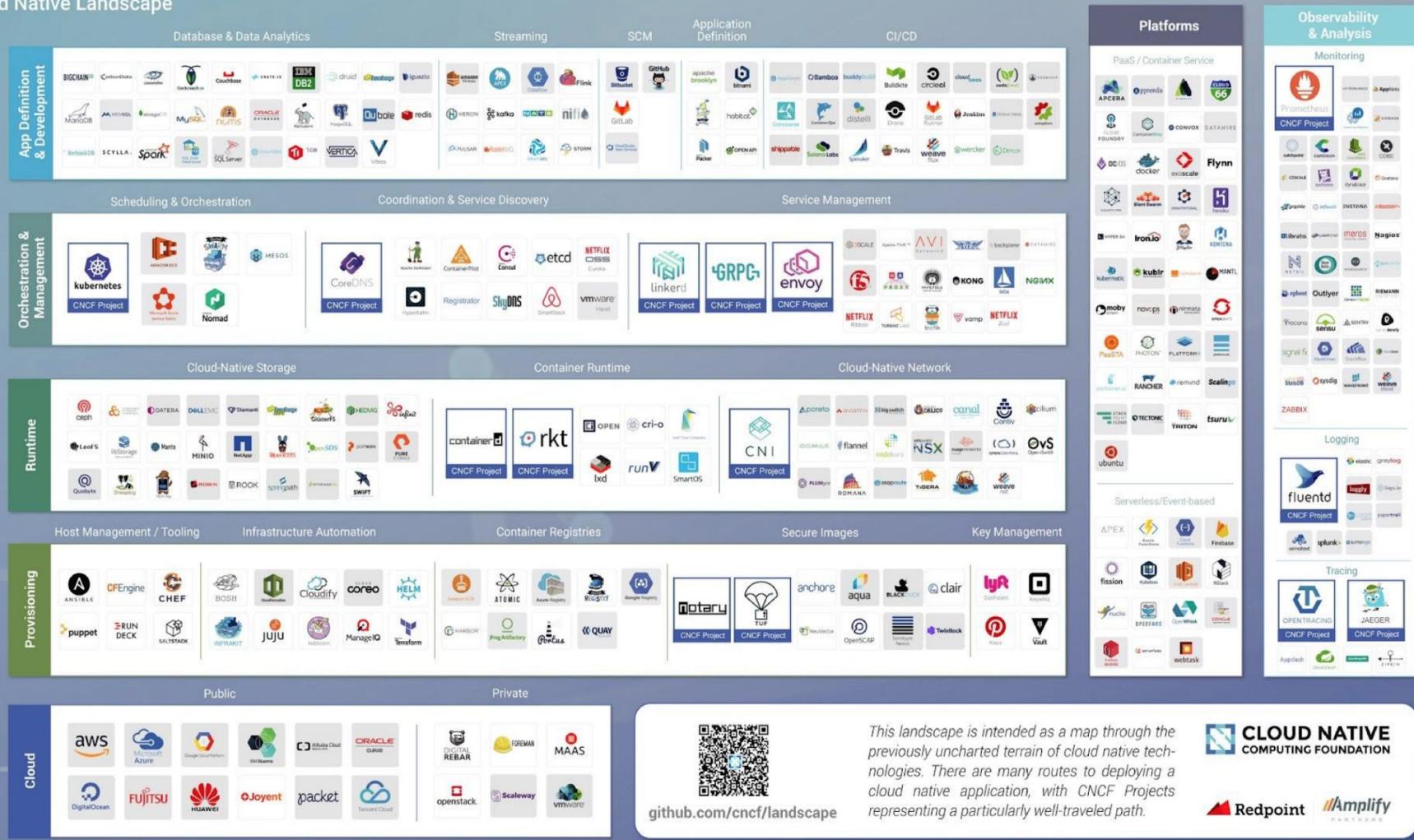
Cloud Native is the set of tools and technologies used to build networked applications on infrastructure outside of the traditional datacenter.

