



KubeCon



CloudNativeCon

Europe 2021

Virtual



Forward Together »



KubeCon



CloudNativeCon

Europe 2021

First Principles of Cloud Native Technology

Ronald Petty (RX-M)

Virtual



whoami

Ronald Petty

Principal Consultant at RX-M (rx-m.com)

Speaker, Instructor, Consultant, Advisor, Coder

Exam Developer: CKAD, CKA

Holds: CKAD, CKA, CKS

Prior experience includes management (CTO) and engineering (coding!) in technology, energy and finance spaces.



Virtual

Problem Statement



Virtual

Congratulations!

You have inherited a system that you did not create.

... think think think ...

Now what? Before first principles, a reasonable next step is to set some goals.

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1038-aws x86_64)
```

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

```
System information as of Sun Apr 4 16:45:52 UTC 2021
```

```
System load: 0.0 Processes: 107
Usage of /: 11.6% of 29.02GB Users logged in: 0
Memory usage: 5% IP address for eth0: 172.31.21.211
Swap usage: 0%
```

- * Introducing self-healing high availability clusters in MicroK8s.
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

```
https://microk8s.io/high-availability
```

- * Canonical Livepatch is available for installation.
 - Reduce system reboots and improve kernel security. Activate at:
<https://ubuntu.com/livepatch>

