Fulfilling The Developers Dream: You Code, We Deploy it on Kubernetes

Tal Neeman
Developer Advocate
IBM

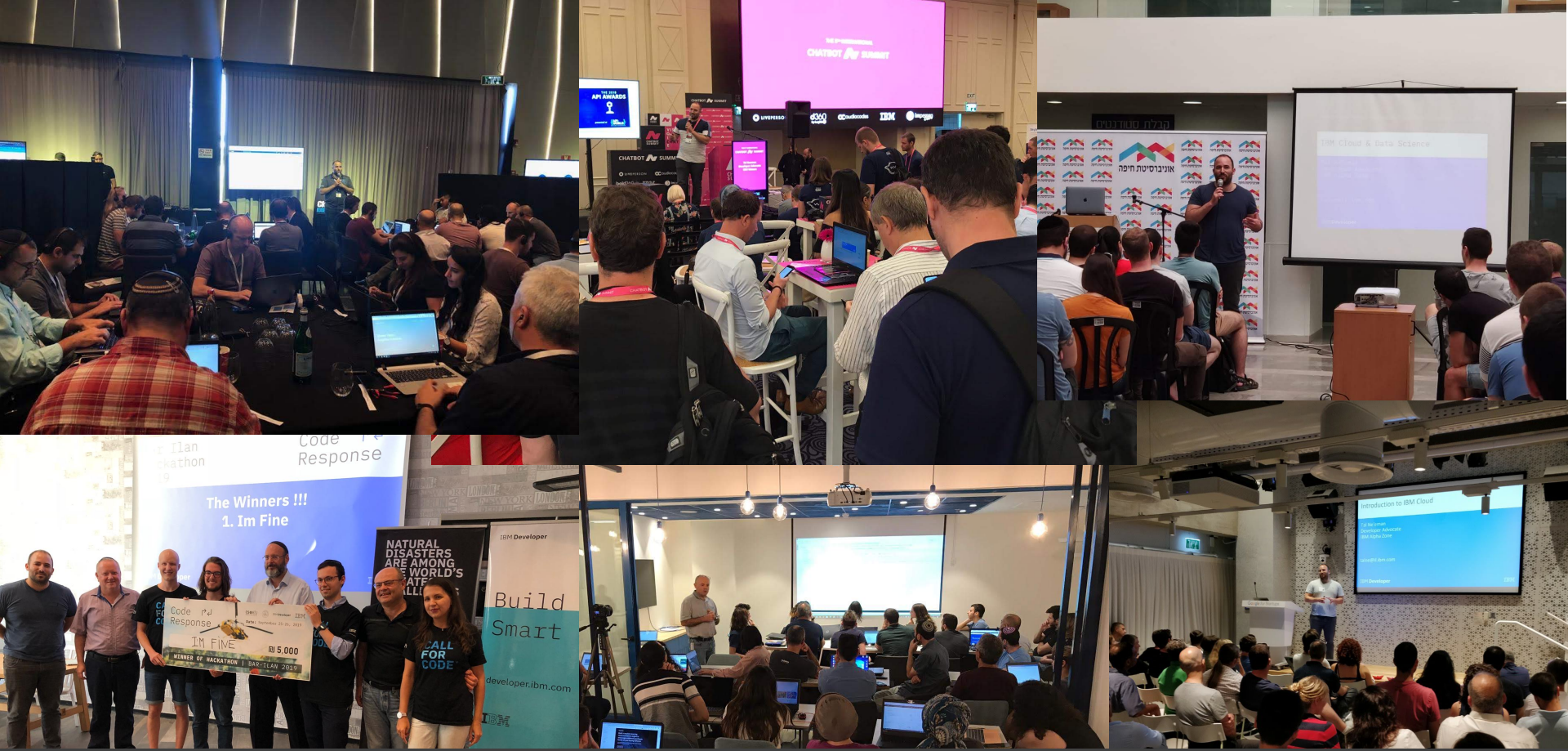
IBM **Developer**

IBM

Hi, I’m Tal

I’m a Developer Advocate at IBM.

I lecture at meetups, participate as mentor at hackathons, having fun in awesome webinars and I also write articles / tutorials about open source or new technologies on IBM Cloud.



developer.ibm.com

Call for Code®

At IBM, we believe that technology has the power to change the world – not for some of us, but for all of us.

https://developer.ibm.com/callforcode/

Skill-building

You can learn about industry-leading open hybrid cloud technologies and develop skills that can enhance your professional career while making a positive impact on the world.

Social Good

By fighting back against the most pressing humanitarian and societal issues of our time, you are building and/or contributing to projects that have the potential to be deployed into communities in need.

$200K Grand Prize

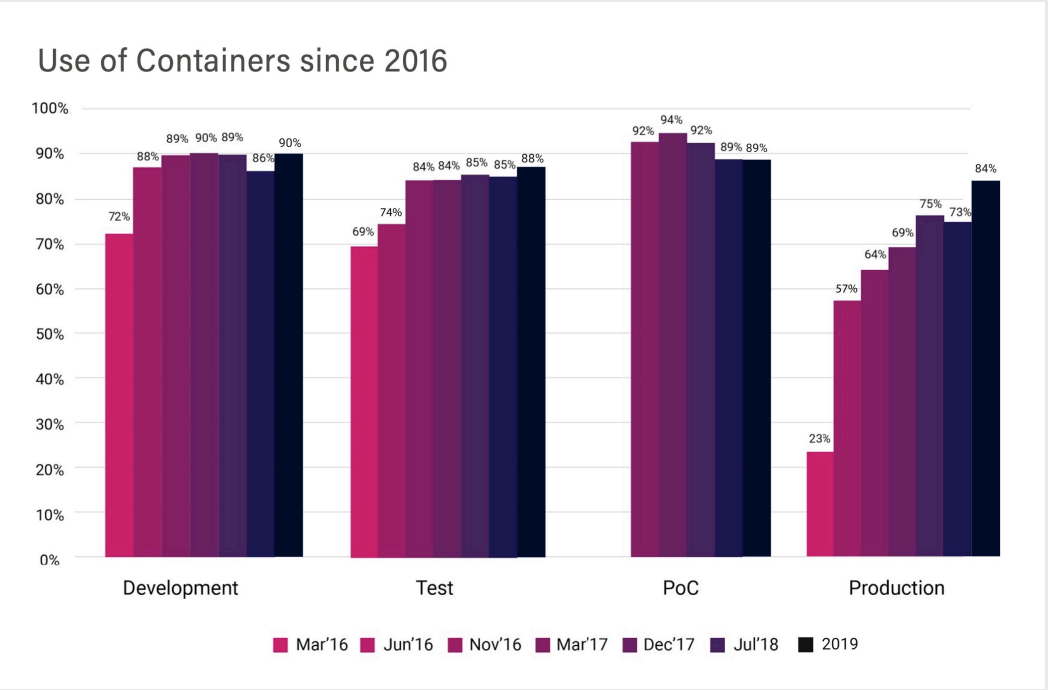
The winner of the Call for Code Global Challenge wins $200k and support from IBM, and its partners help set-up the winning team from concept to company.

