Code @ Think

IBM Code Engine- “Any workload accelerated”

Jaffa Sztejnbok
Developer Advocate
IBM

IBM **Developer**

IBM

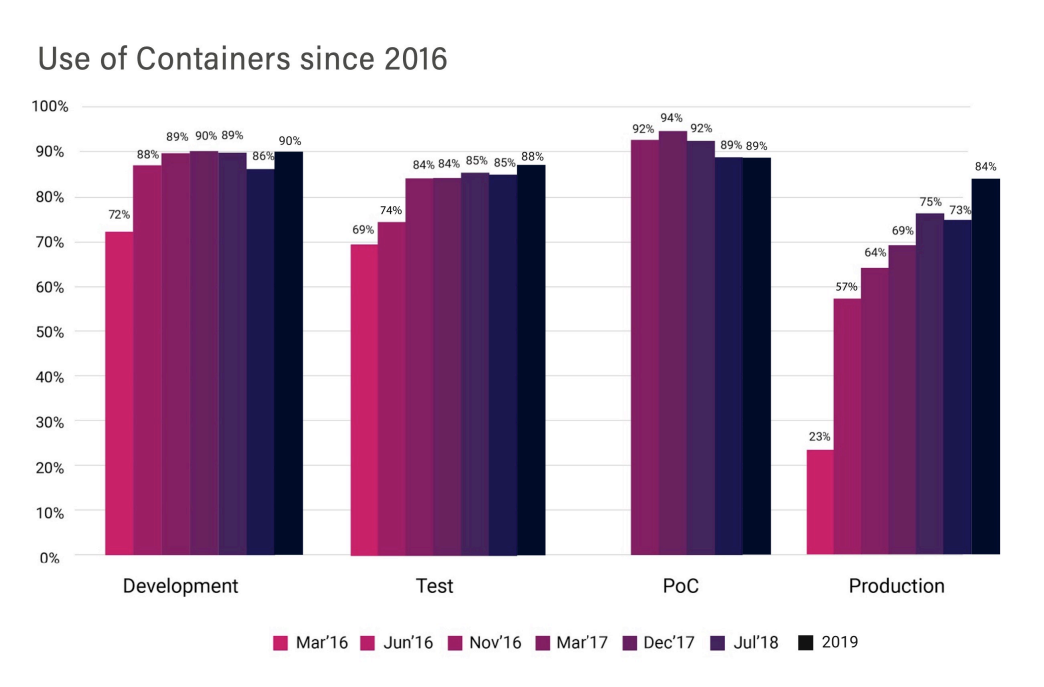
Important Links for this event ...

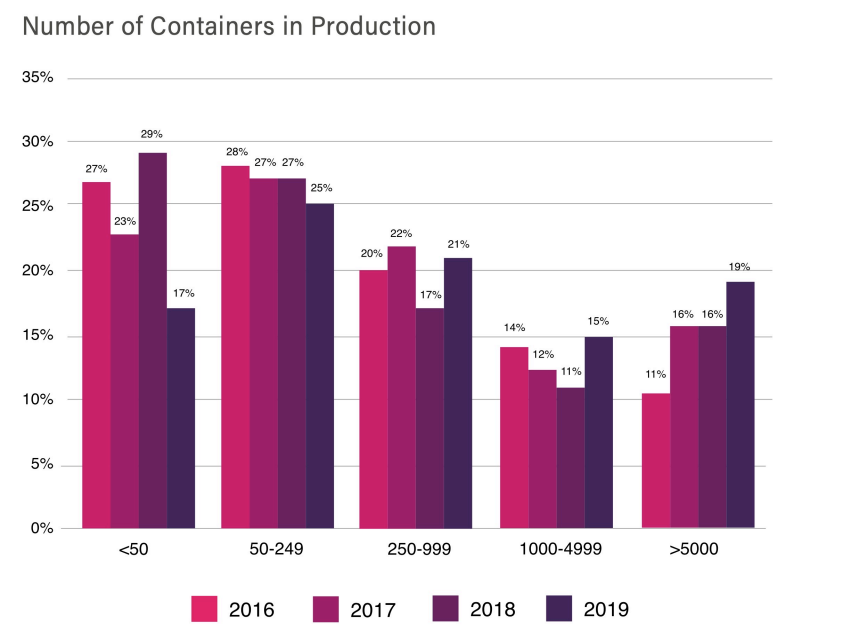
1. PPT https://github.com/jaffaszt/CodeEngineWorkshop

2. Signup for IBM Cloud account : https://ibm.biz/Bdfxxf

3. Please answer a short survey
   https://ibm.biz/BdfxxP

Container Usage is growing – CNCF survey 2019





© 2020 IBM Corporation

But...challenges remain

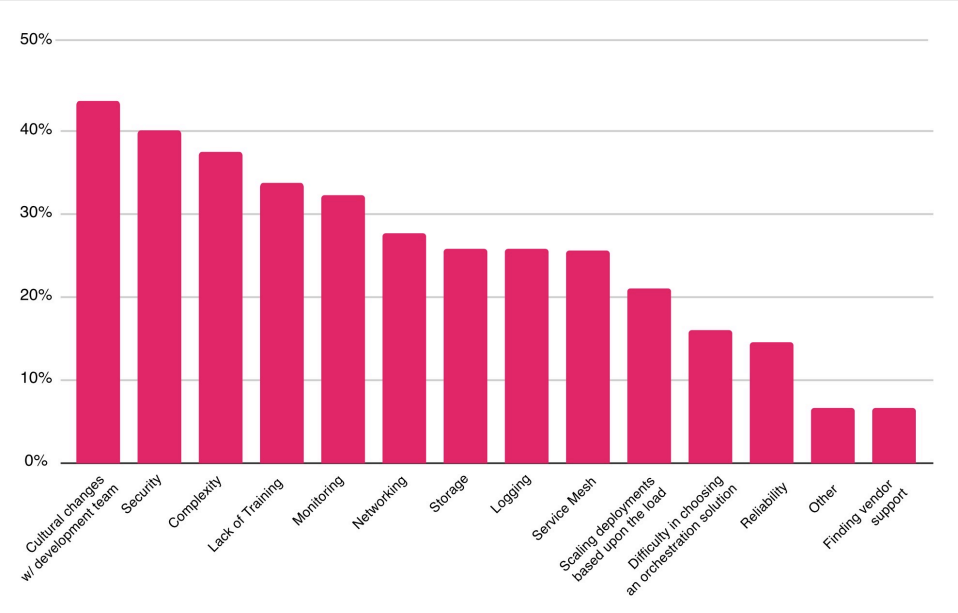
I want to deploy my container-based workloads **easily**, **quickly** and **securely** to a platform that:

