Microsoft

Using Azure DevOps to continuously build, test, and deploy containerized applications with ease

Adrian Todorov, Cloud Solutions Architect
@todorov_adrian | linkedin.com/in/adriantodorov

Hello!

I am Adrian Todorov

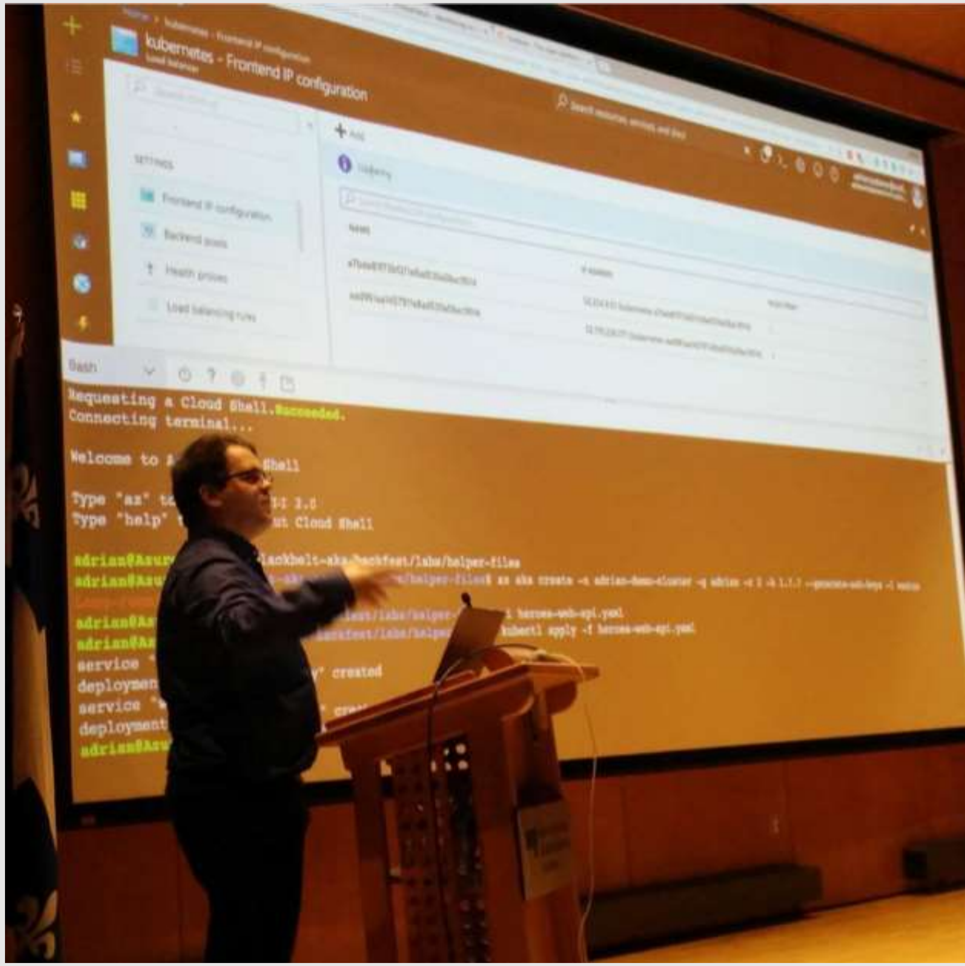
I am here because I love technology and community.

I focus heavily on Azure, OSS, DevOps, Kubernetes and Containers.

I like gaming and movies

I teach people how to use Kubernetes

Connect with me on LinkedIn or/and Twitter (e-book coming soon)


kubernetes - Frontend IP configuration
Frontend IP configuration
Backend pools
Health probes
Load balancing rules
Bash