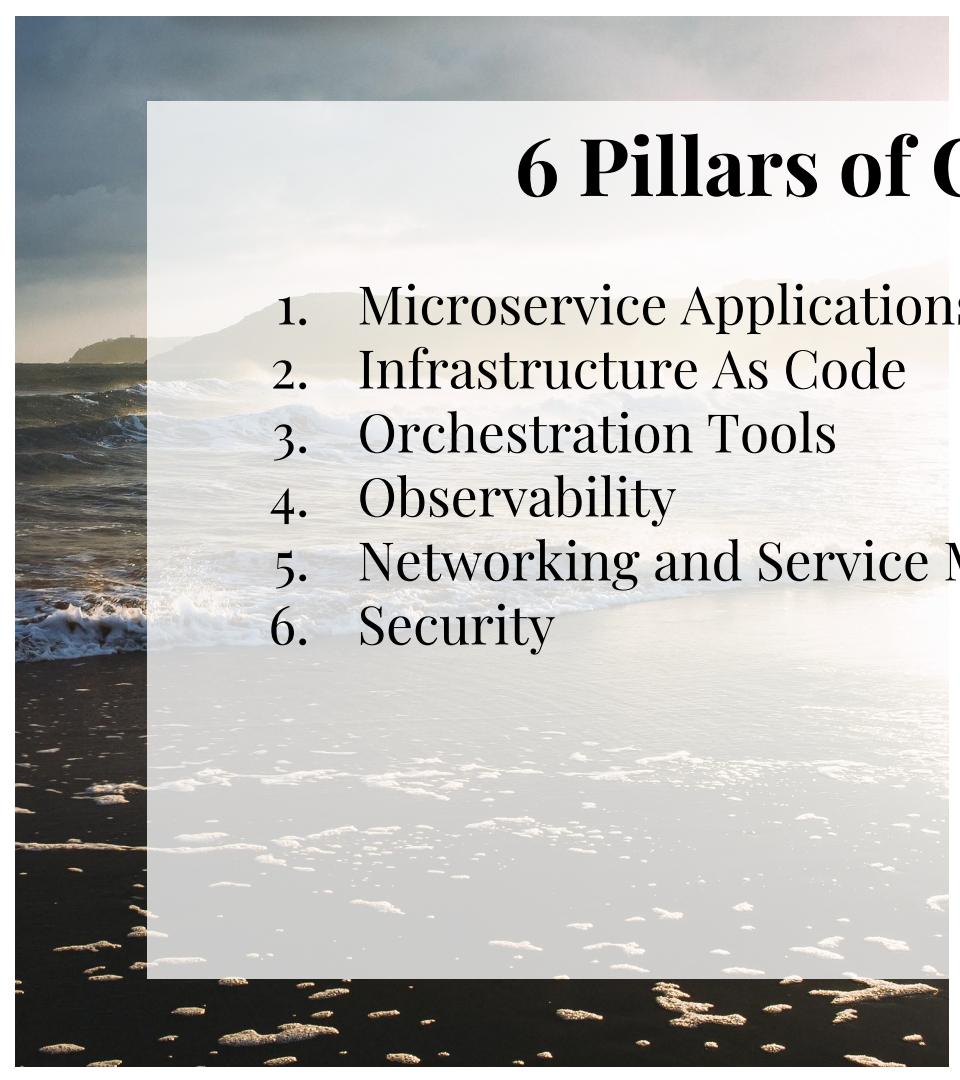
This includes:

- IaaS (AWS, Google Cloud, Azure)
- CNCF projects
- Other leading Cloud projects (Hashicorp, Apache, Monitoring tools, etc)
- Serverless technologies

Going beyond the
CNCF...

Cloud Native Landscape





6 Pillars of Cloud Native

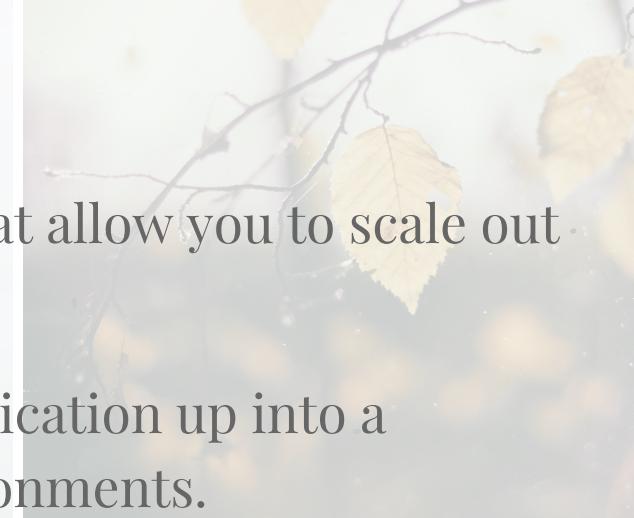
1. Microservice Applications
 2. Infrastructure As Code
 3. Orchestration Tools
 4. Observability
 5. Networking and Service Meshes
 6. Security
- 

Microservices

Breaking up your smaller, decoupled services that allow you to scale out your applications independently and quickly.

The rise of Docker: Allows you to pack your application up into a “binary” that can easily run on most Linux environments.

