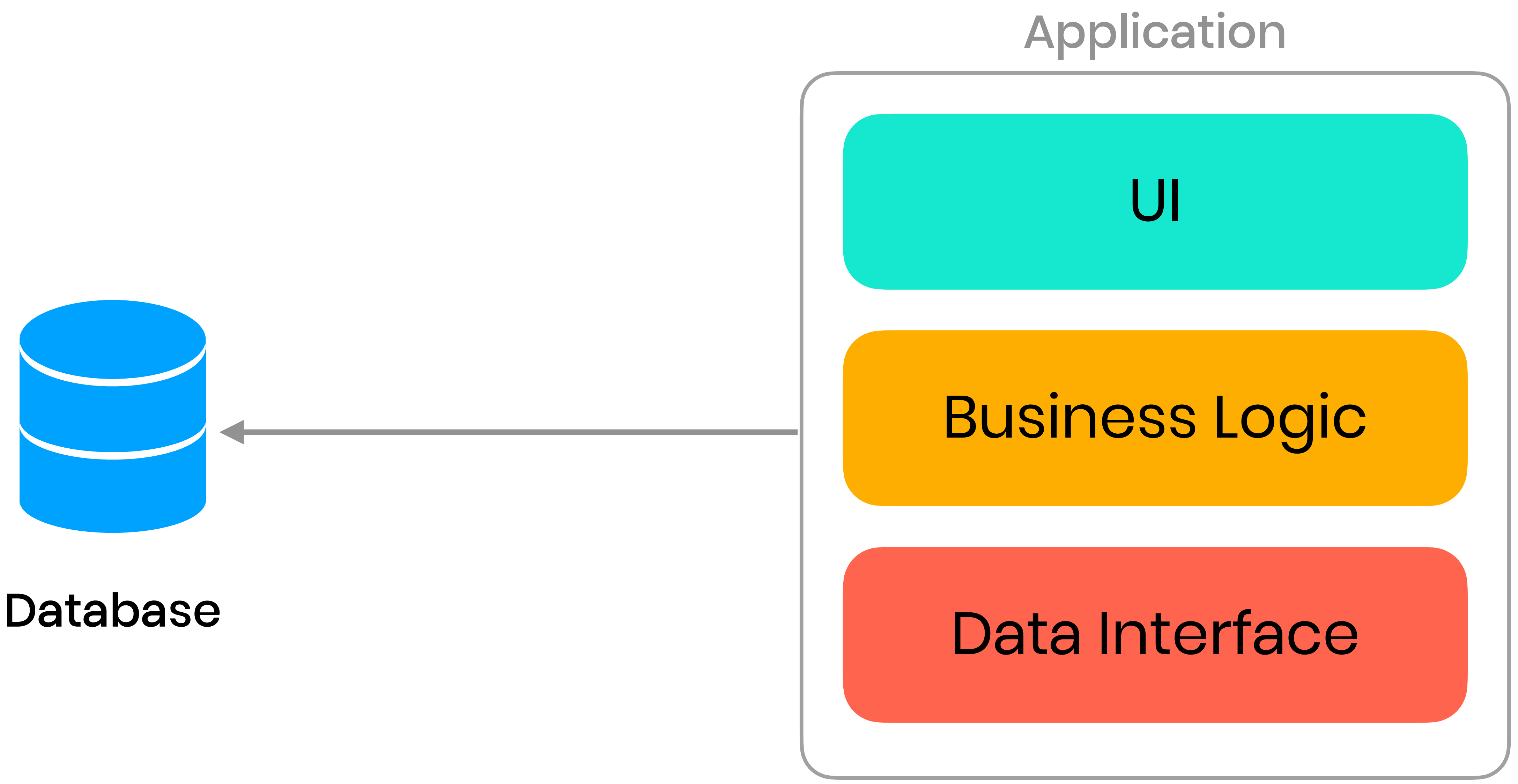
Are Containers Dying? Rethinking Isolation with MicroVMs

Muhammad Yuga Nugraha

Agenda

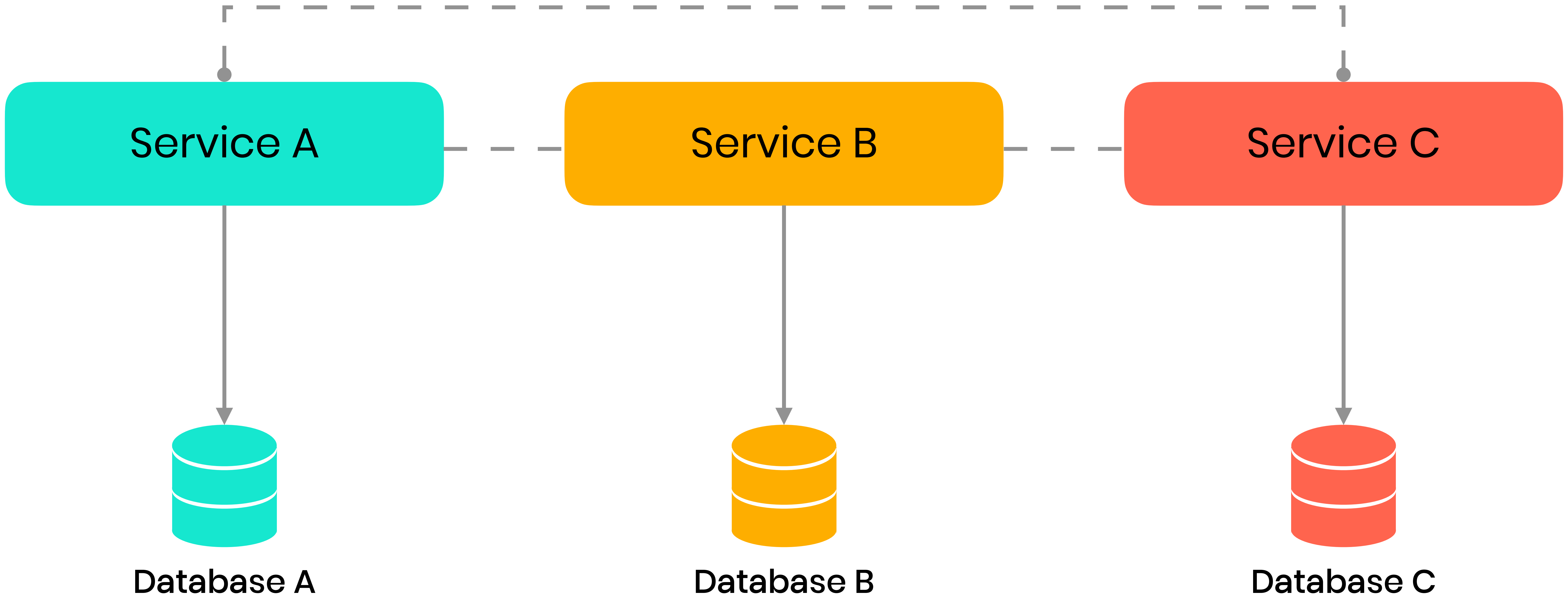
1. How we run applications today
2. Why are we still using containers today?
3. What microVM bring to the table
4. The challenges

How we run applications today



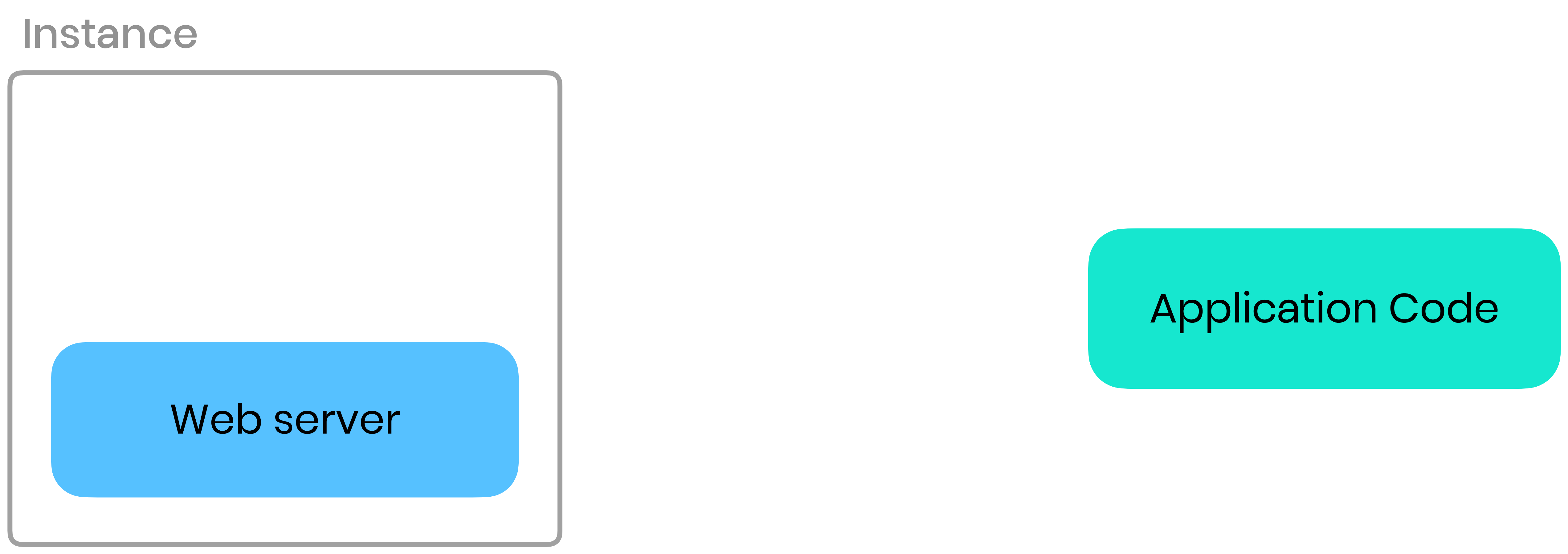
Monolith Architecture

How we run applications today



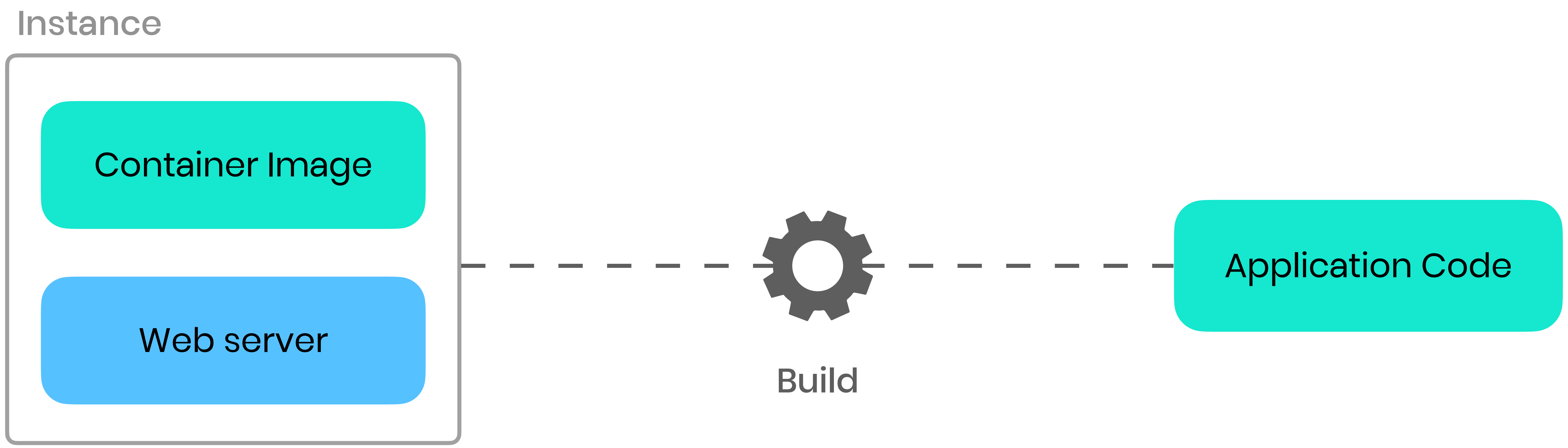
Microservice Architecture

How we deploy applications today



Virtual Machine (VM)