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
```

GOING DIGITAL

1 million/hour
new devices
coming online
by 2020

12 years
average age of S&P
500 corporations
by 2020

60% computing
in the public cloud
by 2025

Monitoring

Location-based Authentication

Serverless

Event driven

Orchestrators

Machine Learning

Compliance

Mixed Reality

Haptics

Containers

Edge

DevOps

GPU

NoSQL

Automation

Quorum

Quantum computing

Microservices

Log telemetry

Digital Twin

Data privacy

IoT

Blockchain

Big data

Hybrid cloud

Beacons

Bots

Artificial Intelligence

Threat Intelligence

Ambient UX

What we hear from developers

I need to create applications at a competitive rate without worrying about IT

New applications run smoothly on my machine but malfunction on traditional IT servers

My productivity and application innovation become suspended when I have to wait on IT

What we hear from IT

I need to manage servers and maintain compliance with little disruption

I’m unsure of how to integrate unfamiliar applications, and I require help from developers

I’m unable to focus on both server protection and application compliance

IT stress points

Security threats

Datacenter efficiency

Supporting innovation

Cloud is a new way to think about a datacenter

Traditional model

Dedicated infrastructure for each application

Purpose-built hardware

Distinct infrastructure and operations teams

Customized processes and configurations

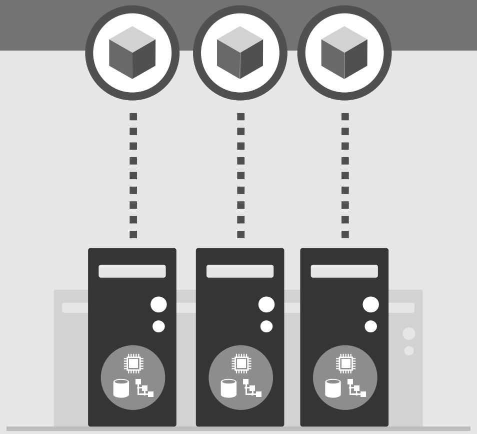
