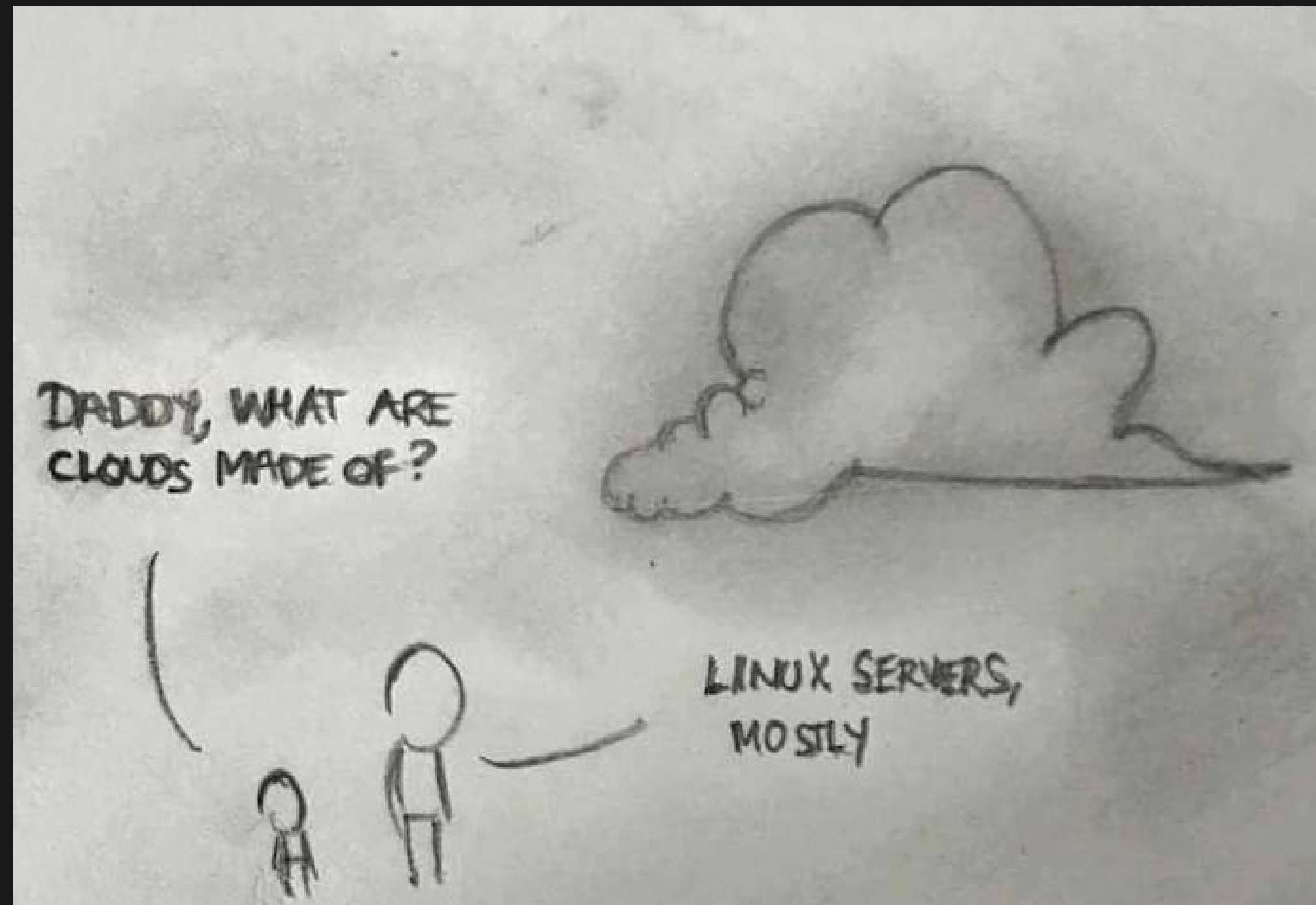


Usługi i platformy deweloperskie dla aplikacji w chmurze



Wojciech Barczyński
wojciech.barczyński@wsb.wroclaw.pl

Whoami

Wojciech Barczyński:

- System Engineer
- Lead @ Codility

Cel

- Azure
- Narzędzia

Zaliczenie

- Projekt - 2x Azure/AWS/GCP service - [github/gitlab](#)
- Egzamin

Wykład & ćwiczenia

- Materiały: [github](#)

Środowisko

- Ubuntu 20.04 albo inna dystrybucja / *nix - MacOS
- Dual-boot?
- Lang: Python, Golang, dotnet core, ...

Ćwiczenia

- Problem
- Mierzymy się z zadaniem samemu - timebox
- Pytamy / prosimy o pomoc.
- Zrobiłam / Zrobiłem - warto pokazać
- Kopiuj&Wklej zakazane (chyba, że wykładowca powie inaczej)

Dlaczego chmura?

- Elastyczność
- Niższe koszty
- Wszystko API, łatwe do automatyzacji
- Pay As You Go
- Gwałtowna demokratyzacja technologii

Dlaczego chmura?

- Potężne narzędzie do rozwiązywania problemów
- Cloud-first, Mobile-first

Ewolucja 1

- Własny hardware

Ewolucja 1

Trendy:

- Rozwój CPU -> CPU idle
- Dużo sprzętu -> szybko się starzeje

Ewolucja 2

- Wirtualizacja na niezawodnym i dedykowanym sprzęcie
- Konsolidacja

Ewolucja 2

- Dużo sprzętu -> szybko się starzeje
- Mała elastyczność
- Trudne do zautomatyzowania
- Amazon i Black Friday
- Google buduje na zwykłym sprzęcie

Ewolucja 3

- Masowa wirtualizacja na zwykłym*/zawodnym sprzęcie

* - tak się zaczęło...

Cattle vs Pets



Cattle vs Pets



D001, D002, D003, D004, Disabled

Cattle vs Pets

- Wirtualne Maszyny może paść
- Wirtualna Maszyna może być przemigrowana
- Chatty neighbours...