How we deploy applications today

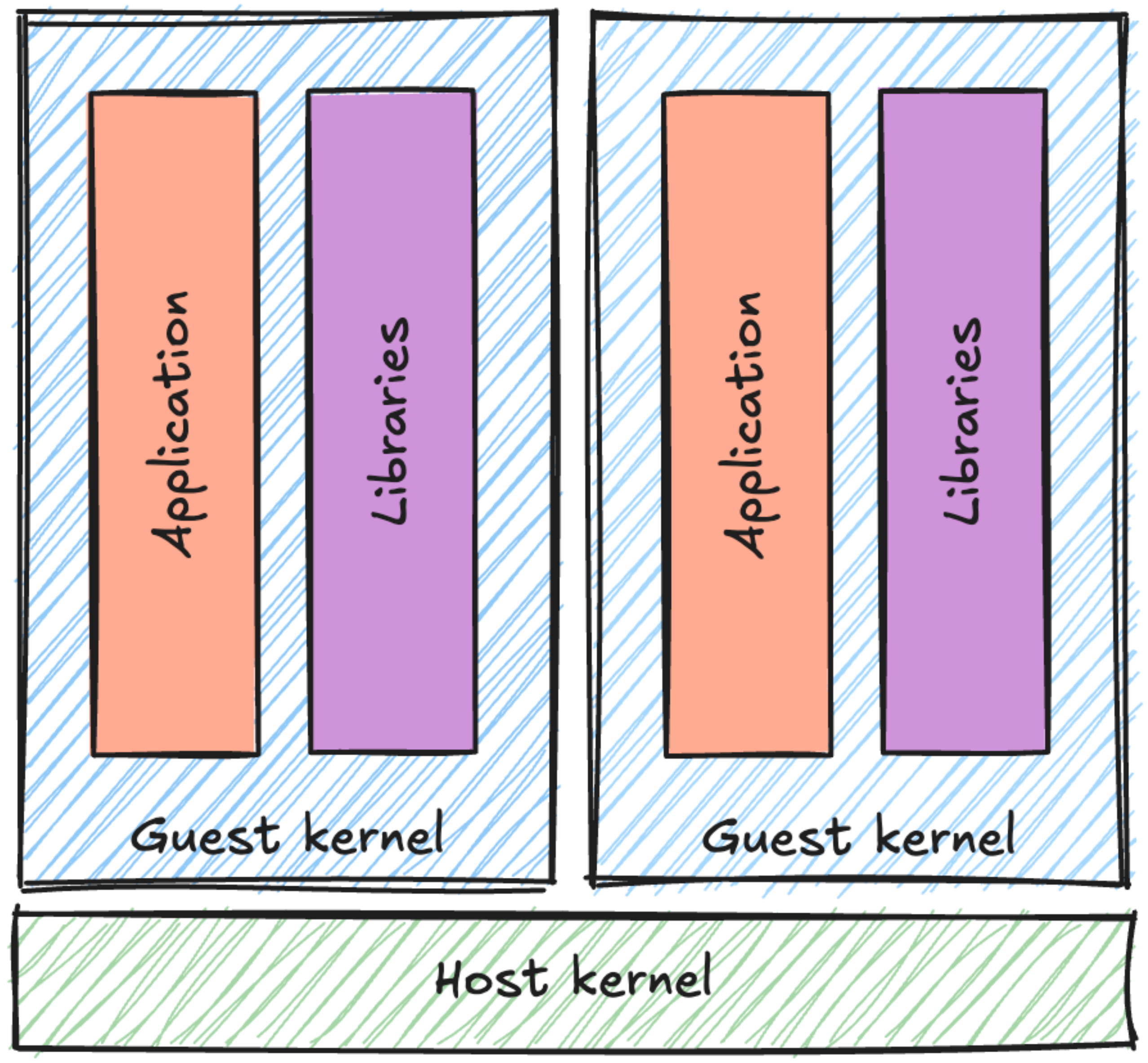


Container

microVM
Container
VM
imgflip.com

Virtual Machine (VM) - The Good

1. Emulates full physical hardware
2. Strong isolation with its own kernel and OS
3. Allows running multiple OS on a single physical host
4. Suitable for legacy apps

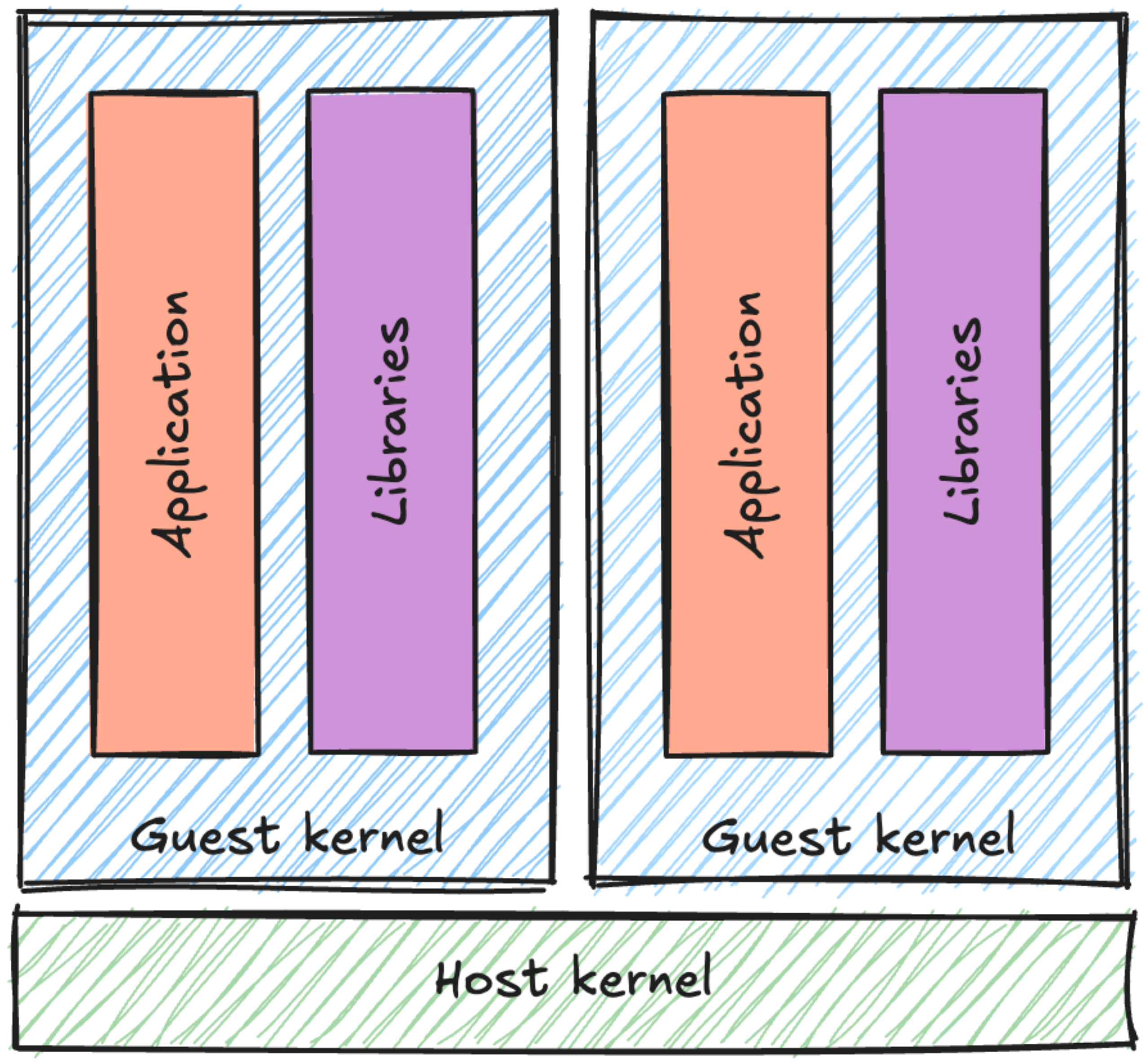


Virtual Machine (VM) - The Bad

1 Heavy resource

2 Long boot times

3 Not ideal for scaling



Container - The Good

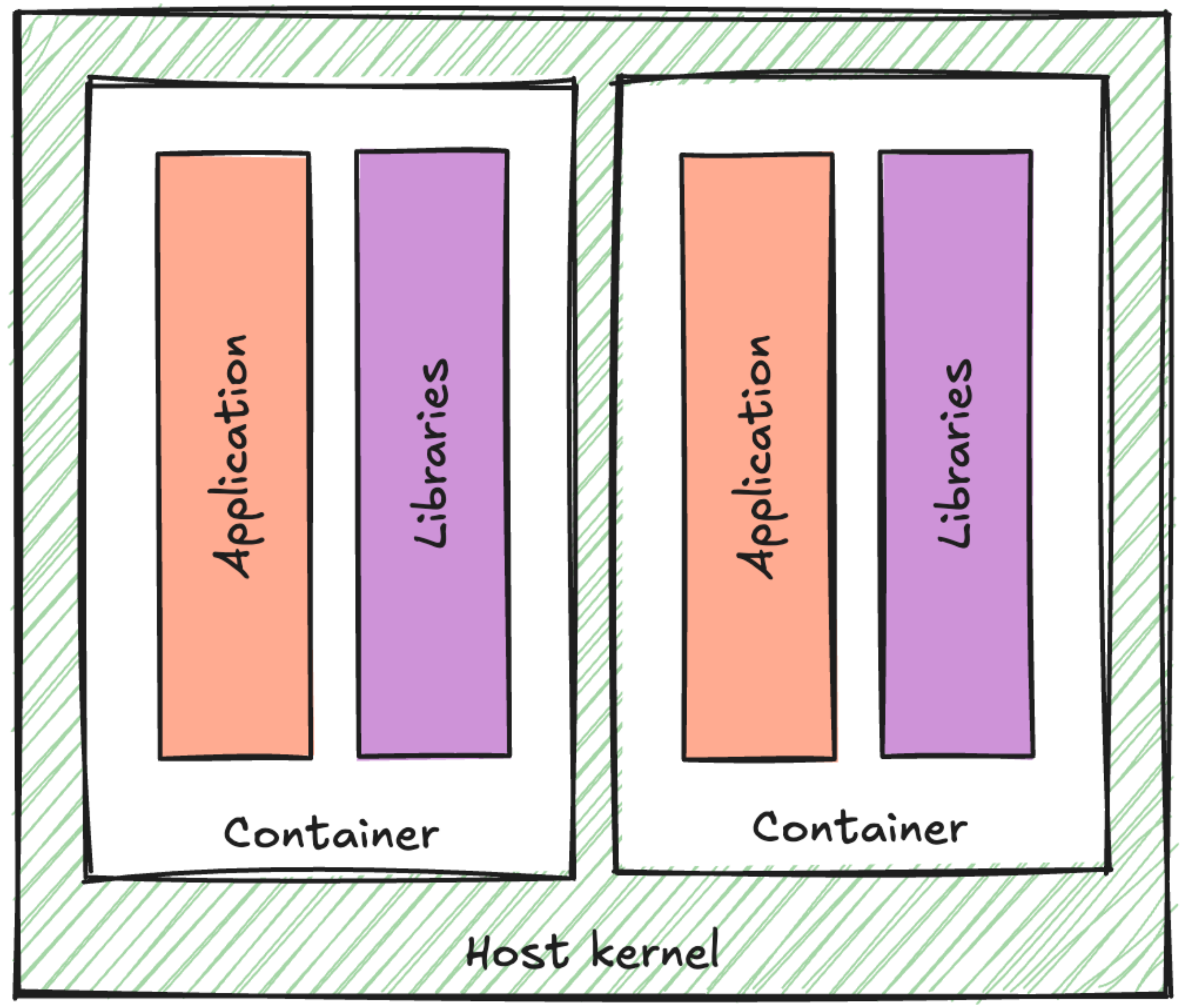
1 Process isolation using a shared kernel

