9th October 2018

Jonathan Relf

DEVEL-OPS

BRINGING DEVELOPMENT PRACTICES TO OPERATIONS TASKS

DEVEL-OPS

THIS TALK IS NOT:

VS

Devs

Ops

THIS TALK IS MORE:


老北京布鞋
老北京布鞋
古文化

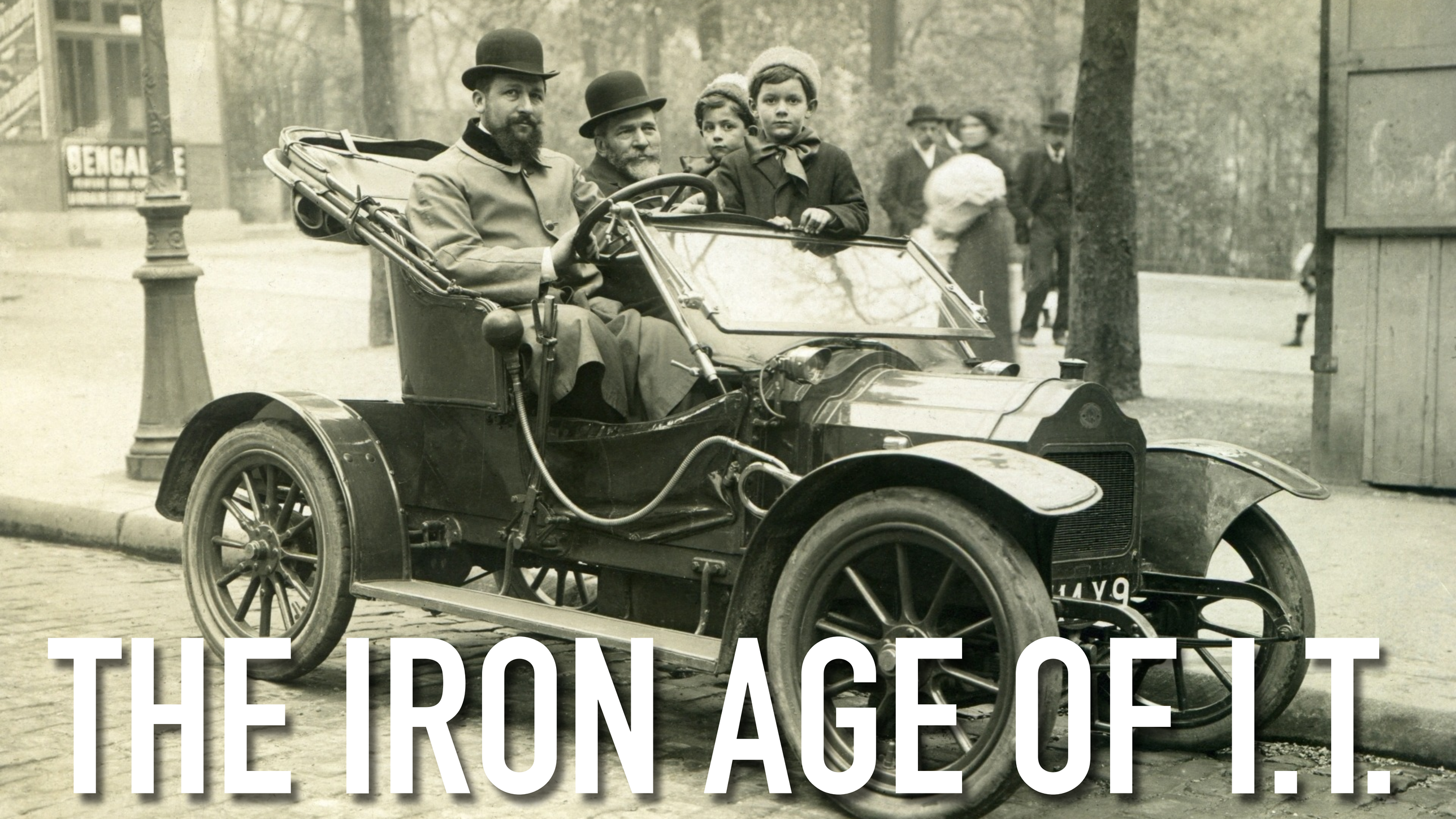
"THE DEFINITION OF INSANITY
IS REPEATING THE SAME MISTAKES
OVER AND OVER AGAIN
AND EXPECTING DIFFERENT RESULTS"

Unknown

"THE DEFINITION OF INSANITY IN I.T. OPERATIONS
IS MANUALLY REPEATING A TASK
OVER AND OVER AGAIN
AND EXPECTING THE SAME RESULTS"