```
11 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable
```

```
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings
```

```
*** System restart required ***
Last login: Sun Apr 4 16:44:53 2021 from 73.241.72.234
~$ echo Now What?
Now What?
~$
```

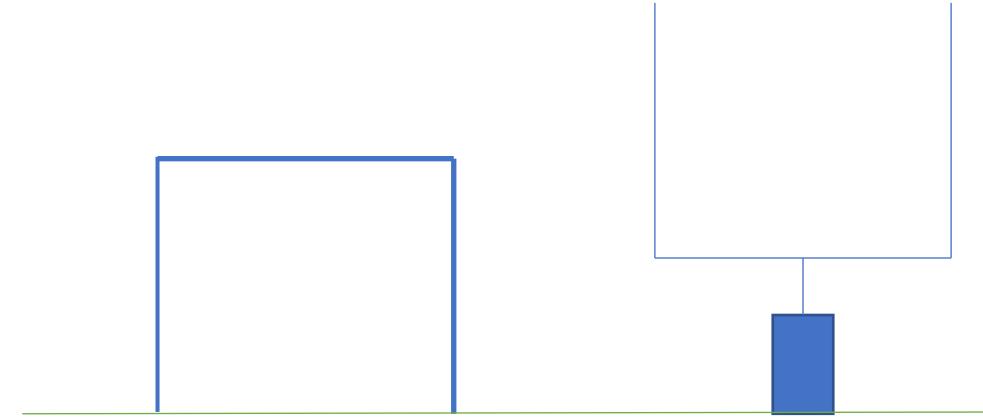
Example Goals

Per Google “goal”:

- The object of a persons ambition or effort; an aim or desired result

Some goals we may set for ourselves:

- Fix bug(s)
- New features
- Maintain
- Improve
- Understand (more on this)
- ... leave the world a better place ... (hope this happens)



What are “First Principles”



Virtual

Related quotes:

3rd millennium of the common era:

“I think it’s important to reason from first principles rather than by analogy”, Musk

2nd millennium of the common era:

“je pense, donc je suis”, Descartes
“I think, therefore I am”

1st millennium before the common era:

“νόησις νοήσεως, nóesis noéseos”, Plato
“knowledge of knowledge”

In general, reassemble from the ground up, question assumptions.

A Few Terms

First we try to recognize what a system is.

Per Google “system”:

- A **set of things working together** as parts of a mechanism or an interconnecting network
- The **prevailing political or social order**, especially when regarded as oppressive and intransigent
- ...

Next, what is Cloud Native Development?

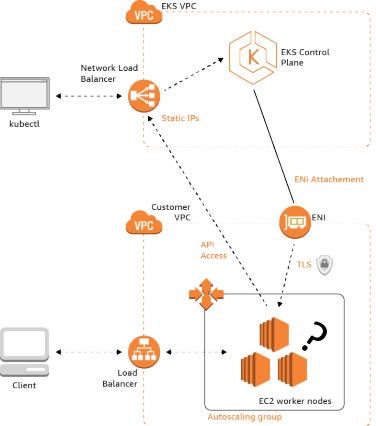
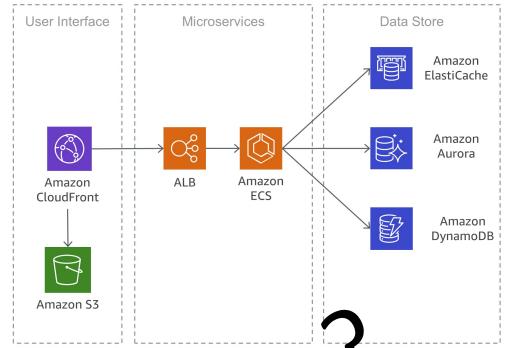
Per Deloitte:

“... leveraging cloud native services drives innovation and increases speed to market. ...”

Per RedHat:

“... More specifically, it’s a way to build and run responsive, scalable, and fault-tolerant apps anywhere – be it public, private, or hybrid cloud. ...”

Enter Reality Check



User reports indicate no current problems at [Roblox](#). Published: 04/02/2021 5:41 p.m. By: [downdetector.com](#)
User reports indicate Roblox is having problems since 5:41 PM EDT. All comments.

A screenshot of the GitHub Issues page for the Kubernetes repository. The search bar shows the query "is:issue is:open sort:comments-desc". The results list several open issues, including one about Facilitate ConfigMap rollouts / management and another about Volumes are created in container with root ownership and strict permissions.

User reports indicate no current problems at [Roblox](#). Published: 04/02/2021 5:41 p.m. By: [downdetector.com](#)
User reports indicate Roblox is having problems since 5:41 PM EDT. All comments.

Elastic Compute Cloud

US East (N. Virginia)

Amazon Elastic Compute Cloud running Linux/UNIX

\$0.00 per Linux t2.micro instance-hour (or partial hour) under monthly free tier
\$0.0058 per On Demand Linux t2.nano Instance Hour
t2.nano Linux instance usage covered by Compute Savings Plans
\$0.0116 per On Demand Linux t2.micro Instance Hour
t2.micro Linux instance usage covered by Compute Savings Plans
\$0.096 per On Demand Linux m5.large Instance Hour
m5.large Linux instance usage covered by Compute Savings Plans
\$0.133 per On Demand Linux r4.large Instance Hour
r4.large Linux instance usage covered by Compute Savings Plans
\$0.20 per On Demand Linux m4.xlarge Instance Hour
m4.xlarge Linux instance usage covered by Compute Savings Plans

Amazon Elastic Compute Cloud running Linux/UNIX Reserved Instances

Linux/UNIX (Amazon VPC), c5.large reserved instance applied, c5.large instance used
USD 0.0 hourly fee per Linux/UNIX (Amazon VPC), c5.large instance
USD 0.0 hourly fee per Linux/UNIX (Amazon VPC), c4.large instance

Amazon Elastic Compute Cloud running Linux/UNIX Spot Instances