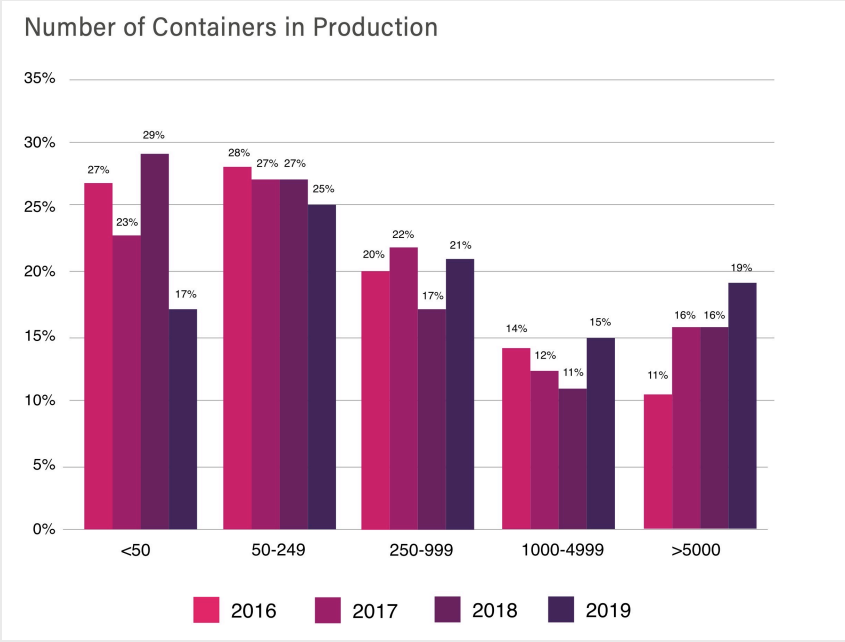
CNCF SURVEY 2019

Deployments are getting larger as cloud native adoption becomes mainstream

https://www.cncf.io/wp-content/uploads/2020/03/CNCF_Survey_Report.pdf

Container Usage is growing





But...challenges remain

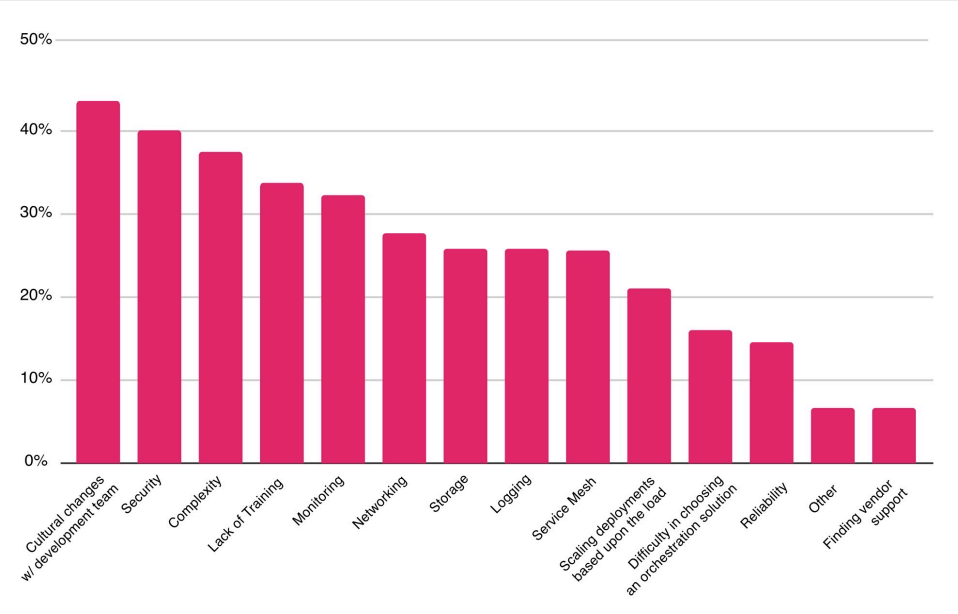
I want to deploy my container-based workloads **easily**, **quickly** and **securely** to a platform that:

... **dynamically scales** my containers according to load.