2 Build once, run anywhere

3 No need for a hypervisor

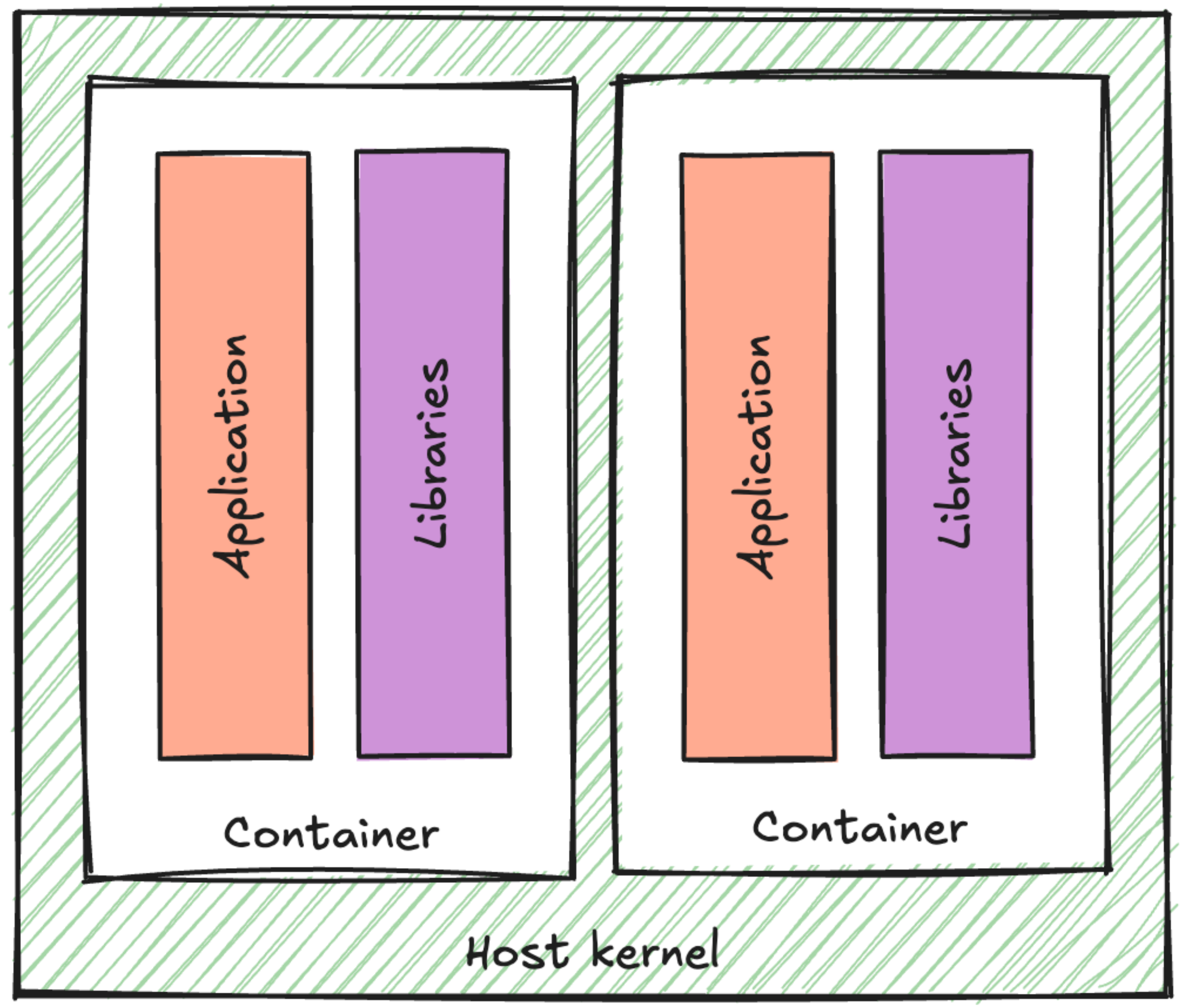
4 Fast startup

5 Scalable with orchestrator

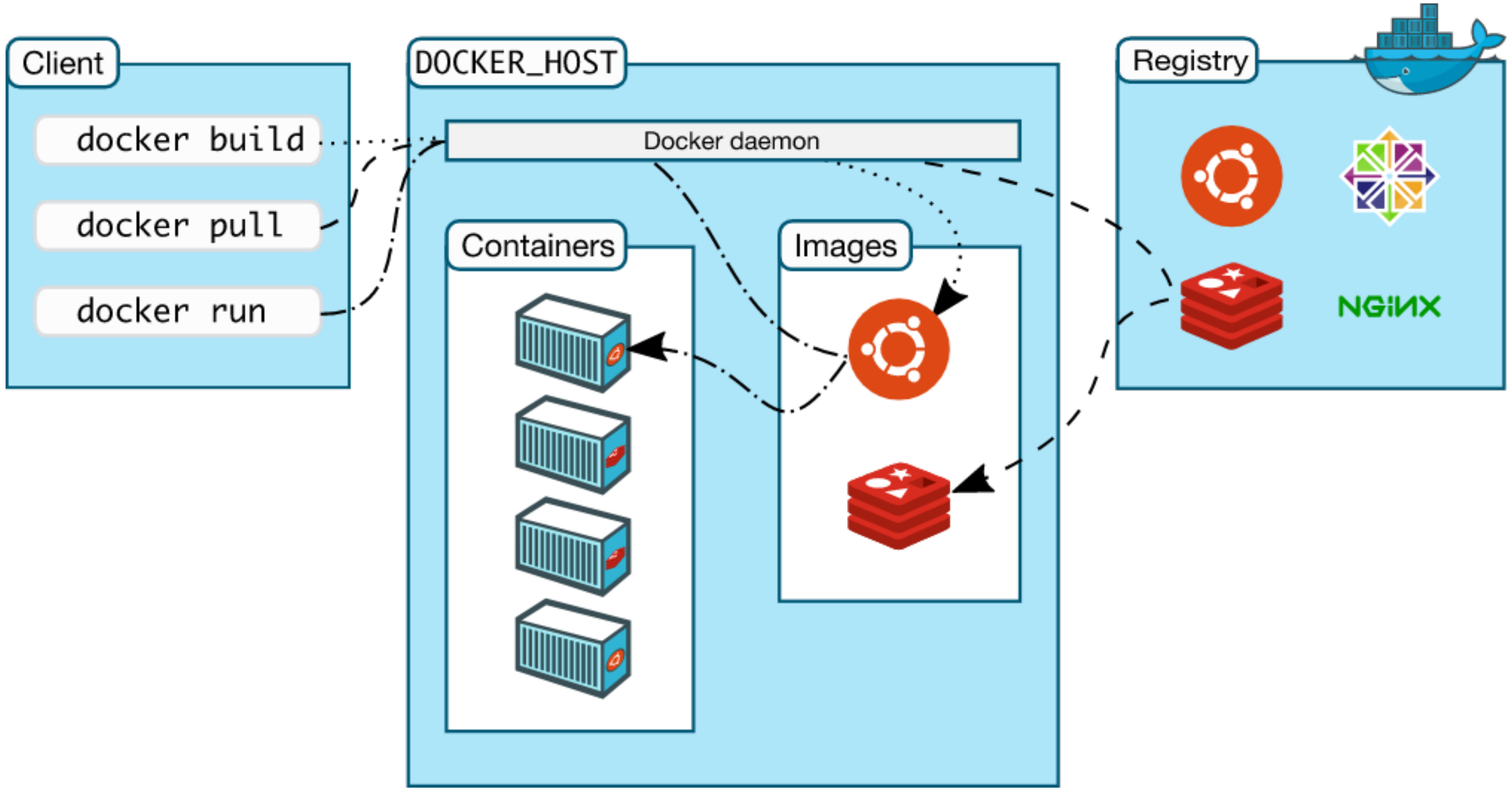


Container - The Bad

1. Shares the host kernel (weaker isolation)
2. Not ideal for untrusted workloads
3. Security misconfigurations are common
4. One container can slow down others
5. Not great for apps that expect a full OS



Docker

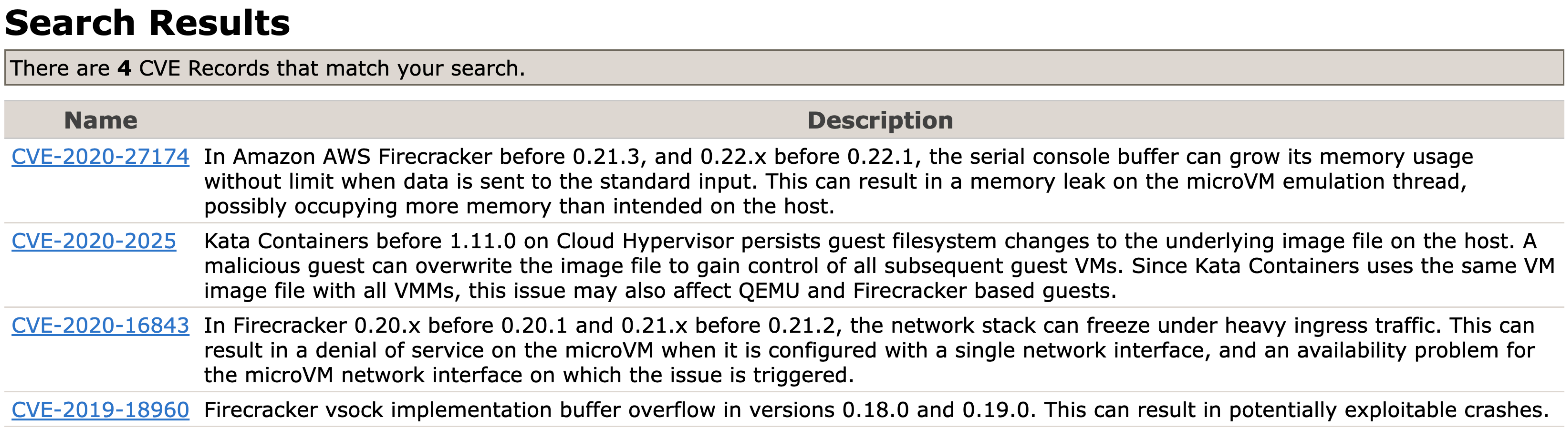


Source: https://docs.edera.dev/concepts/vm-containers/

Container is mainstream
microVM?
imgflip.com

microVM - The Good

1. Strong isolation with less overhead
2. Fast startup (milliseconds)
3. Minimal attack surface
4. Multi-tenancy



Search Results

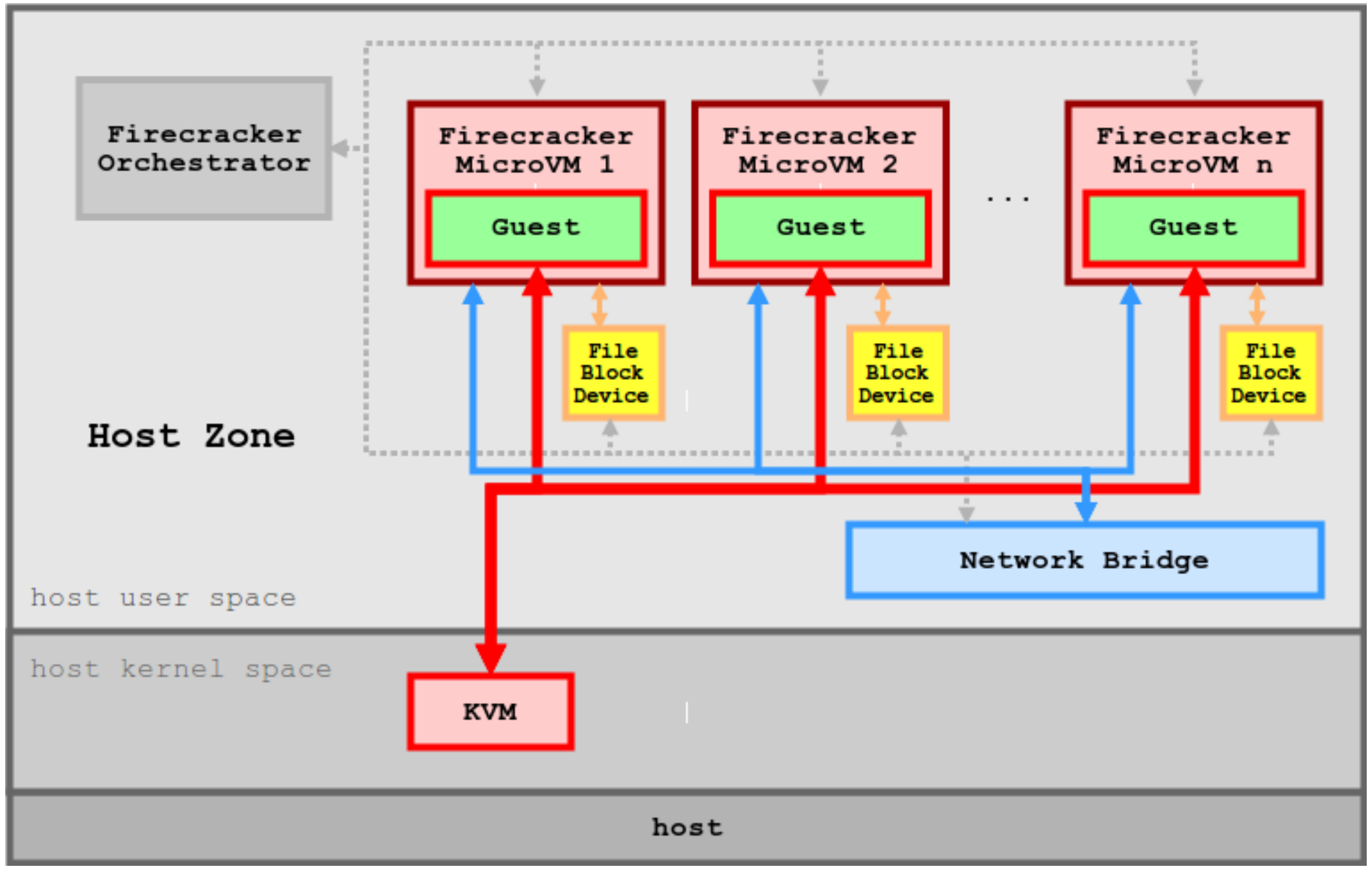
There are **4** CVE Records that match your search.