Cloud model

Loosely coupled apps and micro-services

Industry-standard hardware

Service-focused DevOps teams

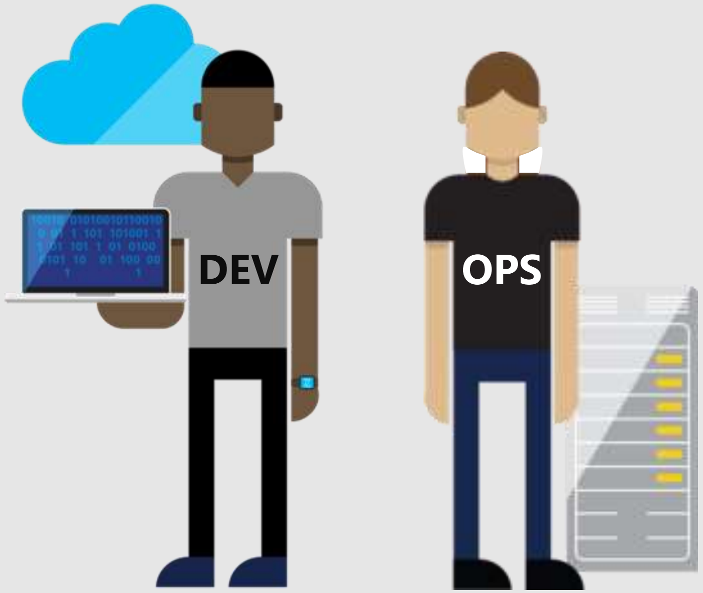
Standardized processes and configurations

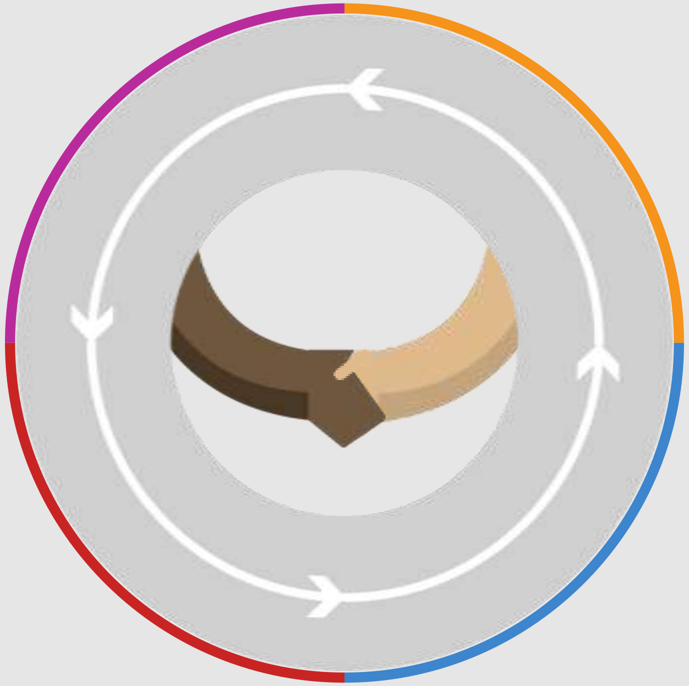


Servers

Services

DevOps: the three stage conversation





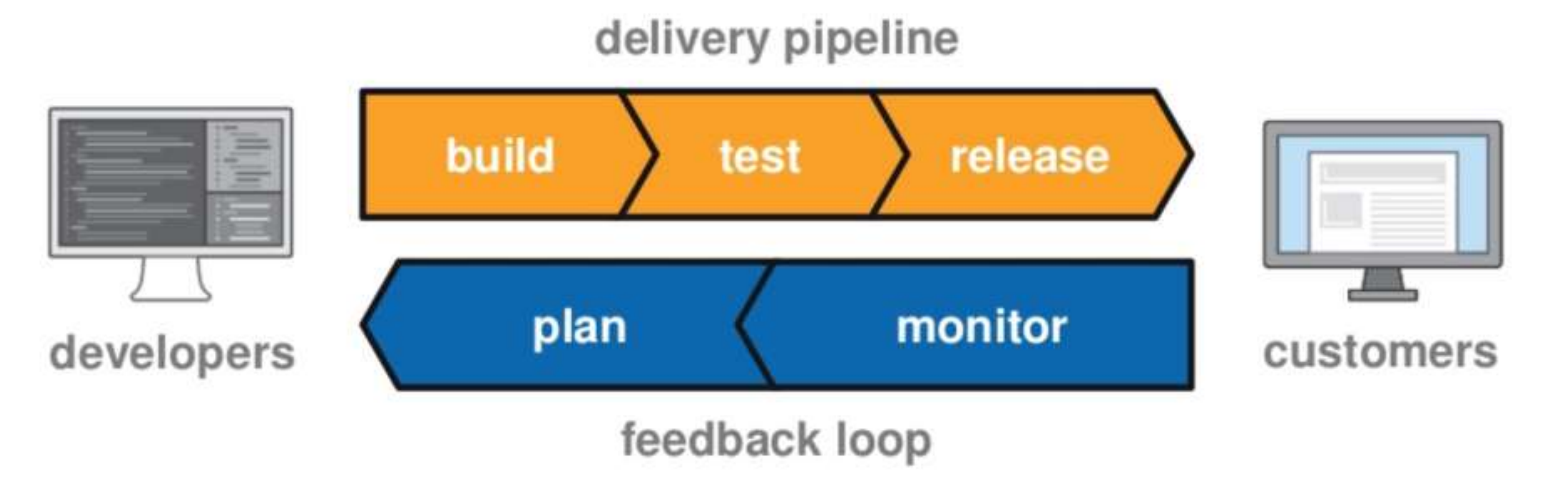
1 People

2 Process

3 Products

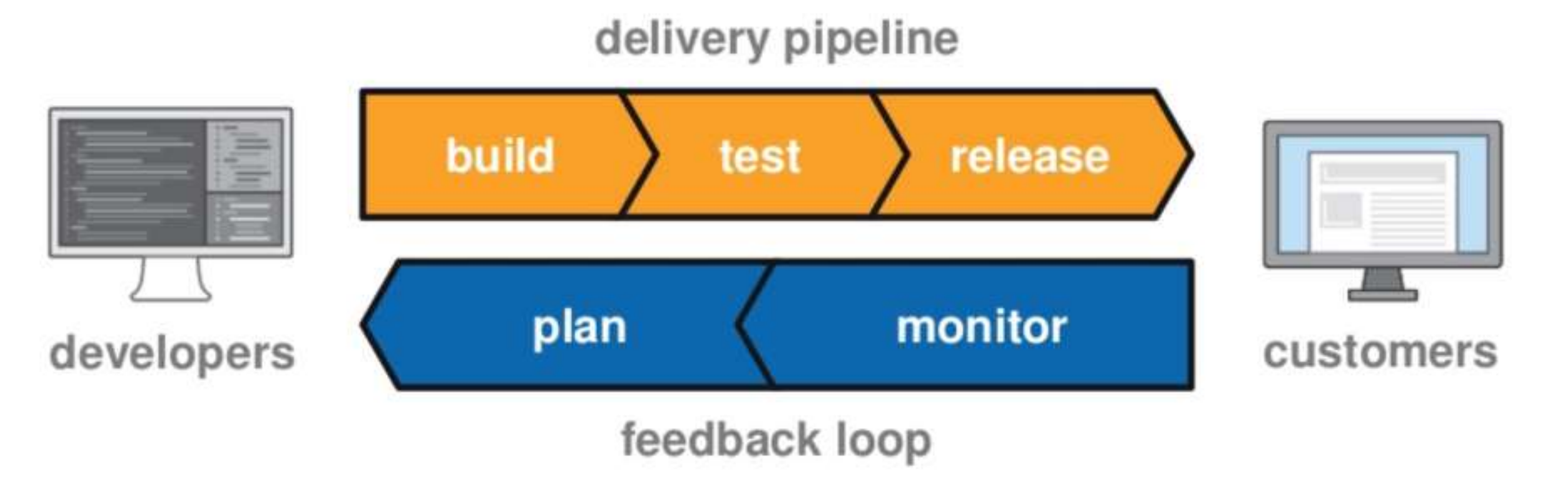
What is DevOps?

Software Development Lifecycle



What is DevOps?

Software Development Lifecycle

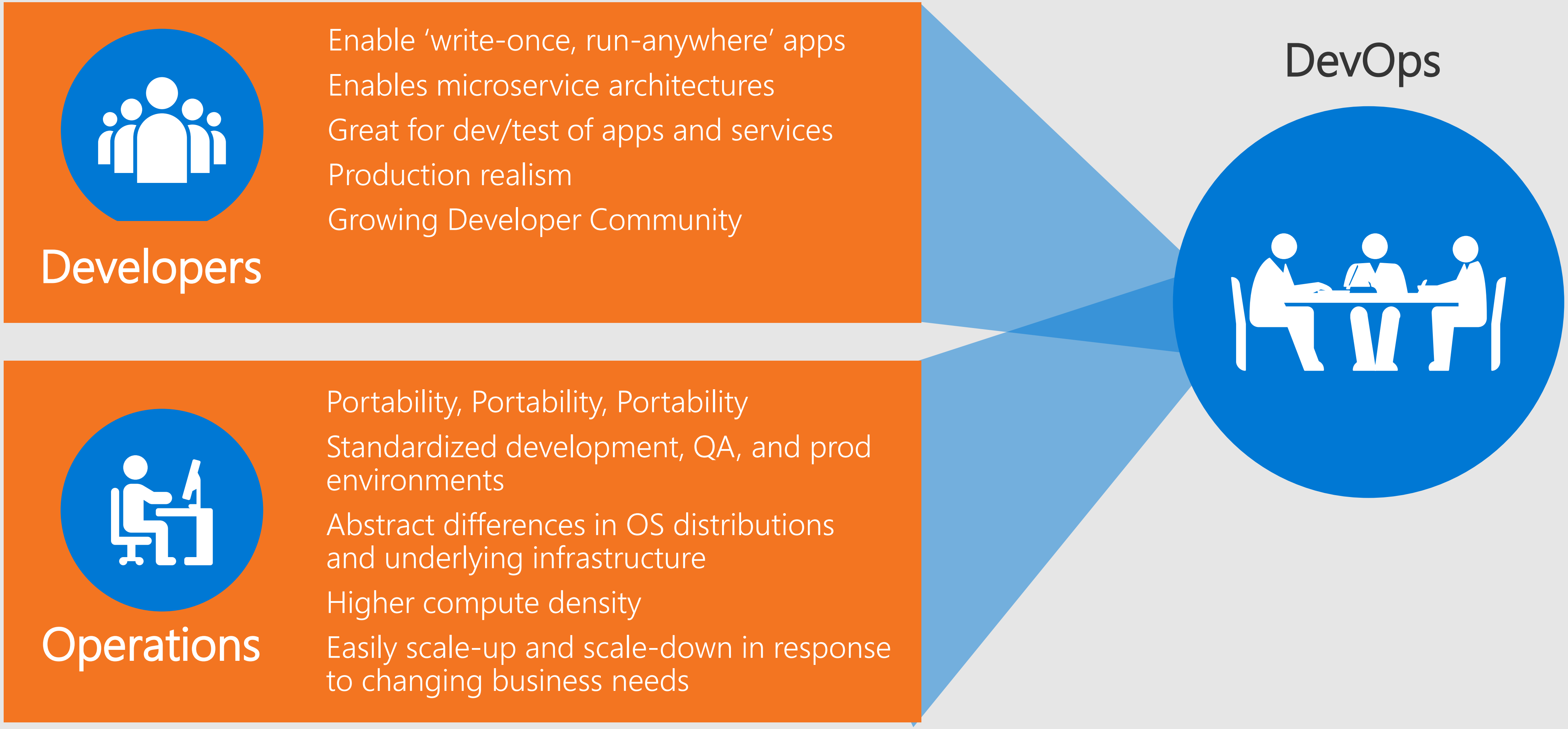


DevOps = efficiencies that speed up this lifecycle

Key DevOps Practices

Infrastructure as Code	Continuous Integration	Continuous Deployment
Automated Testing	Release Management	Performance Monitoring
Availability Monitoring	**Load Testing & Auto Scale**	**Automated Recovery (Rollback & Roll Forward)**

Why Containers?



Developers

- Enable ‘write-once, run-anywhere’ apps
- Enables microservice architectures
- Great for dev/test of apps and services
- Production realism
- Growing Developer Community

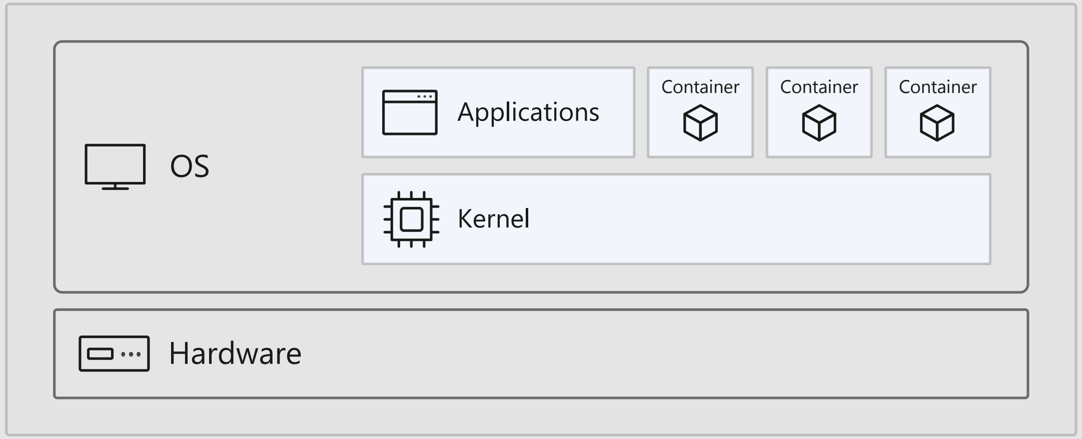
Operations

- Portability, Portability, Portability
- Standardized development, QA, and prod environments
- Abstract differences in OS distributions and underlying infrastructure
- Higher compute density
- Easily scale-up and scale-down in response to changing business needs

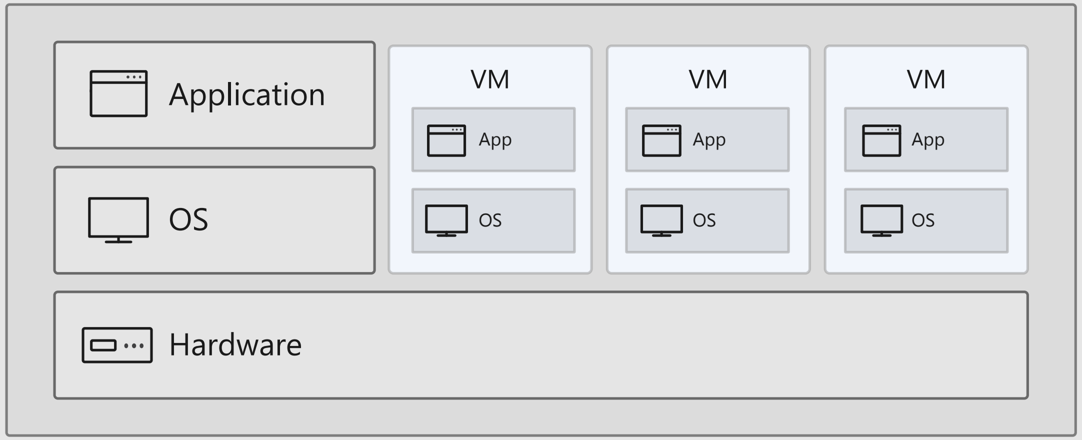
DevOps

What is a container?

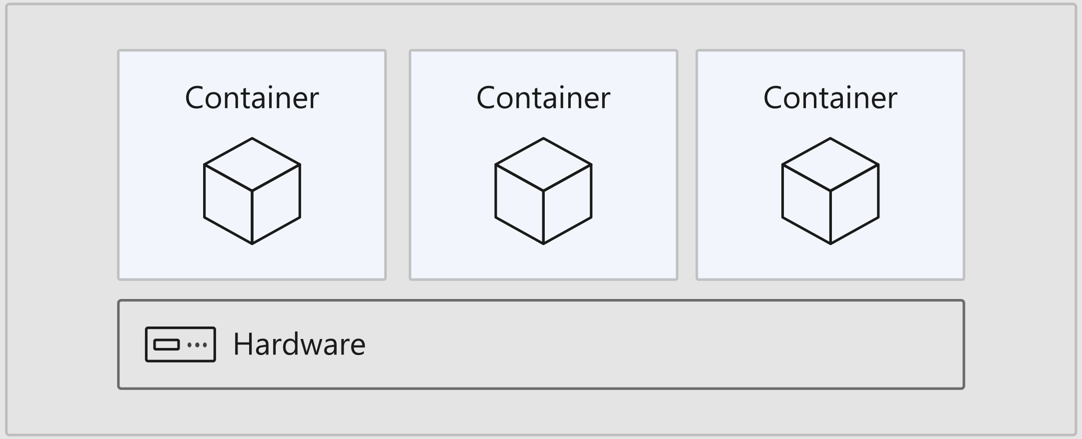