Private
(On-Premise)

Applications

Runtimes

Security

Databases

Servers

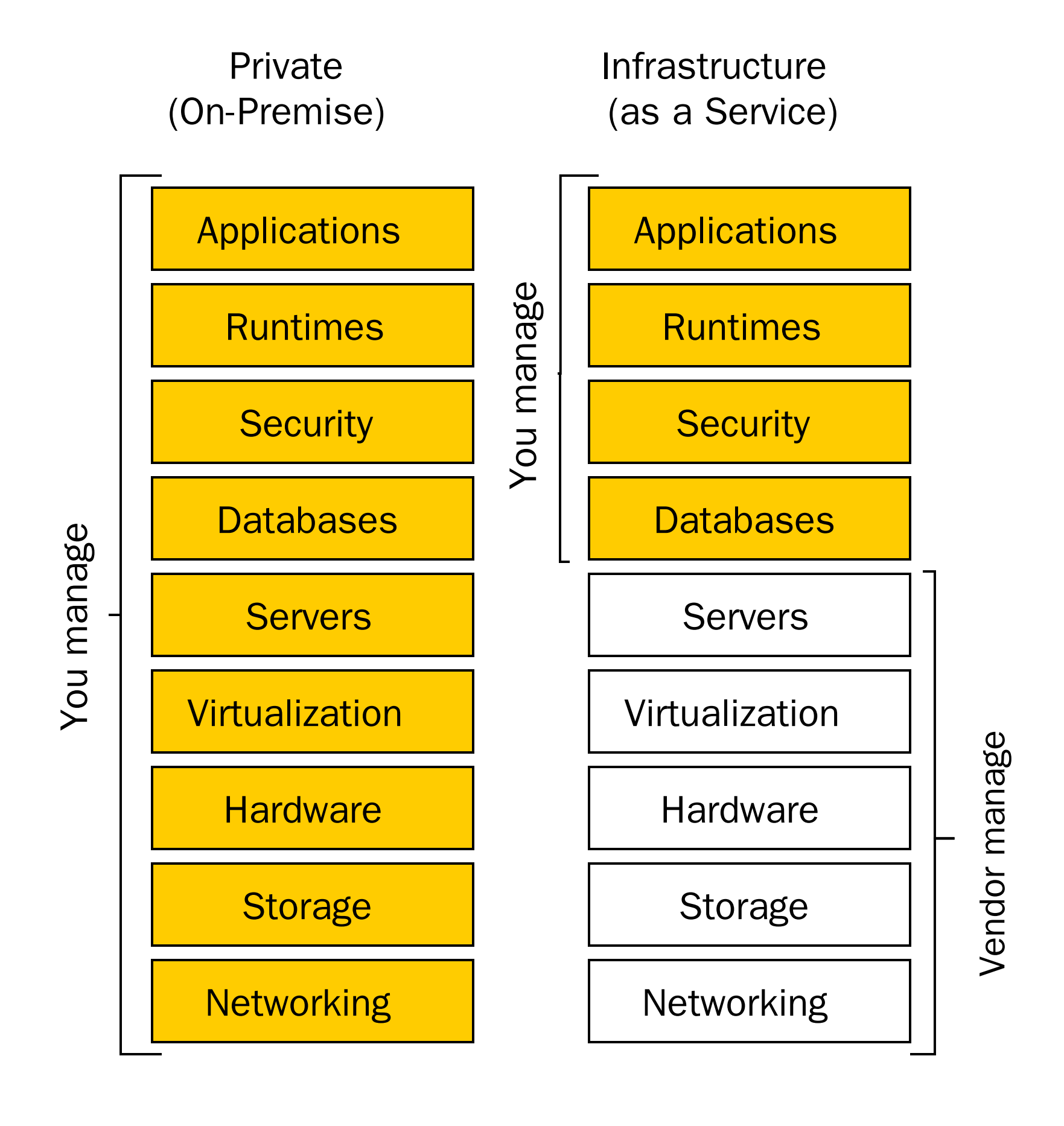
Virtualization

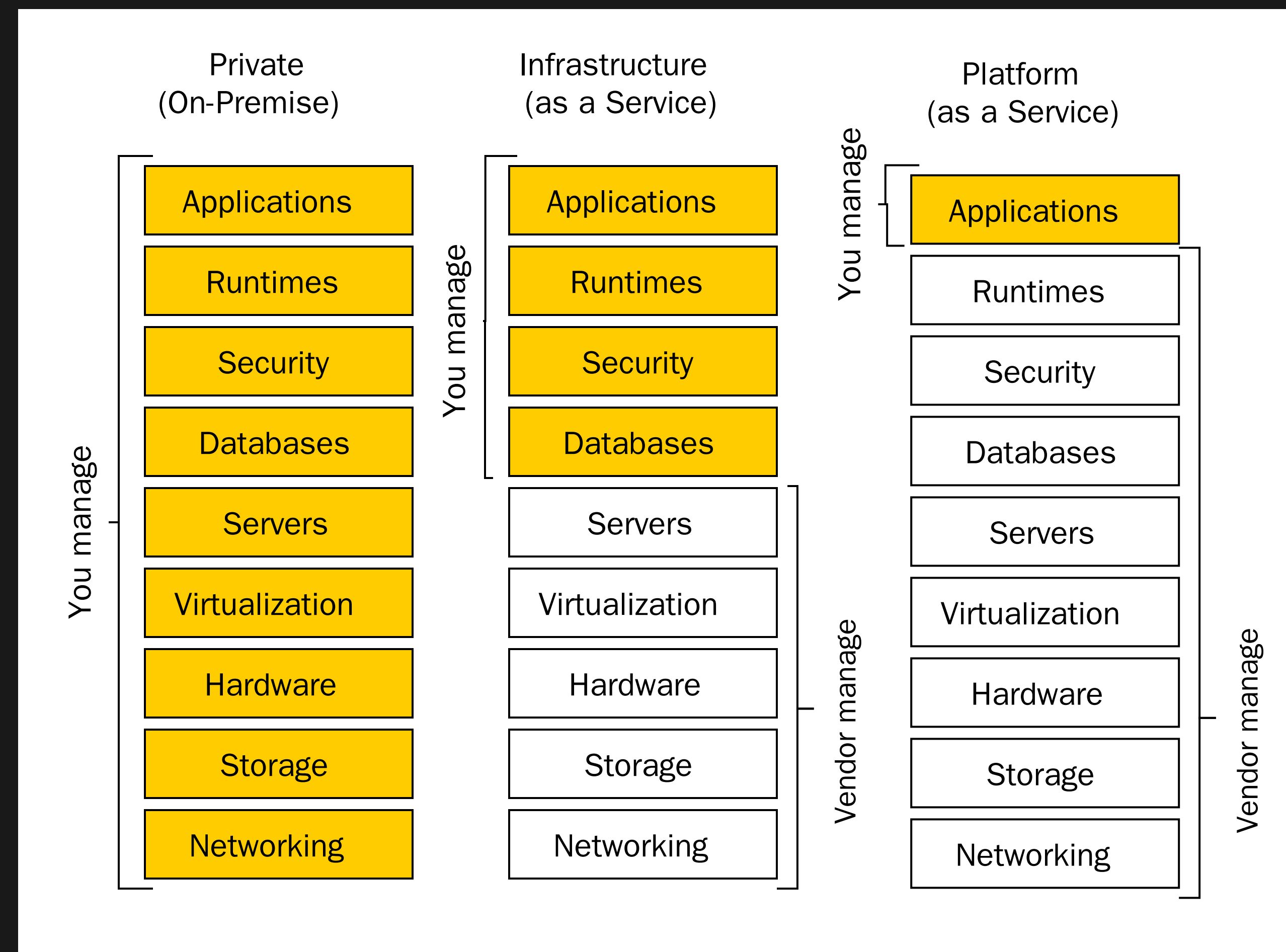