Name	Description
CVE-2020-27174	In Amazon AWS Firecracker before 0.21.3, and 0.22.x before 0.22.1, the serial console buffer can grow its memory usage without limit when data is sent to the standard input. This can result in a memory leak on the microVM emulation thread, possibly occupying more memory than intended on the host.
CVE-2020-2025	Kata Containers before 1.11.0 on Cloud Hypervisor persists guest filesystem changes to the underlying image file on the host. A malicious guest can overwrite the image file to gain control of all subsequent guest VMs. Since Kata Containers uses the same VM image file with all VMMs, this issue may also affect QEMU and Firecracker based guests.
CVE-2020-16843	In Firecracker 0.20.x before 0.20.1 and 0.21.x before 0.21.2, the network stack can freeze under heavy ingress traffic. This can result in a denial of service on the microVM when it is configured with a single network interface, and an availability problem for the microVM network interface on which the issue is triggered.
CVE-2019-18960	Firecracker vsock implementation buffer overflow in versions 0.18.0 and 0.19.0. This can result in potentially exploitable crashes.

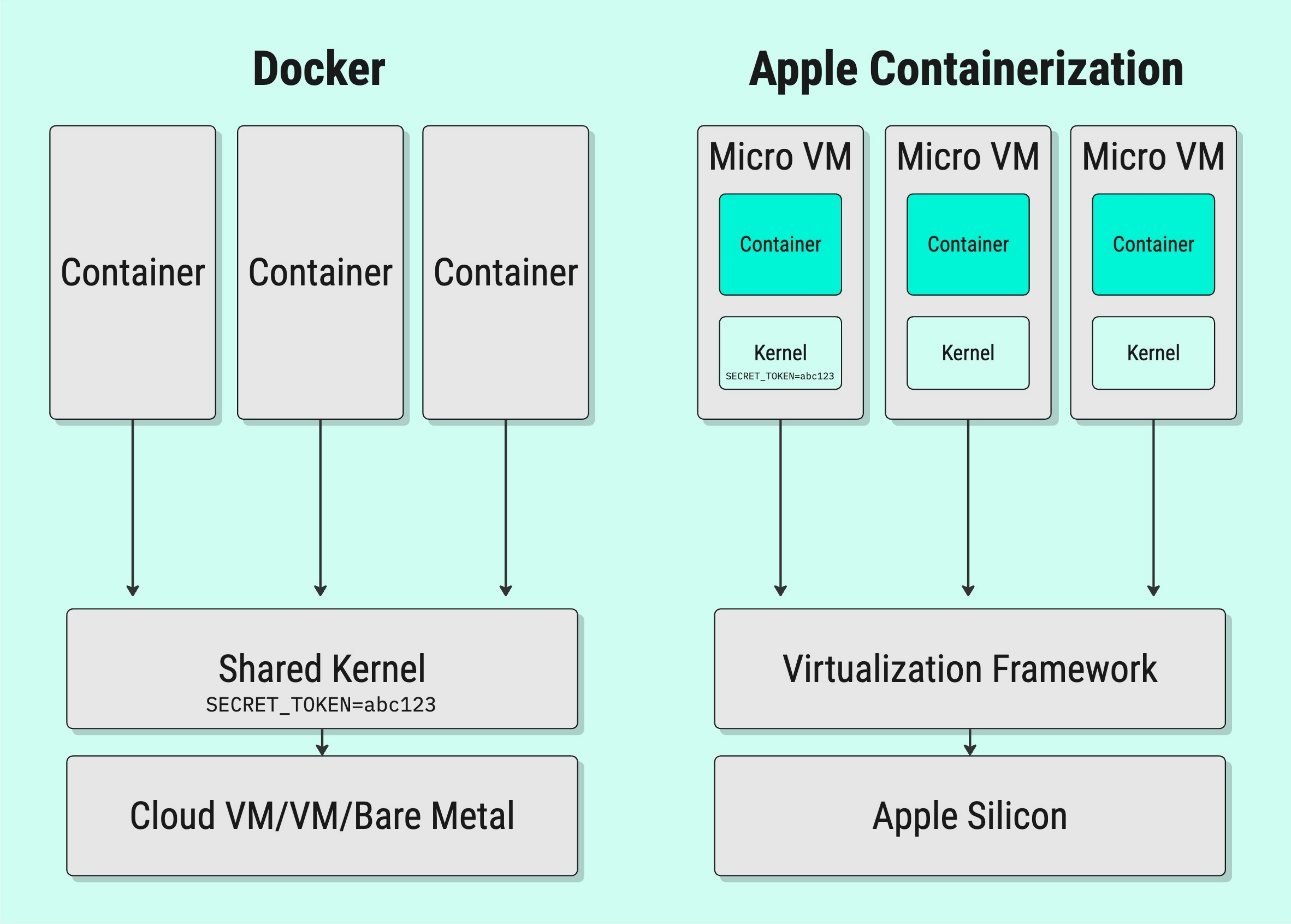
microVM - The Bad

1. Rarely used in general workloads
2. Less ecosystem support
3. Tooling is limited
4. Not easy to integrate
5. Not developer-friendly