r4.large Linux/UNIX Spot Instance-hour in 1st East (Virginia) in VPC Zone #7
t3.medium Linux/UNIX Spot Instance-hr in 1st East (Virginia) in VPC Zone #12

Amazon Elastic Compute Cloud running Windows Spot Instances

c3.large Windows Spot Inst-

EBS

\$0.00 for 480 M⁺
\$0.00 for 5⁺
\$0.00 for 1⁺
\$0.00 for 1⁺

S3 (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

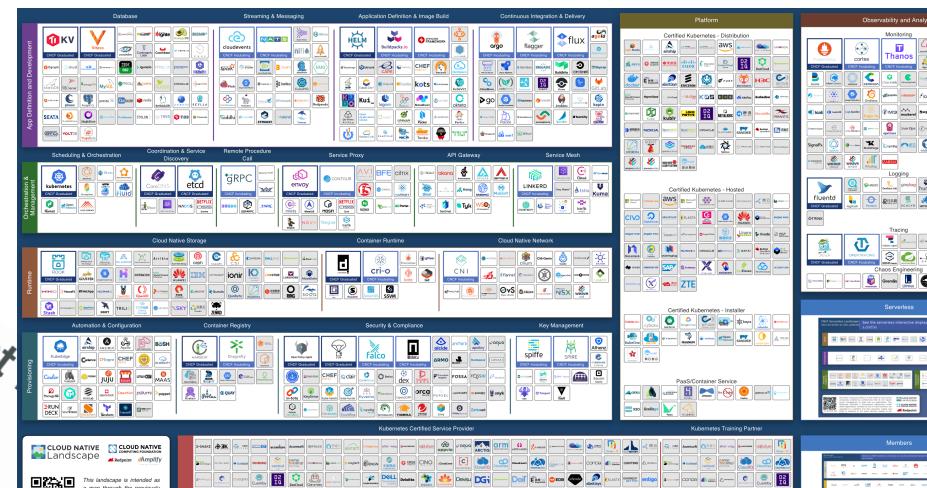
\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month

Amazon Simple Storage Service (Amazon S3) (Standard)

\$0.00 per 1000 objects per month
\$0.00 per 1000 objects per month



Survey Strategy 1

How to go about understanding a system they just inherited?

- Find out what the system is **supposed to be doing**
Who are its users/what do they want from it/**how do they use it/what makes them frustrated about it**
- **Use it yourself**
- Next get a view of the major system components
 - **lookover and verify** (or create) the component diagram, where a component is an independently releasable service / library /etc.)
 - **look over the apis** exposed between components (would not worry about the impl at this level)
 - **examine the schemas** in use (particularly the things passed around in the apis, saved to msg/state stores)
 - **look over the tool choices** for the running system, will have a big impact on the arch and function

How do you know the system is doing what its supposed to be doing?

- Functionally: Using it and, more importantly, **run acceptance tests** (creating them if they don't exist)
- Nonfunctionally: Monitoring and Tracing (you need good metrics and traces to see and **measure what is happening**). Logs won't get your SLI info, move to a metrics culture.

How do you empower the people around you once you know?

- Make monitoring systems **widely available**.

Survey Strategy 2

How does someone learn a system they inherit?

- Dissect the operating environment in terms of **data flow first**, then actual networking connections, then processes.
- Look at the **existing graphs** and then by **building the ones missing** for my desired visibility I get a handle on how stuff actually works

How do you know if that system is solving the problem it's supposed to be?

- Charts and graphs first. If we **don't know the metrics we can't decide** which problem we are supposed to be solving.

How do you move beyond getting yourself up to speed and empower others?

- Involve everyone on the team in the solution space. Empower others by telling them what you see is missing, **hear what they say is missing**, and come up with transparent projects and plans to create sunshine first.

Understanding

Do nothing

Do something

- Self-reflection
- Reflection against others
- Learn and Relearn (unlearn?)

Looking back?

- Talk to someone who has done something similar
- Existing documentation

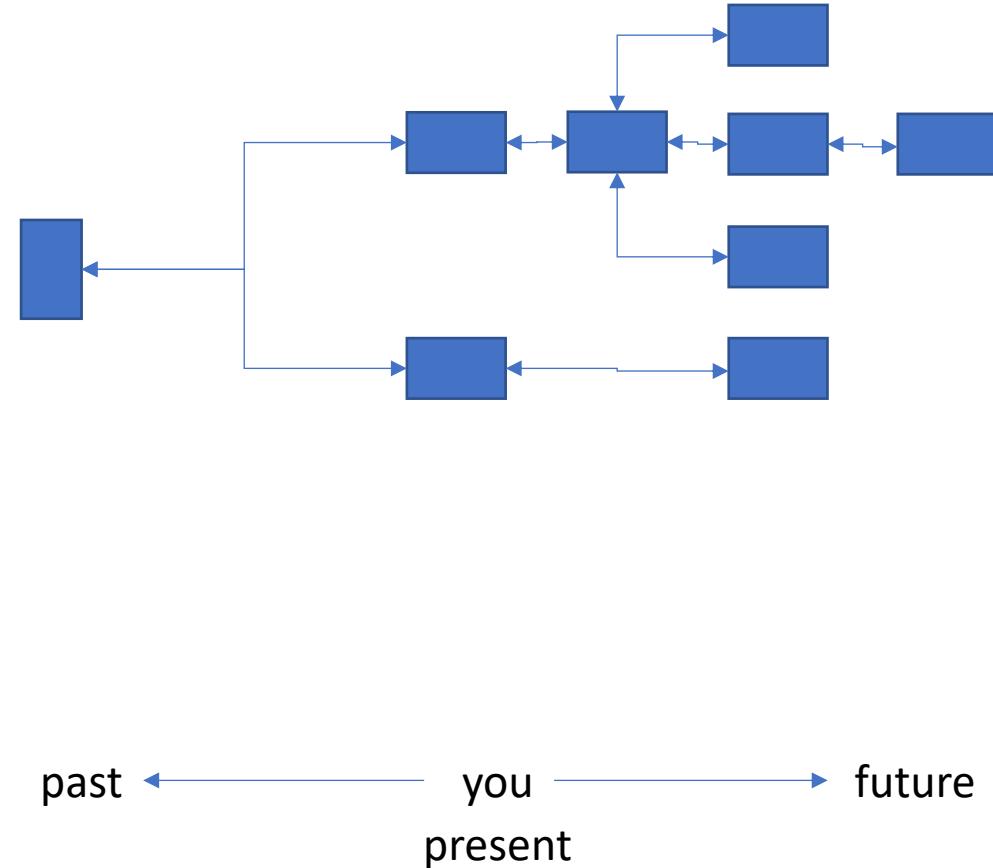
Looking around (last one standing)?

- Authority figure (“because that’s how it works”)
- Characterization Tests

Looking forward?

- Use what I know (alter based on objective)

Lets look at example strategies.



Continuum of Concepts



Virtual

Process < Container < Pod < Kubernetes < Cloud

Break it down!

Know how access cloud console?

- Access bill
- Access dashboards
- Access logs
- Access service controls

Know the basics of Pods?

- Run pods locally

Know how to run a container but not Kubernetes?

- Run containers locally

Don't know how containers work?

- Study how namespaces work
- Study how processes work