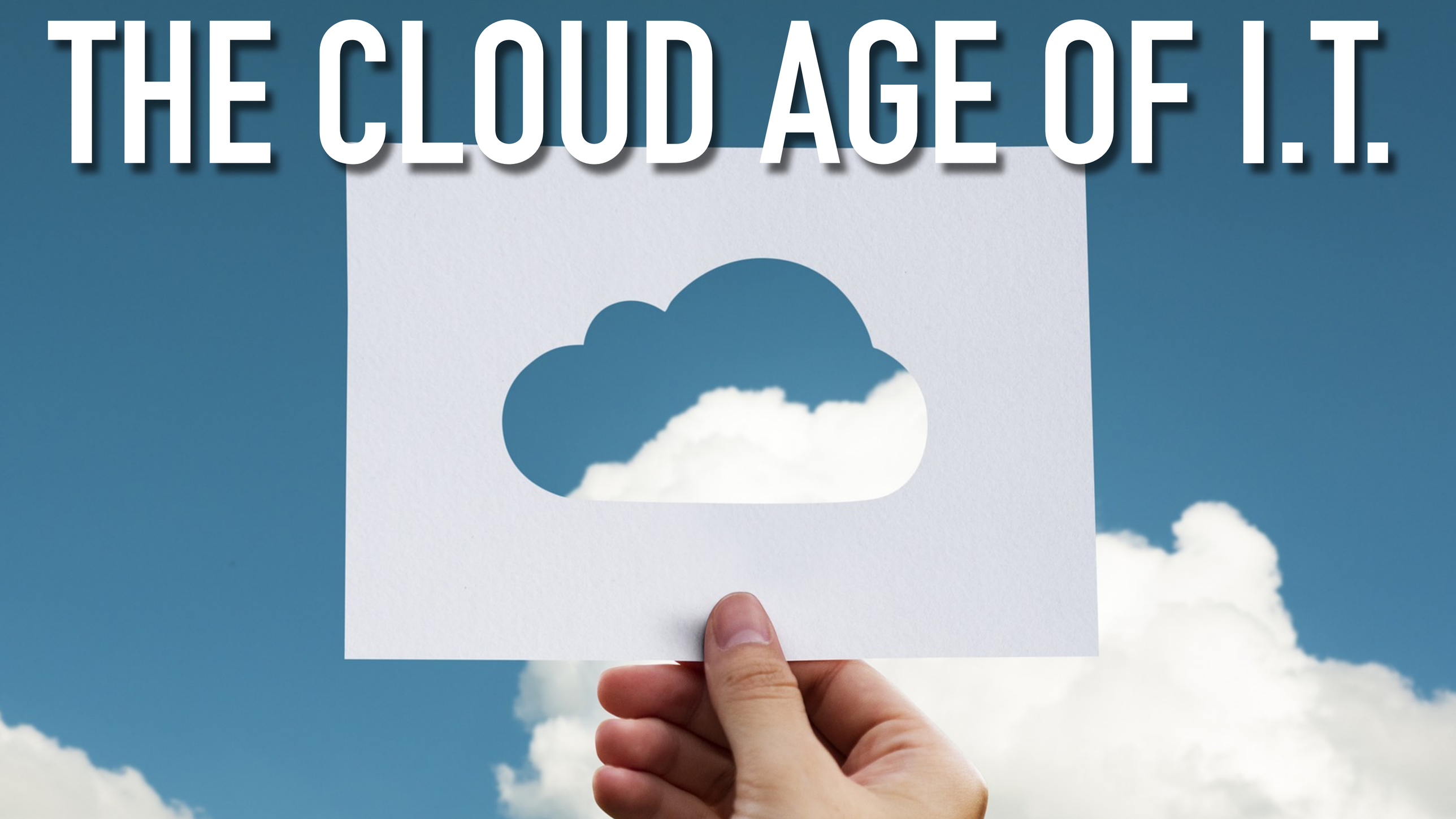
Jonathan Relf


uitgang



