

# Cloud Native and Why It Matters

Alen Komljen  
DevOps Engineer at Sematext Inc.

 @alenkomljen

<https://akomljen.com>

**Netflix** pioneered the concept of cloud native - speed and access at scale

<https://www.slideshare.net/AmazonWebServices/dmg206>

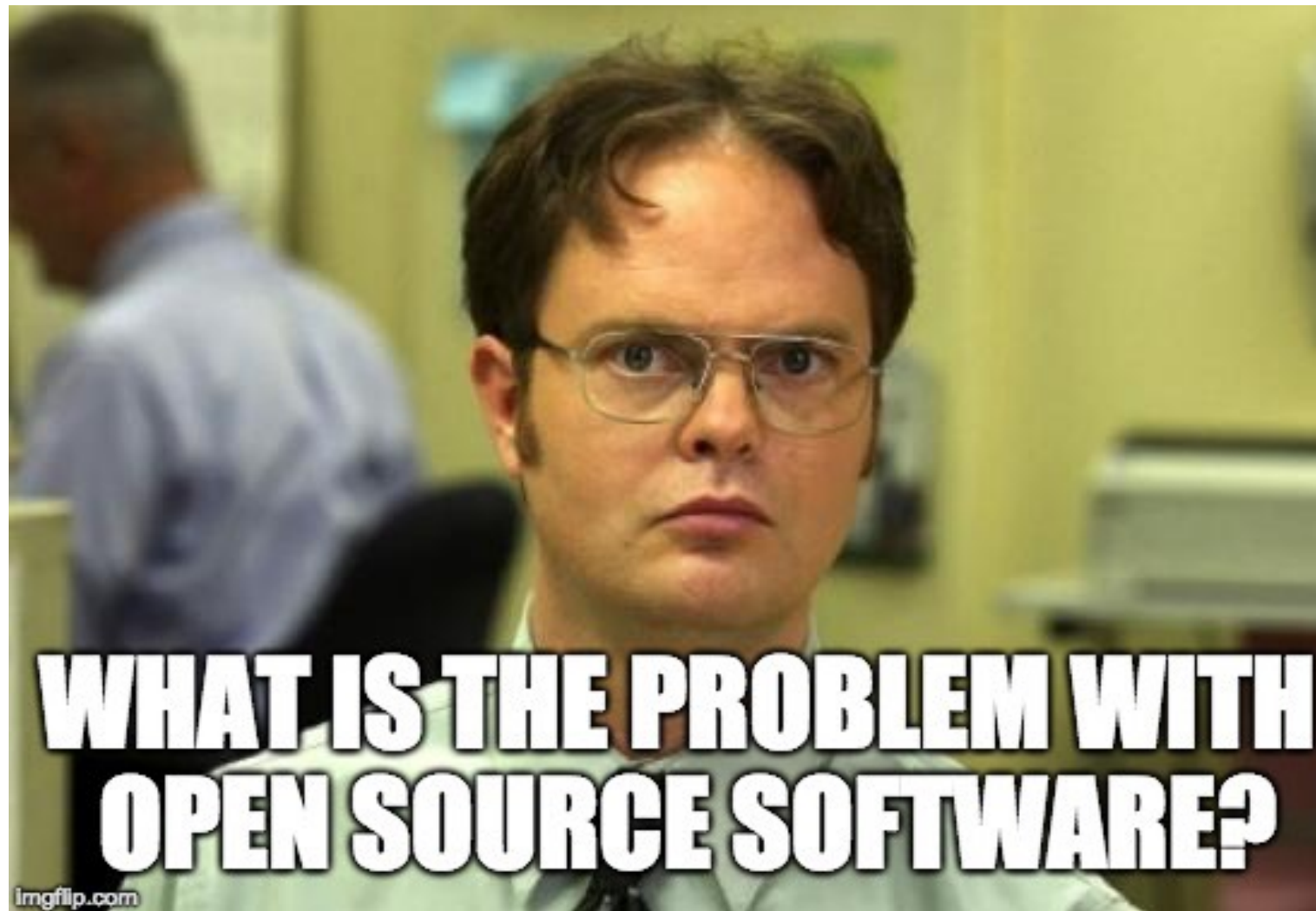


# **CLOUD NATIVE COMPUTING FOUNDATION**

**Enable cloud portability without vendor lock-in!**

<https://www.cncf.io/>

- CNCF is the part of Linux Foundation
- Empowers vendor-neutral and open-source projects