Firecracker microVM



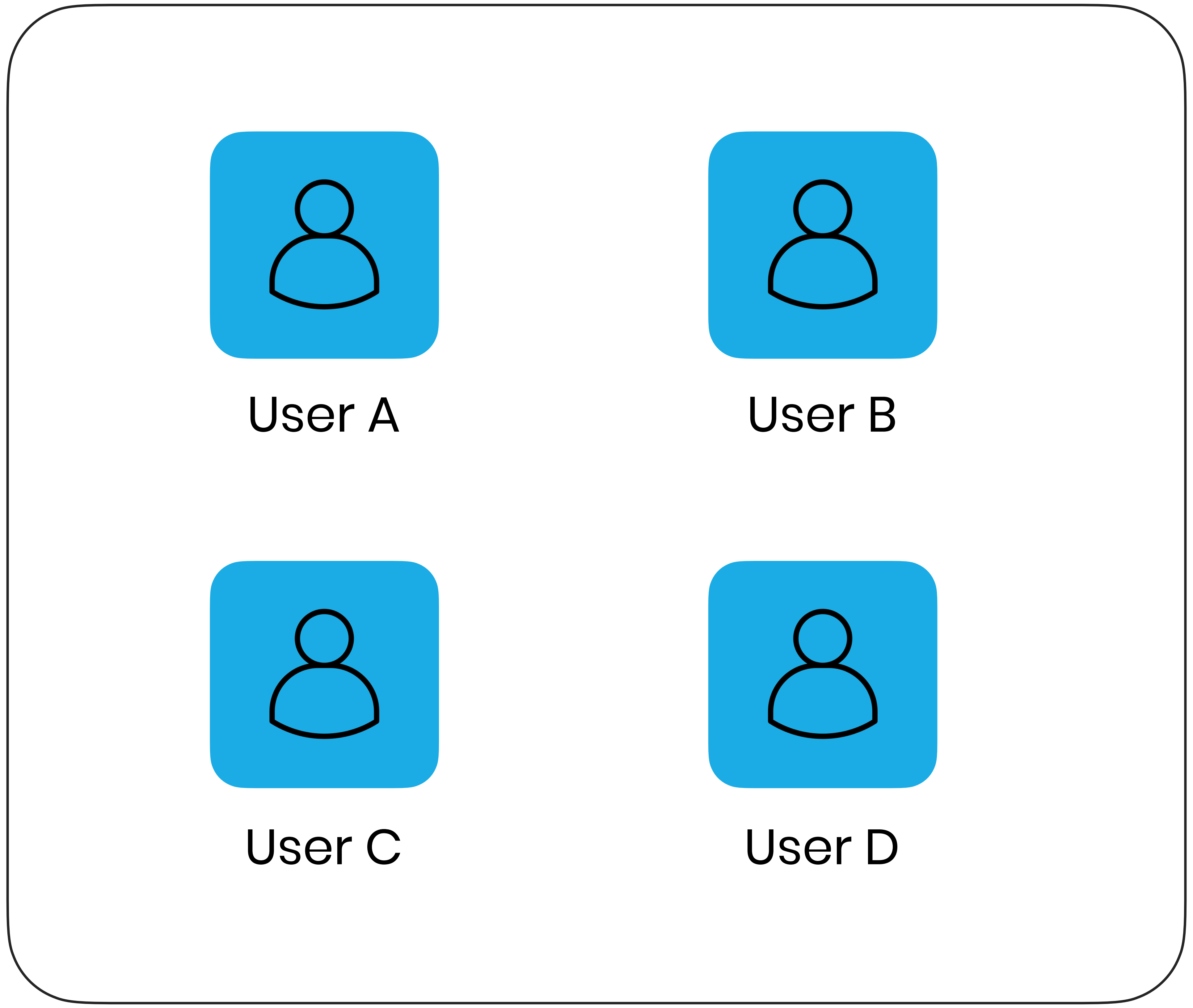
Apple Container



Source: https://docs.edera.dev/concepts/vm-containers/

Why isolation matters now

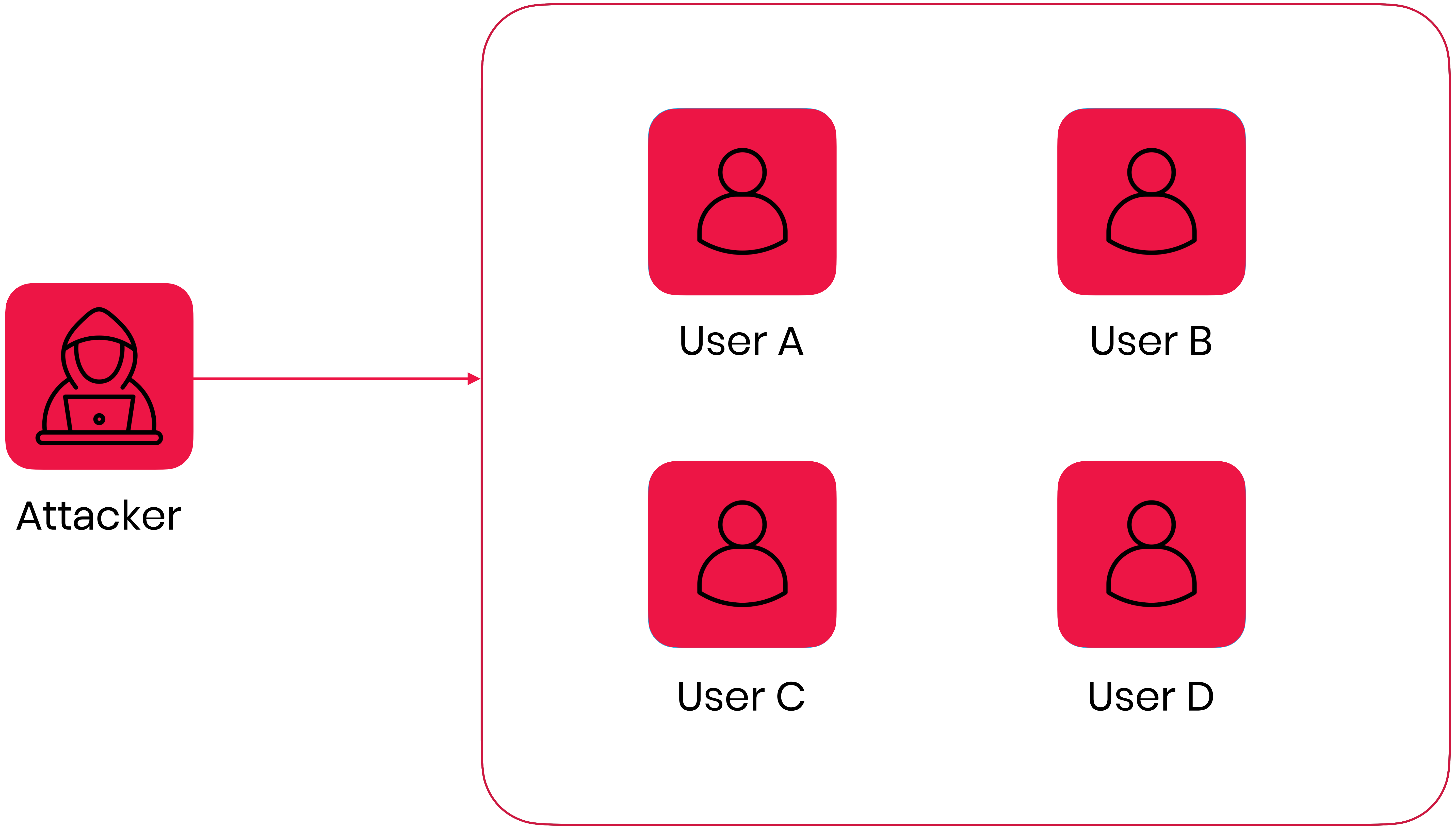
Multi-tenancy



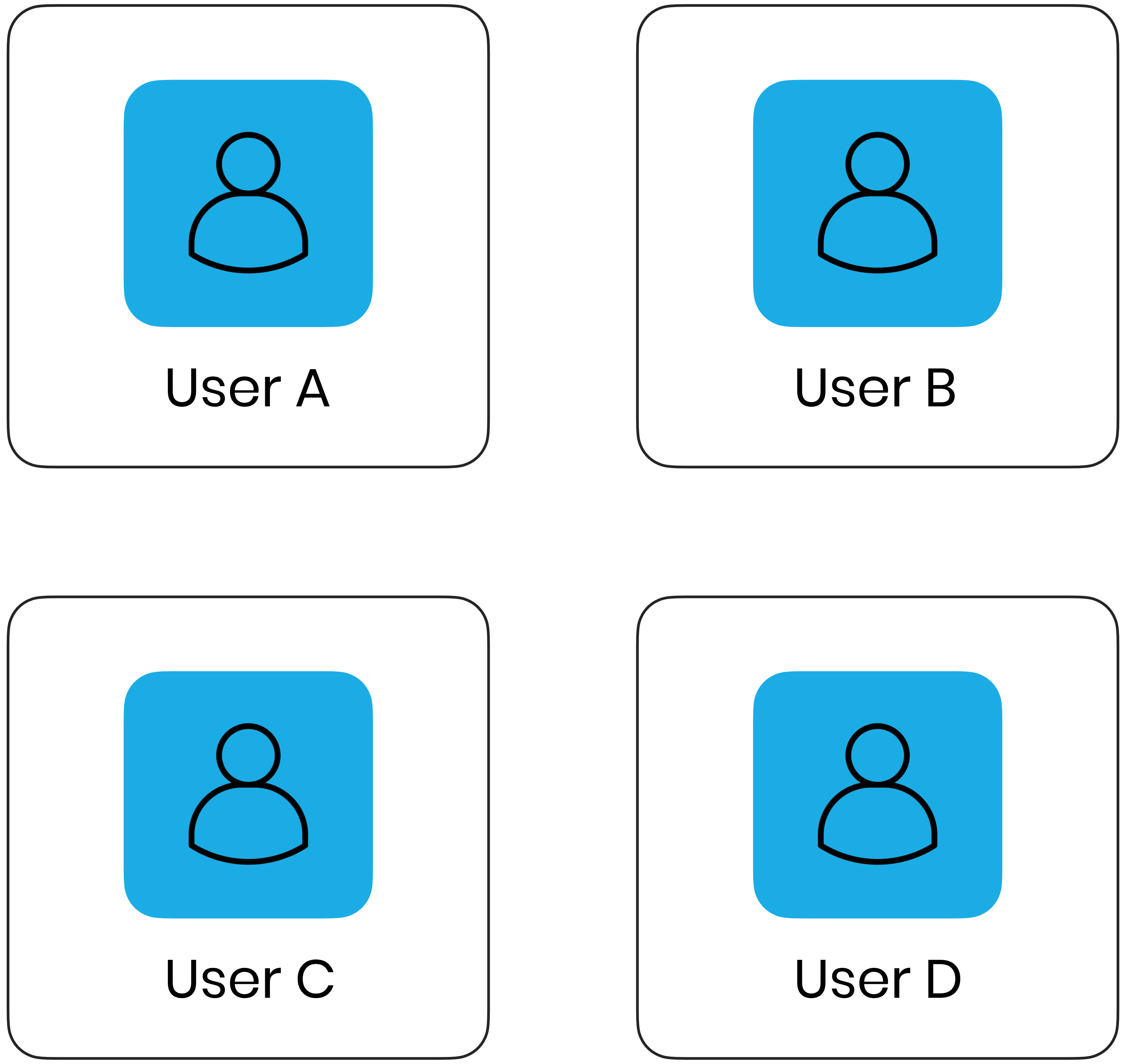
Multi-tenancy

1. Sharing resources (CPU, memory, storage)
2. Saves cost by reducing the number of systems needed
3. Add new users or customers without setting up new servers
4. Easy maintenance, update once for all tenants

Security in multi-tenancy



Single-tenancy



User A

User B

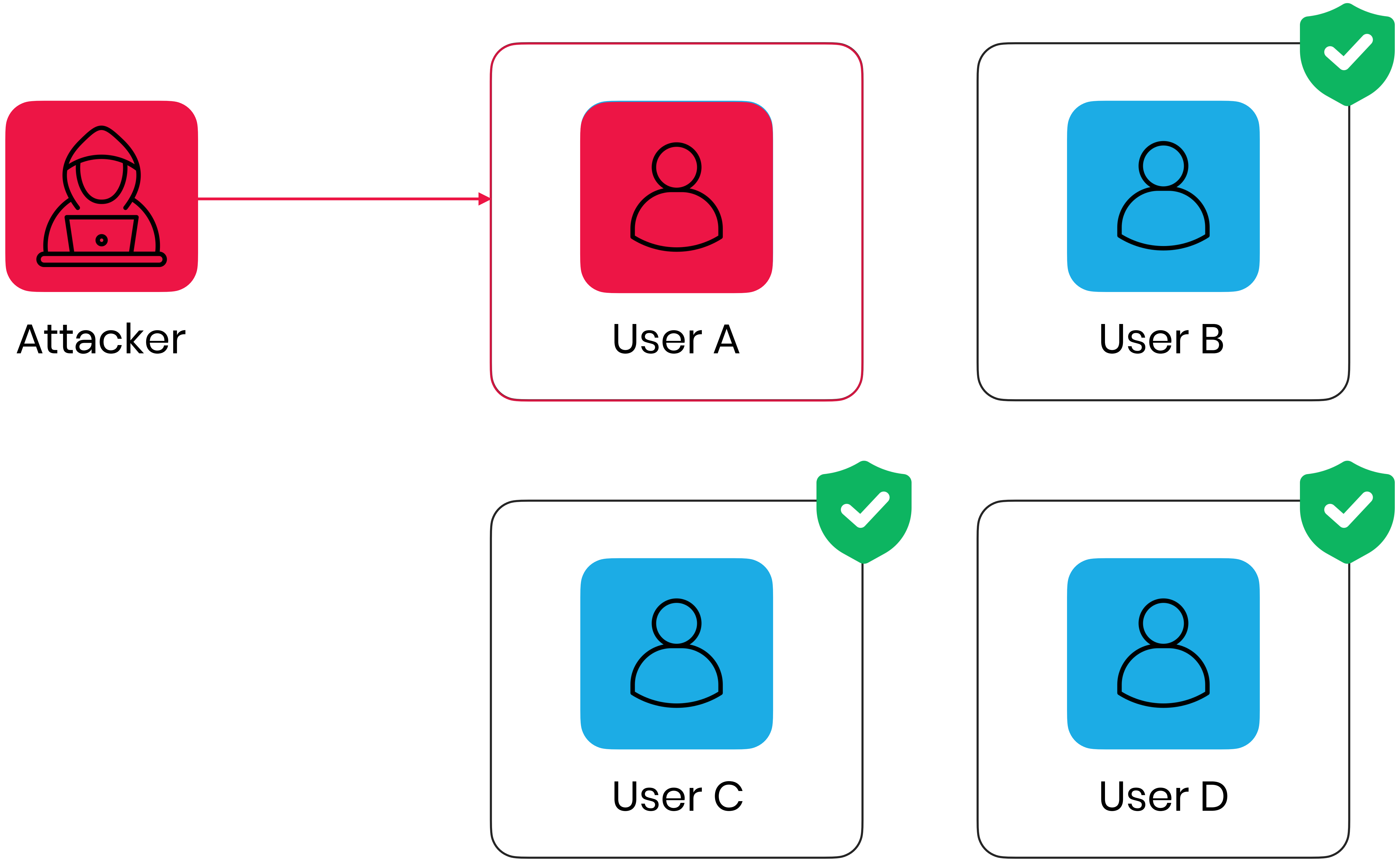
User C

User D

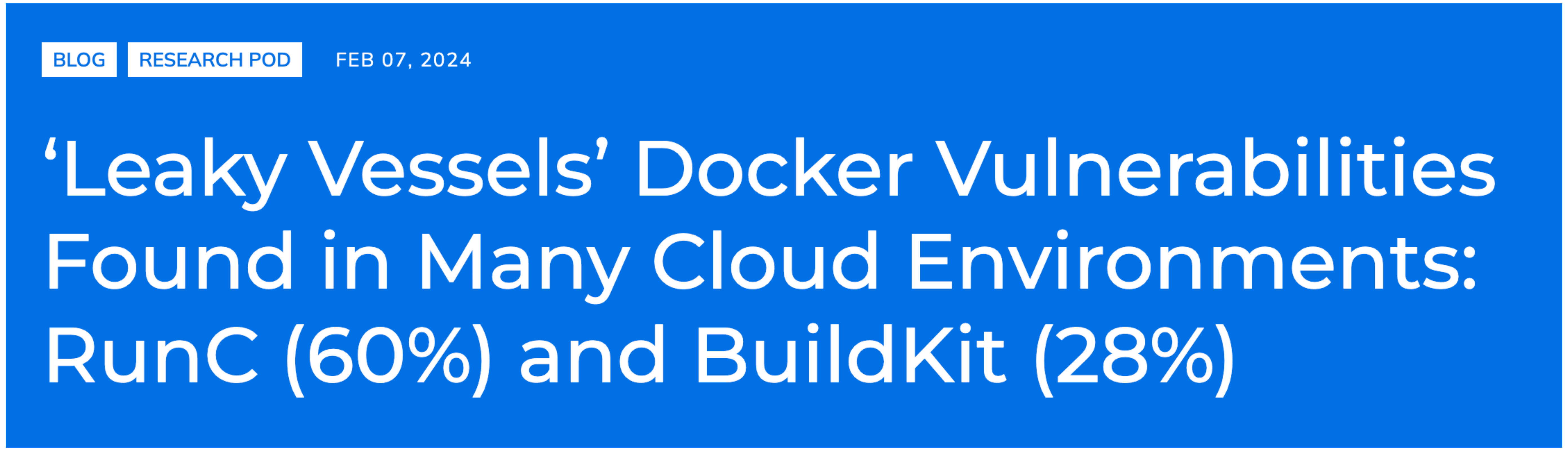
Single-tenancy

1. Own dedicated resources and full control
2. Higher cost because each user needs dedicated resources
3. Setting up new servers to add new users or customers
4. Update each system separately

Security in single-tenancy

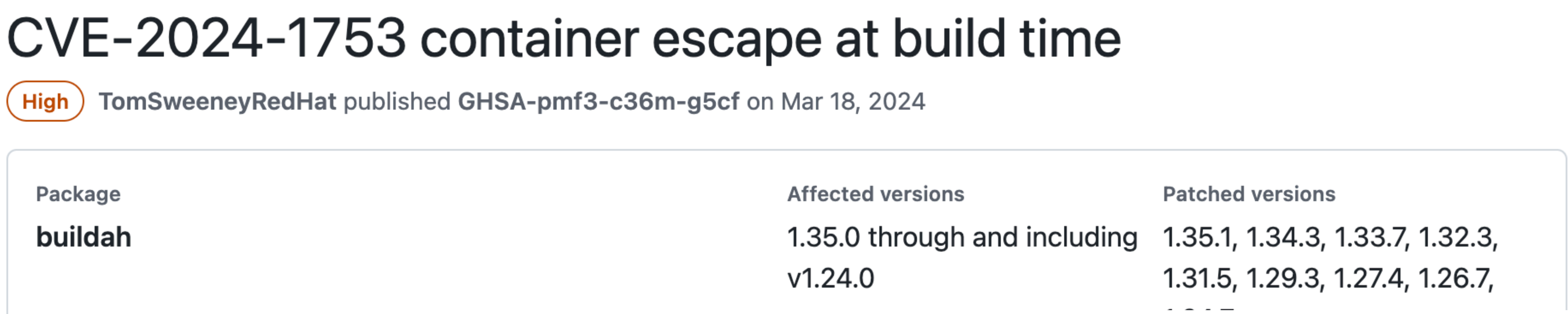


Attacker focus on breaking isolation boundaries



BLOG RESEARCH POD FEB 07, 2024

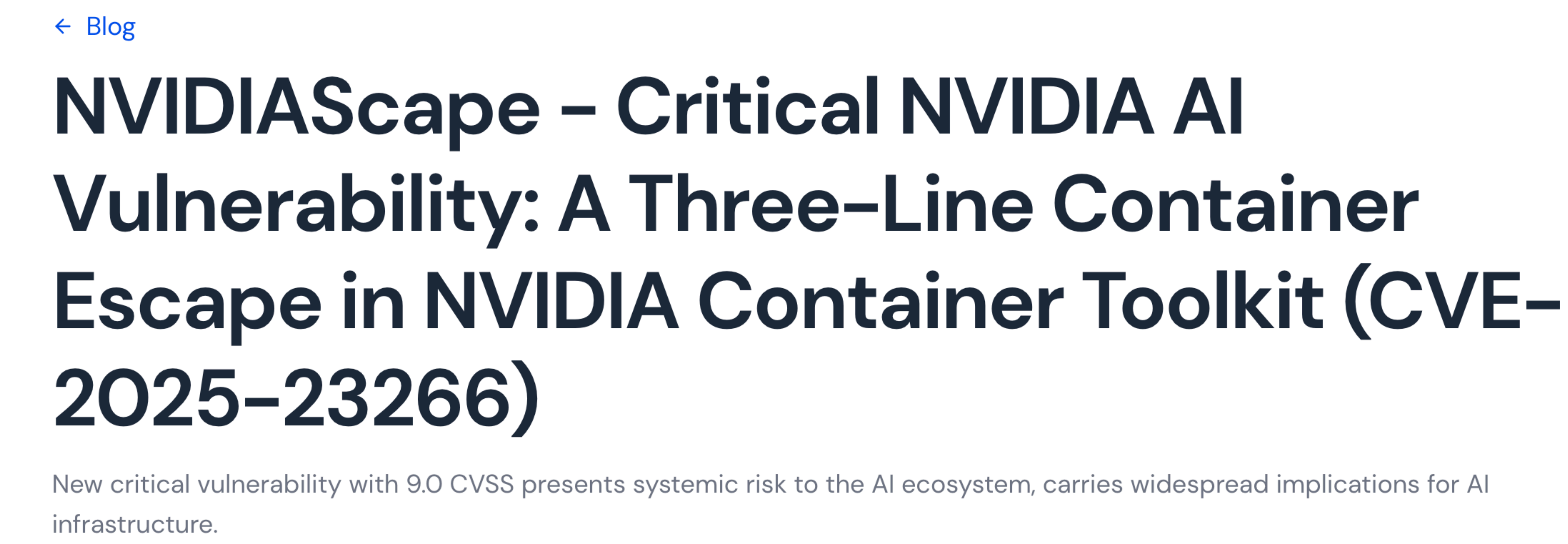
‘Leaky Vessels’ Docker Vulnerabilities Found in Many Cloud Environments: RunC (60%) and BuildKit (28%)