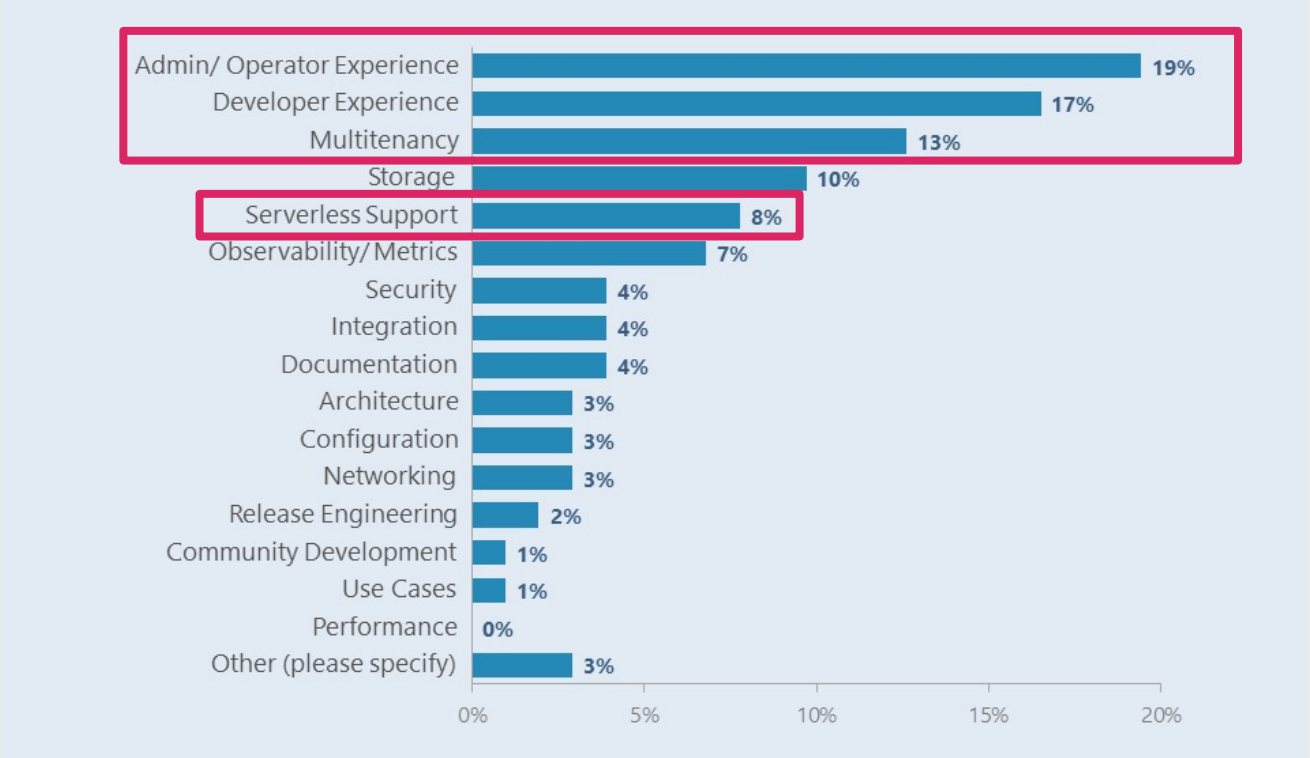
... has **monitoring**, **logging** and a **service mesh** “built in”.

... gives me the **ability to run serverless** workloads.

... does not “lock me in” and is **built on open-source** projects.



Top pain points of Kubernetes



*Top areas the Core Kubernetes Project Needs to Address in 2020, Thenewstack

Source: https://thenewstack.io/ux-is-kubernetes-biggest-short-term-challenge/


in Action
Kubernetes: The Complete Guide
Kubernetes:
Bits You Actually Need

Code Engine is a Kubernetes-based Container Platform that:

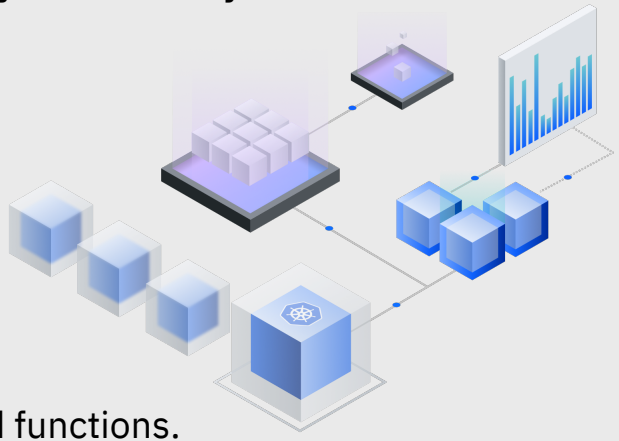
allows me to deploy my container-based workloads **easily**, **quickly** and **securely**.

dynamically scales my containers according to load.

has **monitoring**, **logging** and a **service mesh** “built in”.

gives me the **ability to run serverless** workloads.

does not “lock me in” and is **built on open-source** projects.



... and ...

unifies the deployment of containers, applications, batch jobs and functions.

starts as a **production-grade, multi-tenant shared container service**,
but will be extendable to other locations (via IBM Cloud Satellite) for higher isolation.