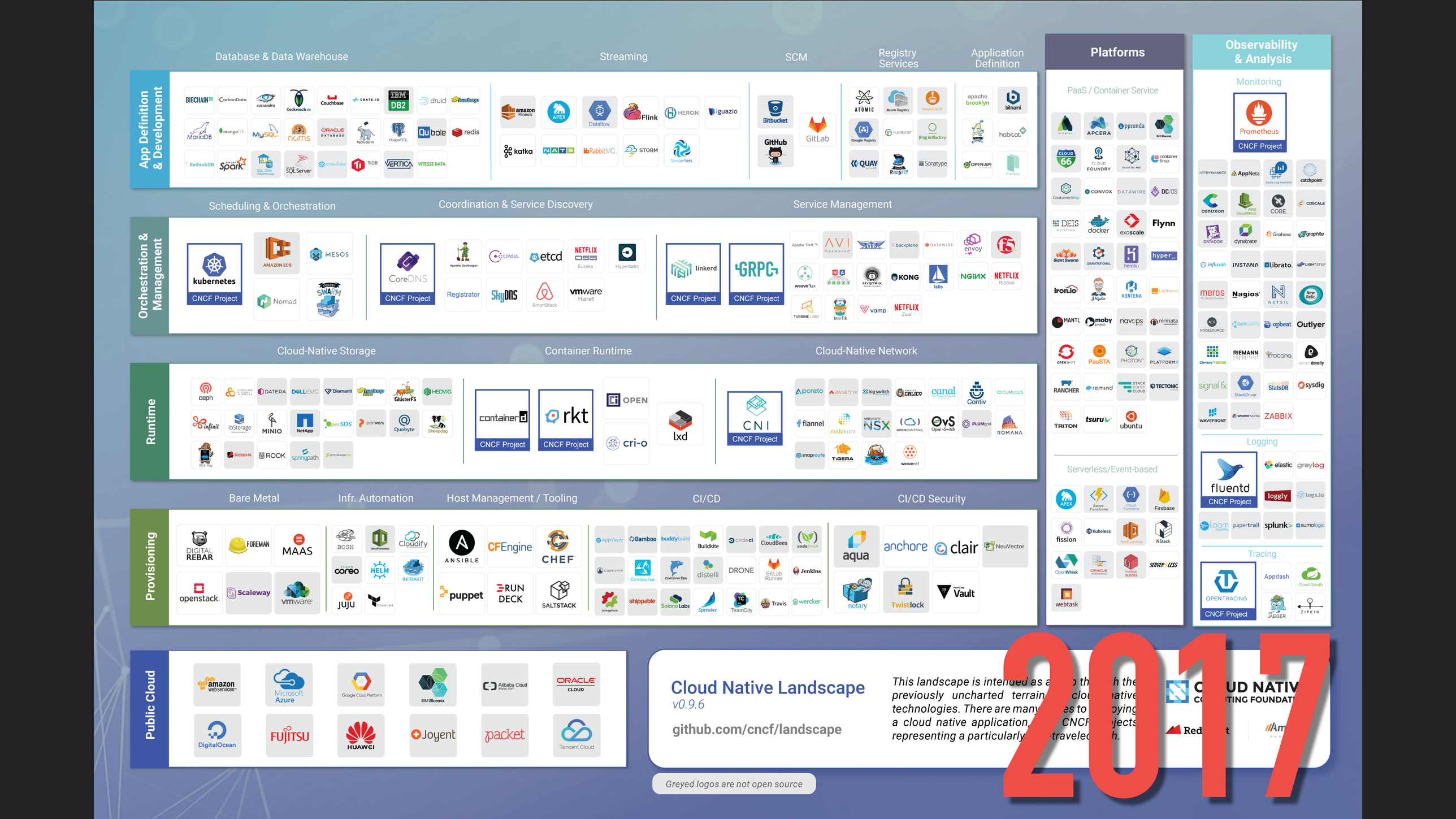
THE IRON AGE OF I.T.