Hardware

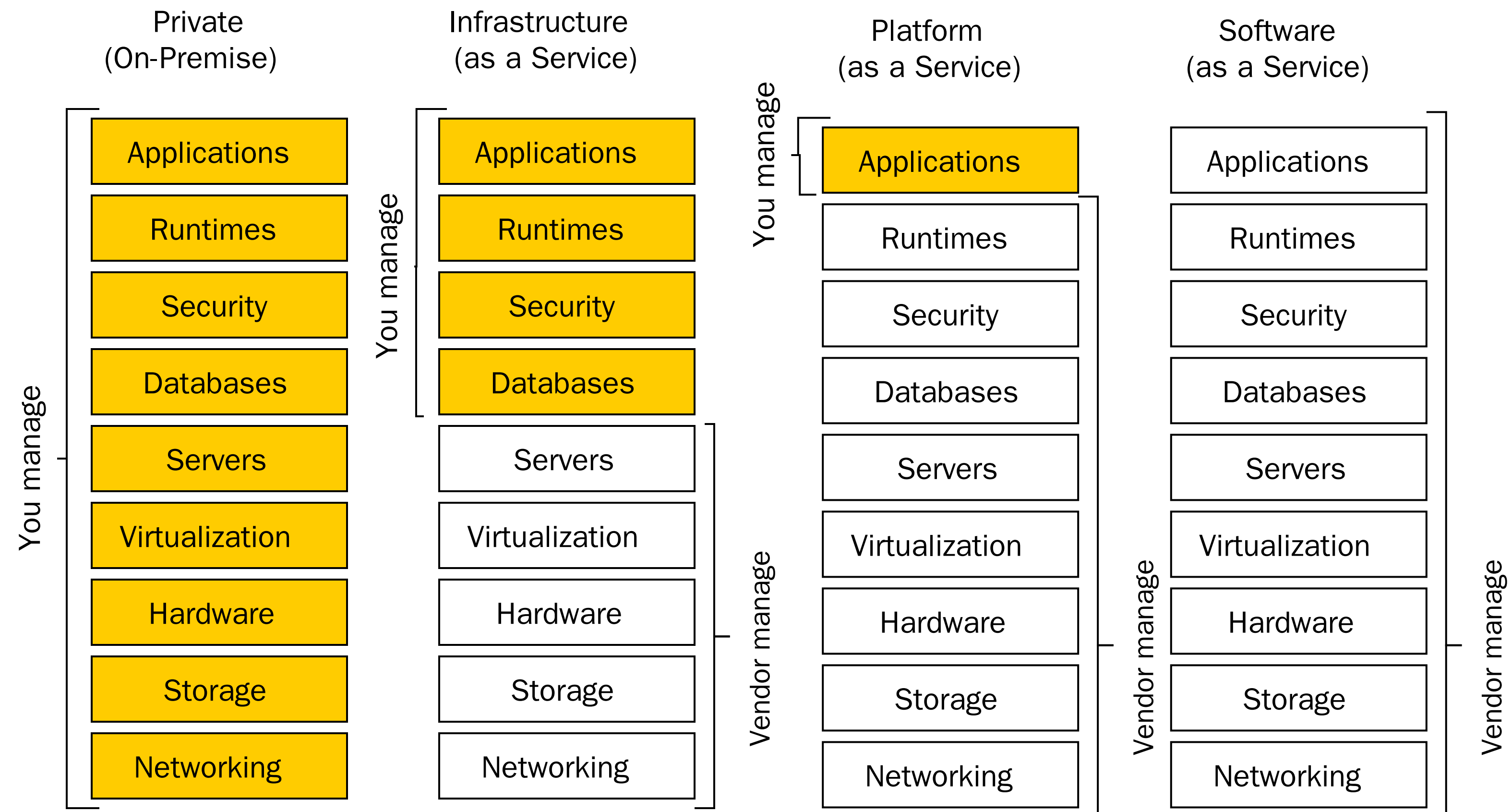
Storage

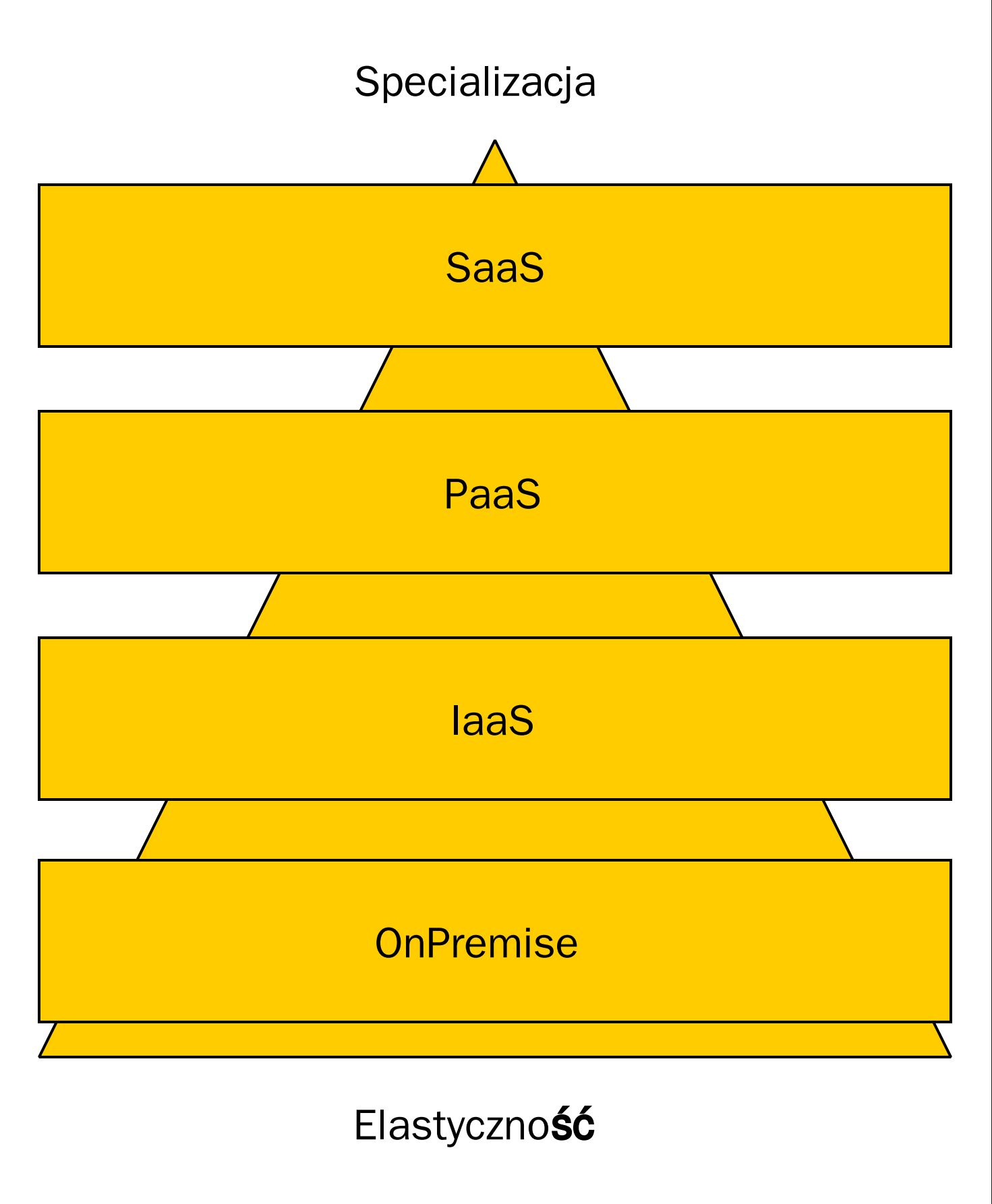
Networking

You manage









MTTR vs MTTF

- Chmura: MTTR