Allows you to use the right tools and languages for each specific business need.



The ability to define and create cloud infrastructure through a configuration language.

Why?

Consistency
Speed (i.e. modules)
Change Management
Cost Control



Common IaC Tools:

- Terraform
- CloudFormation (AWS)
- Pulumi

Infrastructure As Code

Orchestration Tools

An orchestration tool is some software that manages the resources for your application. It is often made up of a control plane (API), a scheduler, service discovery, and a resource provisioner.

API: Used to communicate what and how many resources to allocate to a specific application.

Scheduler: Talks to all the resource provisioners to determine which nodes have the capacity to run the application.

Service Discovery: A service that keeps track of where that application is running, so that applications can communicate with each other. Usually involves IP and port.

Resource Provisioner: Usually an agent that sits on each node in the cluster and does the job of actually deploying the application.

Popular Orchestrators:

- Kubernetes
- Nomad (Hashicorp)
- Mesos
- ECS
- Autoscaling groups

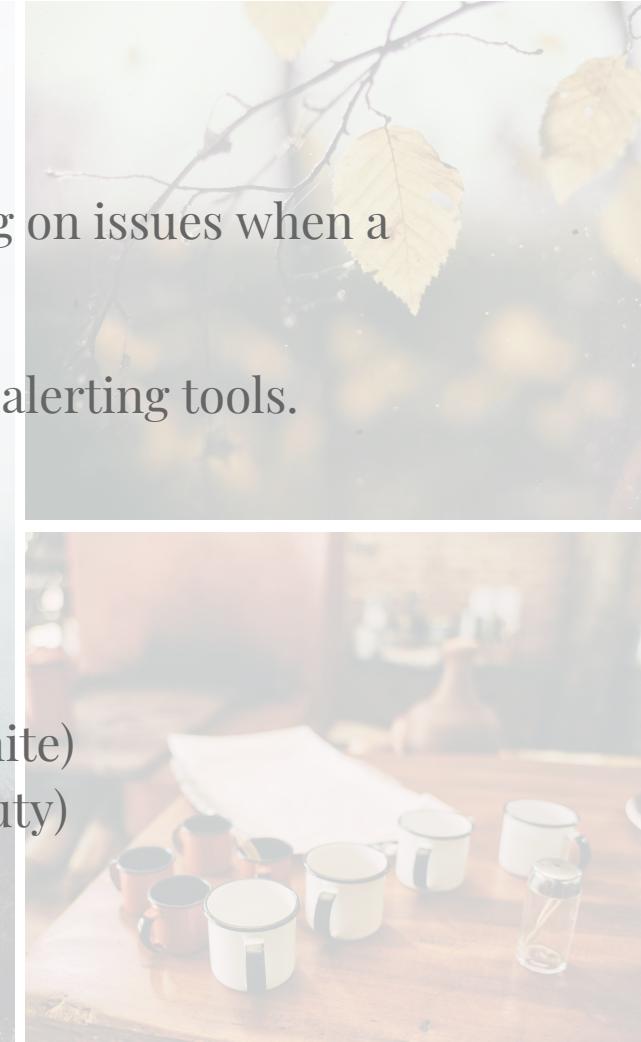
Observability

Observability is the process of monitoring and alerting on issues when a problem arises in your cloud environment.

This often includes logging, time-series, chatops, and alerting tools.

Popular Observability Tools:

- Cloud Providers (CloudWatch, Stackdriver)
- Third Party (Datadog, SumoLogic, New Relic, etc)
- Open Source (Prometheus, Grafana, Influx, Graphite)
- Alerting (Pingdom, OpsGenie, VictorOps, PagerDuty)



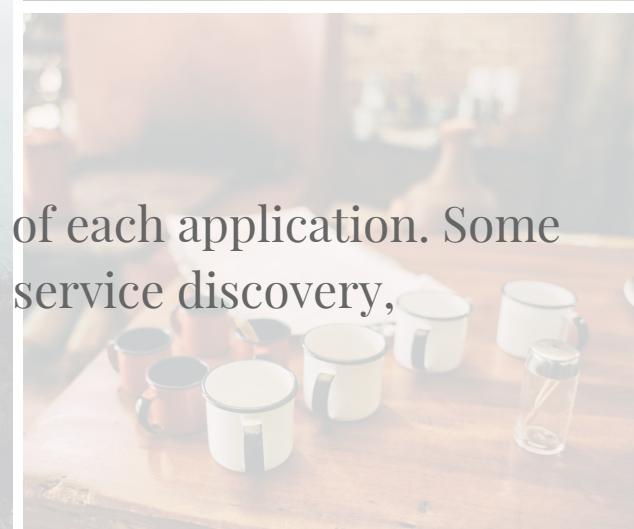
Networking and Service Meshes

Networking is the foundation of cloud native applications. Making sure that your applications can communicate across an ever growing networked environment in a secure way is key to understanding Cloud Native.