CVE-2024-1753 container escape at build time

High **TomSweeneyRedHat** published **GHSA-pmf3-c36m-g5cf** on Mar 18, 2024

Package	Affected versions	Patched versions
buildah	1.35.0 through and including v1.24.0	1.35.1, 1.34.3, 1.33.7, 1.32.3, 1.31.5, 1.29.3, 1.27.4, 1.26.7,



← Blog

NVIDIAScape – Critical NVIDIA AI Vulnerability: A Three-Line Container Escape in NVIDIA Container Toolkit (CVE-2025-23266)

New critical vulnerability with 9.0 CVSS presents systemic risk to the AI ecosystem, carries widespread implications for AI infrastructure.

Why are we still using containers today?

What problems does it solve?

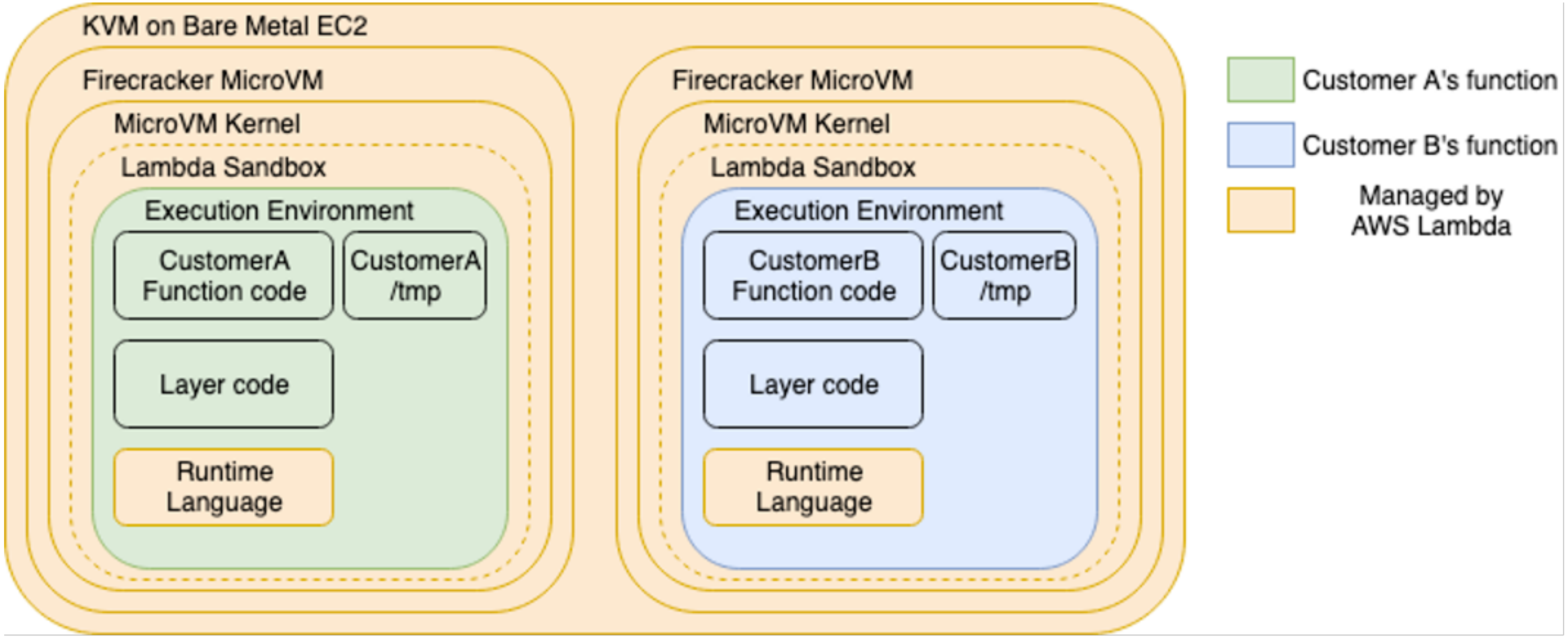
1. Starts quickly, often within seconds
2. No more “**it works on my machine**”
3. Split apps into smaller pieces, easier to manage and update
4. Provides isolation for security and stateless

What microVM bring to the table

Meet “Firecracker”

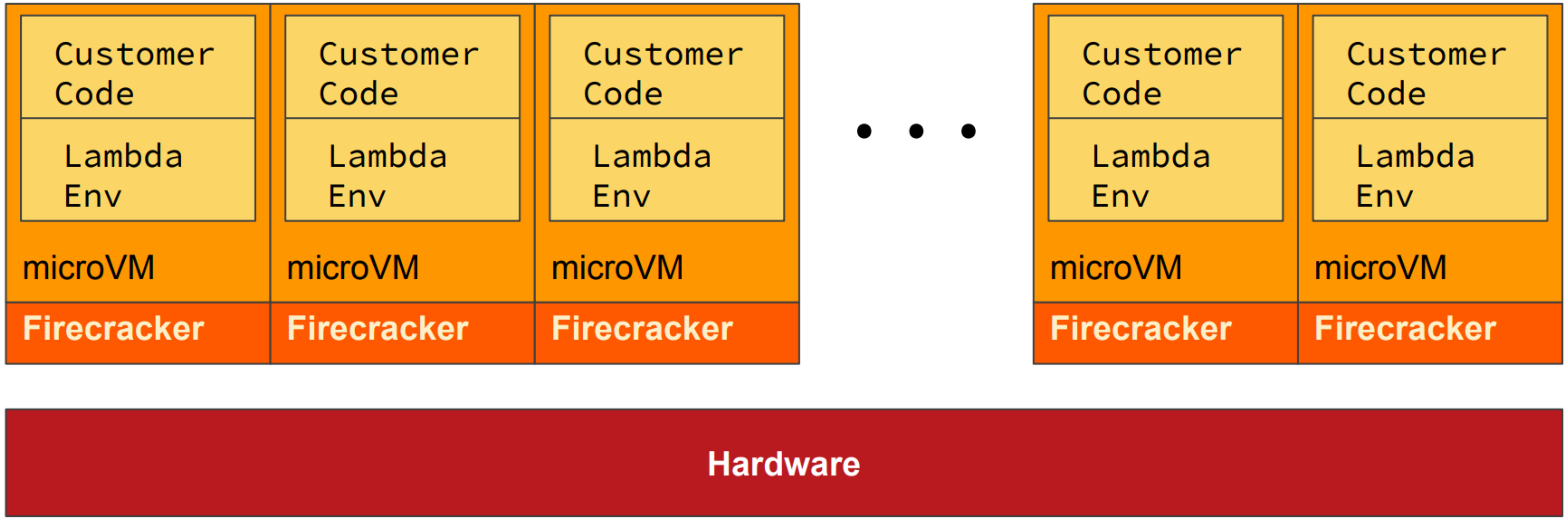
1. Open-source virtualization technology developed by Amazon
2. Powers **AWS Lambda** and **AWS Fargate**
3. Built on Linux KVM and written in **Rust**
4. Combines VM-level isolation with container-like speed

AWS Lambda

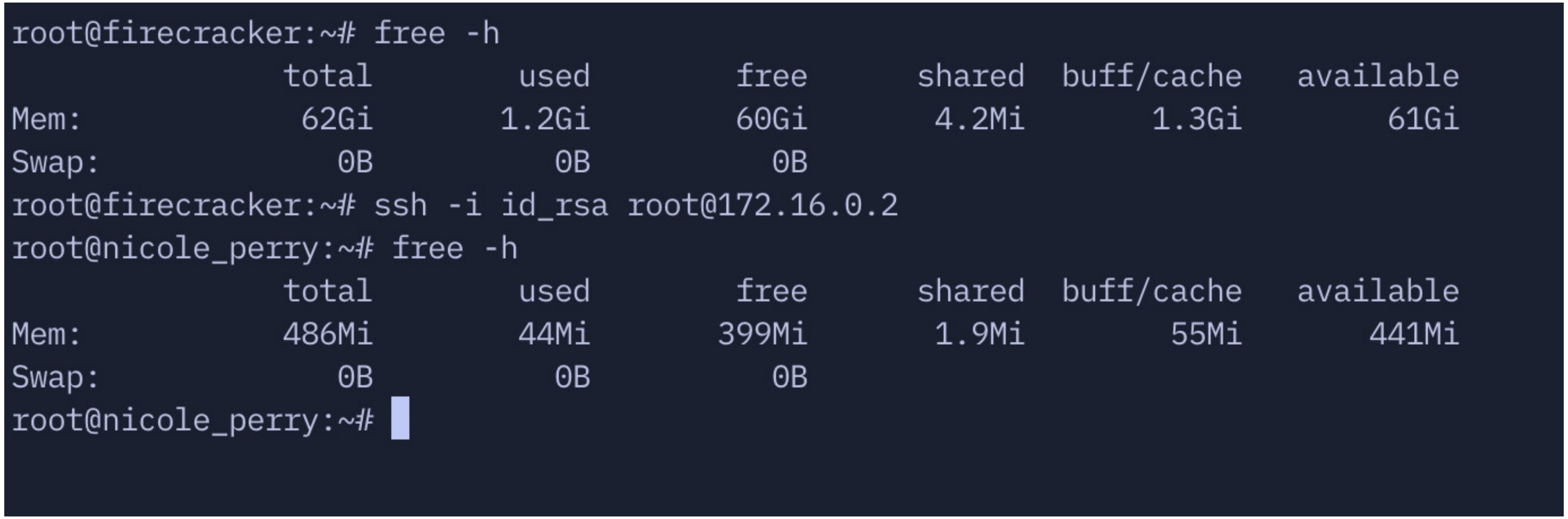


Source: https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/lambda-executions.html