- Wcześniej: MTTF

Fallacies of distributed computing

Jak wybrać?

- 100 serwisów
- K8S? Serverless? EC2 + xyz

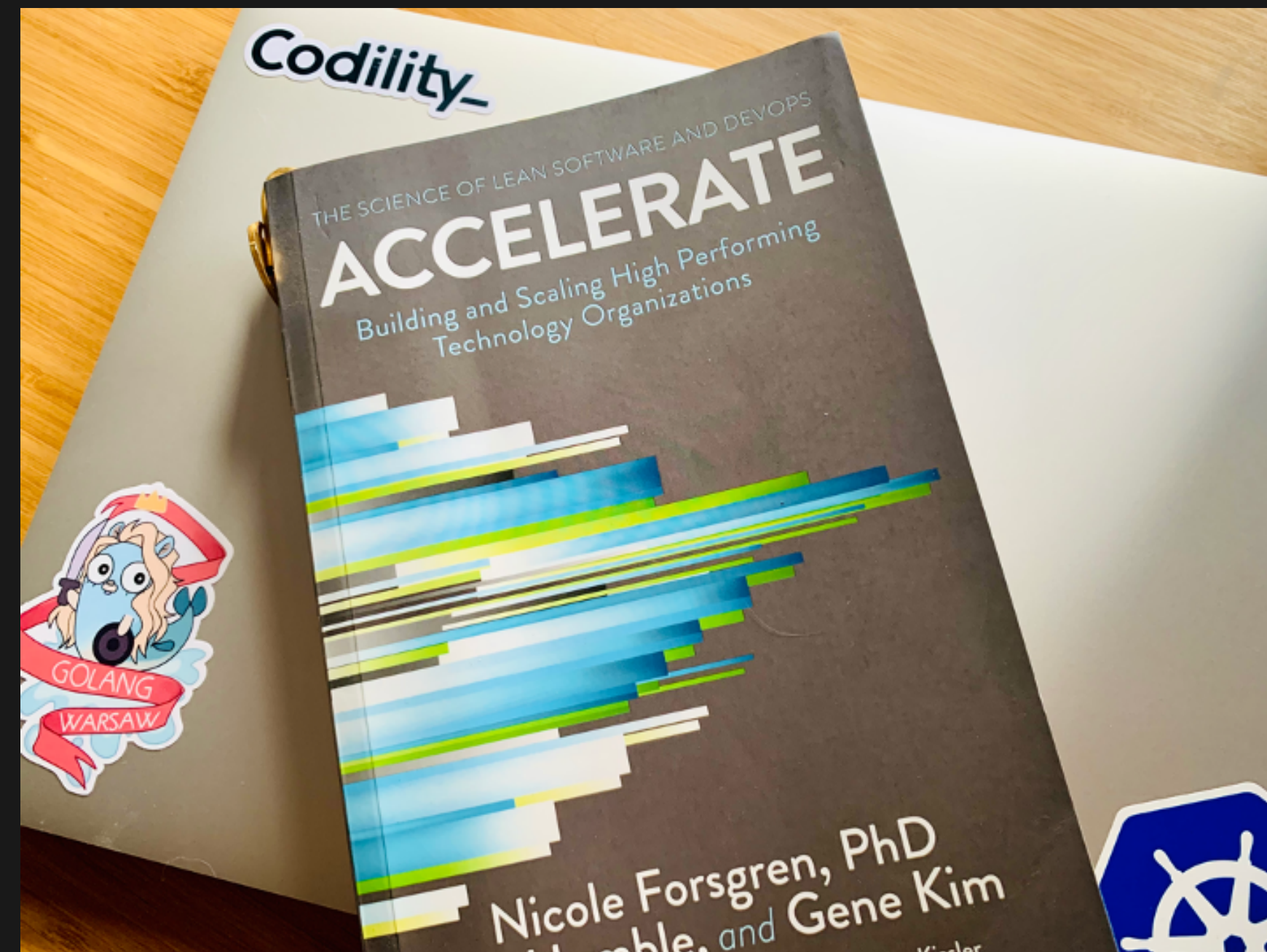
Jak wybrać?