Containers = operating system virtualization



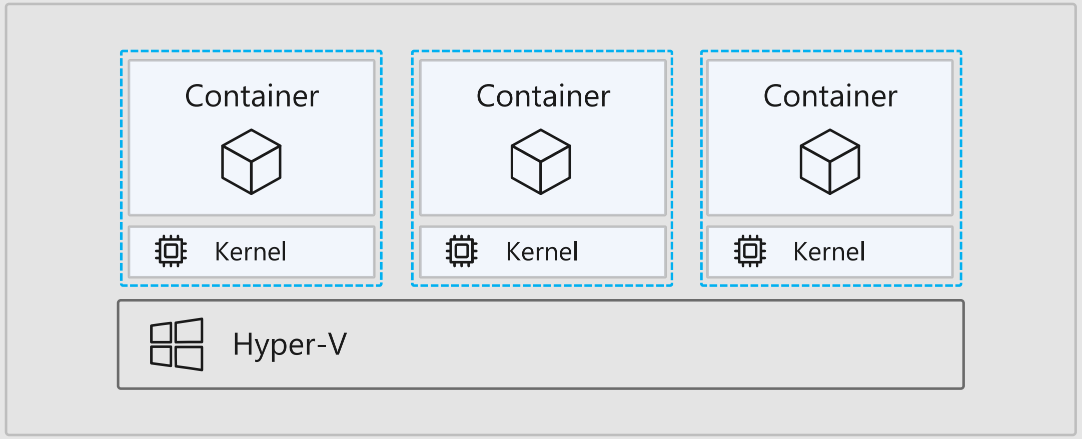
Traditional virtual machines = hardware virtualization



Windows Server containers: maximum speed and density



Hyper-V containers: isolation plus performance



What is a container?

Not a real thing. An application delivery mechanism with **process isolation** based on several **Linux kernel** features.

Namespaces(what a process can see)

- PID
- Mount
- Network
- UTS
- IPC
- User
- Cgroup

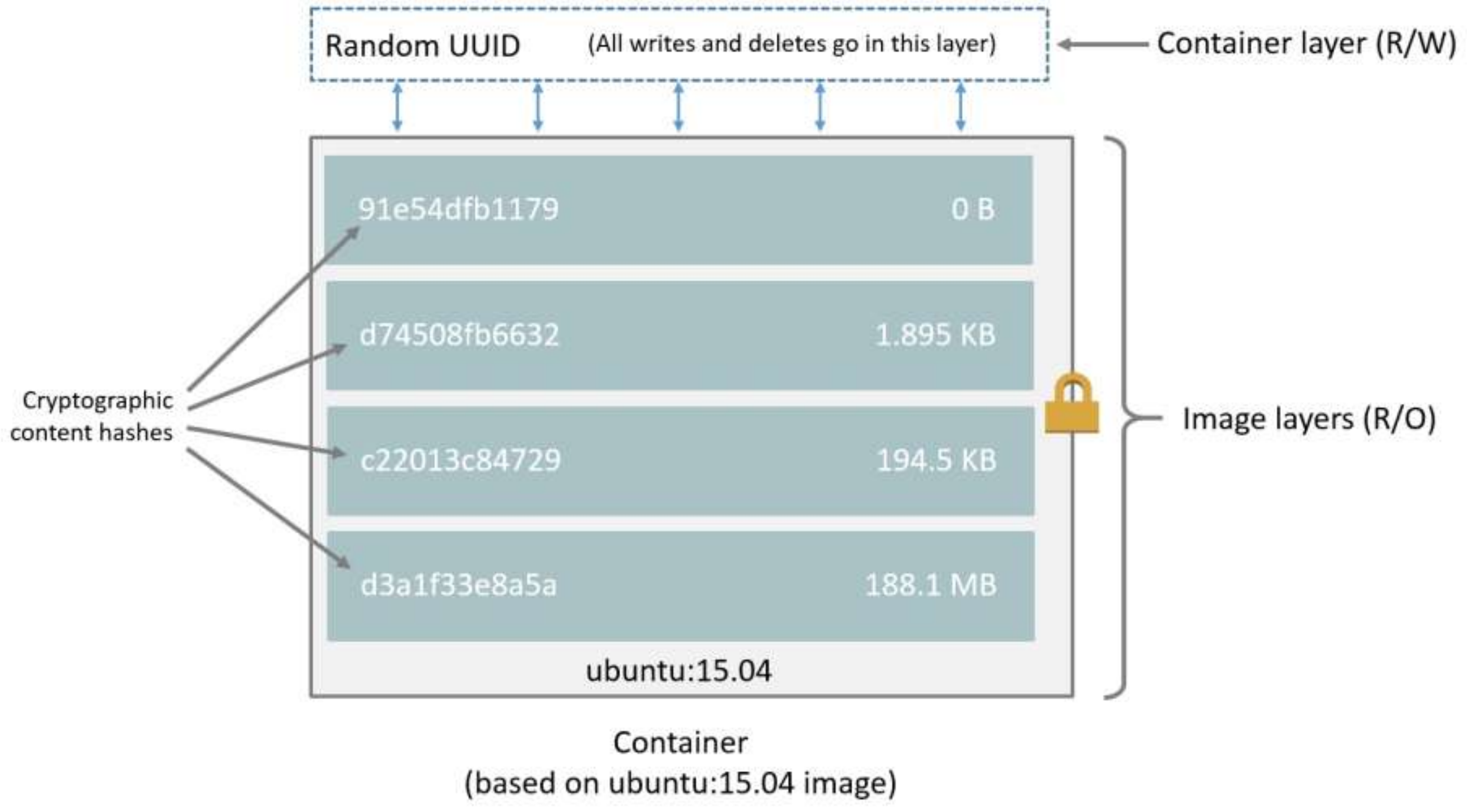
Cgroups (what a process can use)

- Memory
- CPU
- Blkio
- Cpuacct
- Cpuset
- Devices
- Net_prio
- Freezer

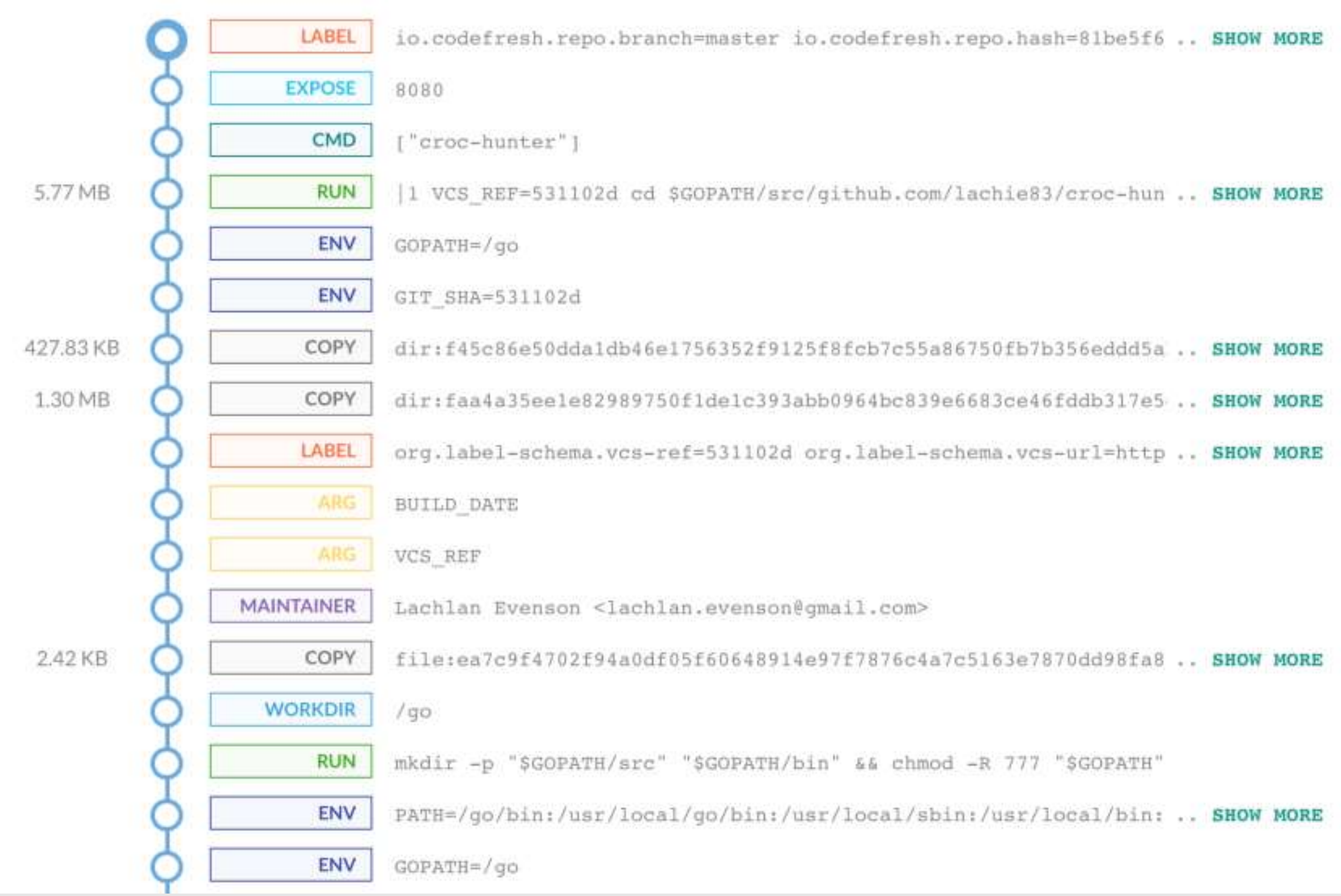
What is docker ?

- An open source container runtime
- Mac, Windows & Linux support
- Command line tool
- “Dockerfile” file format for building container images
- The Docker image format with layered filesystem

Docker Layered Filesystem

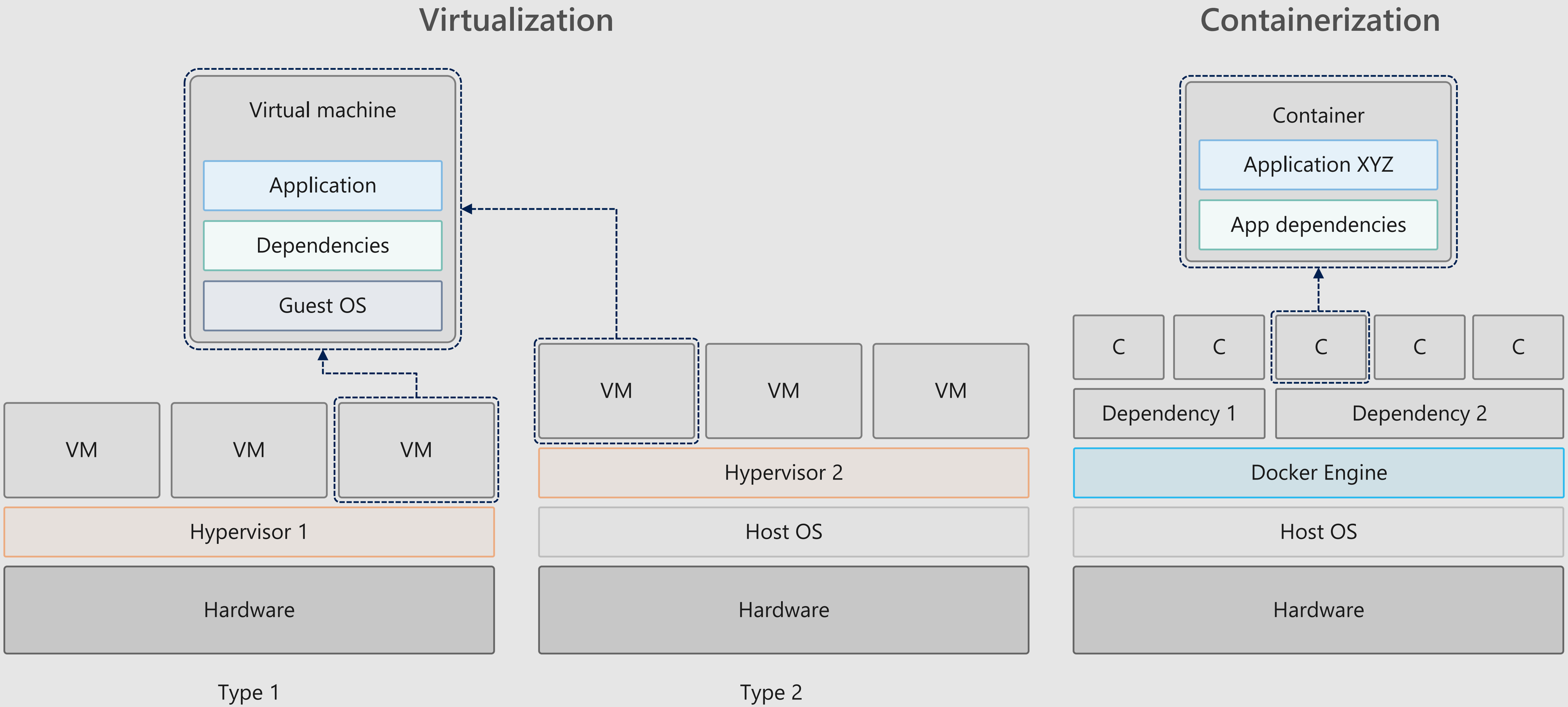


Docker Layered Filesystem

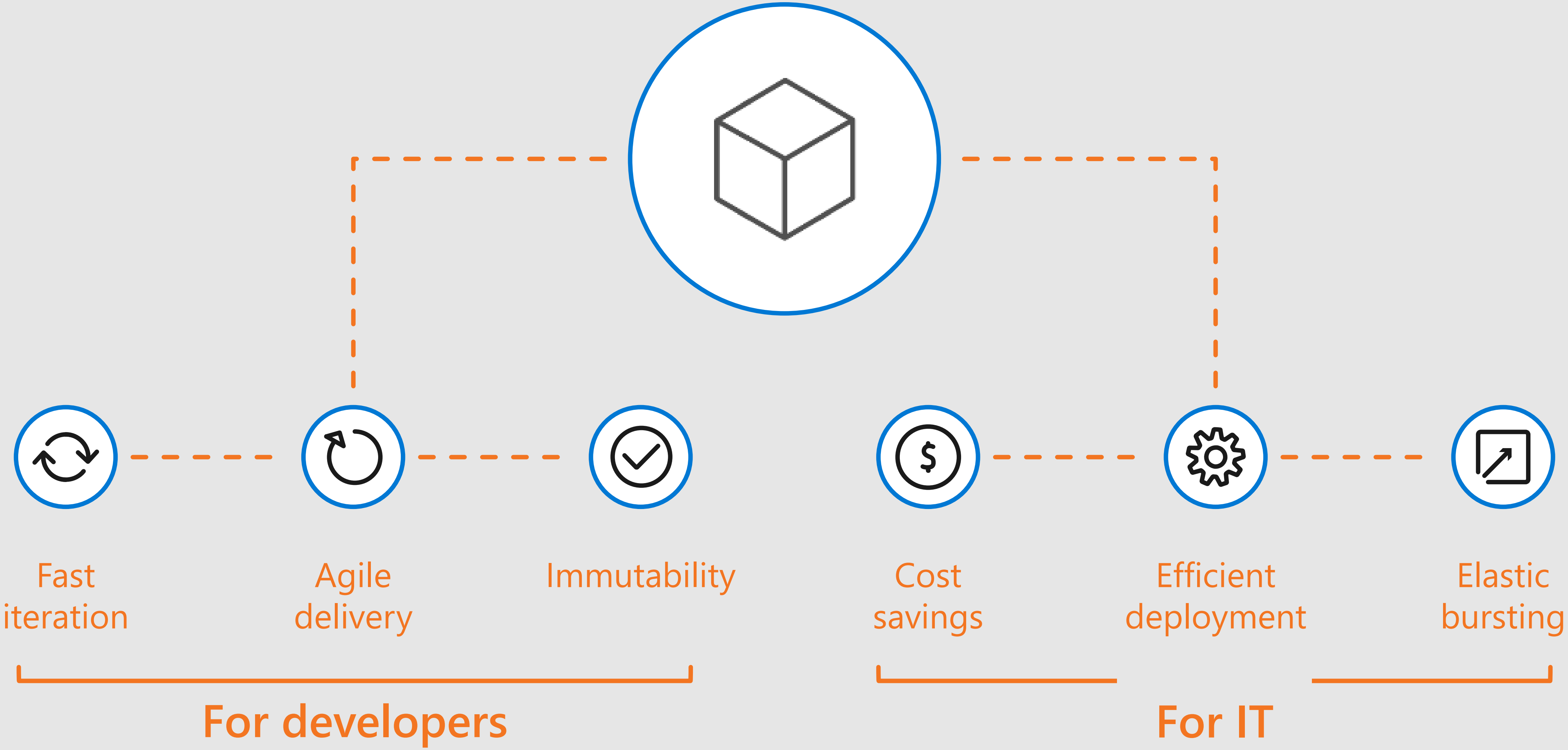


	LABEL	io.codefresh.repo.branch=master io.codefresh.repo.hash=81be5f6 ..	SHOW MORE
	EXPOSE	8080	
	CMD	["croc-hunter"]	
5.77 MB	RUN	|1 VCS_REF=531102d cd $GOPATH/src/github.com/lachie83/croc-hun ..	SHOW MORE
	ENV	GOPATH=/go	
	ENV	GIT_SHA=531102d	
427.83 KB	COPY	dir:f45c86e50dda1db46e1756352f9125f8fcb7c55a86750fb7b356eddd5a ..	SHOW MORE
1.30 MB	COPY	dir:faa4a35ee1e82989750f1de1c393abb0964bc839e6683ce46fddb317e5 ..	SHOW MORE
	LABEL	org.label-schema.vcs-ref=531102d org.label-schema.vcs-url=http ..	SHOW MORE