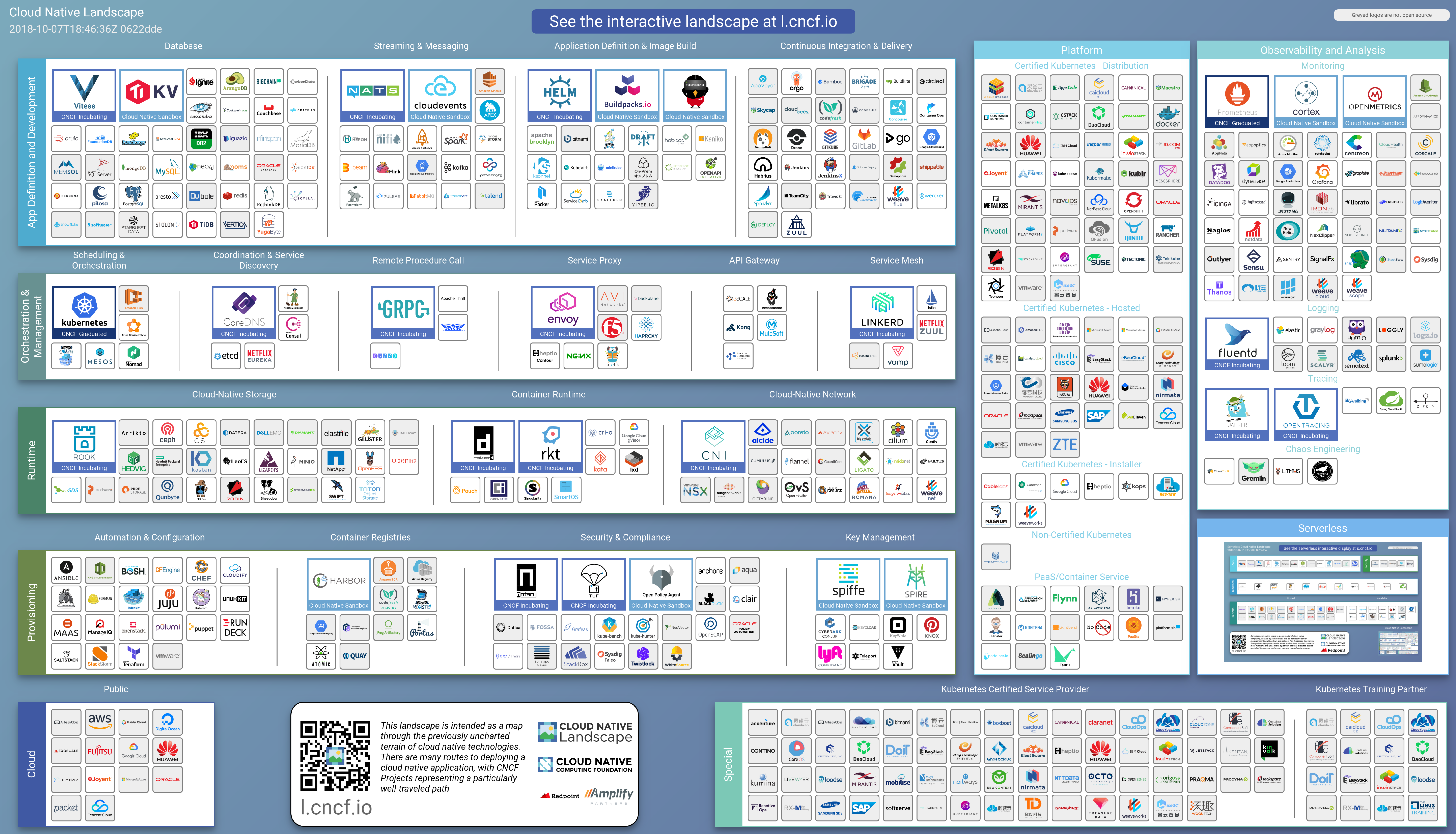
THE CLOUD AGE OF I.T.



"UNRELIABLE SOFTWARE
DEPENDING ON RELIABLE HARDWARE TO
RELIABLE SOFTWARE
RUNNING ON UNRELIABLE HARDWARE"

Infrastructure As Code


App Definition & Development
Database & Data Warehouse
Streaming
SCM
Registry Services
Application Definition
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Service Management
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
Provisioning
Bare Metal
Infr. Automation
Host Management / Tooling
CI/CD
CI/CD Security
Public Cloud
Platforms
PaaS / Container Service
Serverless/Event-based
Observability & Analysis
Monitoring
Logging
Tracing
Cloud Native Landscape
v0.9.6
github.com/cncf/landscape
Greyed logos are not open source
2017


Cloud Native Landscape
2018-10-07T18:46:36Z 0622dde
See the interactive landscape at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
Provisioning
Automation & Configuration
Container Registries
Security & Compliance
Key Management
Cloud
Public
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
Non-Certified Kubernetes
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
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
l.cncf.io
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify PARTNERS

SALTSTACK StackStorm Terraform ATOMIC

Public

Cloud

Alibaba Cloud | aws | Baidu Cloud | DigitalOcean
EXOSCALE | FUJITSU | Google Cloud | HUAWEI
IBM Cloud | Joyent | Microsoft Azure | ORACLE
packet | Tencent Cloud