High performance teams

(Tech) metrics:

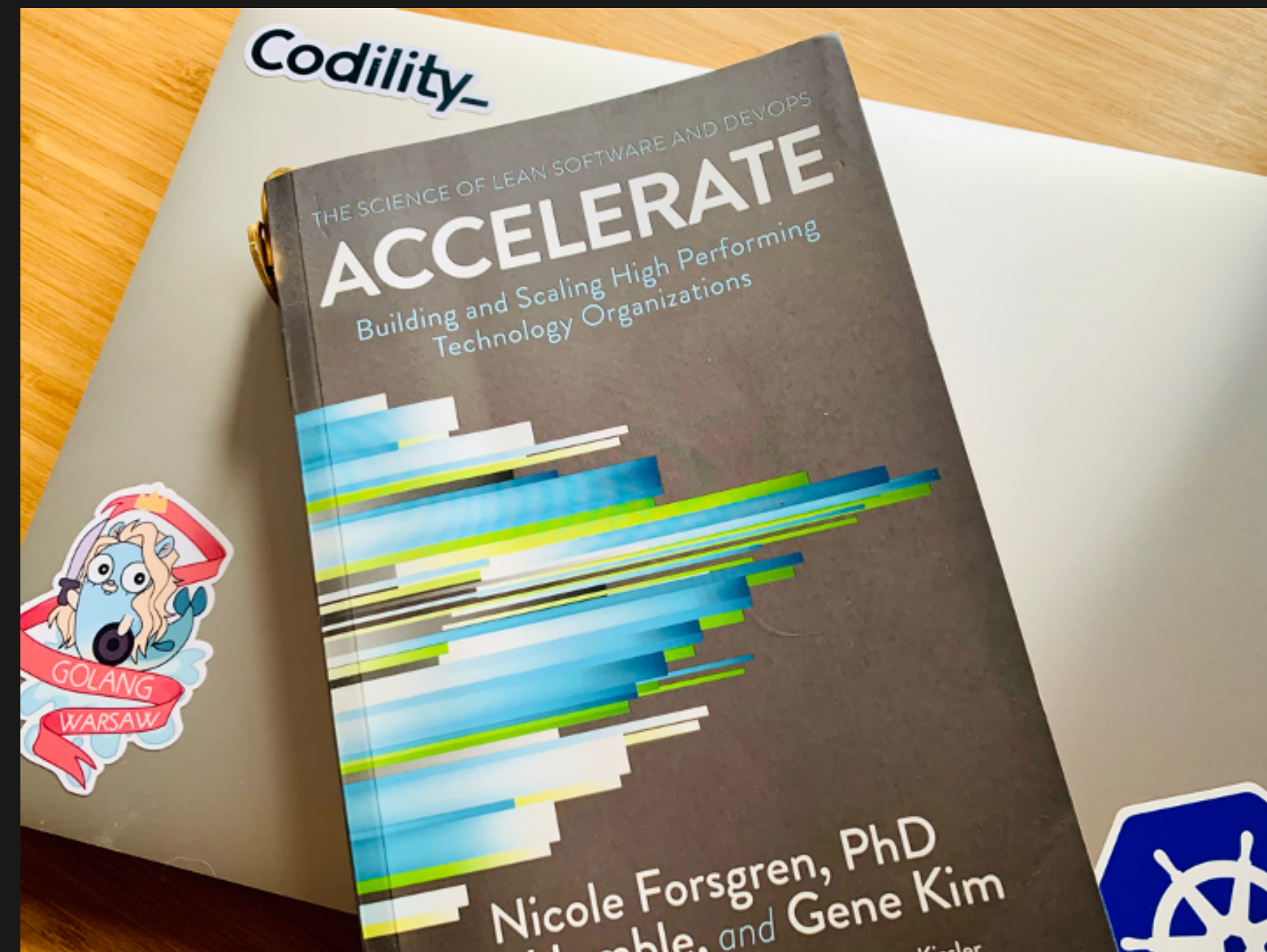
- Lead Time
- Deployment frequency
- Mean time to Recovery
- Change Fail Percent



Choosing tech

Other factors:

- New tech for people to feel that they learn sth new
- What the tech background is



Buy > build

Kandydaci:

- Wysokie TCO, nie jest to częścią naszego produktu
- Skomplikowany i złożony problem np., bazy danych
- Narzędzia

Oczekiwania wobec dev

- Samo programowanie nie wystarcza!

Oczekiwania wobec dev

Świadomość operacyjnej strony oprogramowania:

- monitoring - RED, 4 Golden Signals, USE
- logging - structured
np., Python, Golang, dotnet core, java

Oczekiwania wobec dev

Świadomość operacyjnej strony oprogramowania:

- tracing
- probes

Oczekiwania wobec dev

	Metrics	Logging	Tracing
CapEx	Medium	Low	High
OpEx	Low	High	Medium
Reaction	High	Medium	Low
Investigation	Low	Medium	High

Go for Industrial Programming by Peter Bourgon

Oczekiwania wobec dev

Warto obejrzeć:

Engineering You • Martin Thompson

Oczekiwania wobec dev

Świadomość operacyjnej strony oprogramowania:

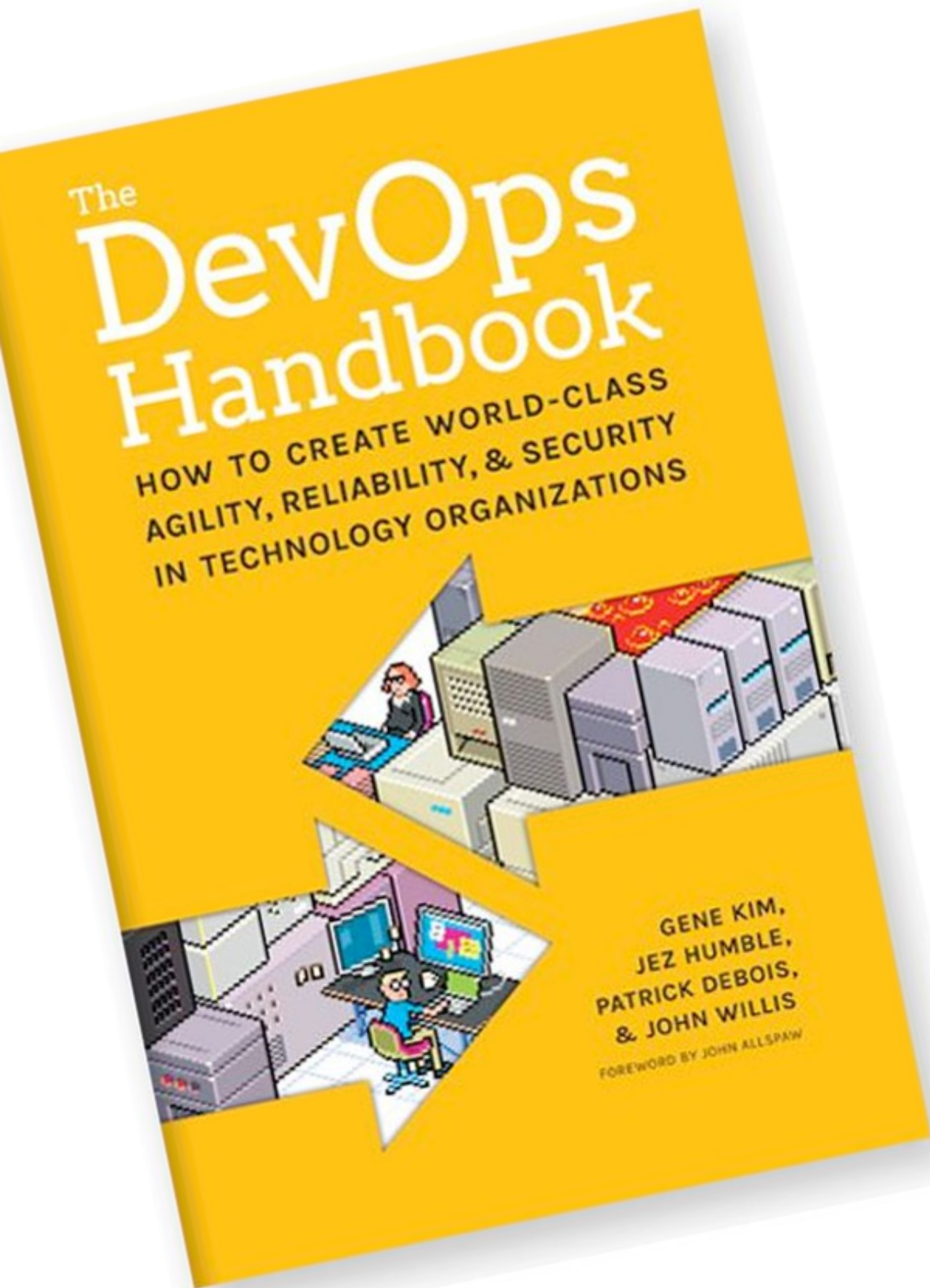
- you build it, you run it
- devs, ops na OnCall

Retry and timeouts

- timeouts
- Retries
- Circute breaker

```
requests.get('https://github.com', timeout=0.001)
```


Warto przeczytać



1

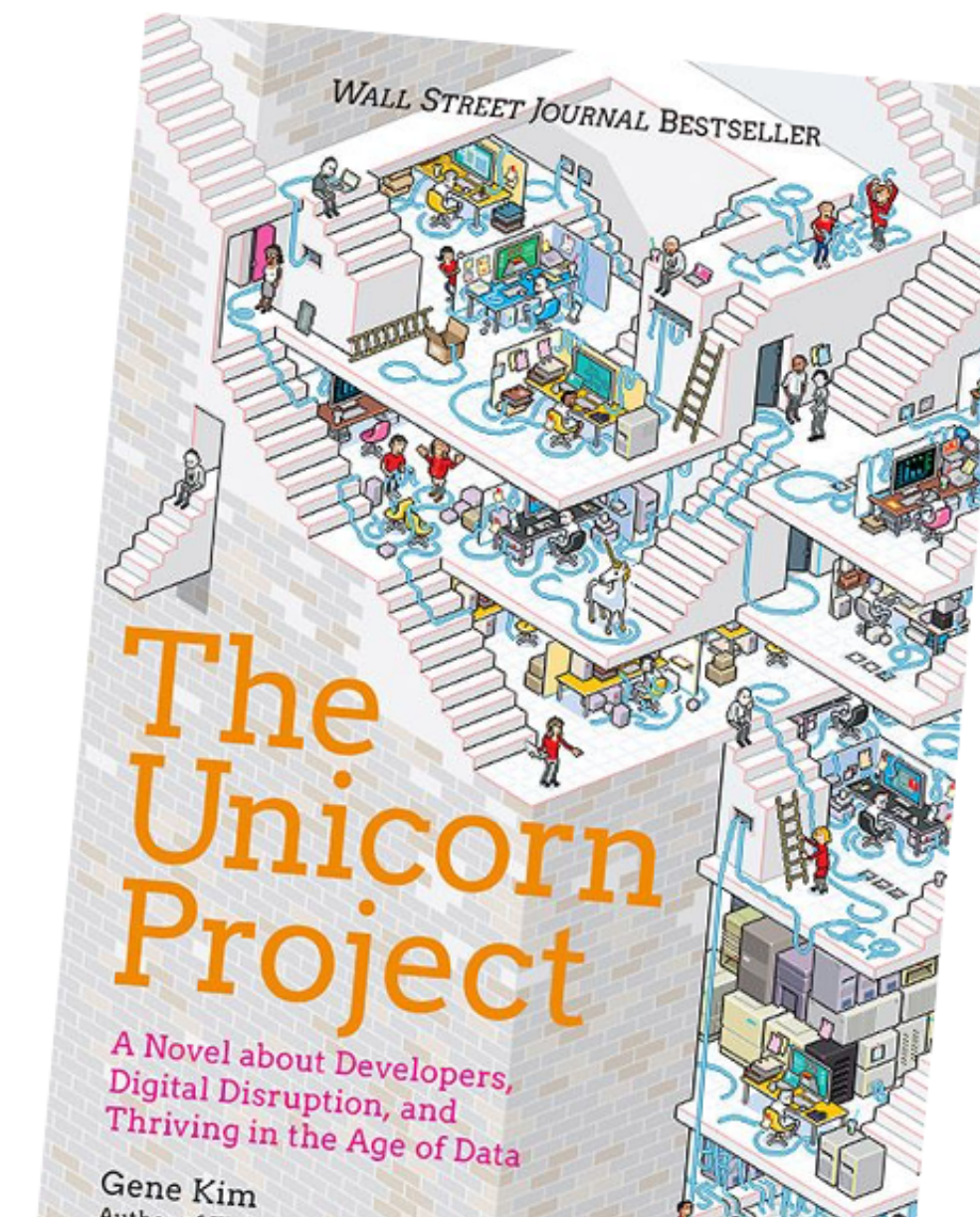
Konkretnie, zrób A, a potem B,
ponieważ...
(DevOps oznacza kulturę, a nie admina 😊)

