

# Notes About Cloud Without Compromise

Adrian Kosmaczewski

2021-11-12

I've been reading "Cloud Without Compromise: Hybrid Cloud for the Enterprise" by Paul Zikopoulos and Christopher Bienko. I think it's a brilliant book; concise, easy to read, funny, and a great resource for non-technical people who would like to understand how the modern "Cloud" works. Highly recommended.

Here are some notes I took.

## Mantra

The book repeats this mantra over and over again:

"Cloud is a capability, not a destination."

## Selected Quotes

About the evolution of terminology and technology:

When was the last time you heard someone at work say, "I found it on our intranet?" Answer: you were probably dancing the Macarena and weren't embarrassed doing it.

Page 12

---

Remember to turn off your unused services:

Our pro tip: always remember that public cloud pricing is utility pricing—the meter is running, just like how you're billed for electricity in your home (and that can be a good or a bad thing as illustrated earlier). Running a nonstop application, or even a developer forgetting to shut down their test instances (one of us, whose name isn't Chris, did that and cost his department \$5,000), is like walking around your empty house with all the lights on and wondering aloud about all the money you could be saving. (If you have kids, this is a well-known feeling to you.)

Page 29

---

Probably the most brilliant quote from the book is this one:

Every client we talk to is doing one of two things: spending money to save money or spending money to make money. When you spend it (money) to save it, you're renovating and when you spend it to make it, you're innovating.

Page 31

---

The main goal of all this is speed:

The theme in modern application development is speed. Speed of scale, speed of delivery, and so on. The very nature of speed is the DevSecOps delivery nuance—bringing together the speed developers love (they think project to project), and the resiliency and assuredness that operations seek (they think about things like service level agreements, upgrades, patching, and security).

Page 44

---

About Kubernetes:

We'd be so bold as to say that Kubernetes is the operating system for distributed systems of containers regardless of the hardware architecture. What does an operating system like Linux, Windows, or iOS really do? It manages resources (CPU, memory, storage, communications, and so on) and that's exactly what K8s does: it manages the resources of a cluster of nodes—each of which is running its own operating system.

Page 51

---

About the Microservices architecture:

But it's more than that. In a modern system, you might need several databases, an authentication service, a service that manages stock in the warehouse, a service that does billing, a service that queues orders to be shipped, a service that computes taxes, a service that manages currency exchange, and so on. Make your own list for your own business: it will be very extensive. And if your application is used heavily, you may need many copies of these services to handle the load.

Page 52

---

Putting technical terms in a language that managers can understand is an underrated gift. Paragraphs such as the following are really, really helpful:

Moving to the cloud gives you increased agility and an opportunity for capacity planning that's similar to the concept of using electricity: it's "metered" based on the usage that you pay for (yes, you'll yell at developers who leave their services running when not in use the same way you do your kids for leaving the lights on in an empty room). In other words, IaaS enables you to consider compute resources as if they are a utility like electricity.

Page 66