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

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

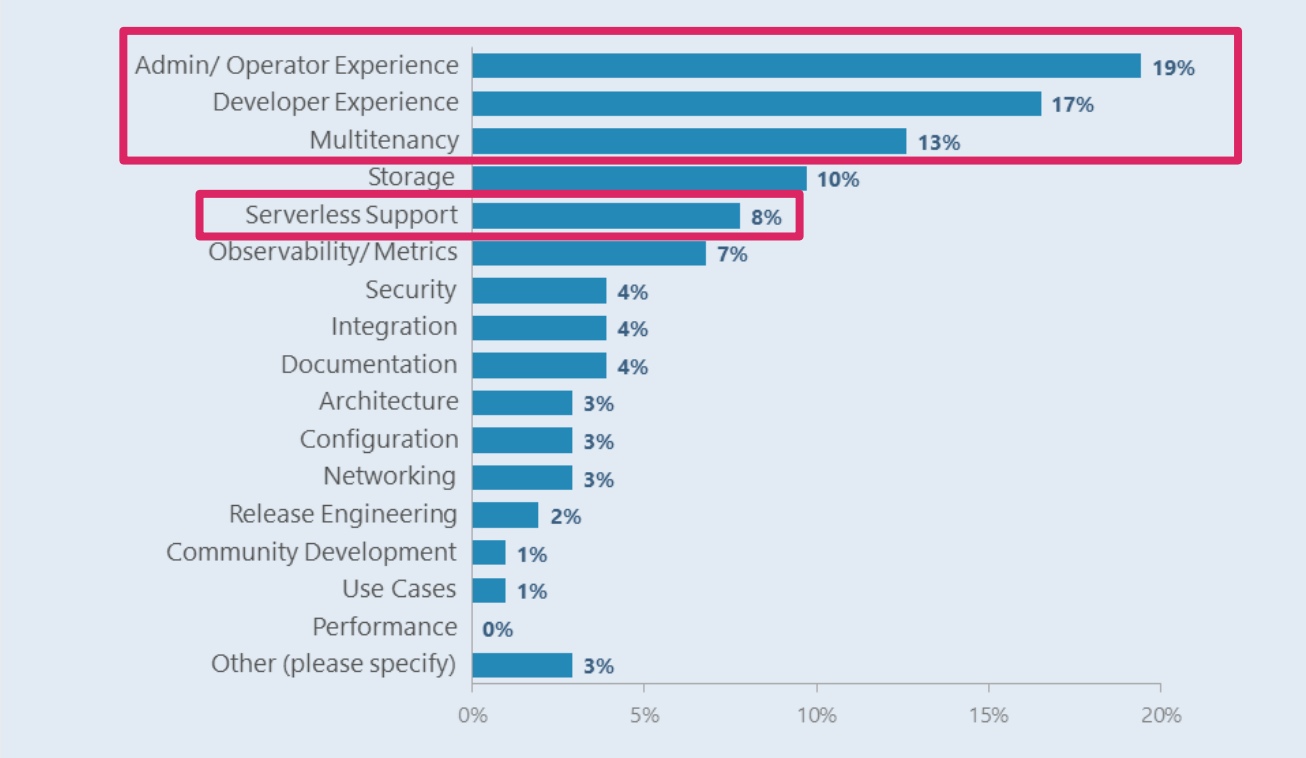
… gives me the **ability to run serverless** workloads.

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



© 2020 IBM Corporation

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/

© 2020 IBM Corporation

Kubernetes: The Complete Guide
Kubernetes:
Bits You Actually Need

Kubernetes FTW ?

Missing features compared to other *aaS
Simplified UX
Build
Event driven/tooling
Async tasks

DIY Features
Endpoint + Load Balancing
On-demand infrastructure - auto-scaling

Resources
Containers
Pods
Replica Sets
Deployments
Services
Endpoints
Secrets
Networks
Volumes/PV/PVC
Ingress/LBs

Tooling & Other "stuff"
yaml
Spec vs Status
helm
kubectl
Istio
...

Blue/green deployments

Kubernetes

~~Kubernetes abstracts & exposes the infrastructure~~
~~But all I want is to deploy my code~~
I just want to code

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”.

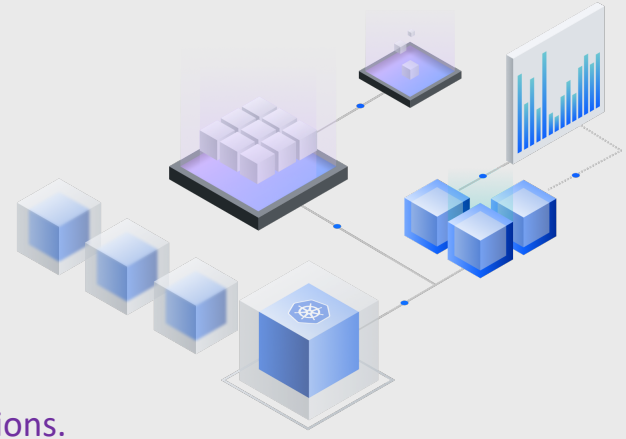
gives me the **ability to run serverless** workloads.