IBM Cloud Code Engine

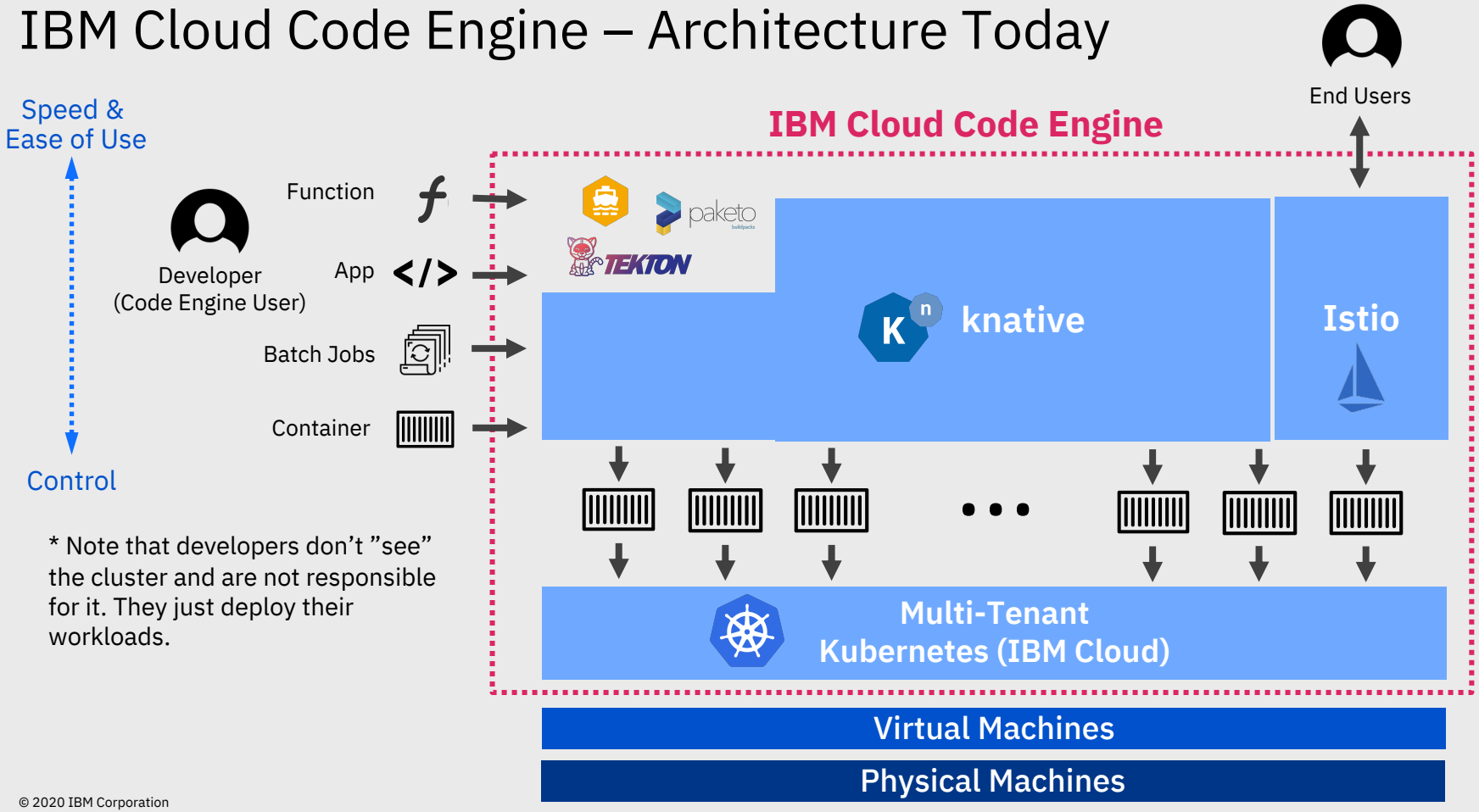
A developer can deploy **any type of application**
- container
- batch job
- source code
- function

on a unified platform **without**
- provisioning
- configuring
- managing
- securing

any
- clusters
- networks
- VMs
- certificates

and **only pay** when their application is active.

IBM Cloud Code Engine – Architecture Today



* Note that developers don't "see" the cluster and are not responsible for it. They just deploy their workloads.

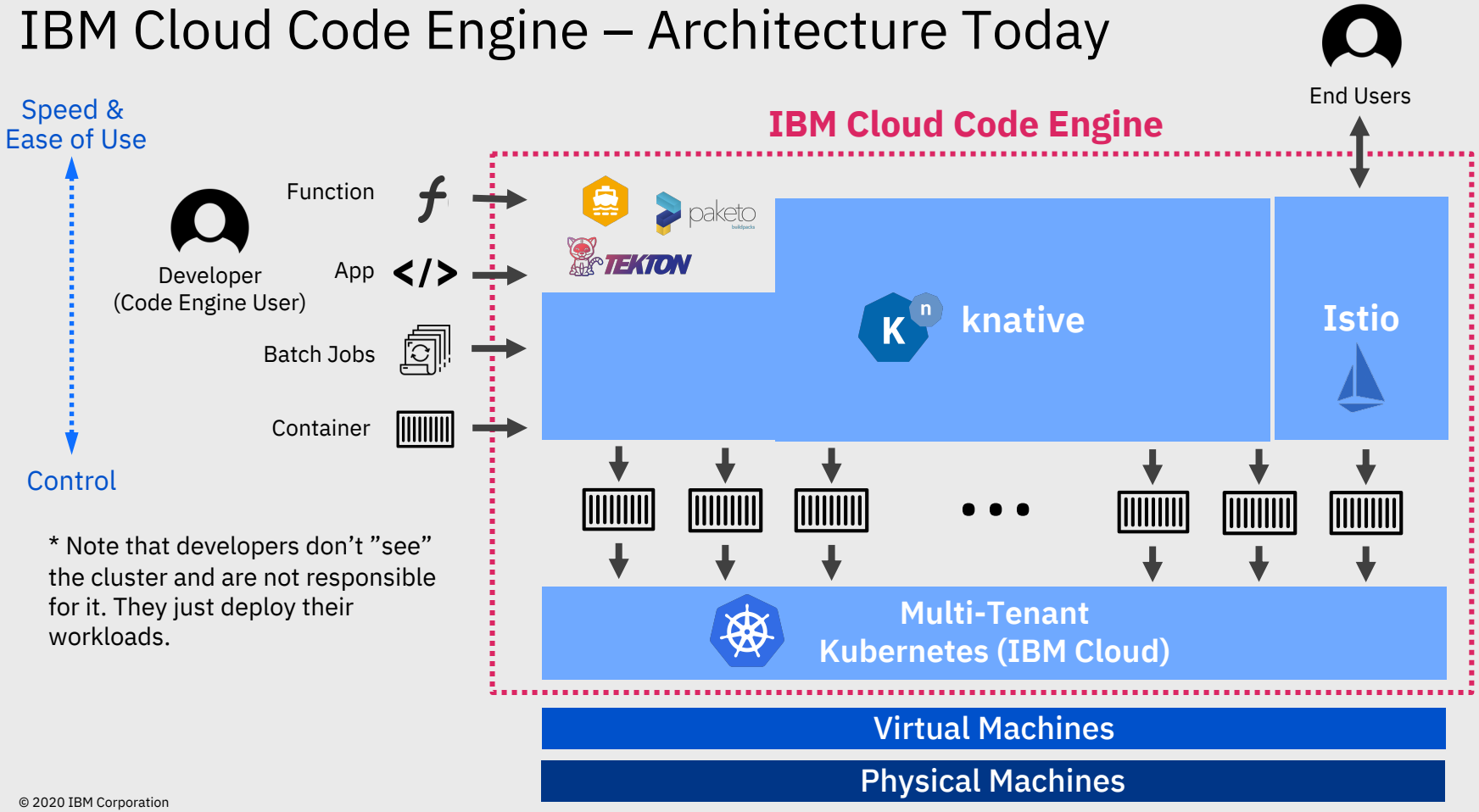
Let’s talk about Container Images

A container image is a **snapshot of the filesystem** for your application or job