	ARG	BUILD_DATE	
	ARG	VCS_REF	
	MAINTAINER	Lachlan Evenson <lachlan.evenson@gmail.com>	
2.42 KB	COPY	file:ea7c9f4702f94a0df05f60648914e97f7876c4a7c5163e7870dd98fa8 ..	SHOW MORE
	WORKDIR	/go	
	RUN	mkdir -p "$GOPATH/src" "$GOPATH/bin" && chmod -R 777 "$GOPATH"	
	ENV	PATH=/go/bin:/usr/local/go/bin:/usr/local/sbin:/usr/local/bin: ..	SHOW MORE
	ENV	GOPATH=/go	

Virtualization versus containerization



The container advantage



Fast iteration

Agile delivery

Immutability

Cost savings

Efficient deployment

Elastic bursting

For developers

For IT

Demo - Containers

Container Orchestration

The elements of orchestration

Scheduling

Affinity/anti-affinity

Health monitoring

Failover

Scaling

Networking

Service discovery

Coordinated app upgrades

Kubernetes: the de-facto orchestrator

Portable

Public, private, hybrid, multi-cloud

Extensible

Modular, pluggable, hookable, composable

Self-healing

Auto-placement, auto-restart, auto-replication, auto-scaling

Kubernetes: empowering you to do more

- Deploy your applications quickly and predictably
- Scale your applications on the fly
- Roll out new features seamlessly
- Limit hardware usage to required resources only

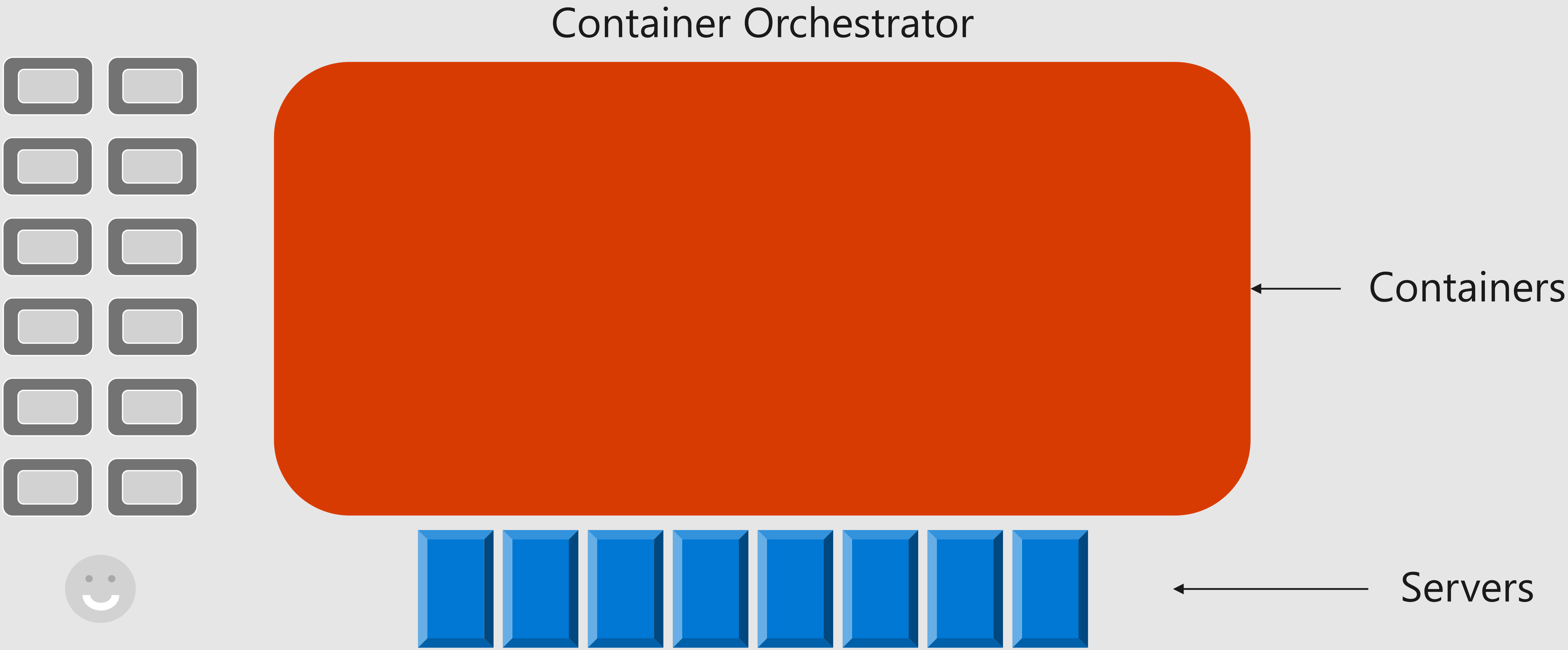
Container Orchestration: Kubernetes

What is Kubernetes?

Open source container orchestrator that automates deployment, scaling, and management of applications

- Features include:
  - Automatic bin-packing
  - Self-healing