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.



© 2020 IBM Corporation

Our Users

Container-Savvy Developer

I can run my containerized application without having to worry about sizing, creating or managing a cluster.

“Run my container” vs. “Give me a cluster, that I can then run my container on”.

Functions Developer

I love Functions-as-a-Service and can now run them with almost no limits.

I now have a single platform to securely combine Functions with Apps and other containerized workloads

Batch Job Creator

I can create powerful batch jobs and easily combine them with events and other services.

The underlying platforms scales out and allows me to run massively parallel jobs … and I only pay for what I use.

PaaS Developer

I can start utilizing a new powerful platform and:
- keep using a “push source code” experience
- do not have to worry about containers
- can easily connect my code to backing services

With Code Engine you can run **any workload serverlessly**. That includes long-running containers and batch jobs as well as bursty and short-lived functions. Other platforms force you to choose between long execution times, high resource limits, private networking and scale-to-zero. We don't. All on **one fully-managed platform**, which means that **you don't need any infrastructure skills and can get back to coding**.

© 2020 IBM Corporation

A developer can deploy **any type of application**

container
batch job
source code
function

on a unified platform **without**

provisioning
configuring
managing
securing

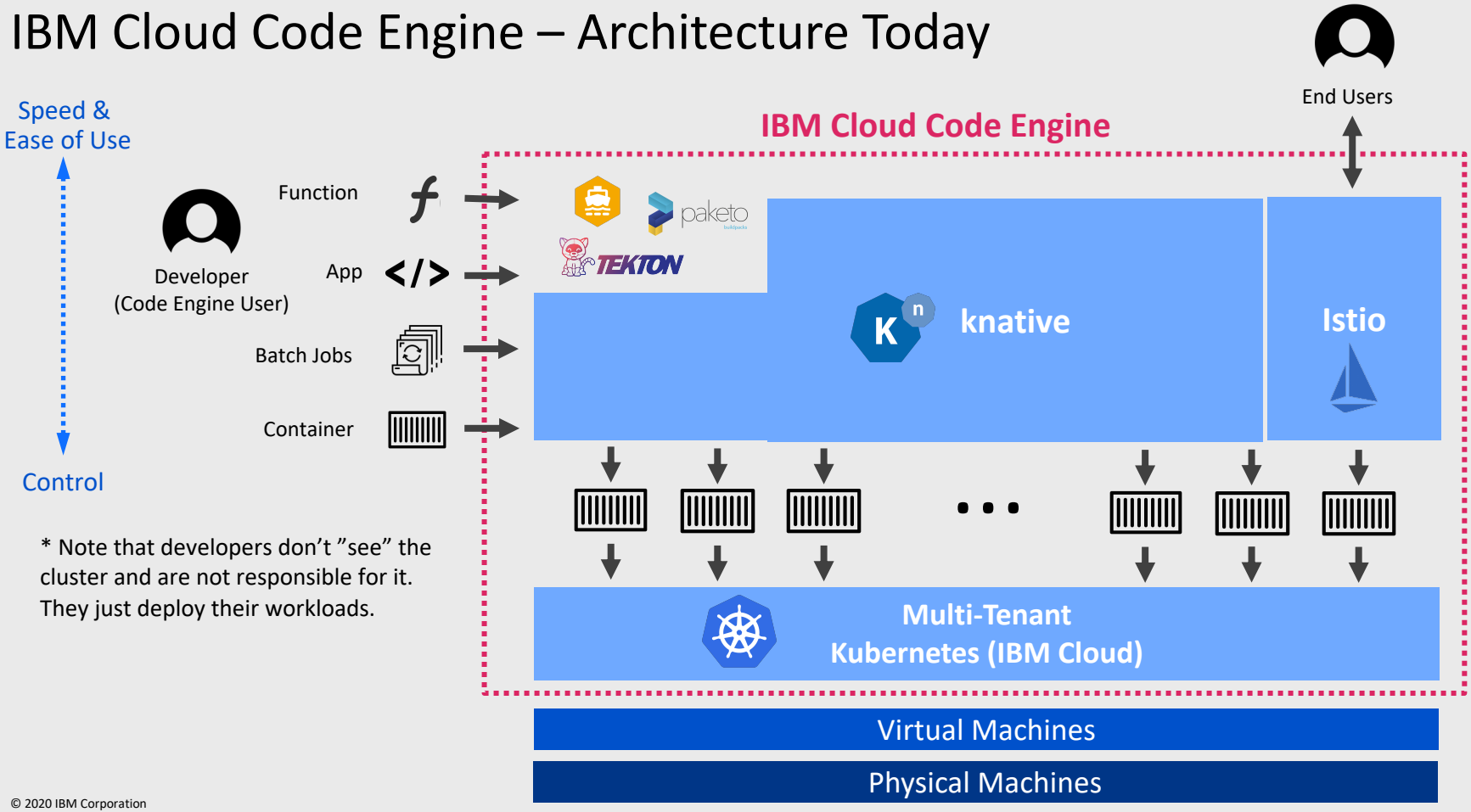
any

clusters
networks
VMs
certificates

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

© 2020 IBM Corporation

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.