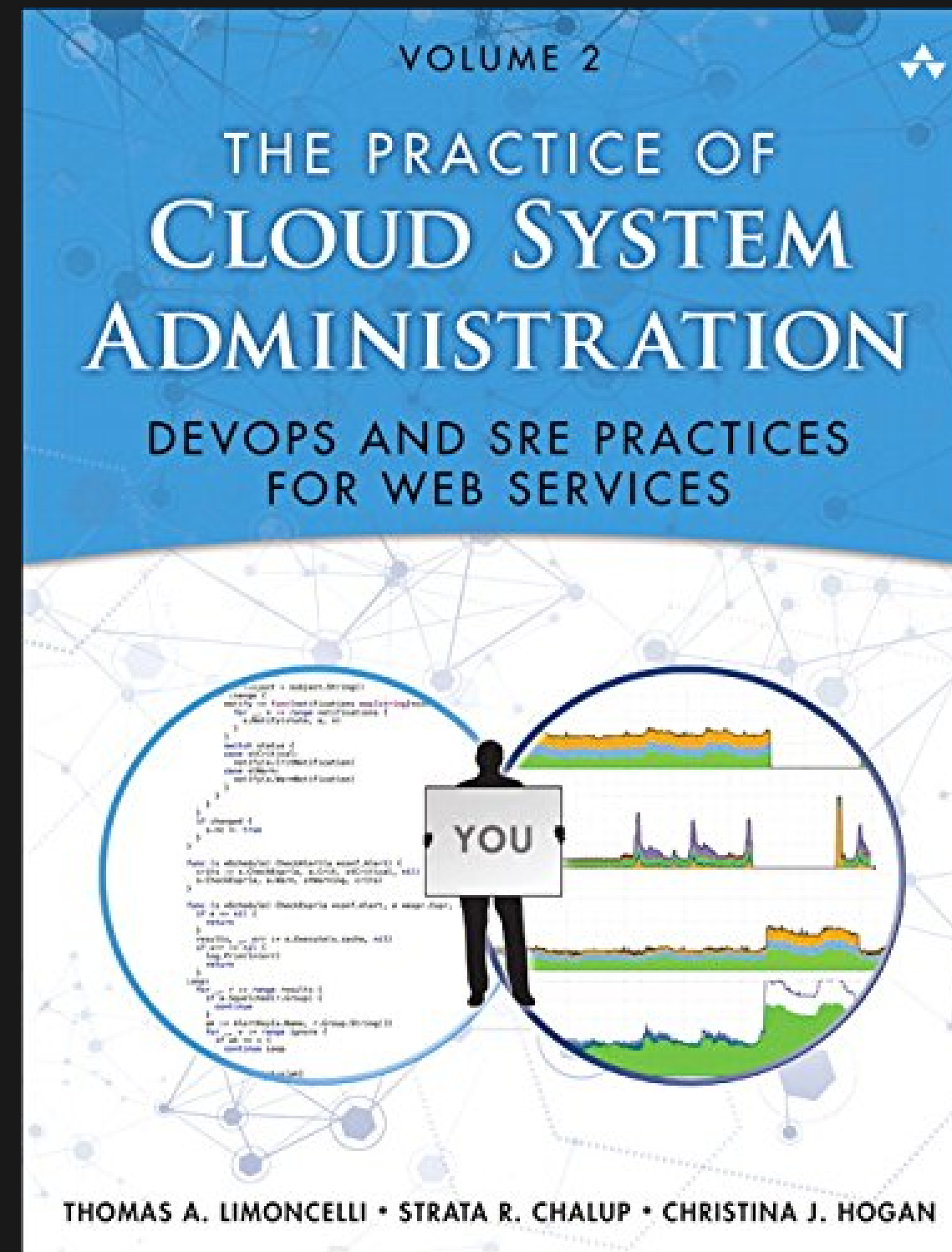
2

Pierwsze 4 rozdziały najwięcej
wartości, szybko się czyta,
reszta jak macie czas.

3

Przypowieść,
nie jest to prosto
z mostu jak 1





Practice of Cloud System Administration, The: DevOps and SRE Practices for Web Services, Volume 2

12factor apps

- Good to know what [12factorapp](#) is.
- Not new, might look obvious.

12factor apps

Few that are very relevant for Kubernetes:

1. III. **Config** - environment variable, injected by environment
2. VIII. **Concurrency by process**
3. IX. **Disposability** (MTTR vs MTTF)

12factor apps

- 4. X. **Dev/Prod parity** including time, tooling, and backing services
- 5. XI. **Logs** - push your logs to stdout and let log routers to take care of them

Praca z chmurą

Automate everything?



best practices

Automate everything?

- A journey

Automate everything?

Kroki:

1. Skrypty/README
2. Infrastructure-as-a-code w iteracjach
3. Unikać rzeczy utworzonych z ręki bez śladu w naszych git (doc, kod)

Automate everything?

Kroki:

4. Dobrze mieć z tyły głowy: x2, x5
5. Rozdzielić środowiska dev, staging, prod na osobne konta wcześniej niż później

Technologie

1. Terraform
2. Terragrunt
3. Ansible / Saltstack
4. AWS CDF?
5. [Pulumi](#) jest ciekawym projektem

Continuous deployment

- CI/CD zaczyna się na komputerze programisty
- Komponenty powinny być uruchamialne lokalnie

Continuous deployment

- Działający ci/cd jest najwyższym priorytetem!