- More a characteristic than a problem
- Who can guarantee that the open source project will be active after 1 year or more?
- CNCF Provides the tools that we can trust

# Cloud Native Landscape

v20180525

See the interactive landscape at [l.cncf.io](http://l.cncf.io)

Greyed logos are not open source

App Definition and Development

Database and Data Warehouse

Streaming

Source Code Management

Application Definition and Image Build

Continuous Integration / Continuous Delivery (CI/CD)

Orchestration & Management

Scheduling & Orchestration

Coordination & Service Discovery

Service Management

Runtime

Cloud-Native Storage

Container Runtime

Cloud-Native Network

Provisioning

Host Management / Tooling

Infrastructure Automation

Container Registries

Secure Images

Key Management

Cloud

Public

Private

Platforms

Monitoring

Logging

Tracing

Serverless

Special

Kubernetes Certified Service Provider

Kubernetes Training Partner



This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

**CLOUD NATIVE Landscape**  
**CLOUD NATIVE COMPUTING FOUNDATION**  
Redpoint Amplify

Special



Serverless Cloud Native Landscape  
v20180525

See the serverless interactive landscape at [s.cncf.io](https://s.cncf.io)

Greyed logos are not open source

Tools

dashbird

Epsagon

event gateway

Gloo

IO|pipe

Iron.io

Node Lambda

OVERCLOCK

python-λ

SIGMA

STACKERY

THUNDER

Security

intrinsic

Protego

PURESEC

snyk

Framework

ΔPEX

.arc Architect

aws Chalice

AWS SAM

Claudia.js

FLOGO™

gun.io

serverless

Shep

SPARTA

Spring Cloud Function

Platform

Hosted

ALGORITHMIA

AWS Lambda

Azure Functions

BINARIS

Clay

cloudboost

CLOUDFLARE

Google Cloud Functions

HUAWEI

HYPER.SH

IBM Cloud

NANO-LAMBDA

PubNub FUNCTIONS

spotinst

stdlib

syncano

twilio

WEBLAB

ZEIT

Installable

APACHE OpenWhisk™

AppScale

DISPATCH

fission

fn

GALACTIC FOG

Kubeless

LunchBadger

nuclio

OPENFAAS

OpenLambda

riff

Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment.

# Traditional vs Cloud Native applications

Cloud native is the new way of thinking about applications:

- Different in architecture
- Written in the stateless manner (shared state when needed)
- Using load balancers everywhere
- Nodes are disposable



# How to get there?

- Use common ~~standards~~ patterns
- Monitor everything - metrics, logs (Prometheus, ELK, etc.)
- Use modern practices, micro-services (if applicable)
- Use containers
- Automate processes
- Use pipelines

## CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape [Landscape](https://landscape.cncf.io) has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

### HELP ALONG THE WAY

#### A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer [cncf.io/training](https://cncf.io/training)

#### B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider [cncf.io/kcsp](https://cncf.io/kcsp)

#### C. Join CNCF's End User Community

For companies that don't offer cloud native services externally [cncf.io/enduser](https://cncf.io/enduser)

### WHAT IS CLOUD NATIVE?

Cloud-native technologies, such as containers and microservices, empower organizations to develop and deploy scalable, agile applications and services in dynamic, distributed environments. By taking into account these characteristics, such systems are designed to be resilient, elastic, and loosely coupled, via manageable abstractions and declarative APIs, thereby enabling effective, reliable automation. This allows engineers to observe the applications and to safely make impactful changes, and results in processes and workflows that fully take advantage of these environments and minimize toil.

The Cloud Native Computing Foundation seeks to drive adoption of these techniques by fostering an ecosystem of open-source, vendor-neutral projects that align with these objectives, and which are portable to public, private, and hybrid clouds. We democratize the state-of-the-art patterns and practices to ensure innovations remain open and accessible for everyone.