© 2020 IBM Corporation

IBM Cloud Code Engine – The bigger Picture

Serverless 1.0

- Transparent scaling (including scale-to-zero)
- No infrastructure mgmt.
- No capacity mgmt.
- Pay-by-consumption

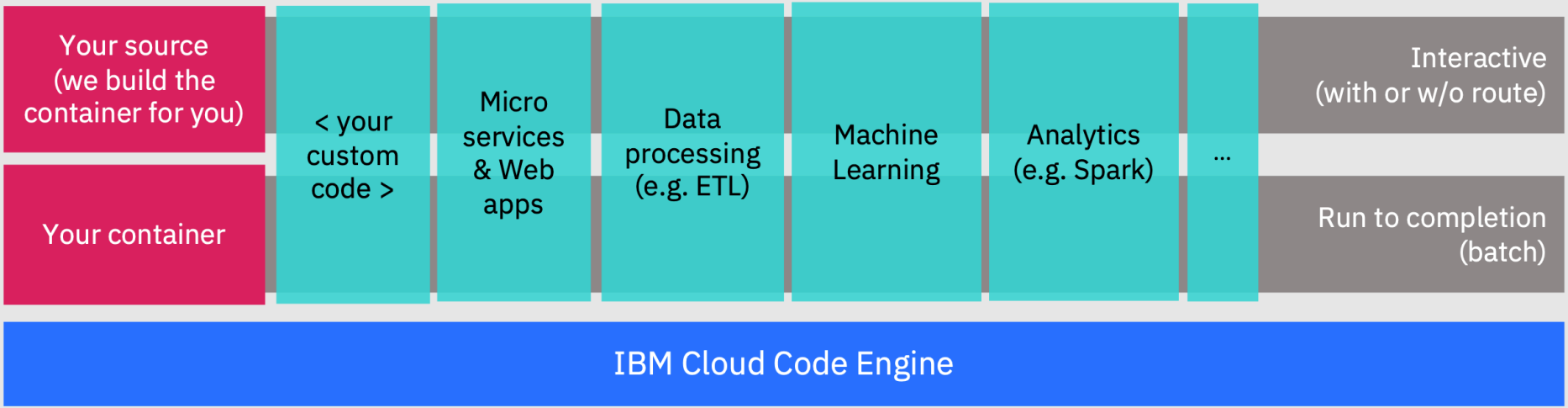
+

Elimination of limits

- (Almost) no CPU, memory or disk limits
- No duration limit

=

Serverless 2.0



© 2020 IBM Corporation

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...

© 2020 IBM Corporation

Let’s talk about Container Images

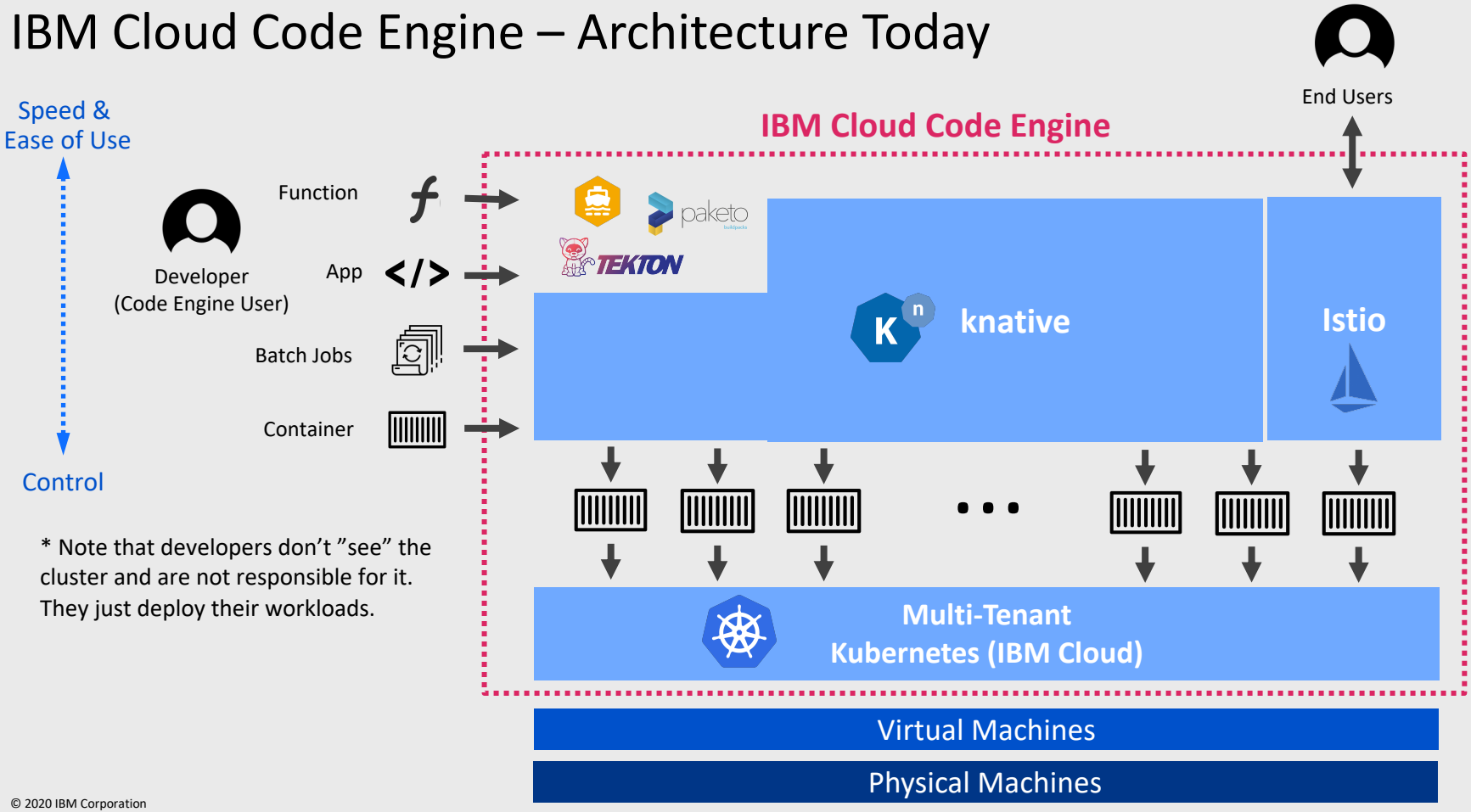
A container image is a **snapshot of the filesystem** for your application or job