  - Horizontal scaling
  - Service discovery and load balancing
  - Automated rollouts and rollbacks
  - Secret and configuration management
  - Storage orchestration
  - Batch execution

Kubernetes - Agility

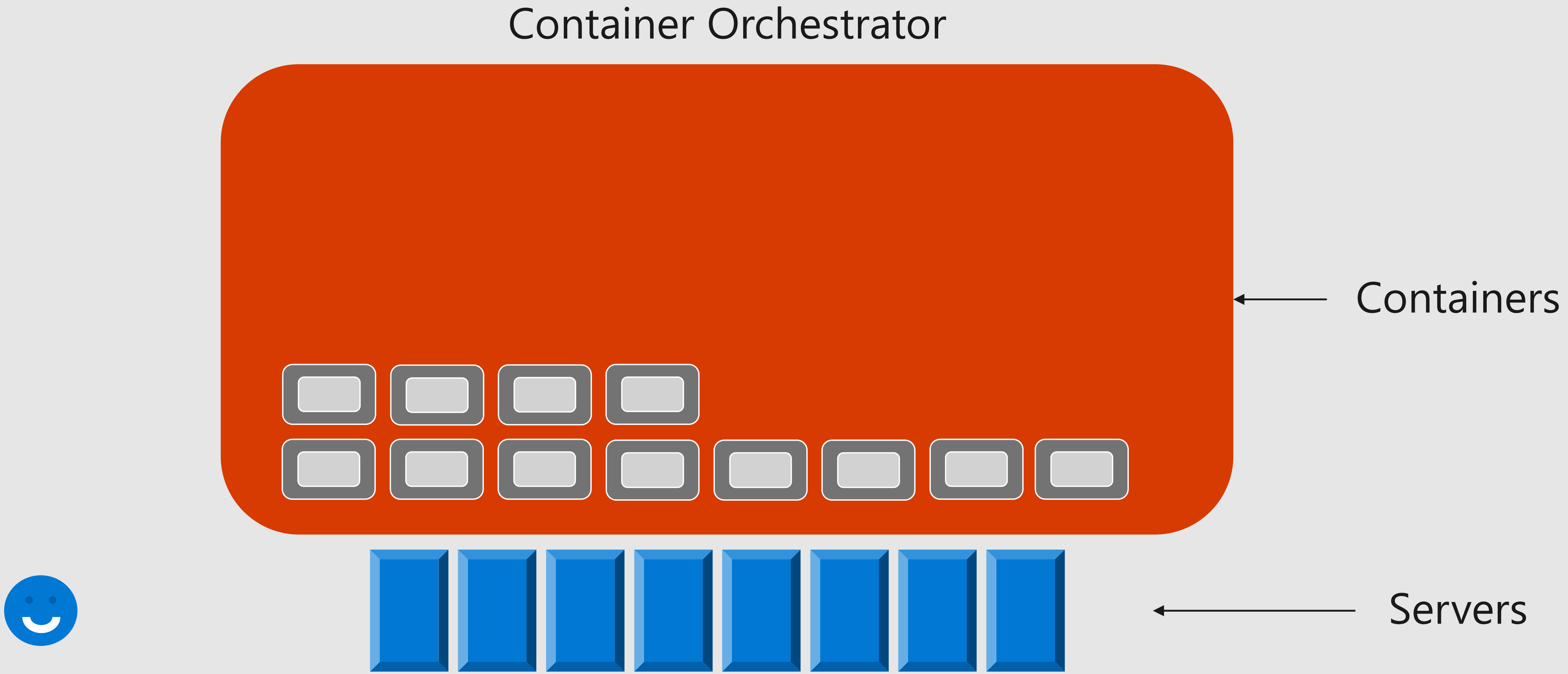


Container Orchestrator

Containers

Servers

Kubernetes - Agility

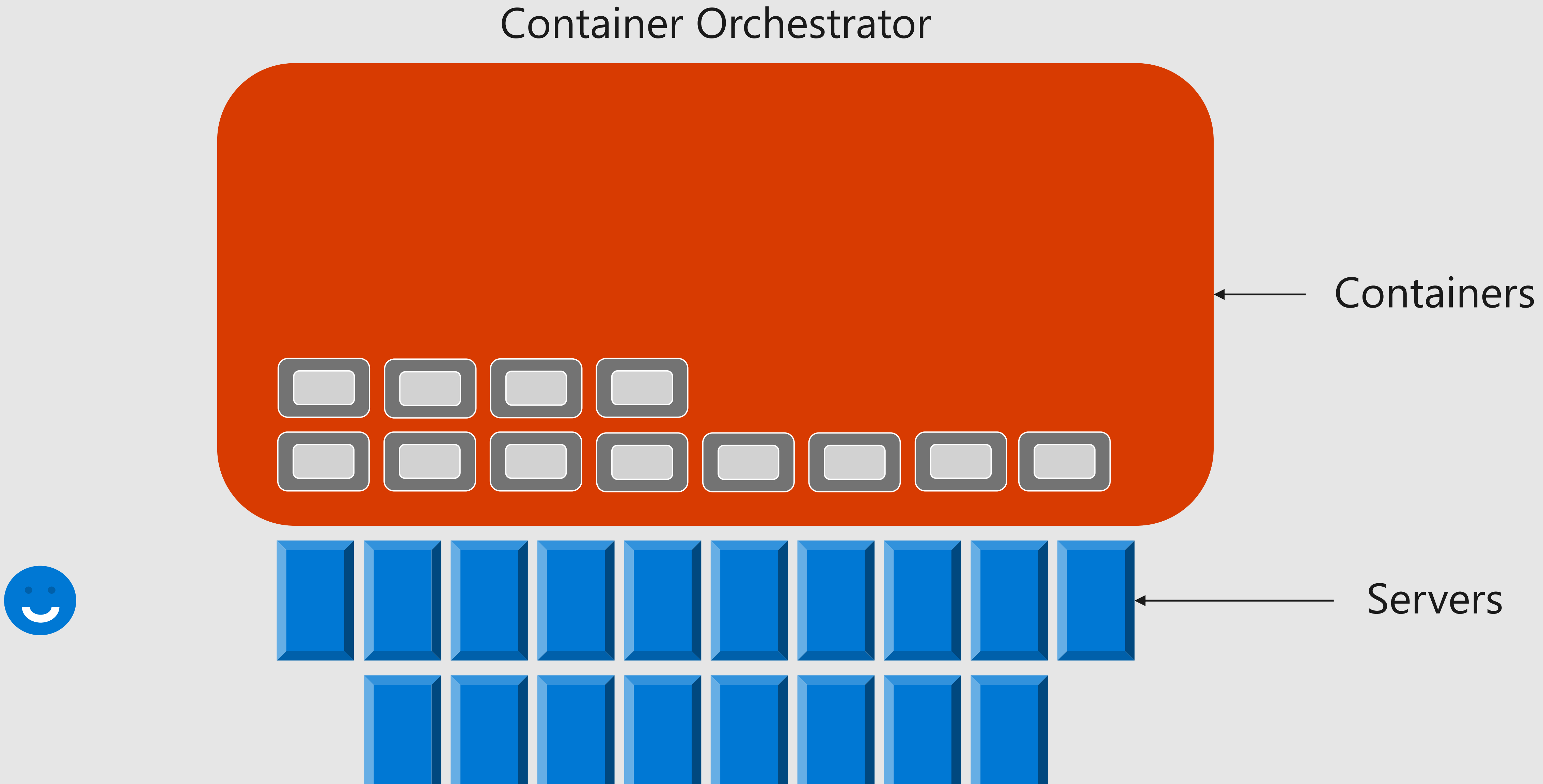


Container Orchestrator

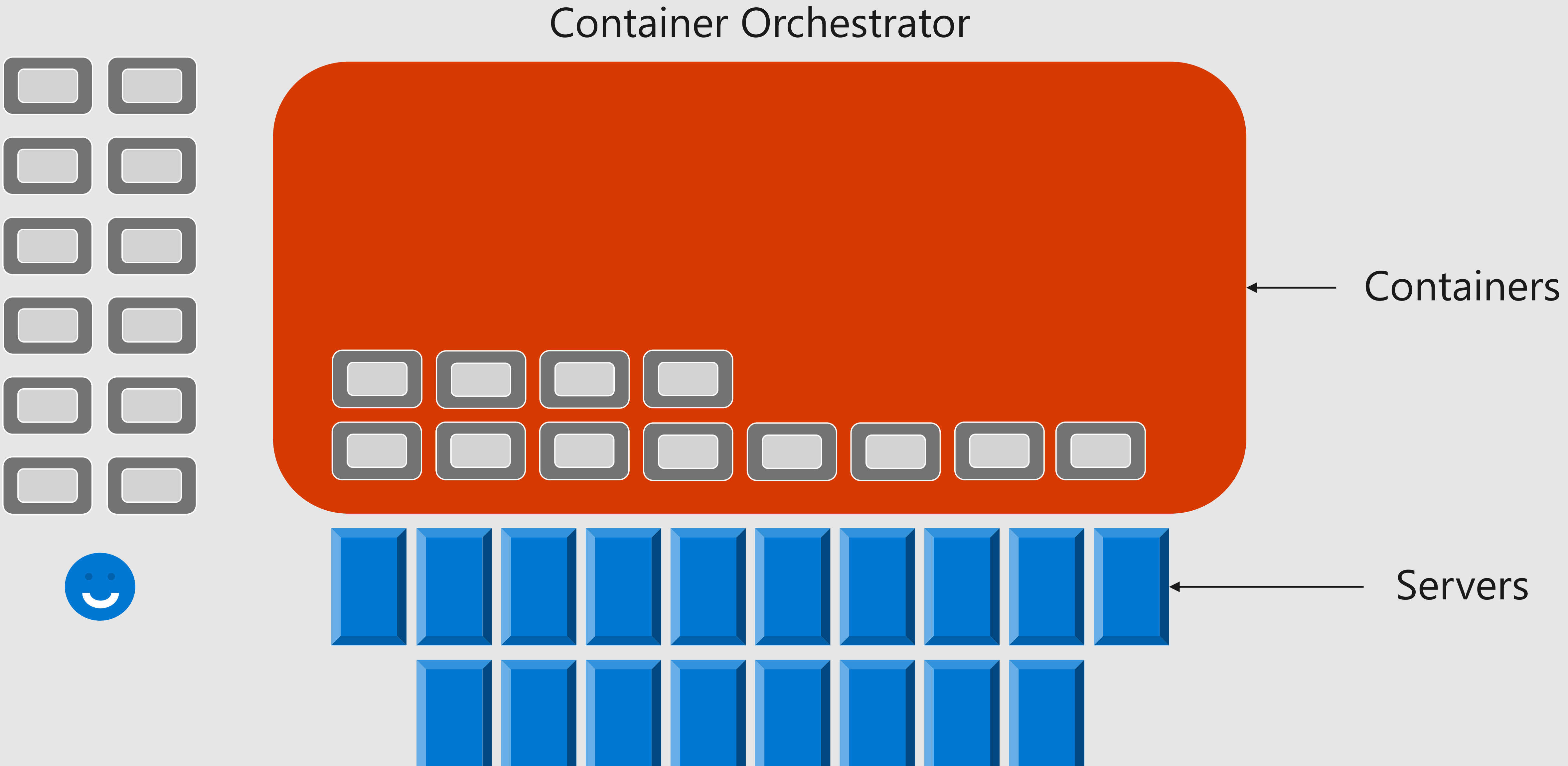
Containers

Servers

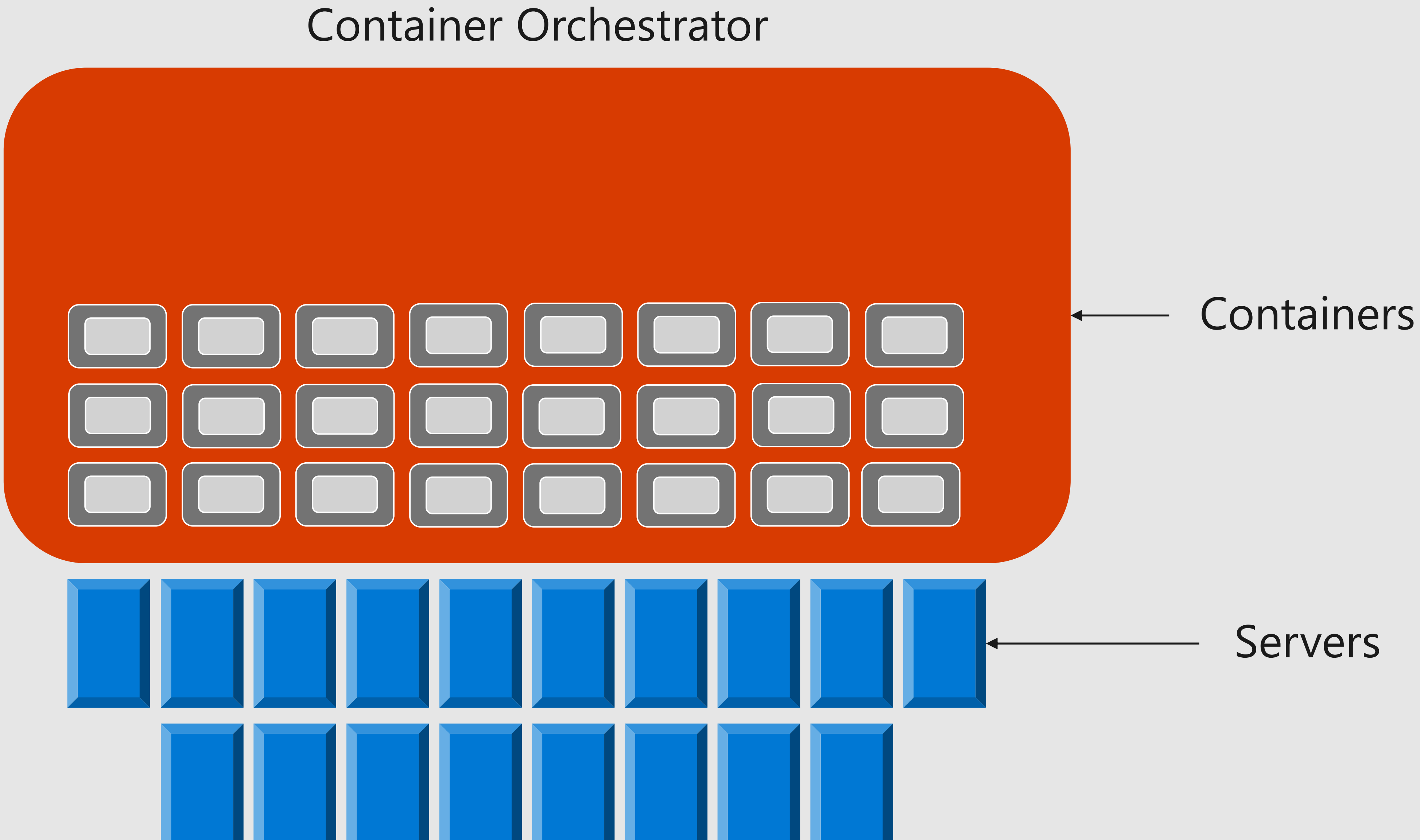
Kubernetes - Scalability



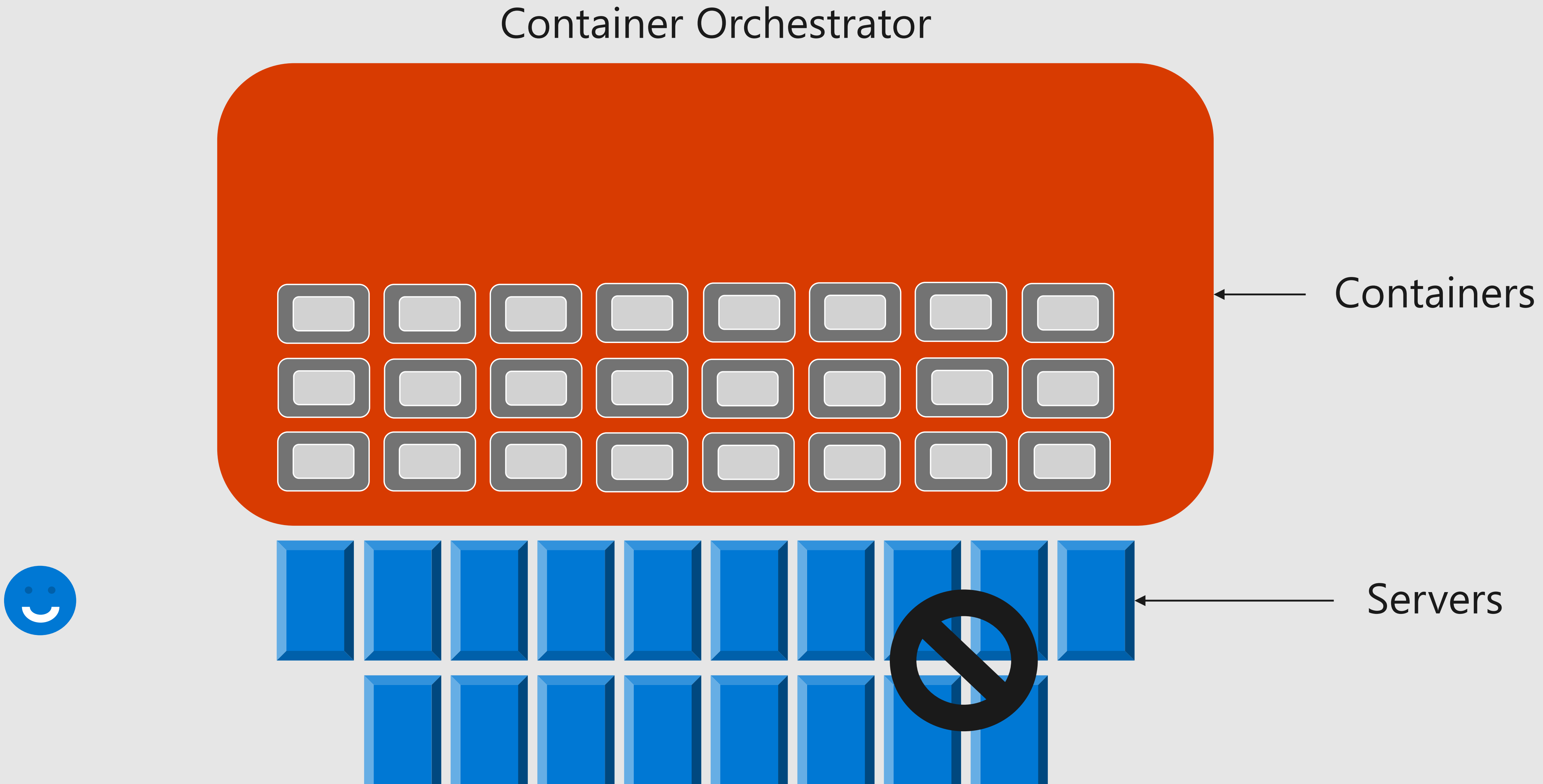
Kubernetes - Scalability



Kubernetes - Scalability



Kubernetes - Reliability



Azure Container Service (AKS)

Your Kubernetes cluster, managed by Azure

Why AKS?

- Easy to use
  - Fastest path to Kubernetes on Azure
  - Up and running with 3 simple commands

- Easy to manage
  - Automated upgrades and patching
  - Easily scale the cluster up and down
  - Self-healing control plane

- Uses Open APIs
  - 100% upstream Kubernetes

Getting Started with AKS

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys
\ Running ..

$ az aks install-cli
Downloading client to /usr/local/bin/kubectl ..

$ az aks get-credentials -g myResourceGroup -n myCluster
Merged "myCluster" as current context ..

$ kubectl get nodes
NAME                       STATUS    AGE       VERSION
aks-mycluster-36851231-0   Ready     4m        v1.8.1
aks-mycluster-36851231-1   Ready     4m        v1.8.1
aks-mycluster-36851231-2   Ready     4m        v1.8.1
```