l.cncf.io

This
thro
terra
The
clou
Proj
well

& Image Build

Continuous Integration & Delivery

s.io

andbox

Cloud Native Sandbox

RAFT

habitat by CHEF

Kaniko

n-Prem
ンプレム

OPEN SERVICE BROKER API

OPENAPI INITIATIVE

PEE.IO

AppVeyor

argo

Bamboo

BRIGADE

Buildkite

circleci

Skycap

cloudbees

codefresh

CODESHIP

Concourse

ContainerOps

DeployHub

Drone

GITKUBE

GitLab

go

Google Cloud Build

Habitus

Jenkins

JenkinsX

Octopus Deploy

Semaphore

shippable

Spinnaker

TeamCity

Travis CI

HYSCALE wavemaker

weave flux

wercker

DEPLOY

ZUUL

Certifie

AGILESTACKS

灵雀云 alauda.cn

CLOUDFOUNDRY CONTAINER RUNTIME

containership

Giant Swarm

HUAWEI

Joyent

KONTENA PHAROS

METALK8S

MIRANTIS

Pivotal

PLATFORM9

STACKPOINT

API Gateway

Service Mesh

Platform

d Kubernetes - Distribution

AppsCode
caicloud 才云
CANONICAL
Maestro
CSTACK 酷栈科技
DaoCloud
DIAMANTI
docker
IBM Cloud
inspur 浪潮云
inwinSTACK
JD.COM TIG
kube-spawn
Kubermatic
kublr
MESOSPHERE
navops by Univa
NetEase Cloud
OPENSHIFT
ORACLE
portworx
QFusion
QINIU
RANCHER
SUPERGIANT
SUSE
TECTONIC
Telekube MADE BY GRAVITATIONAL

Observability and Analysis

Monitoring

Prometheus
CNCF Graduated

cortex
Cloud Native Sandbox

OPENMETRICS
Cloud Native Sandbox

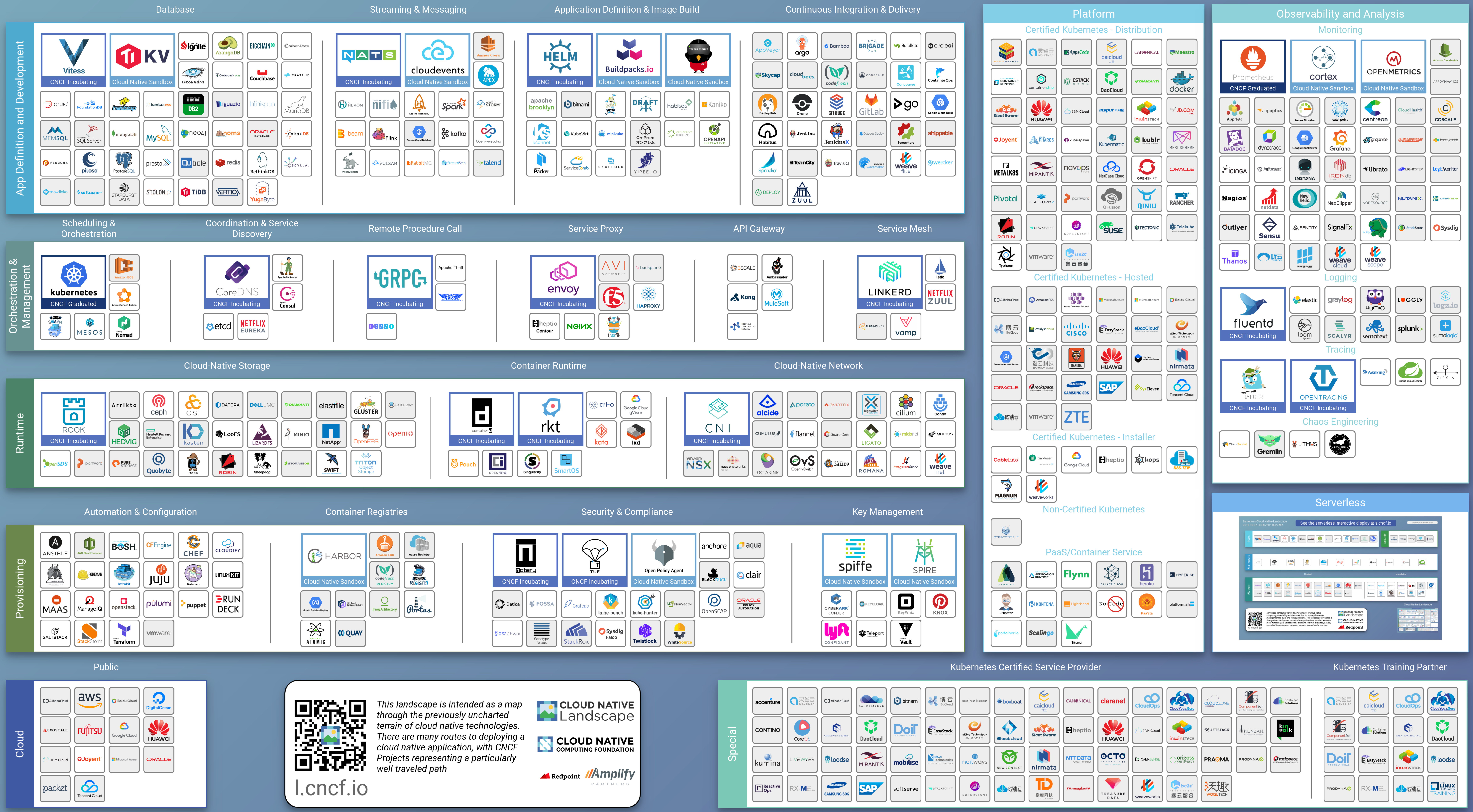
Amazon Cloudwatch
APPDYNAMICS
AppNeta
appoptics
Azure Monitor
catchpoint
centreon
CloudHealth TECHNOLOGIES
COSCALE
DATADOG
dynatrace
Google Stackdriver
Grafana
graphite
Honeybadger
honeycomb
ICINGA
influxdata
INSTANA
IRONdb POWERED BY CIRCONUS
librato
LIGHTSTEP
LogicMonitor
Nagios
netdata
New Relic.
NexClipper
NODESOURCE
NUTANIX
OPENTSDB
Outlyer
SENTRY
SignalFx
snap
StackState
Sysdig