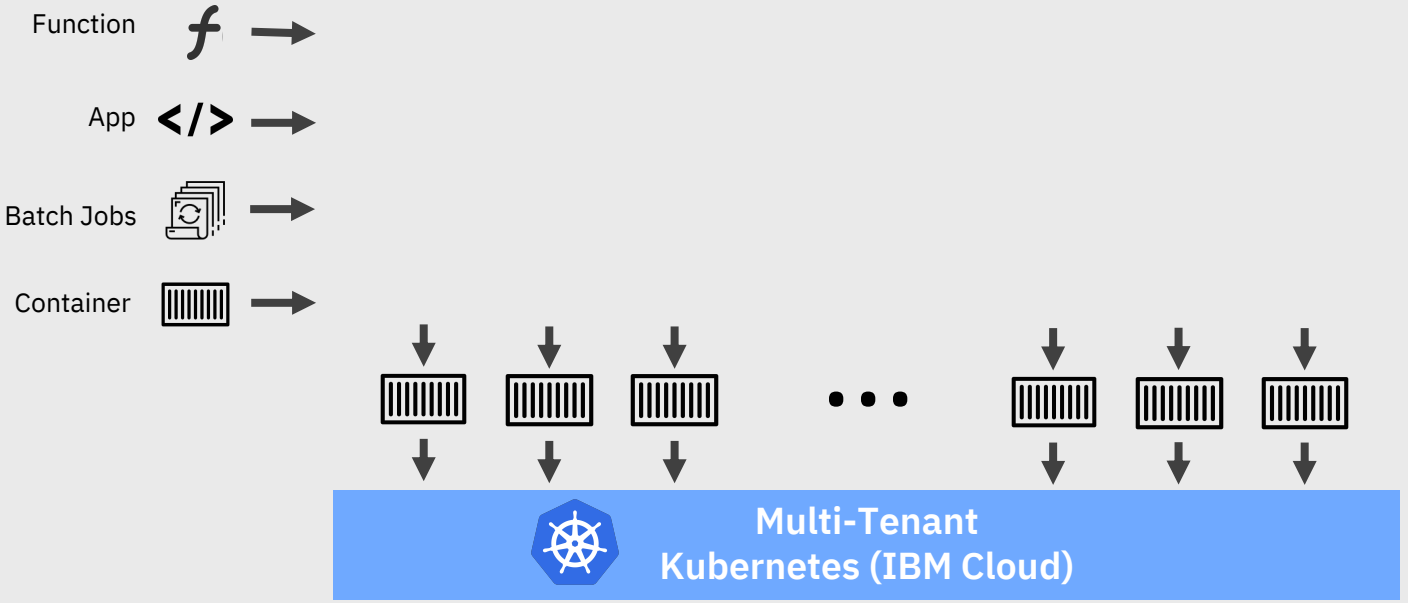
Plus some **extra metadata**, to be used at runtime

- Memory, CPU, ...
- Default command to run
- Environment variables
- ...

IBM Cloud Code Engine – Architecture Today



* Note that developers don't "see" the cluster and are not responsible for it. They just deploy their workloads.