Managing an AKS cluster

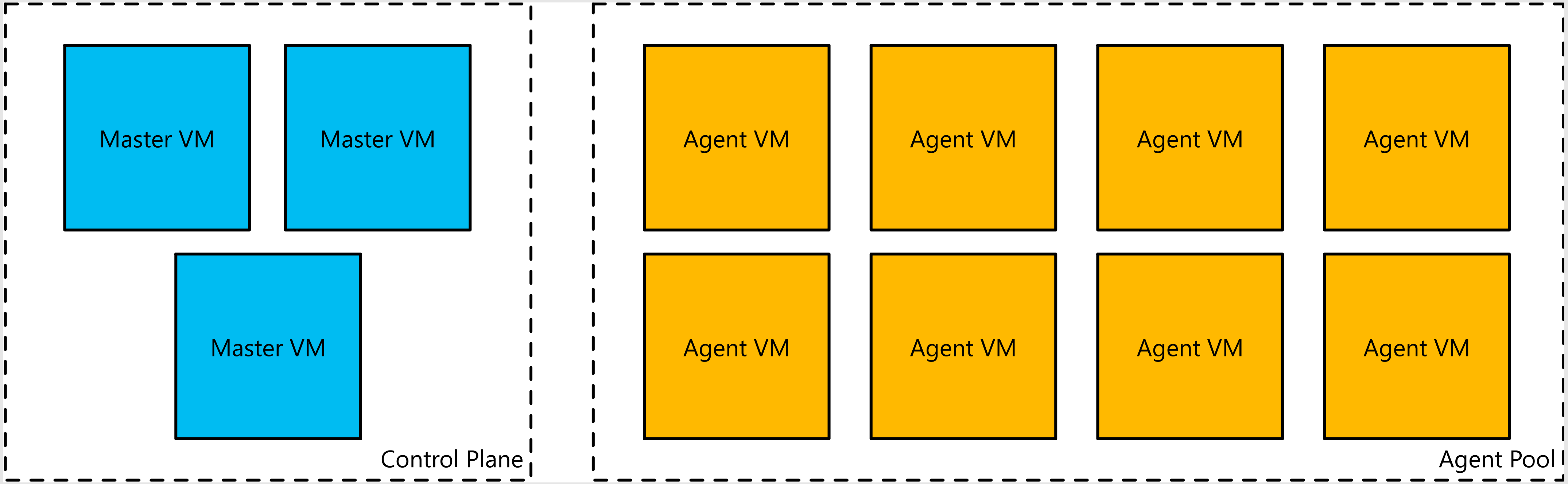
```
$ az aks list –o table
Name                Location    ResourceGroup      KubernetesRelease  ProvisioningState
------------------  ----------  --------------  --------------------  -------------------
myCluster           westus2     myResourceGroup                1.7.7  Succeeded
```

```
$ az aks upgrade -g myResourceGroup -n myCluster –-kubernetes-version 1.8.1
\ Running ..
```