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

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

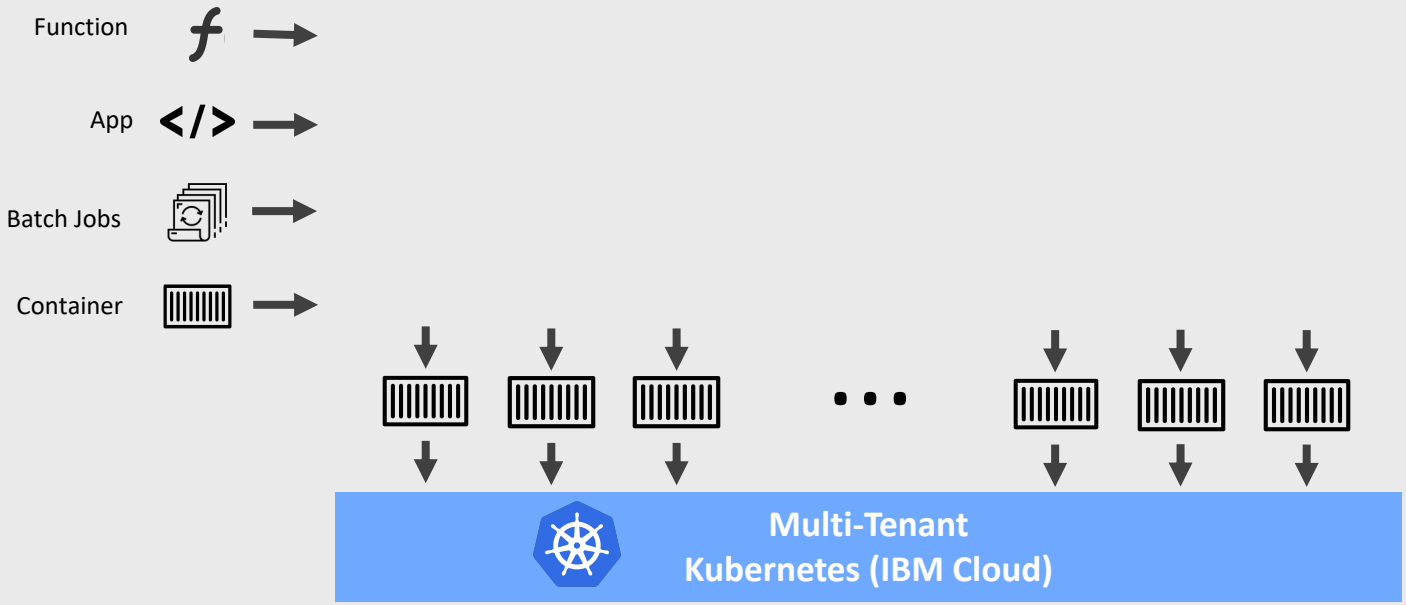
© 2020 IBM Corporation

IBM Cloud Code Engine – Architecture Today



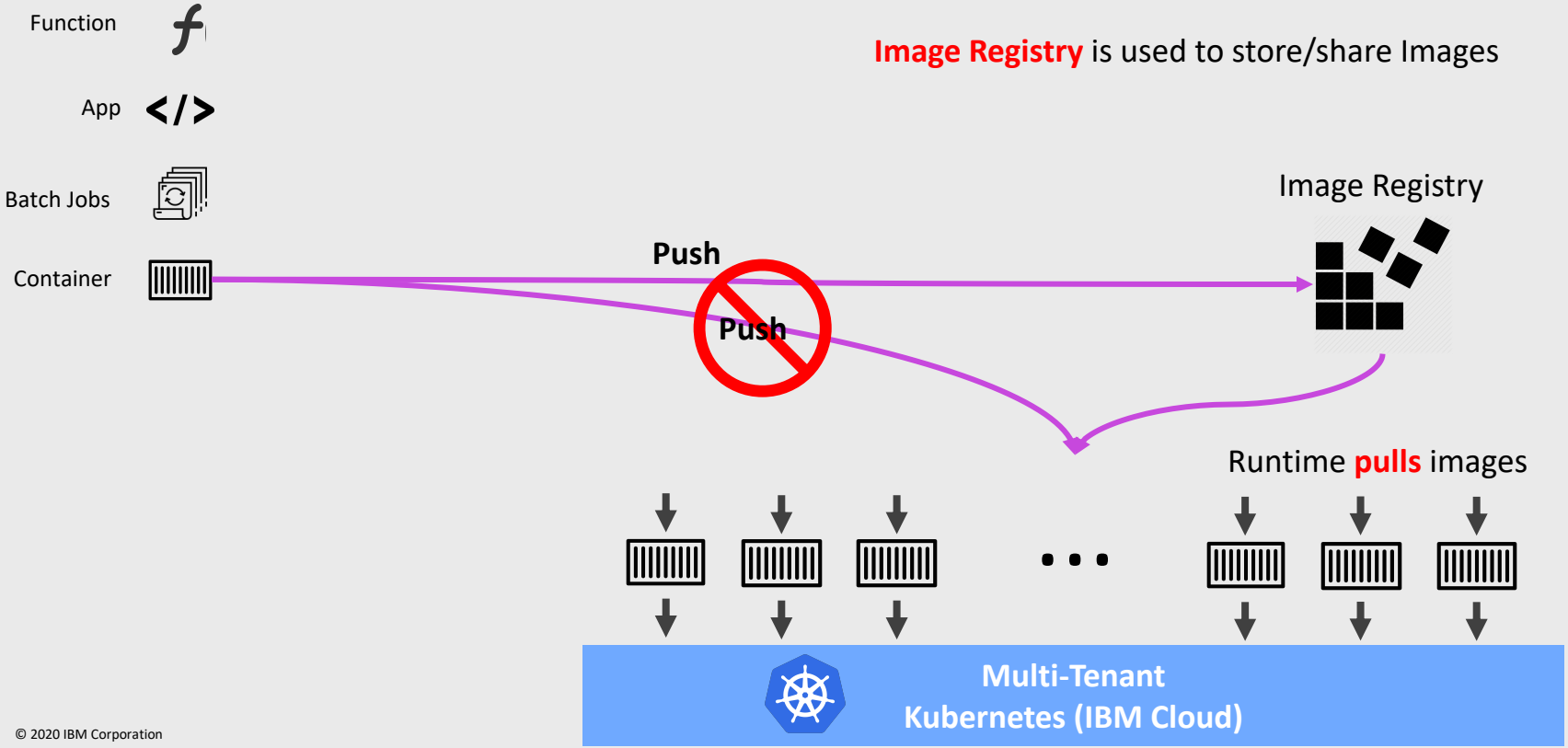
* Note that developers don't "see" the cluster and are not responsible for it. They just deploy their workloads.