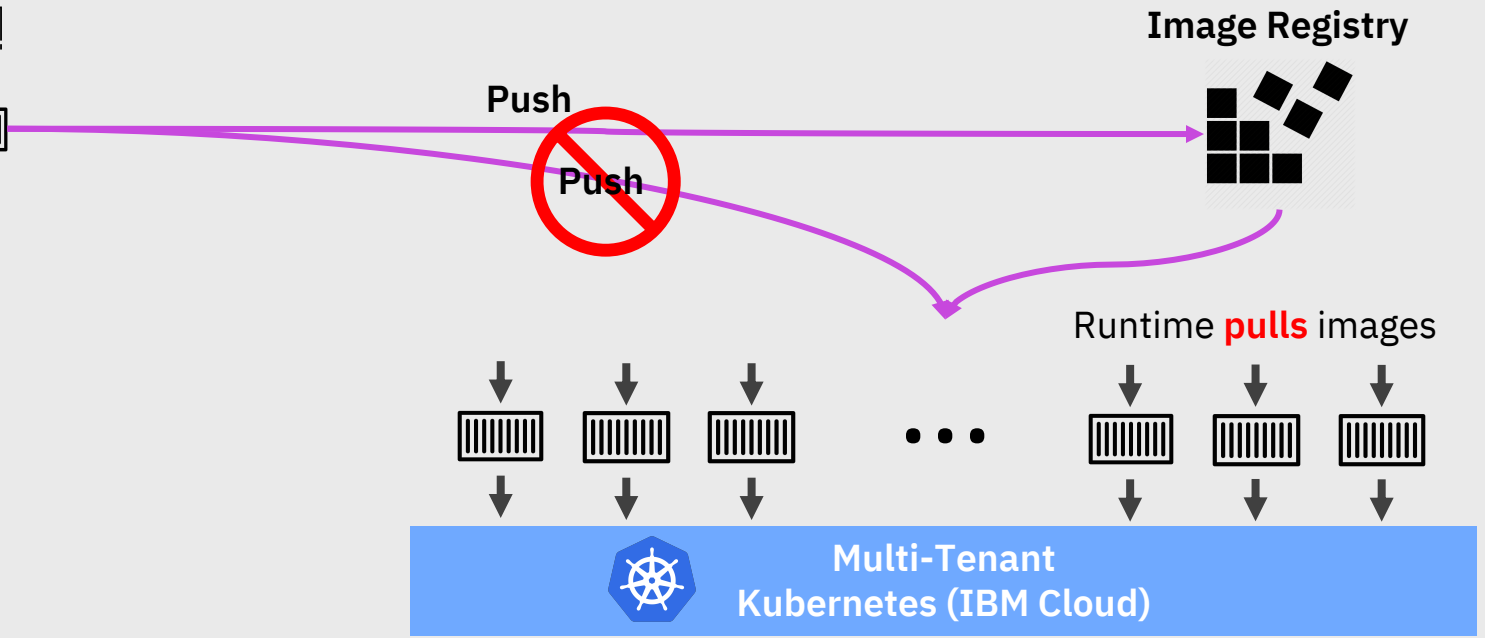
Function

App

Batch Jobs

Container

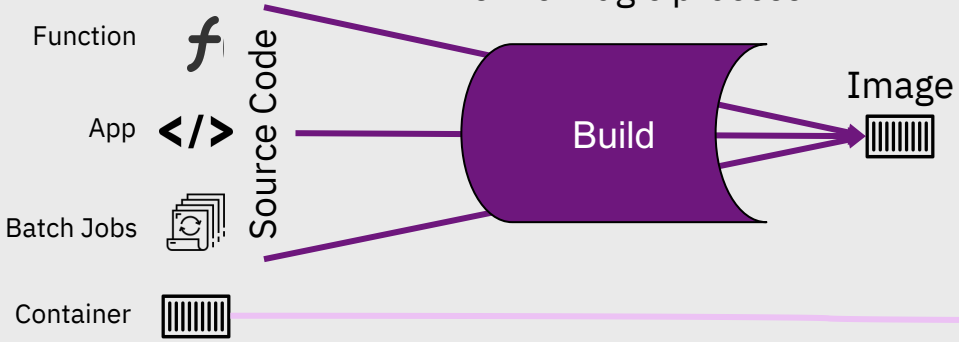
Image Registry is used to store/share Images

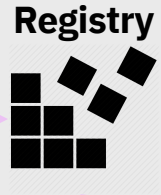


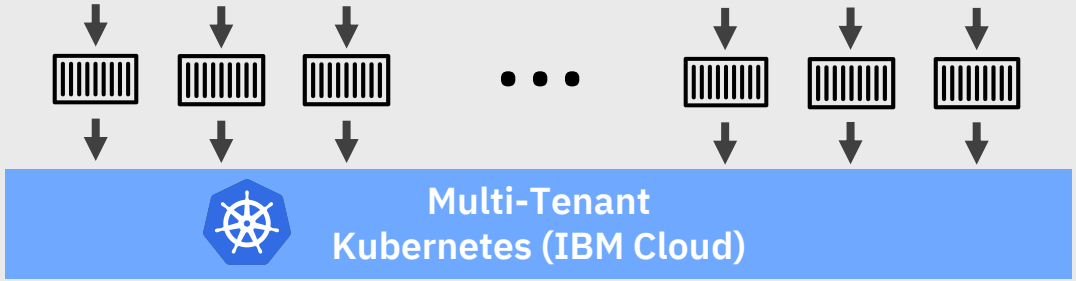
However, developers should focus on writing code, not building images

Plus, creating a good and secure container image can be complicated and time-consuming

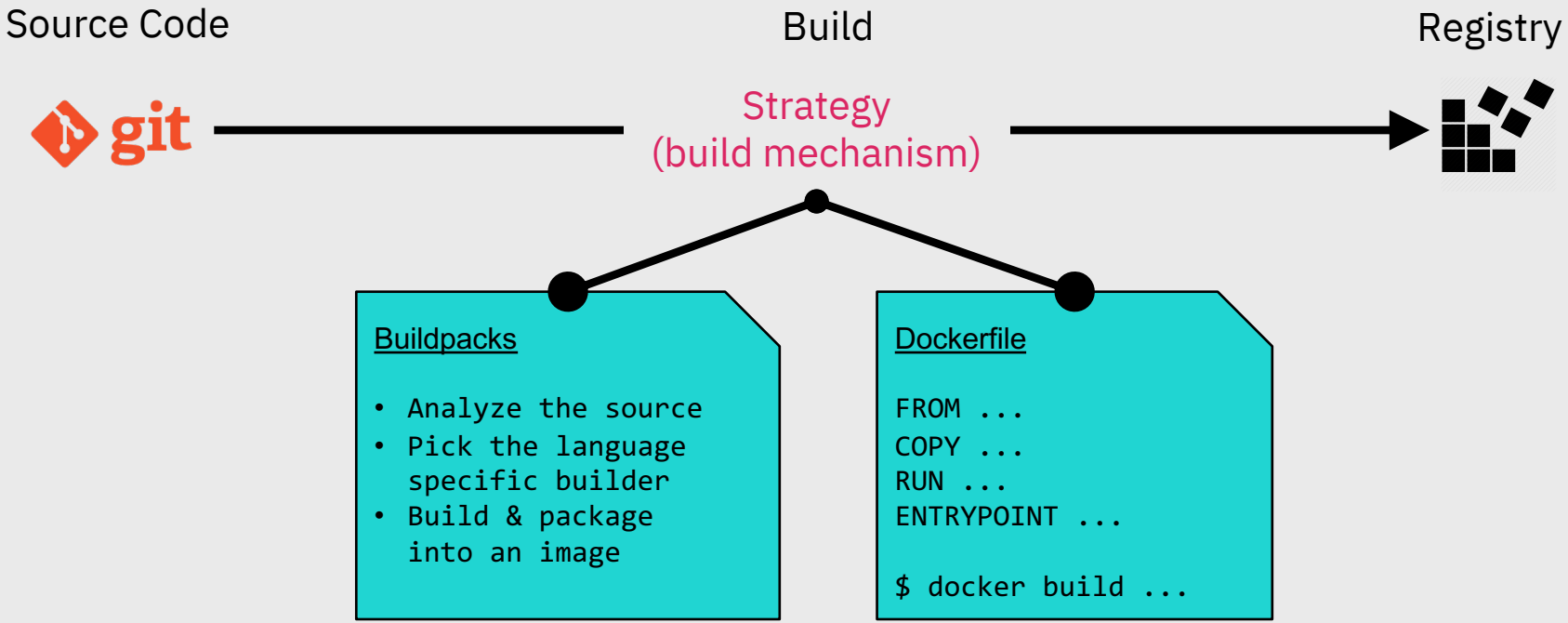
Build (aka “Source-to-Image”)
is the magic process