```
~$ kubectl run mytest --image=nginx
~$ kubectl logs mytest | tail -2
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up

~$ pidof nginx
20183 20130

~$ echo hi | sudo tee -a /proc/20183/fd/1
hi

~$ kubectl logs mytest | tail -2
/docker-entrypoint.sh: Configuration complete; ready for start up
hi
```

Cloud Focus Tooling



Virtual

Process < Container < Pod < Kubernetes < Cloud

CLI

- Terminal
- Shell
- Processes
- Namespaces
- Networking

CRI/OCI/CNI

- Namespaces
- Networking
- Logging/Metrics

System

- ex. Falco (process/system interaction)
- systemd

Kubernetes

- kubectl top
- Ex. Prometheus (process activity)
- Ex. Istio/LinkerD (process activity on network)

Cloud

- Dashboard
- Logs

Which area is the easiest to study?
Which area has the most to learn?
Which area will empower us the most?
Which area is in our control?

What else?



Virtual

What about studying the CI or CD pipeline?

If you have them, these can be a great start to understanding an existing system.
Be warned it can represent an outdated view, go beyond!

What about everything else?

Outside of understanding, we have additional considerations such as right-sizing, security, optimization, etc.
Tools affect outcomes. A complex tool may give more detail but will take more time (typically).

Approach your system from viewpoints aside your own.
Do not forget about the users, both internal (your company) and external (your clients).

Where to go next?



Virtual

Make a plan.

Share the plan with your colleagues.

Remember, we don't have to be perfect or even always agree. Sharing a plan and acting on it will move you forward.

Concepts:

- Linux Processes: <https://devconnected.com/understanding-processes-on-linux/>
- Linux User Space: <https://www.kernel.org/doc/html/latest/userspace-api/index.html>
- /proc: <https://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html>
- Namespaces: <https://man7.org/linux/man-pages/man7/namespaces.7.html>
- OSI Model: https://en.wikipedia.org/wiki/OSI_model
- Microservices: <https://en.wikipedia.org/wiki/Microservices>

Tools/Systems:

Observability and Analysis: <https://landscape.cncf.io/card-mode?category=observability-and-analysis&grouping=category>

Security and Compliance: <https://landscape.cncf.io/card-mode?category=security-compliance&grouping=category>

Service Mesh: <https://landscape.cncf.io/card-mode?category=service-mesh&grouping=category>

Thank You!

Find us at:

- <https://rx-m.com>
- <https://www.linkedin.com/in/ronaldpetty/> | ronald.petty@rx-m.com

Certification Training:

CKAD / CKA / CKS – self-study and instructor-led

RX-M Cloud Native Expertise:

Training, Consulting/Advisory, Staffing

- Largest breadth of cloud native / open source training curriculum
- Custom consultative training programs aligned to business goals and job roles



KubeCon



CloudNativeCon

Europe 2021

Virtual



cloud native
training &
consulting



CLOUD NATIVE
COMPUTING FOUNDATION
SILVER MEMBER

THE LINUX
FOUNDATION
AUTHORIZED TRAINING
PARTNER