© 2020 IBM Corporation



© 2020 IBM Corporation

Image Registry is used to store/share Images

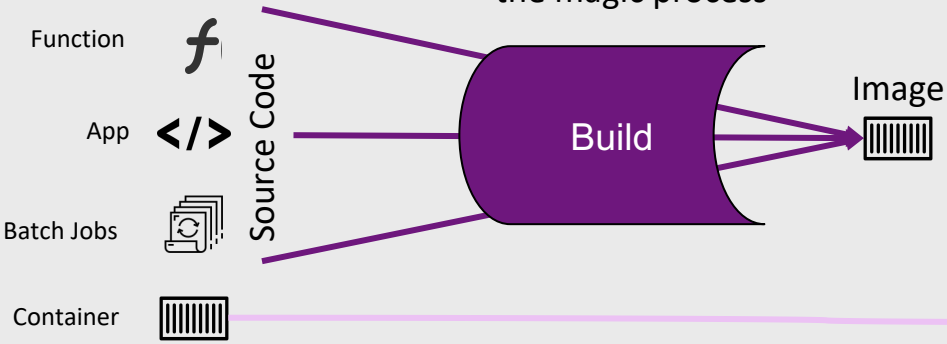


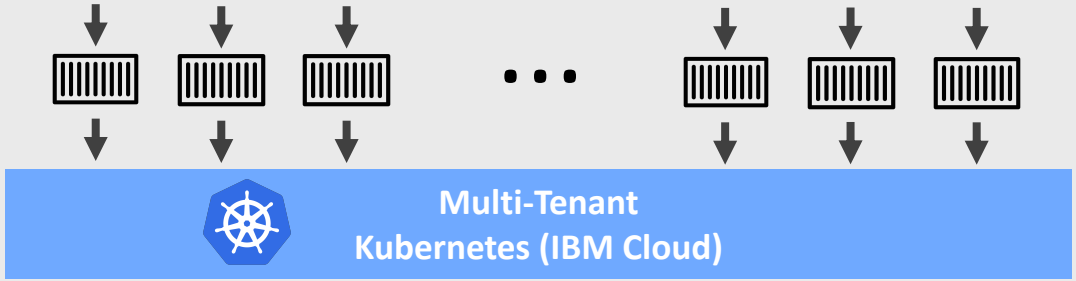
© 2020 IBM Corporation

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





© 2020 IBM Corporation

Build / Source-to-image Terminology

Source Code

Build

Strategy
(build mechanism)

Registry

Buildpacks

- Analyze the source
- Pick the language specific builder
- Build & package into an image