Build / Source-to-image Terminology



Choosing a Build Strategy

Dockerfile (flexible option)

But comes with costs & responsibilities:

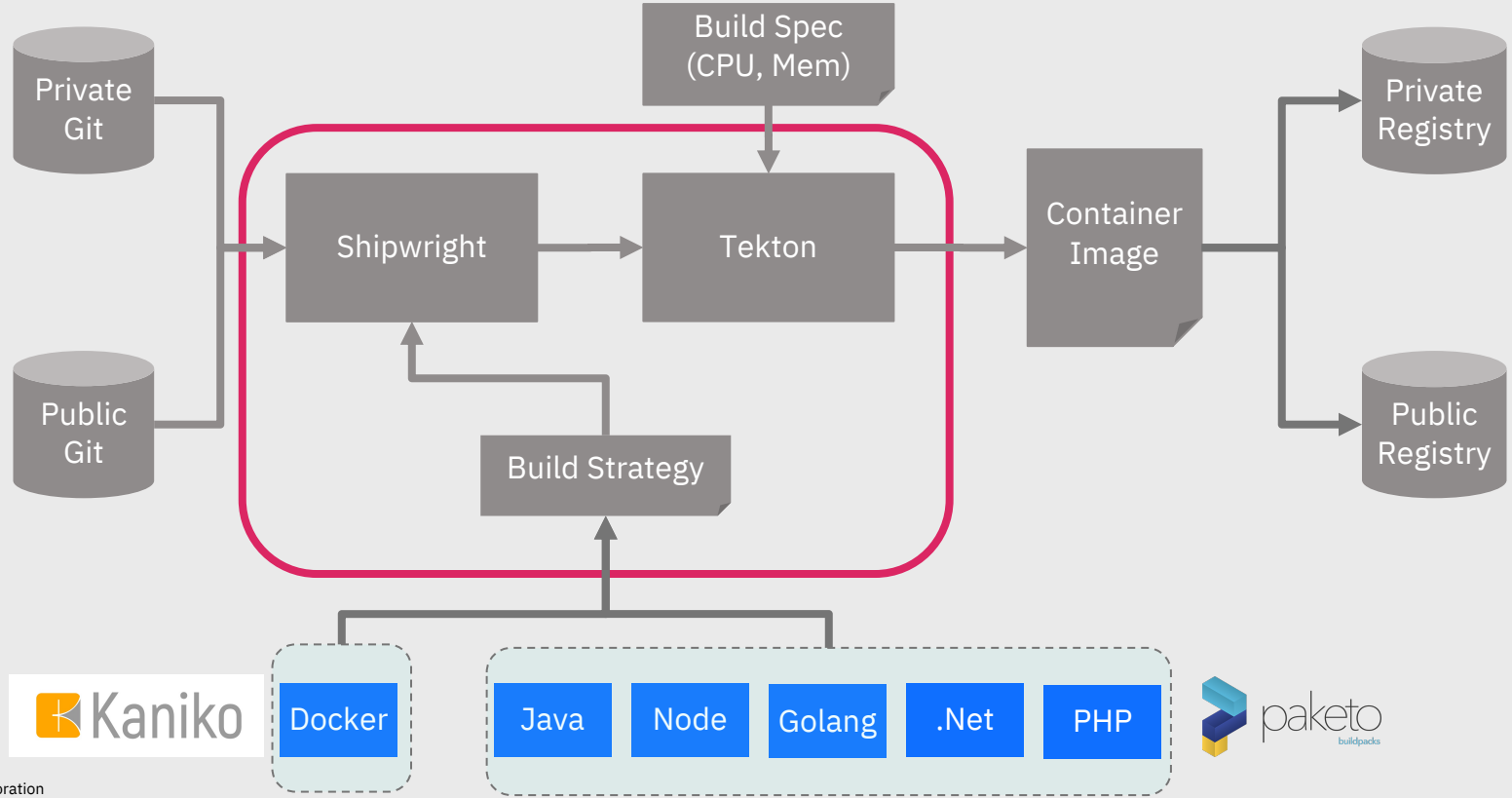
- More complex (need to understand Dockerfiles)
- You are in total control and responsible for
  - Base Image
  - Security **(!)**
  - Managing vulnerabilities in dependencies

Buildpacks (user-friendly option)

But comes with less choices & freedom:

- Nothing to learn (it all happens automagically)
- Completely managed build process
- Reliable & secure base images

Let’s have a look ”inside” Code Engine



Shipwright

A **framework for building container images**

- Highly extensible
- Developers can define and reuse build strategies for container images (used by CI/CD pipelines)
- In Code Engine, we support Kaniko and Cloud Native Buildpacks as builder tools

Open Source

- https://github.com/shipwright-io/build
- Collaboration between the IBM Code Engine and the Red Hat OpenShift teams
- In the future, Shipwright will replace “Source-2-image” in OpenShift

Tekton

A **framework for creating CI/CD systems**

- Powerful and highly extensible
- Kubernetes-native
- Provides components to standardize CI/CD tooling and processes

Open Source

- https://tekton.dev/
- Collaboration managed through the CD Foundation

Paketo

A **collection of buildpacks**

- Leveraging the Cloud Native Buildpacks framework to make image builds easy, performant and secure
- Ensuring that upstream languages, runtimes and frameworks are continuously patched in response to vulnerabilities and updates