Service Meshes are becoming a key part in enabling applications to easily communicate with each other.

So, what is a service mesh?

A **service mesh** (or sidecar) is a proxy that sits in front of each application. Some of its benefits include: ssl termination, authentication, service discovery, monitoring, and distributed tracing.

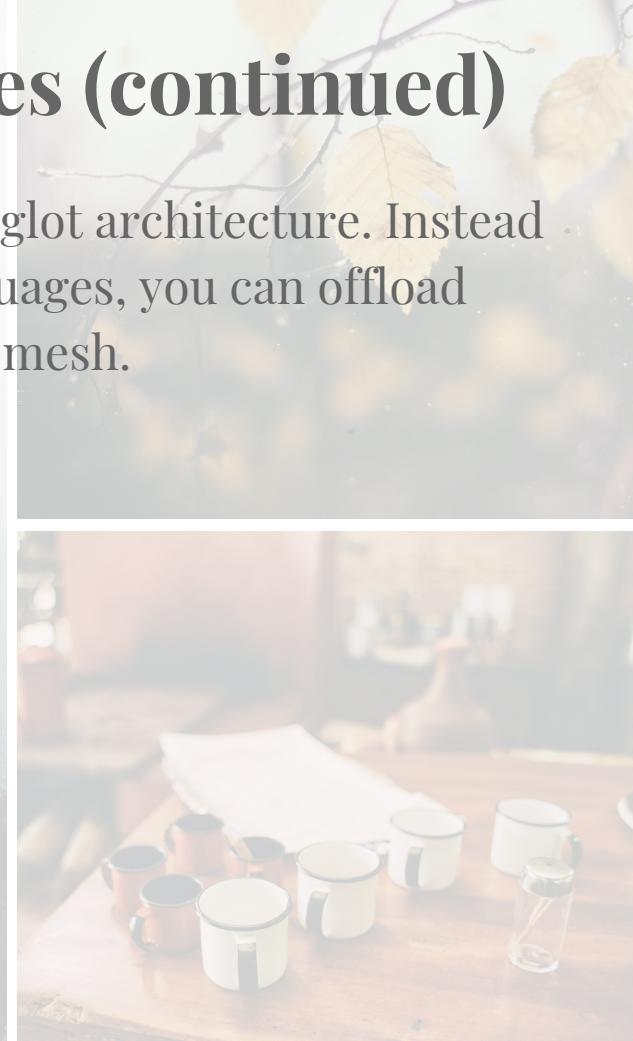


Networking and Service Meshes (continued)

Another huge benefit is it allows for a much more polyglot architecture. Instead of creating and maintaining libraries for multiple languages, you can offload many of these middleware type services to the service mesh.

Popular Service Meshes:

- Envoy (really a proxy used to build the mesh)
 - Istio
 - AWS App Mesh
- Linkerd / Conduit
- Kong (based off Nginx)