Dockerfile

```
FROM ...
COPY ...
RUN ...
ENTRYPOINT ...

$ docker build ...
```

© 2020 IBM Corporation

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

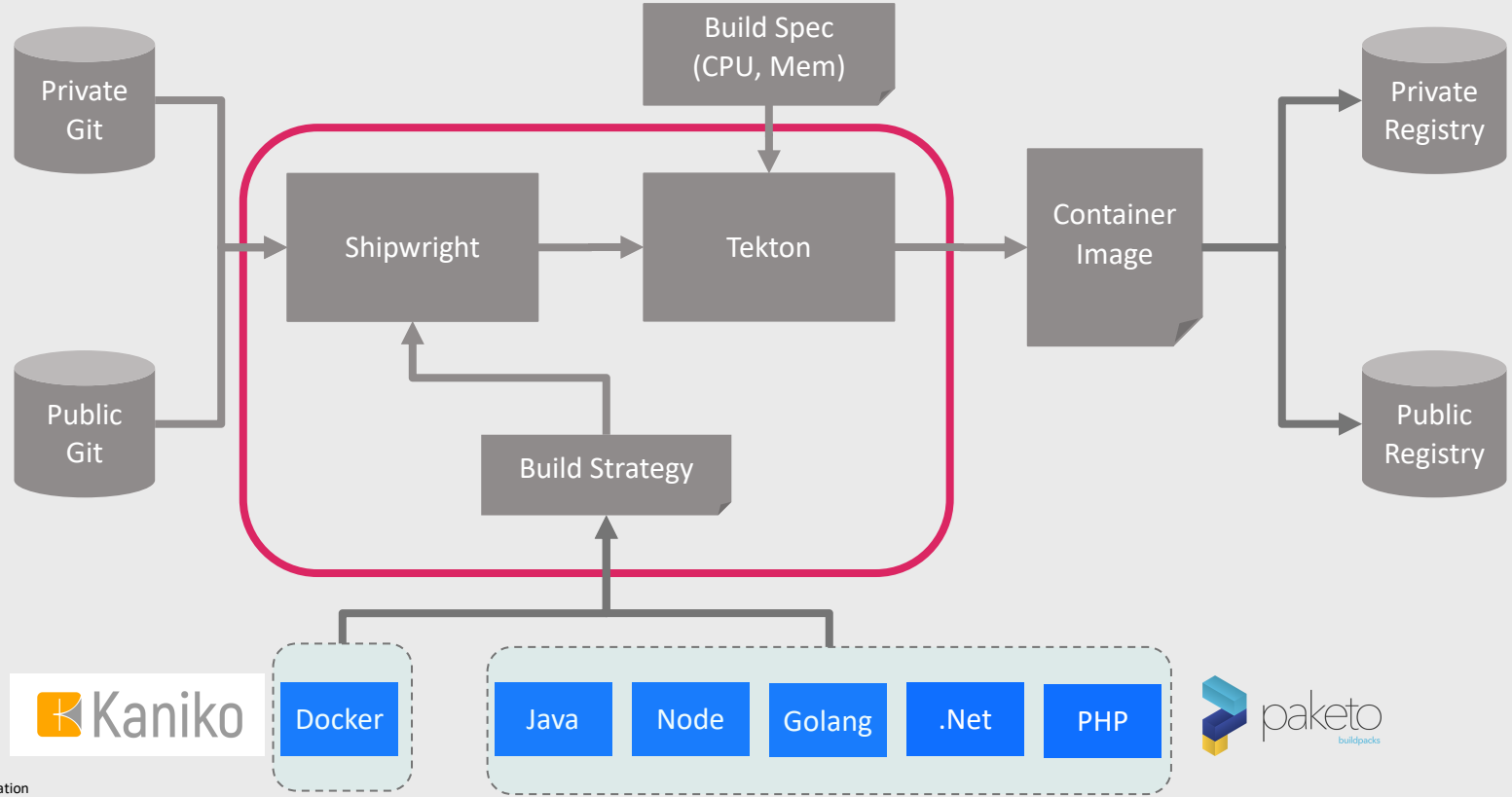
Buildpacks (user-friendly option)

But comes with less choices & freedom:

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

© 2020 IBM Corporation

Let’s have a look ”inside” Code Engine



© 2020 IBM Corporation

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

© 2020 IBM Corporation

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

© 2020 IBM Corporation

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”

© 2020 IBM Corporation

Kaniko

A **tool to build container images**

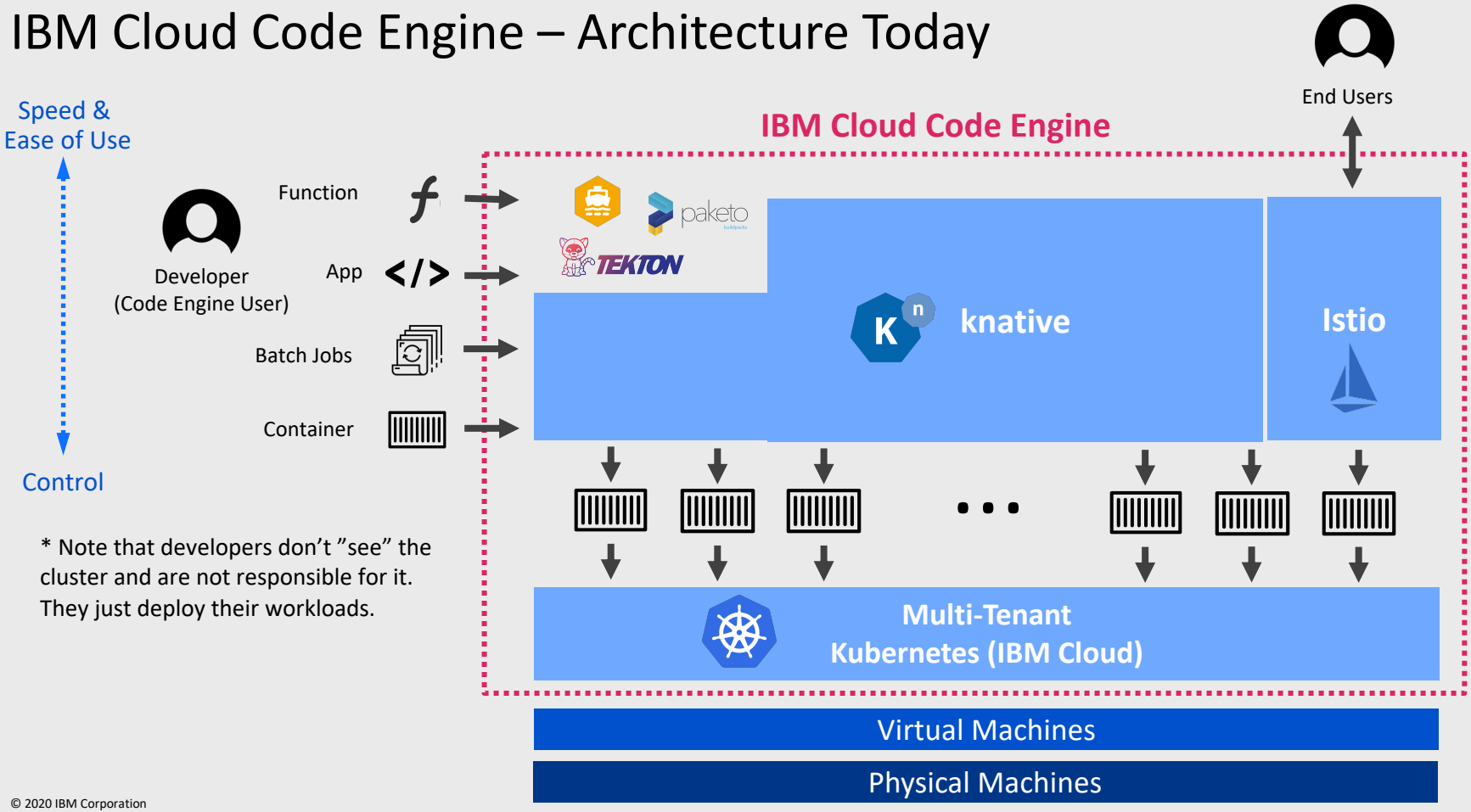
- Builds container images from Dockerfiles
- Executes each command within the Dockerfile completely in userspace
- Enables building container images in environments that can't easily or securely run a Docker daemon (such as a standard Kubernetes cluster)