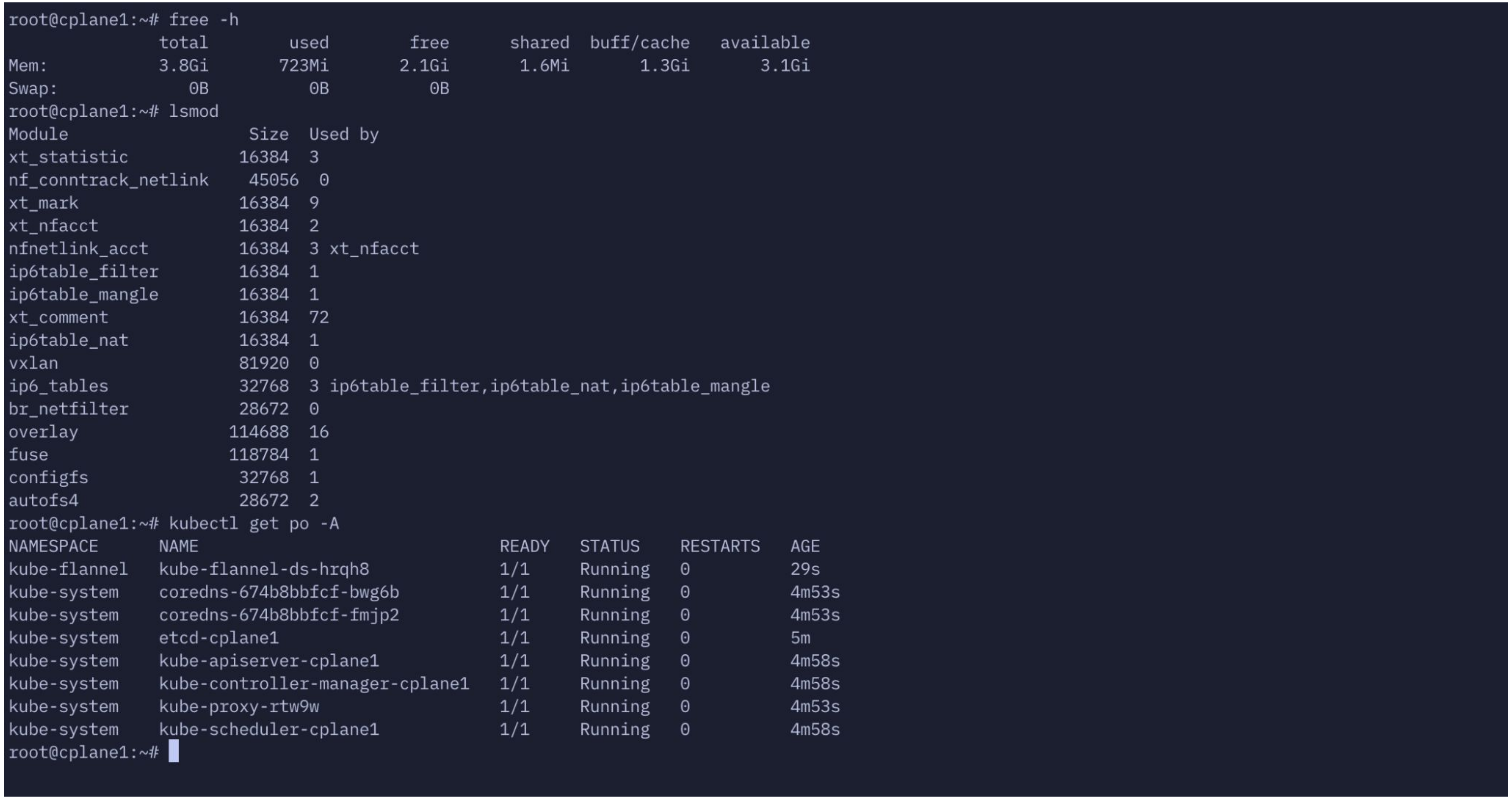
AWS Lambda



Customer Code
Lambda Env
microVM
Firecracker
Customer Code
Lambda Env
microVM
Firecracker
Customer Code
Lambda Env
microVM
Firecracker
Customer Code
Lambda Env
microVM
Firecracker
Customer Code
Lambda Env
microVM
Firecracker
Hardware

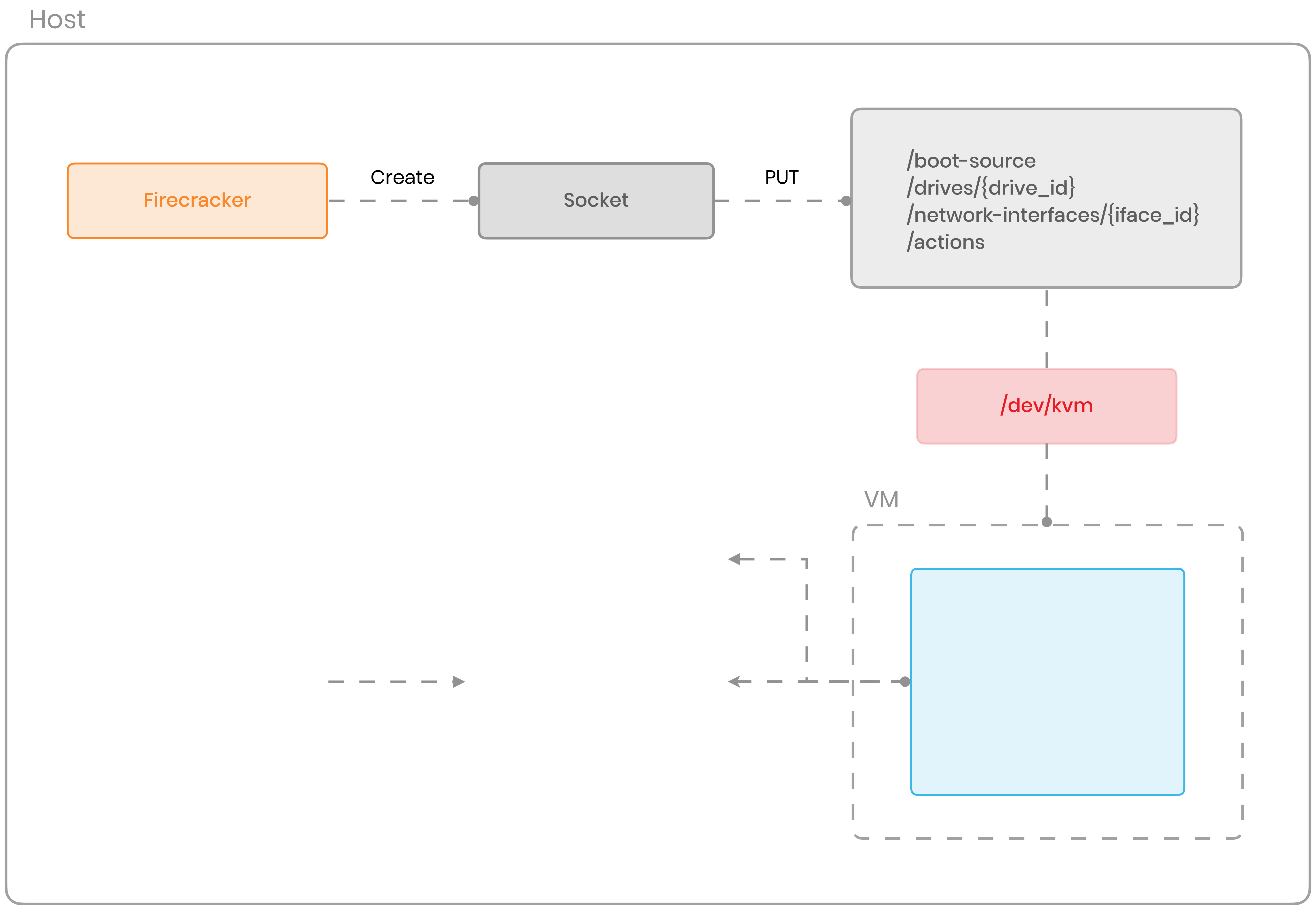


```
root@firecracker:~# free -h
               total        used        free      shared  buff/cache   available
Mem:            62Gi       1.2Gi        60Gi       4.2Mi       1.3Gi        61Gi
Swap:             0B          0B          0B
root@firecracker:~# ssh -i id_rsa root@172.16.0.2
root@nicole_perry:~# free -h
               total        used        free      shared  buff/cache   available
Mem:           486Mi        44Mi       399Mi       1.9Mi        55Mi       441Mi
Swap:             0B          0B          0B
root@nicole_perry:~# 
```



```
root@cplane1:~# free -h
               total        used        free      shared  buff/cache   available
Mem:           3.8Gi       723Mi       2.1Gi       1.6Mi       1.3Gi       3.1Gi
Swap:             0B          0B          0B
root@cplane1:~# lsmod
Module                  Size  Used by
xt_statistic           16384  3
nf_conntrack_netlink    45056  0
xt_mark                16384  9
xt_nfacct              16384  2
nfnetlink_acct         16384  3 xt_nfacct
ip6table_filter        16384  1
ip6table_mangle        16384  1
xt_comment             16384  72
ip6table_nat           16384  1
vxlan                  81920  0
ip6_tables             32768  3 ip6table_filter,ip6table_nat,ip6table_mangle
br_netfilter           28672  0
overlay               114688  16
fuse                  118784  1
configfs               32768  1
autofs4                28672  2
root@cplane1:~# kubectl get po -A
NAMESPACE      NAME                                  READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-hrqh8                 1/1     Running   0          29s
kube-system    coredns-674b8bbfcf-bwg6b              1/1     Running   0          4m53s
kube-system    coredns-674b8bbfcf-fmjp2              1/1     Running   0          4m53s
kube-system    etcd-cplane1                          1/1     Running   0          5m
kube-system    kube-apiserver-cplane1                1/1     Running   0          4m58s
kube-system    kube-controller-manager-cplane1       1/1     Running   0          4m58s
kube-system    kube-proxy-rtw9w                      1/1     Running   0          4m53s
kube-system    kube-scheduler-cplane1                1/1     Running   0          4m58s
root@cplane1:~# 
```

Firecracker in Action

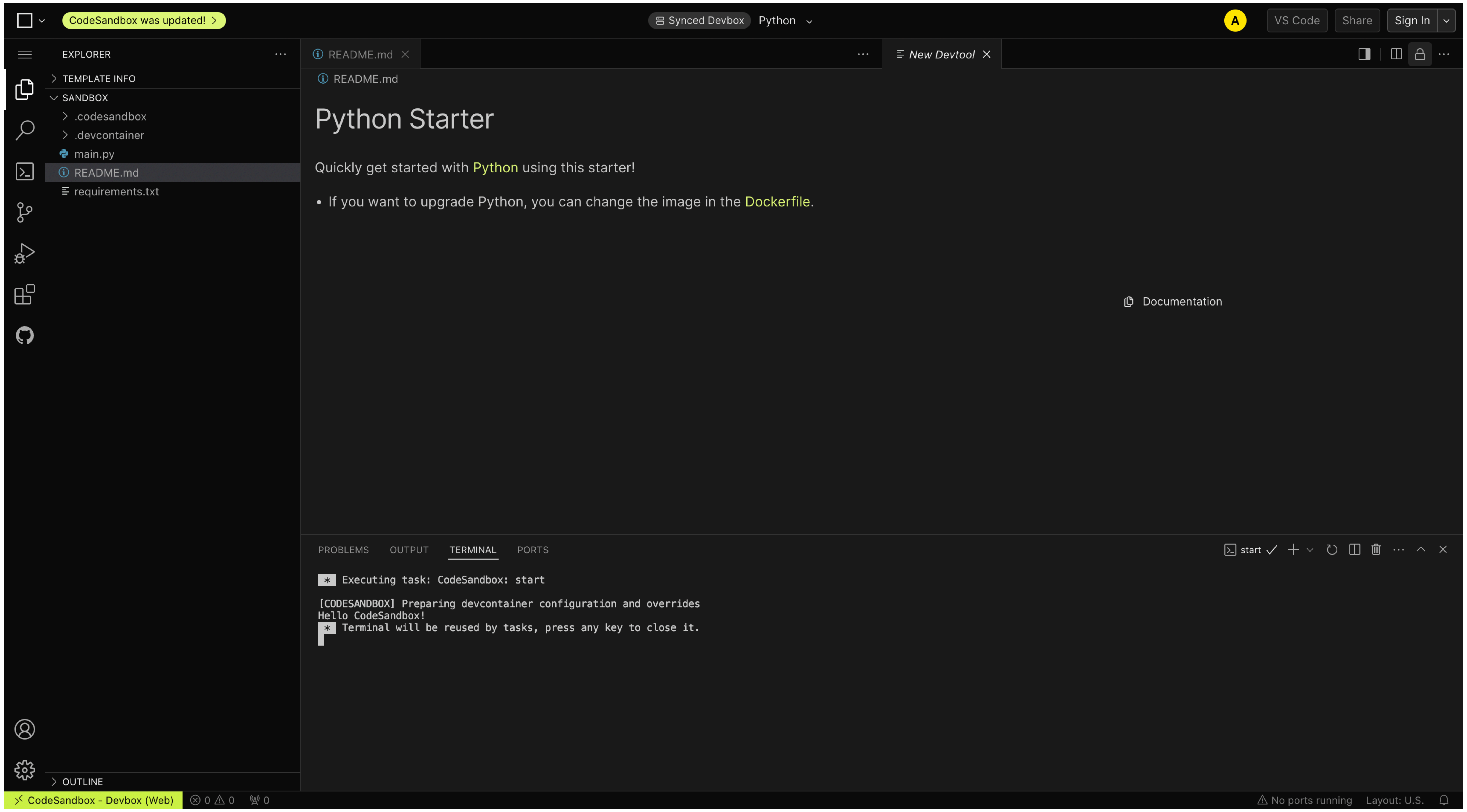


```
(venv) root@firecracker:~#
```

1. Fast like container, isolated like VM
2. Multi-tenancy with single-tenancy-level isolation
3. Minimal attack surface for better security

Who are using Firecracker?