[l.cncf.io](https://l.cncf.io)

v20180604



### 1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

### 3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: [cncf.io/ck](https://cncf.io/ck)
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



### 5. SERVICE MESH AND DISCOVERY

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



### 7. DISTRIBUTED DATABASE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding.



### 9. CONTAINER RUNTIME

You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.

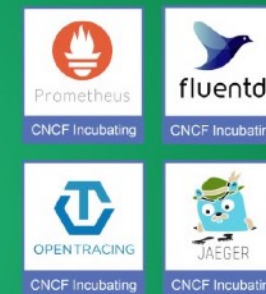


### 2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing

### 4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



### 6. NETWORKING

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.



### 8. MESSAGING

When you need higher performance than JSON-RPC, consider using gRPC. NATS is publish/subscribe message-oriented middleware.



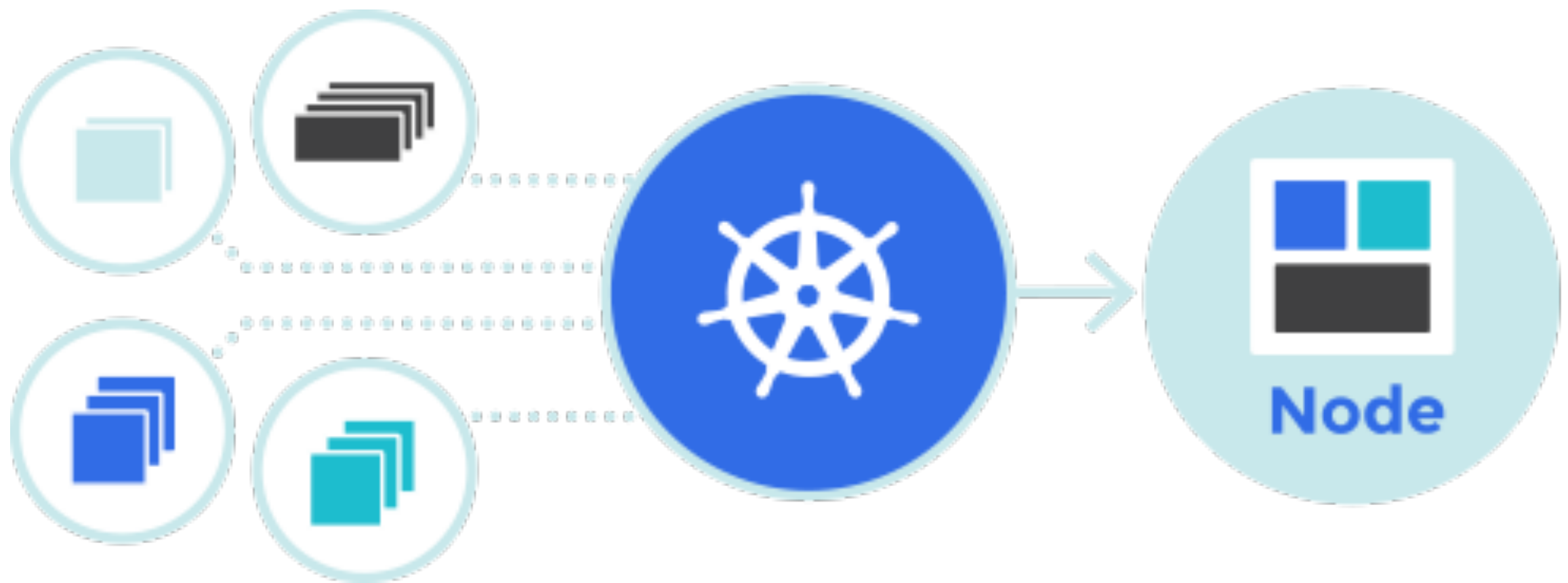
### 10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



**We shouldn't be proud  
of long-running services**

# kubernetes.io



- People are asking the wrong questions when considering Kubernetes for their workloads
- How to embrace Kubernetes successfully?

**The cloud native focus is on  
applications!**

# Thanks!

 @alenkiljen

<https://akomljen.com>