Cloud Native Landscape

2018-10-07T18:46:36Z 0622dde

See the interactive landscape at l.cncf.io

Greyed logos are not open source



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

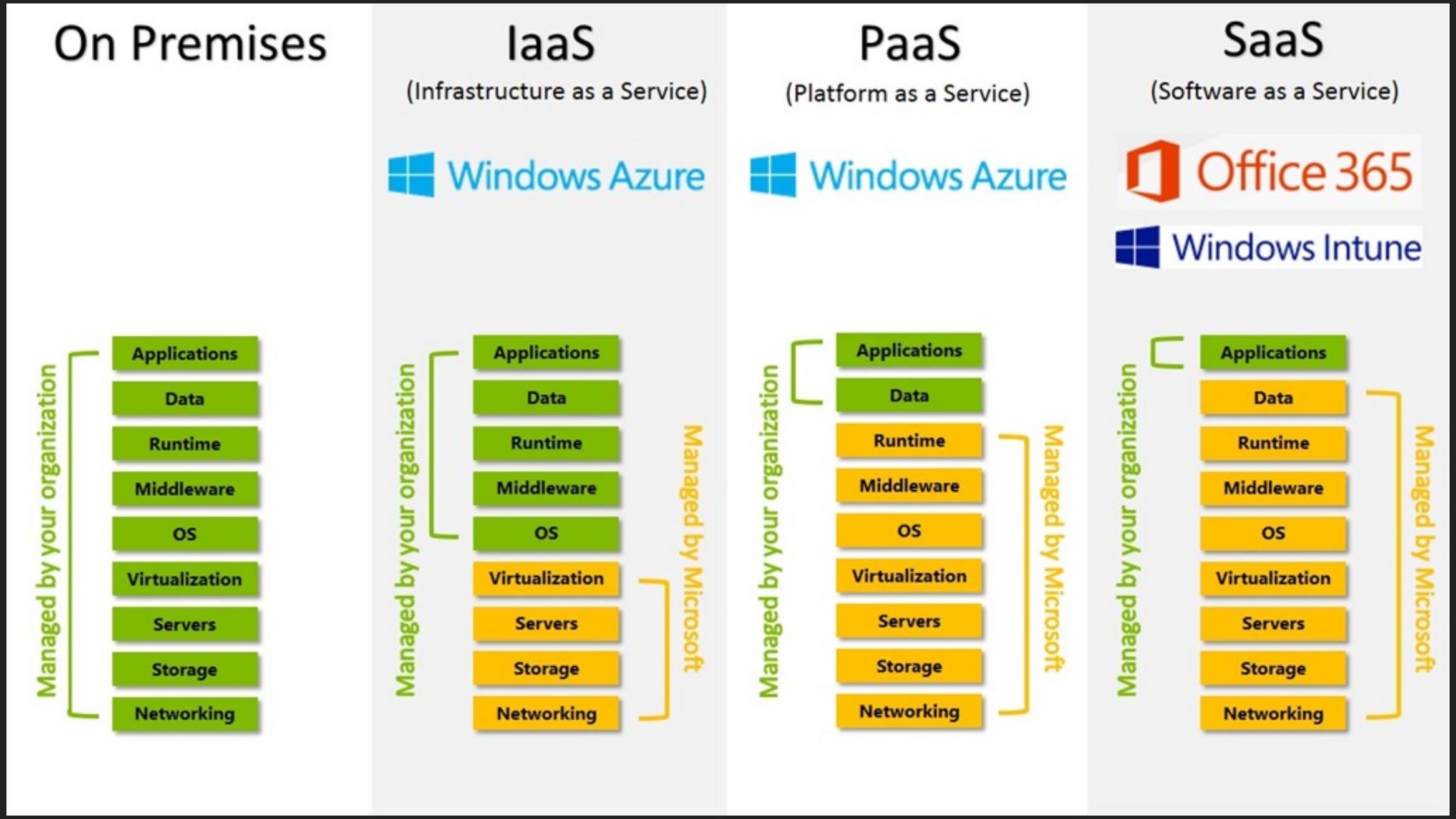
CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint

Amplify PARTNERS

l.cncf.io



https://blogs.technet.microsoft.com/xdot509/2013/07/24/getting-started-with-windows-azure-part-2-what-are-cloud-services/

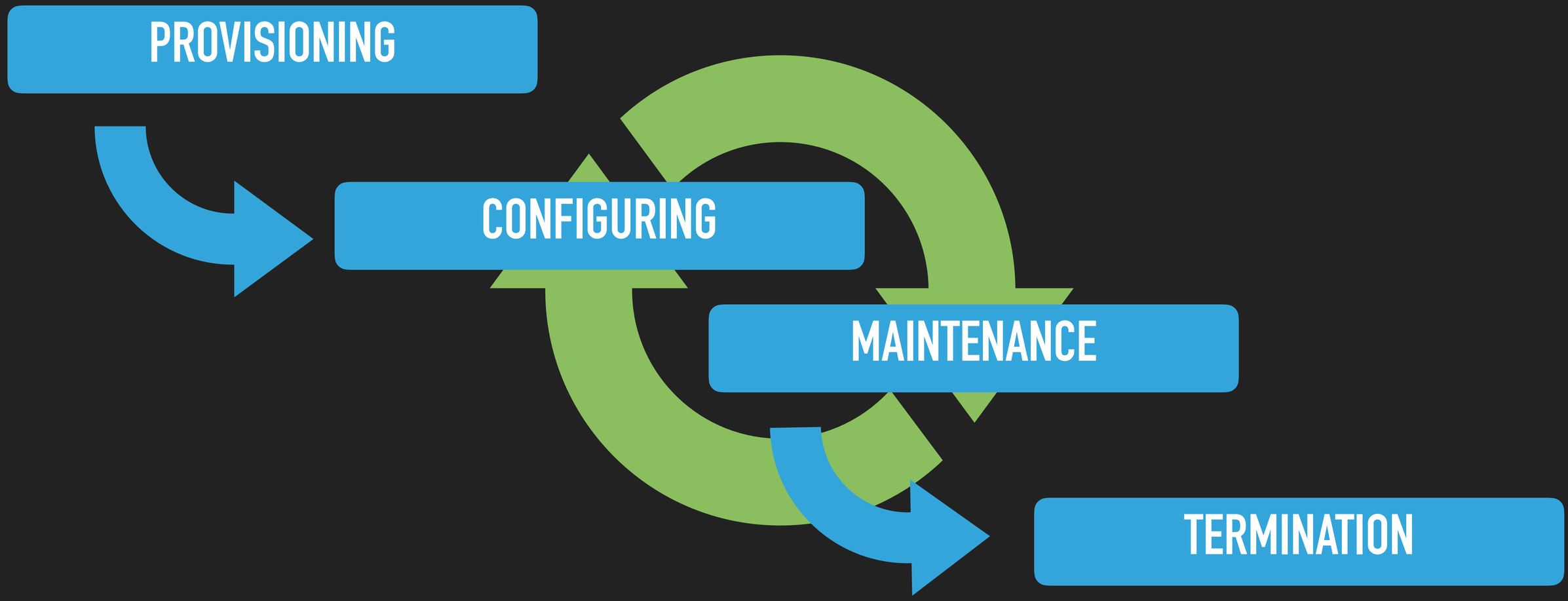
WHY BOTHER DEFINING YOUR OWN INFRASTRUCTURE IN CODE?