CodeSandbox - Instant Cloud Development Environments



Python Starter

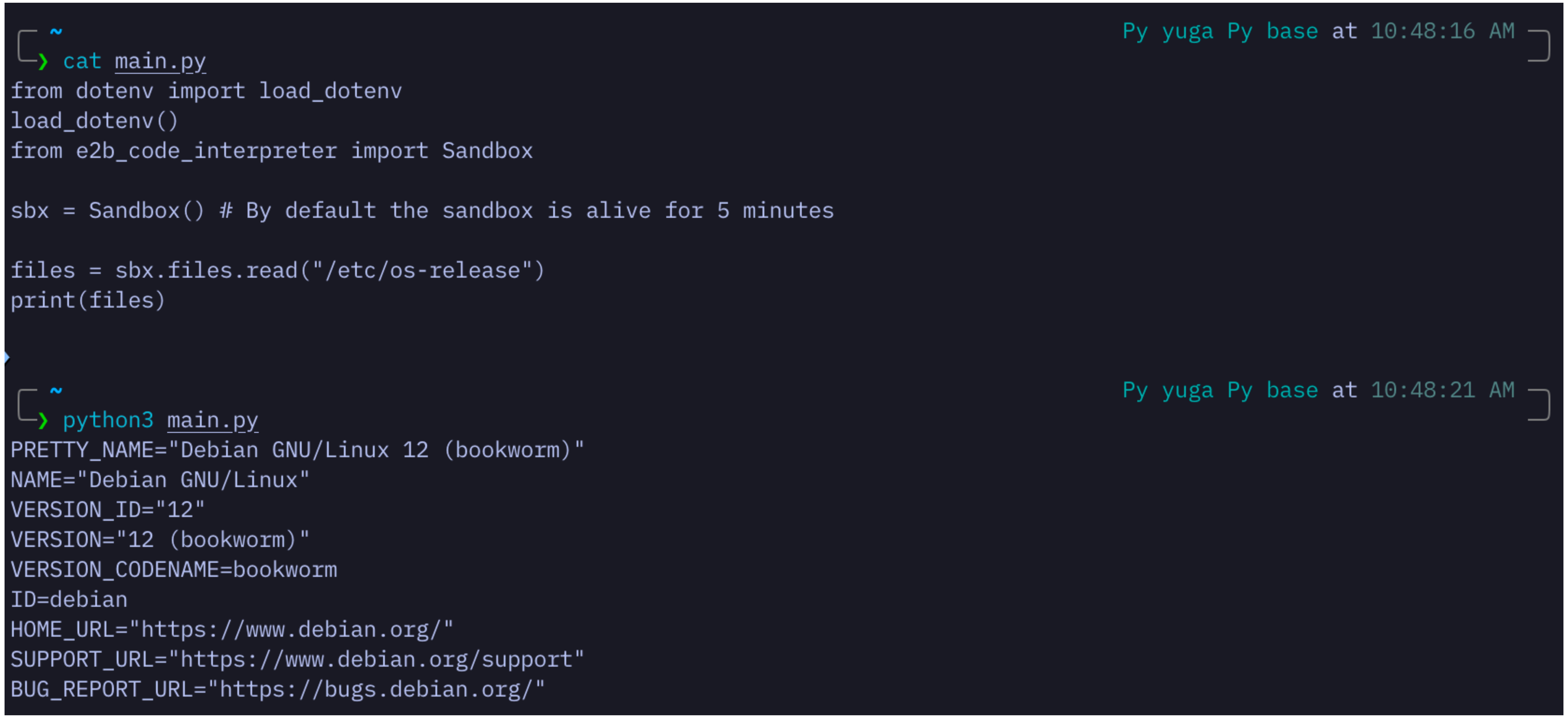
Quickly get started with Python using this starter!

- If you want to upgrade Python, you can change the image in the Dockerfile.

```
 *  Executing task: CodeSandbox: start

[CODESANDBOX] Preparing devcontainer configuration and overrides
Hello CodeSandbox!
 *  Terminal will be reused by tasks, press any key to close it.
```

E2B - Code Interpreting for AI apps



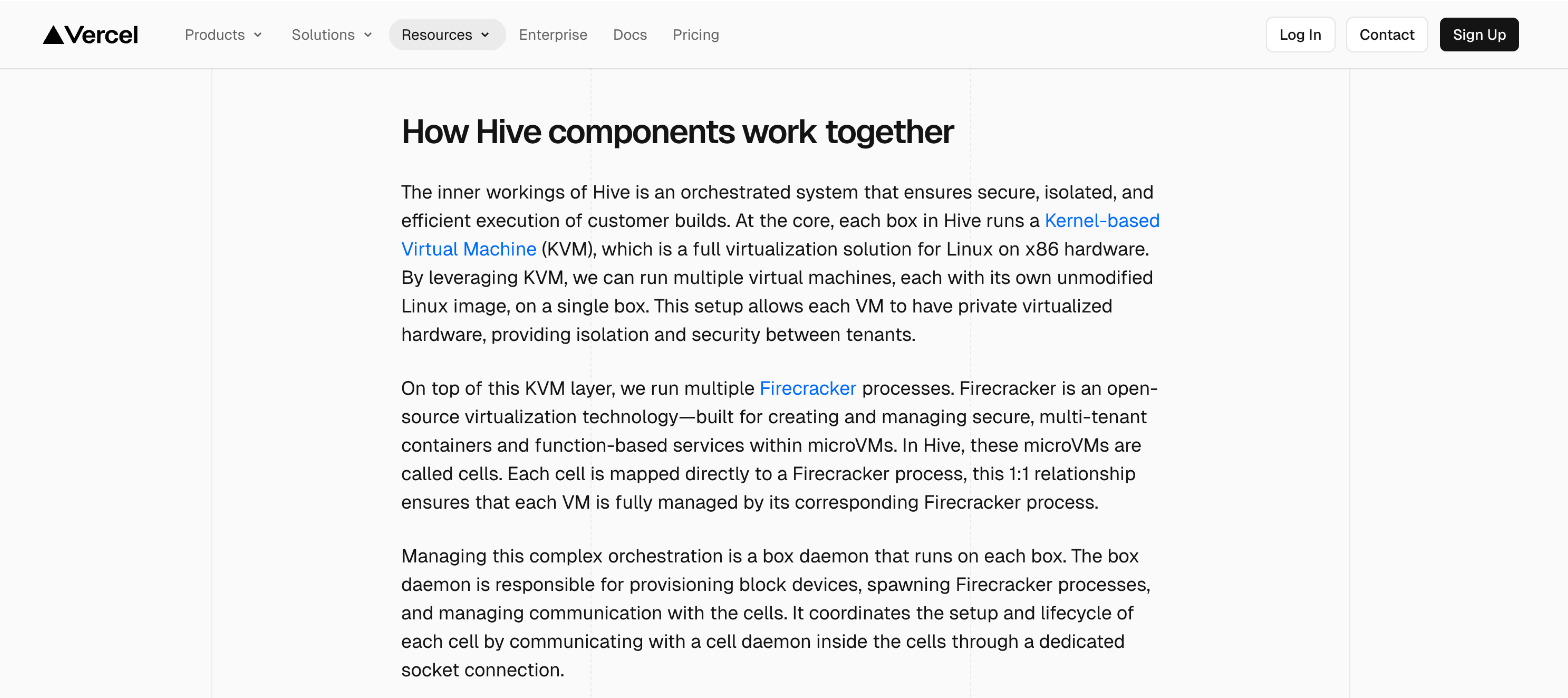
```
╭─ ~                                                                    Py yuga Py base at 10:48:16 AM ─╮
╰─❯ cat main.py
from dotenv import load_dotenv
load_dotenv()
from e2b_code_interpreter import Sandbox

sbx = Sandbox() # By default the sandbox is alive for 5 minutes

files = sbx.files.read("/etc/os-release")
print(files)


╭─ ~                                                                    Py yuga Py base at 10:48:21 AM ─╮
╰─❯ python3 main.py
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Vercel - Hive



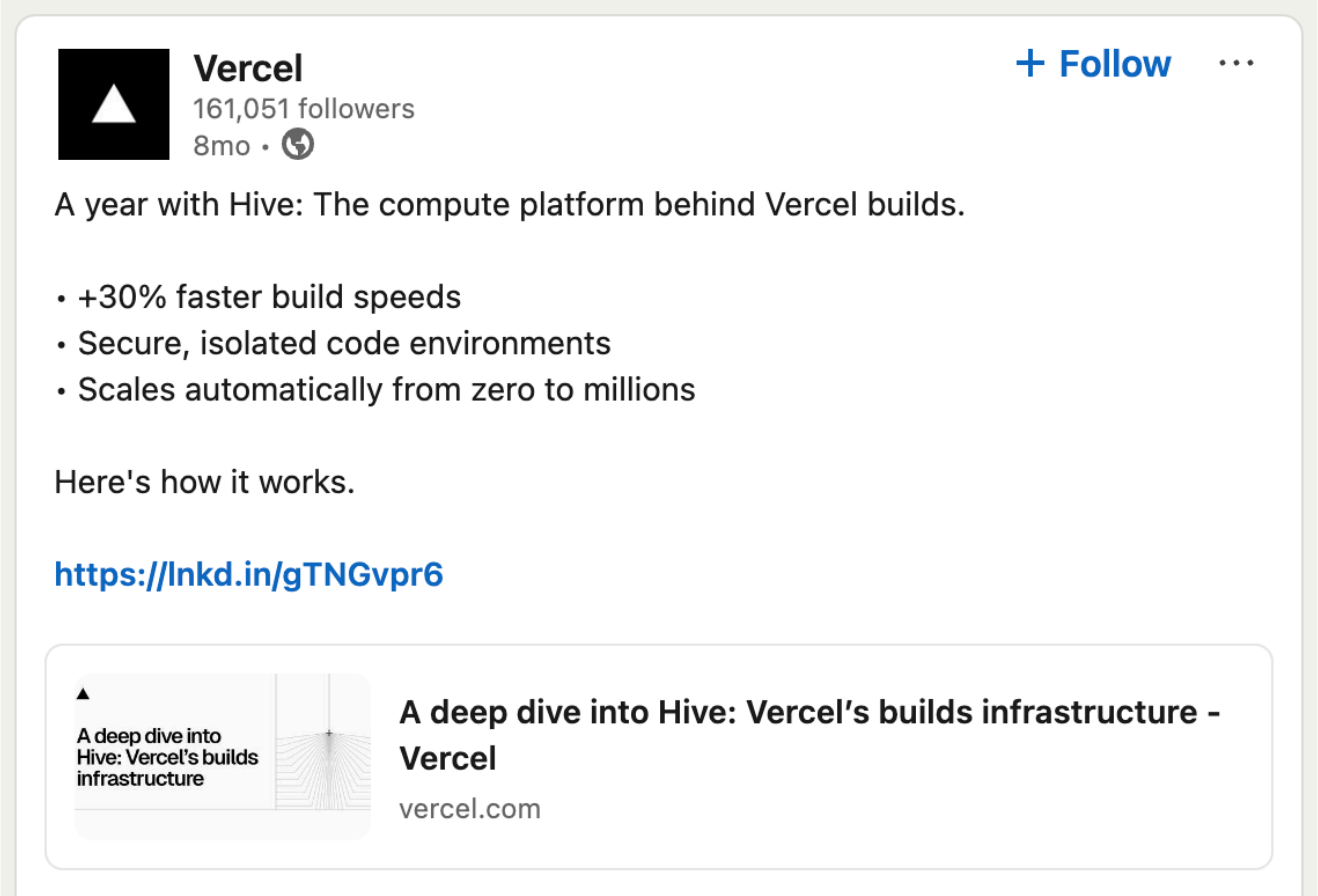
How Hive components work together

The inner workings of Hive is an orchestrated system that ensures secure, isolated, and efficient execution of customer builds. At the core, each box in Hive runs a Kernel-based Virtual Machine (KVM), which is a full virtualization solution for Linux on x86 hardware. By leveraging KVM, we can run multiple virtual machines, each with its own unmodified Linux image, on a single box. This setup allows each VM to have private virtualized hardware, providing isolation and security between tenants.

On top of this KVM layer, we run multiple Firecracker processes. Firecracker is an open-source virtualization technology—built for creating and managing secure, multi-tenant containers and function-based services within microVMs. In Hive, these microVMs are called cells. Each cell is mapped directly to a Firecracker process, this 1:1 relationship ensures that each VM is fully managed by its corresponding Firecracker process.

Managing this complex orchestration is a box daemon that runs on each box. The box daemon is responsible for provisioning block devices, spawning Firecracker processes, and managing communication with the cells. It coordinates the setup and lifecycle of each cell by communicating with a cell daemon inside the cells through a dedicated socket connection.

Source:https://vercel.com/blog/a-deep-dive-into-hive-vercels-builds-infrastructure



Vercel
161,051 followers
8mo •

A year with Hive: The compute platform behind Vercel builds.

- +30% faster build speeds
- Secure, isolated code environments
- Scales automatically from zero to millions

Here's how it works.

https://lnkd.in/gTNGvpr6

A deep dive into Hive: Vercel's builds infrastructure - Vercel
vercel.com

Firecracker isn’t the only microVM out there

The challenge(s)

1. Not developer friendly
2. Integration is more complex
3. Limited ecosystem and tooling unlike other technology
4. Less adoption and community support
5. Runs only on KVM (though PVM is an alternative option)
6. And many more…

Recap

1. VM offers strong isolation, containers are fast and both have trade-offs
2. microVM bridge the gap, combining VM isolation with container speed
3. Ideal for serverless, CI/CD pipelines, and short-lived workloads
4. Could microVM be the future of how we run workloads?

QnA