Open Source

- https://github.com/paketo-buildpacks
- Widely used in the industry and quickly becoming the ”standard”

Kaniko

A **tool to build container images**

- Builds container images from Dockerfiles
- Executes each command within the Dockerfile completely in userspace
- Enables building container images in environments that can't easily or securely run a Docker daemon (such as a standard Kubernetes cluster)

Open Source

- https://github.com/GoogleContainerTools/kaniko

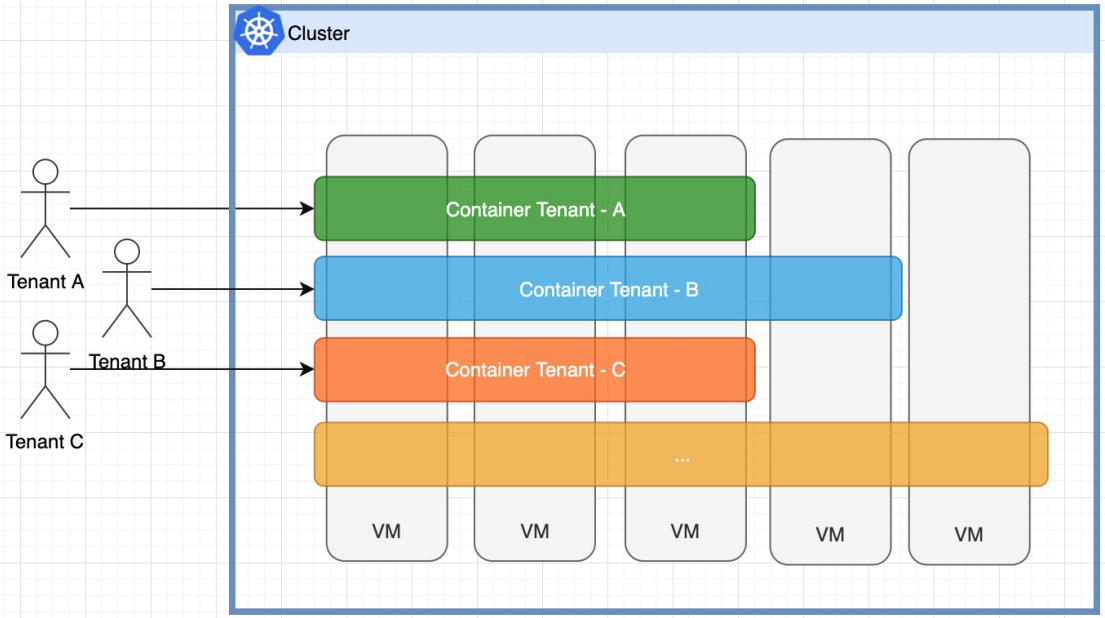
Code Engine Security

In our multitenant environment we

run (untrusted) code

from **different tenants**

in the **same network and compute infrastructure**



where one tenant is **separated** from the others by using **namespace isolation**

Authentication & Authorization (RBAC)

- Admission Control
- Subject, Cluster Role Bindings, Roles, Namespaces

Network Security Policies

- Control the network flows within your environment
- Prevent untrusted code to access private network endpoints
- Prevent untrusted code to access other tenants' endpoints
- Prevent workload from public network attacks

Pod Security Policies and Settings

- Controls security sensitive aspects of the pod specification
- Enabled and enforced by admission controller
- Prevents privilege escalation, Sysctl and capabilities like CHOWN
- Enforces RunAsUser
- Only allows specific kind of volume mounts for he pods

Resource Quotas and Limit Ranges

- Avoid running out of capacity
- Cost control of your infrastructure
- Each tenant gets fair share of the capacity
- Allows resource distribution based on tenant contract

Use Cases - Abstractly

Applications

- Short-lived, event driven calls
- Infrequent or has sporadic demand, with large, unpredictable variance in scaling requirements
- Stateless, ephemeral
- Highly dynamic in terms of changing business requirements that drive a need for accelerated developer velocity

Batch

- Asynchronous, concurrent, easy to parallelize into independent units of work

Use Cases - Concretely

- Executing logic in response to database changes (insert, update, trigger, delete)
- Performing analytics on IoT sensor input messages (such as MQTT messages)
- Handling stream processing at scale (analyzing or modifying data in motion)
- Managing single time extract, transform, and load jobs that require a lot of processing for a short time (ETL)
- Providing cognitive computing via a chatbot interface (asynchronous, but correlated)
- Scheduling tasks performed for a short time, such as cron or batch style invocations
- Serving machine learning and AI models (retrieving one or more data elements such as tables, NLP, or images and matching against a pre-learned data model to identify text, faces, anomalies, etc.)
- Continuous integration pipelines that provision resources for build jobs on-demand, instead of keeping a pool of build slave hosts waiting for jobs to be dispatched

Supporting a wide variety of Workloads

- HTTP endpoints, e.g. REST API, web/mobile/IoT backend
  - Web Hooks
- Model serving
- Batch
- Any kind of background worker workload
- Any kind of ‘fan-out/fan-in’ workload
- HPC
- Monte Carlo simulations
- Financial risk modeling
- Highly parallel
  - Document/pdf processing
  - Image processing
  - Video processing
  - Audio processing
  - File processing
- Map/(reduce) workloads
- Media transcoding
- Web scraping
- Model scoring for large amounts of images
- OCR
- Parallel data processing
- Rendering
- Genetic sequence analysis
- Object storage / data lake processing
- Batch processing (incl. mainframe modernization)
- High Throughput Computing (HTC)
- Computationally heavy tasks
- Post-trade analysis
- Fraud surveillance
- Batch processing of Object storage
- ETL
- Scientific computing
- Molecule simulation
- Drug screening
- DNA sequencing
- Transcoding
- Media supply chain
- Hyperparameter tuning
- ...and many more...

https://github.com/tal2k4xj/CodeEngine