Logowanie do VM

1. Zawsze ssh z kluczami
2. Przez bastion
3. AWS Systems Manager Session Manager / AWS EC2 Instance Connect lub alternatywa

ssh z kluczen rsa

1. Tworzymy parę: prywatny i publiczny klucz

```
# wygenerowany klucz znajdzie się  
# w ~/.ssh/id_rsa  
ssh-keygen -t rsa -b 4096 -C "wbjt@wsb.wroclaw.pl"  
  
# jeśli chcemy mieć więcej niż jeden klucz  
ssh-keygen -t rsa -b 4096 -C "wbjt@wsb.wroclaw.pl" \  
-f id_rsa_wsb
```

ssh z kluczem rsa

- Publiczny `id_rsa.pub` umieszczamy na serwerze
- Prywatny `id_rsa` chronimy jak oka w głowie

ssh z kluczem rsa

2. Przegląd

```
ls ~/.ssh  
id_rsa # nigdy nie dzielimy z ni  
id_rsa.pub
```

3. Debugowanie:

```
ssh -vv github.com
```

ssh z kluczem rsa

Ćwiczenie:

używanie klucza ssh do pracy z repozytorium github
[instrukcja.](#)

Jak wybrać x?

1. warto mieć **ADR** lub inną lekką formę dokumentacji decyzji
2. osobne konto na eksperymenty dla teamu

Jak wybrać x?

- warto mieć **ADR** lub inną lekką formę dokumentacji decyzji

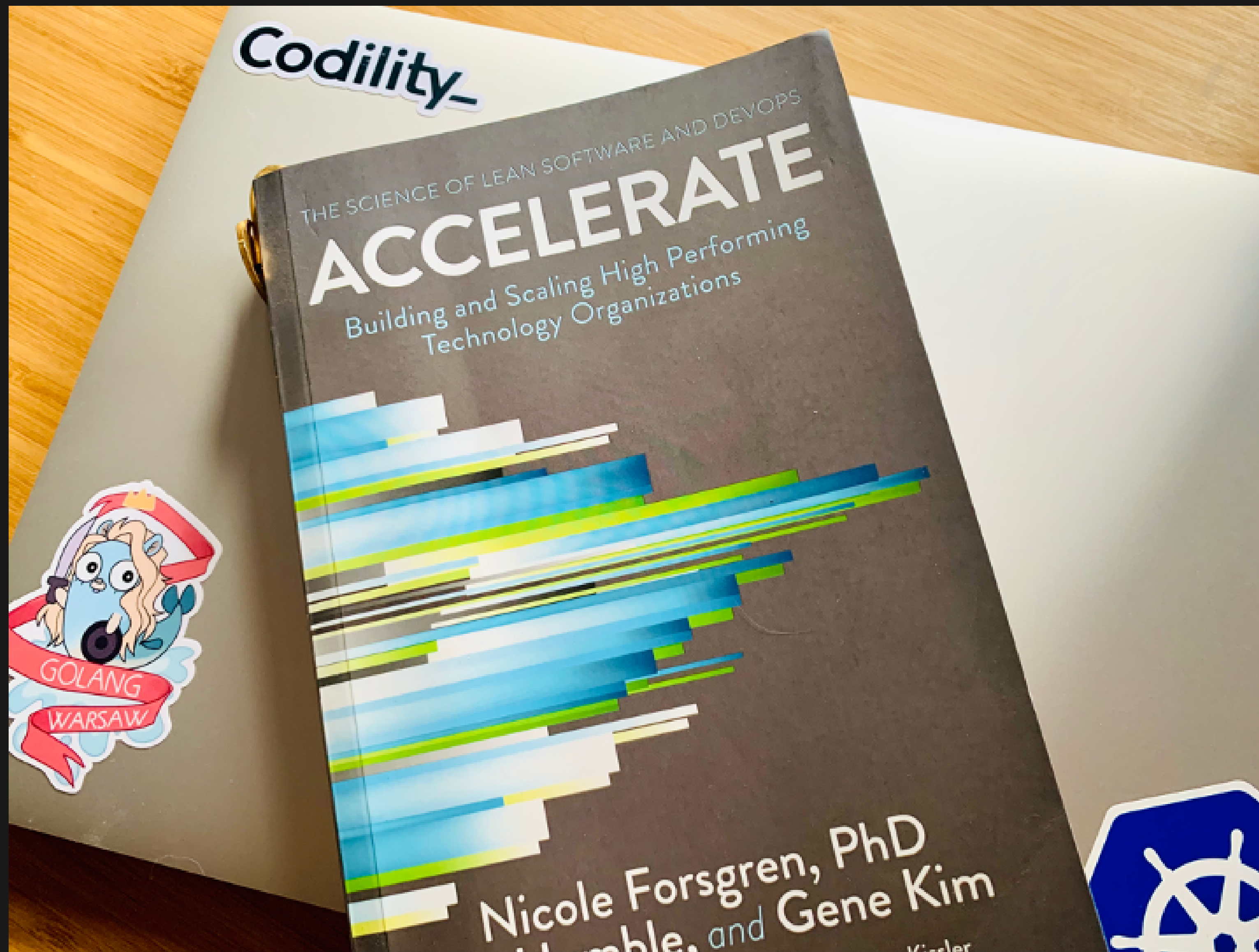
Jak wybrać region

- Aspekty prawne
- Gdzie jest mój klient? (latency)
- Cena i dostępność usług

Jak wybrać vm

- Cena
- Wymagania aplikacji
- VM types

Dziękuję za uwagę



Backup slides

Azure batch

Azure batch

Zestaw narzędzi wiersza poleceń Azure CLI udostępnia polecenie `az batch` do pracy z usługą Azure Batch. Pozwala ono na zarządzanie kontem, pulami węzłów oraz zadaniami.

Azure batch

```
az batch account create \  
  --resource-group <nazwa-grupy-zasobów> \  
  --name <nazwa-konta> \  
  --location <region>
```

Azure batch

```
az batch account login \  
  --resource-group <nazwa-grupy-zasobów> \  
  --name <nazwa-konta> \  
  --shared-key-auth
```

Azure batch

```
az batch pool create \  
  --id <id-puli> \  
  --vm-size <rozmiar-węzła > \  
  --target-dedicated-nodes <liczba-węzłów> \  
  --image <obraz-maszyny> \  
  --node-agent-sku-id "batch.node.ubuntu 18.04"
```

Azure batch

```
az batch task create \  
  --task-id <id-zadania> \  
  --job-id <id-zadania-nadrzędnego> \  
  --command-line <zdanie-do-wykonania>
```