Open Source

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

© 2020 IBM Corporation

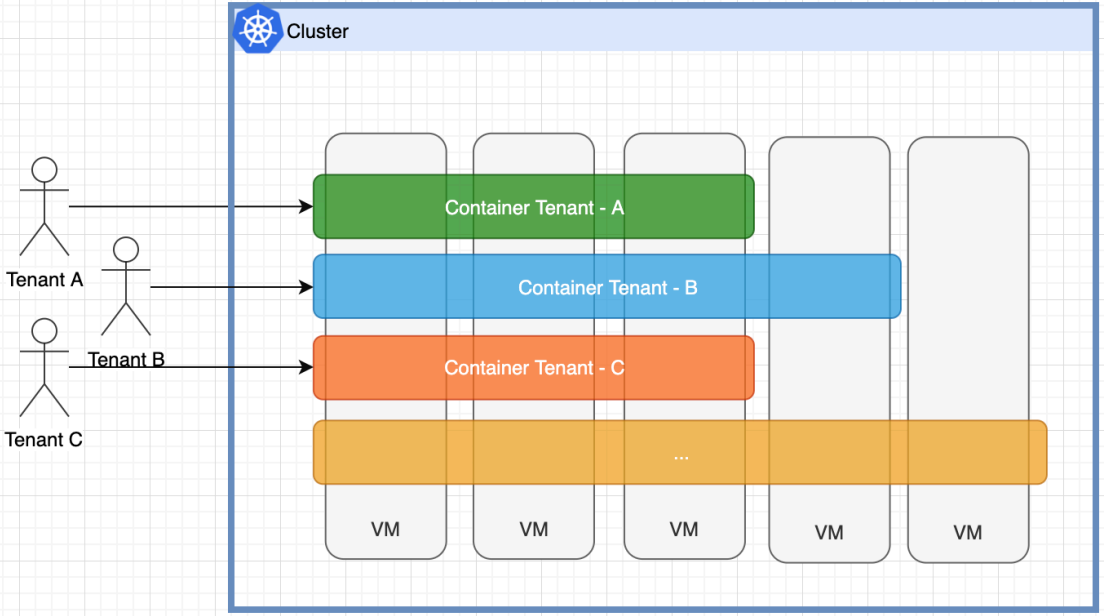
IBM Cloud Code Engine – Architecture Today



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

© 2020 IBM Corporation

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

© 2020 IBM Corporation

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

© 2020 IBM Corporation

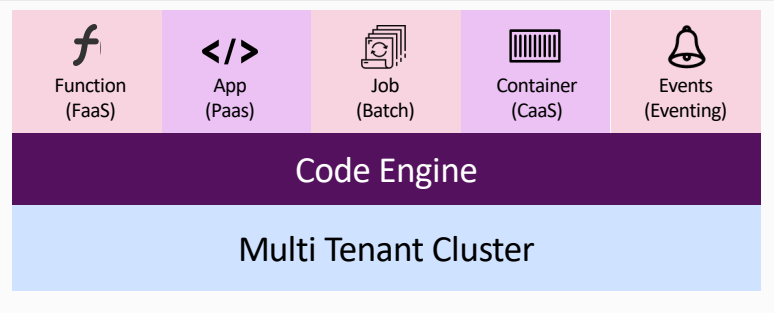
"I just want to write & run my applications"

"I don't want to learn Kubernetes"

"I want smart infrastructure that does what it needs to do, not only what/when I tell it to do it"

Code Engine:

- **Managed, simplified, intelligent hosting experience**
- **All your container-based workloads in one environment**
- **Pay only for what you use - Free while in Beta**
- **Designed to let your developers get back to coding**
- **The power of Kubernetes without the complexity**



Try it today for free: https://cloud.ibm.com/codeengine

Why?

- Kubernetes, while feature rich, is too complex
- Kubernetes is missing some newer features - e.g. serverless
- Forcing a choice of *aaS is artificial and unnecessary
- Developers should code, not manage infrastructure

What?

- Public managed container hosting platform
- Supports containerized workloads (web, APIs, batch, ...)
- Key features
  - Request-based auto-scaling - including to zero
  - Built-in security - e.g. networking, runtime isolation
  - App versioning
  - Build images from Source - Dockerfiles or Buildpacks
  - Integration with IBM Cloud Services (service bindings)
  - No artificial limits - e.g. execution time
  - Runtime choice - e.g. cpu, memory
  - Event driven integrations (Cloud Object Storage, Ping)
  - Private Registry Support

How?

- Built on open-source technologies - Kube, Istio, Knative, ...
- Infrastructure still available if necessary, goal is to hide it
- Integrates with existing Kubernetes workloads

Getting Started

IBM Cloud Code Engine in the cloud catalog:
https://cloud.ibm.com/codeengine/overview

Samples for IBM Cloud Code Engine
https://github.com/IBM/CodeEngine

© 2020 IBM Corporation

Important Links for this event ...

1. PPT https://github.com/jaffaszt/CodeEngineWorkshop

2. Signup for IBM Cloud account : https://ibm.biz/Bdfxxf

3. Please answer a short survey
   https://ibm.biz/BdfxxP