- May have regulatory or data sovereignty limitations
- Unsure of what you are getting using 3rd party solutions
- Allowing you to ‘harden‘ application hosting
- Reducing the risk of ‘vendor lock‘

MAINTAINING SERVERS OVER TIME

OWN SOFTWARE	CONFIGURATION	UPDATES	
3RD PARTY SOFTWARE	CONFIGURATION	UPDATES	
OPERATING SYSTEM	FEATURES	CONFIGURATION	UPDATES
HYPERVISOR	CONFIGURATION	UPDATES	
PHYSICAL SERVER	CONFIGURATION	UPDATES	

THE FOUR STAGES OF INFRASTRUCTURE



PROVISIONING

- Since virtualisation, one of the easiest of the phases
- Traditionally based off ISO images
- Out of date, unpatched images

CONFIGURATION

- Can be where most configuration drift is added if not done through automation
- It may involve ‘tinkering’ to get a server working
- Can end up with ‘Snowflake servers’
- Configuration Management software like Puppet & Chef

MAINTENANCE

- Updates or upgrades of software components
- From security patches to in-place upgrades of O.S.
- Ordering of patches may affect outcome
- Scripting can help ensure consistency

TERMINATION

- Fear of shutting off servers
- Treating servers like “pets, not cattle“
- Anti-pattern: Celebrating up-time
- Plan for the fact an instance could disappear

GOALS OF INFRASTRUCTURE AS CODE – MORRIS

- IT infrastructure supports & enables change
- Changes to the system are routine, without drama or stress
- IT staff spend their time on valuable things.. not repetitive tasks
- Users are able to define, provision, and manage the resources they need, without needing IT staff to do it for them

DEFINITION TOOLS

- Good Infrastructure As Code tools
  - have scriptable interfaces
  - can be run unattended
  - can be tailored through config
  - allow tasks to be defined in code
  - the definition files become ‘living documentation’

DEVELOPMENT PRACTICES

- Version Control
- Continuous Integration
- Build Pipelines
- Automated Testing
- Continuous Delivery
- Code Analysis

VERSION CONTROL

- Natural part of development workflow
  - branching, rollbacks, ownership
- Single point of truth
- Living documentation

CONTINUOUS INTEGRATION

- Early feedback about potentially breaking changes
- Changes tested in isolation from production
- Can apply to infrastructure, server, and configuration changes
  - *"Does this produce the instance I expect?"*
  - *"Does this instance have all the features I expect?"*
  - *"Is this instance configured for its role correctly?"*

BUILD PIPELINES

- Avoid 'automation fear'
- Build regularly as well as on changes
- Pipelines maintaining templates ensures up-to-date images
- Reduces manual knowledge fading
- Services used to build can be provisioned temporarily
- Packer supports building machine images

AUTOMATED TESTING

- One of the best practices to borrow from Development
- Not relying on ‘Green builds‘
- Frameworks like ServerSpec (http://serverspec.org)


```
describe service('apache2'), :if => os[:family] == 'ubuntu' do
  it { should be_enabled }
  it { should be_running }
end

describe service('org.apache.httpd'), :if => os[:family] == 'darwin' do
  it { should be_enabled }
  it { should be_running }
end

describe port(80) do
  it { should be_listening }
end
```

CONTINUOUS DELIVERY

- Not ‘Continuous Deployment’
- Being able to update Test / Lab environments regularly
- Risk increases with ‘Time Since Last Success’

CODE ANALYSIS

- Emerging area drawing from Dev practices
- Config management declarative languages have ‘lint’ tools

FC045: Metadata does not contain cookbook name

`correctness` `metadata` `chef12`

This warning is shown when your cookbook does not define a name within the cookbook `meta`
breakage if the name of the containing directory changes. Additionally, Chef 12 requires the nam

Metadata without the name attribute

This example matches the FC045 because it lacks the name property

```
# Don't do this
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'All Rights Reserved'
description 'Installs/Configures test'
long_description 'Installs/Configures test'
version '0.1.0'
```

"A NETFLIX TEAM KNEW THAT A PERCENTAGE OF AWS INSTANCES, WHEN PROVISIONED, WILL PERFORM MUCH WORSE THAN THE AVERAGE INSTANCE SO THEY WROTE THEIR PROVISIONING SCRIPTS TO IMMEDIATELY TEST THE PERFORMANCE OF EACH NEW INSTANCE. IF IT DOESN'T MEET THEIR STANDARDS, THE SCRIPT DESTROYS THE INSTANCE AND TRIES AGAIN WITH A NEW INSTANCE."

Kief Morris

WAYS TO START INTRODUCING THIS

- Start small
- Script everything
- Automate the process
- Run it regularly
- Test the changes in a safe environment
- Monitor all the things


แบบหัดอ่าน A B C สำหรับเด็ก


QUESTIONS?

ABOUT ME

- Jonathan Relf
- Solutions Architect @ Commify
- https://about.me/jbjon

@jbjon