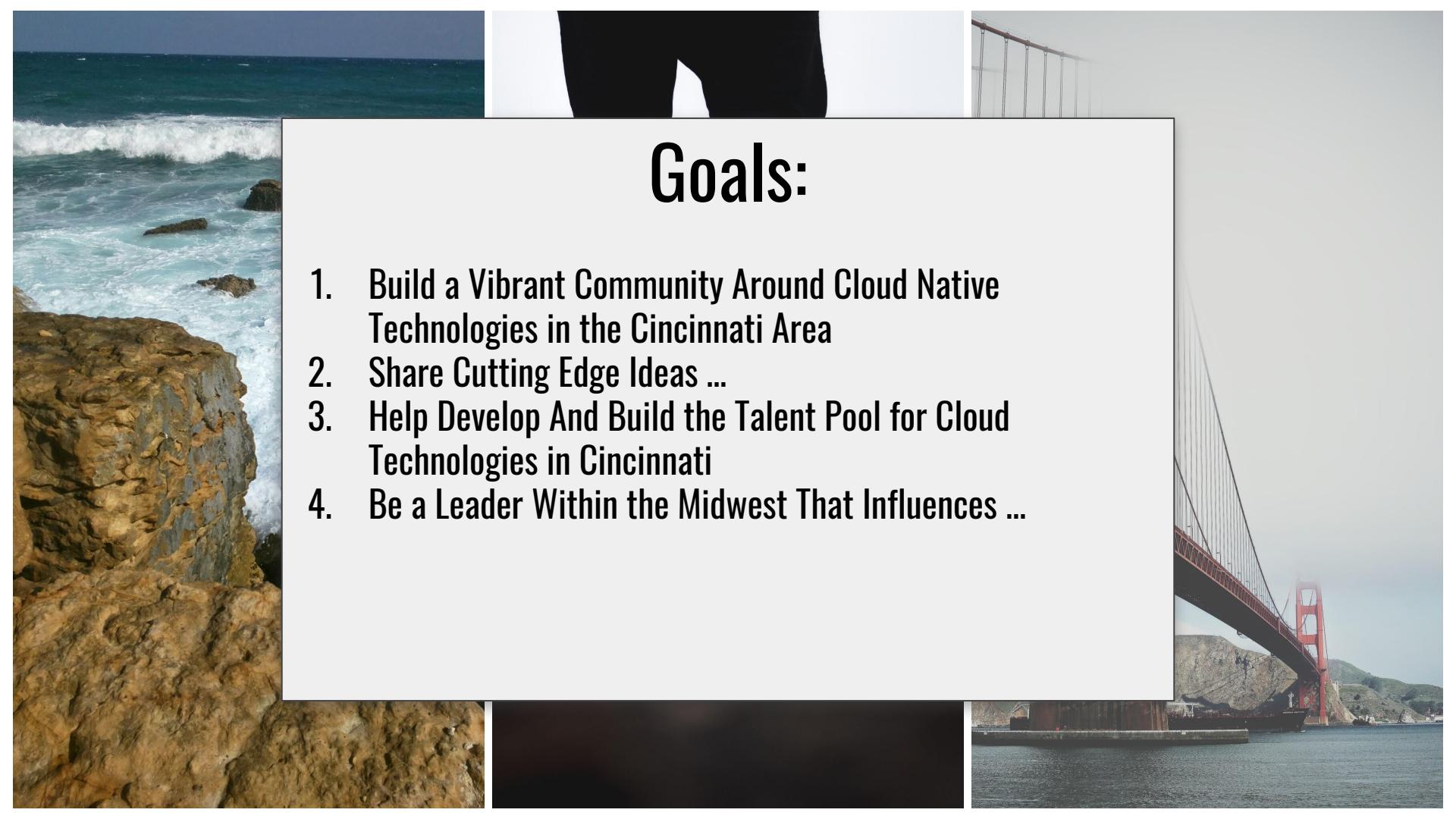


Security

Security is core to the success of building microservices in the cloud. We must be careful to secure our applications at every layer. This is often called **Defence in Depth**.

Key areas that we should be consistently checking:

1. Application (proper auth, SQL injection, SSX, etc)
2. Encrypt All the Things (Zero Trust vs DMZ)
3. Infrastructure (VPC, subnets, ACLs, firewalls)
4. Containers (don't use root user, vulnerability scanning, rbac/policies [kubernetes])
5. Secrets Management (Vault, KMS, etc)



Goals:

1. Build a Vibrant Community Around Cloud Native Technologies in the Cincinnati Area
2. Share Cutting Edge Ideas ...
3. Help Develop And Build the Talent Pool for Cloud Technologies in Cincinnati
4. Be a Leader Within the Midwest That Influences ...

Ideas for Workshops:

- Build an AWS/GCP/Azure VPC from Scratch (NAT, pub/priv subnets, routes, etc)
- Deploy two services behind Envoy and Nginx and compare them
- K8s workshop (using kubectl, svc, deploy, secrets, cm, etc)
- Vault w/ Consul setup

Talks

- Security in the Cloud
- Proxies (Envoy, Kong, etc)
- Compare AWS, GKE networking
- Understanding networking and microservices
- Service Meshes
- Distributed Systems - HA with Consensus Algorithms (Raft, etc)
- Distributed Systems - Gossip Protocols
- Kubernetes
-

Questions?

Formats we will try:

1. Occasionally Speakers (long and lightning)
2. Lots of Workshops and tutorial style, hands-on
3. State of Cloud Native (occasionally) - Panels by Leaders in space, Reviews and News (maybe do Review & News each month - 5ish minutes)
4. Q & A sections
5. Trivia/Fun Nights ???
6. Intense Course Style (but with fun Hands-on activities) - Lab

Workshops:

Short Intro

Theory (example What is CIDR?)

Lab

Answer and Questions

Deploy Firecracker on GCP