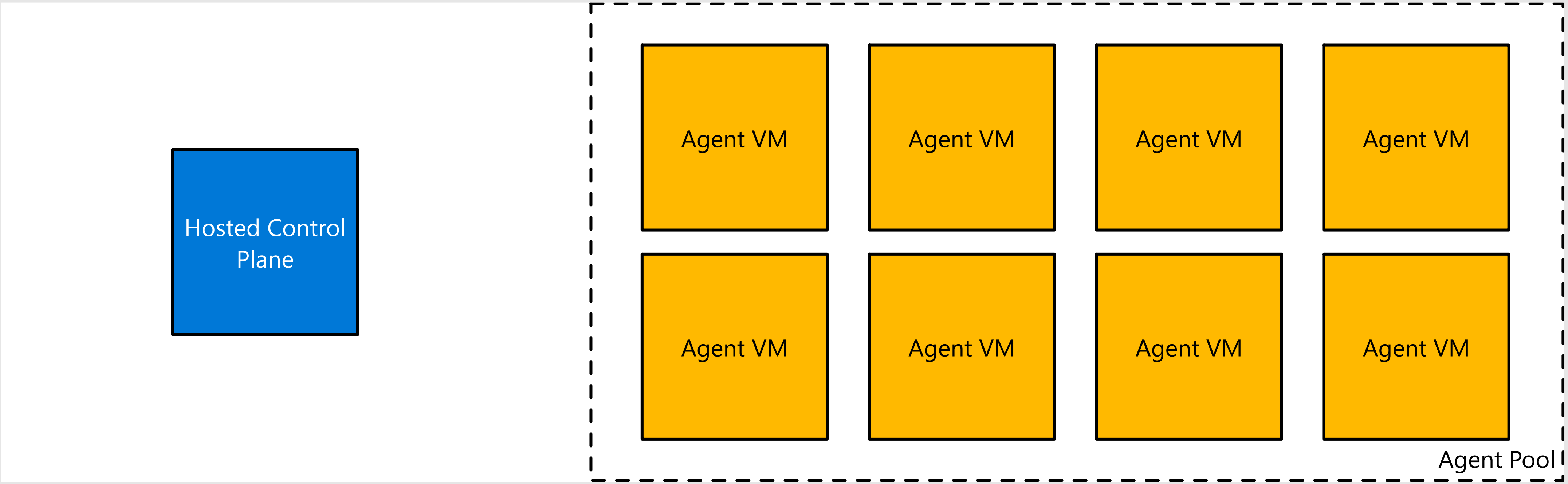
```
$ kubectl get nodes
NAME                       STATUS    AGE       VERSION
aks-mycluster-36851231-0   Ready     12m       v1.8.1
aks-mycluster-36851231-1   Ready     8m        v1.8.1
aks-mycluster-36851231-2   Ready     3m        v1.8.1
```

```
$ az aks scale -g myResourceGroup -n myCluster --agent-count 10
\ Running ..
```

Kubernetes without AKS



Kubernetes with AKS



Hosted Control Plane

Agent VM

Agent VM

Agent VM

Agent VM

Agent VM

Agent VM

Agent VM

Agent VM

Agent Pool

Demo – Azure DevOps Release

MAERSK

Maersk uses AKS for a customer service process to elevate NSAT, an industry-wide challenge

Needs:

- Get near-real-time data to provide better customer service
- Collect data for future Machine Learning driven features

Challenges:

- Compute & memory intensive features
- Data integration difficulties
- Limited organisational experience in Cloud & Kubernetes

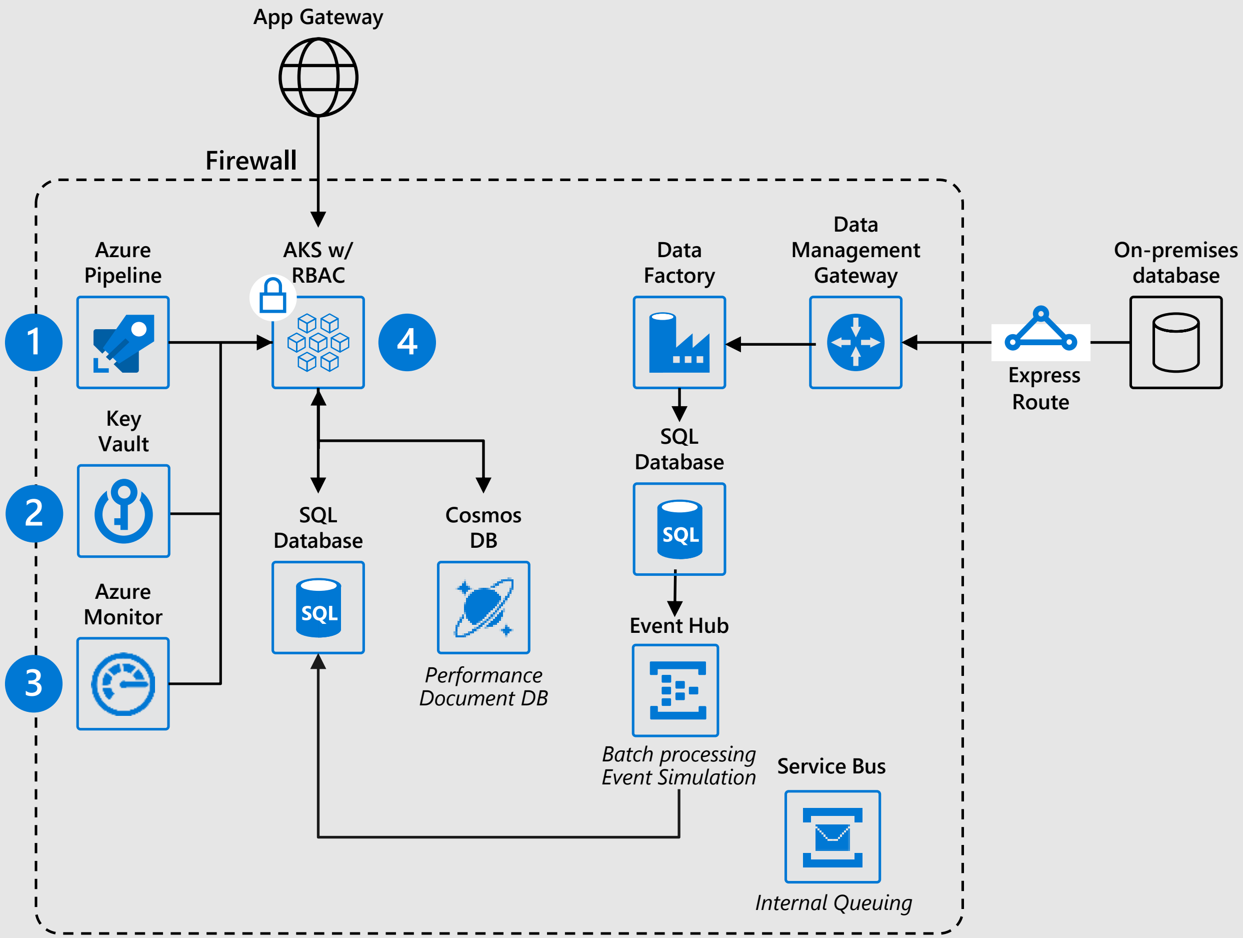
Requirements:

- Spend less time on container software management
- Automation and continuous delivery
- Full visibility to application, container and infrastructure
- Fine grained security and access control

Click icon to learn more

Architectural approach MAERSK

1. Azure Pipelines for automation and CI/CD pipelines; adding Terraform for further automation
2. Key Vault to secure secrets and for persistent configuration store
3. Azure Monitor for containers provides better logging, troubleshooting, with no direct container access
4. RBAC control for fine grained Kubernetes resources access control



Results

MAERSK

Reduced environment provisioning time from **1+** weeks to **2.5** hours

Deploy times reduced to minutes with the introduction of Terraform

Increased developer autonomy with ARM Templates and Terraform

Less time spend on managing secrets with AKS and Key Vault

AKS and CaaS can potentially save **33%** on run cost

100% automated production deployments

Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Release automation tools

Simplifying the Kubernetes experience

DRAFT

Streamlined Kubernetes development

HELM

The package manager for Kubernetes

BRIGADE

Event-driven scripting for Kubernetes

KASHTI

Visualization dashboard for Brigade

Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Helm

The best way to find, share, and use software built for Kubernetes

Manage complexity

Charts can describe complex apps; provide repeatable app installs, and serve as a single point of authority

Easy updates

Take the pain out of updates with in-place upgrades and custom hooks

Simple sharing

Charts are easy to version, share, and host on public or private servers

Rollbacks

Use `helm rollback` to roll back to an older version of a release with ease

Azure Container Service (AKS)

Azure Container Instances (ACI)

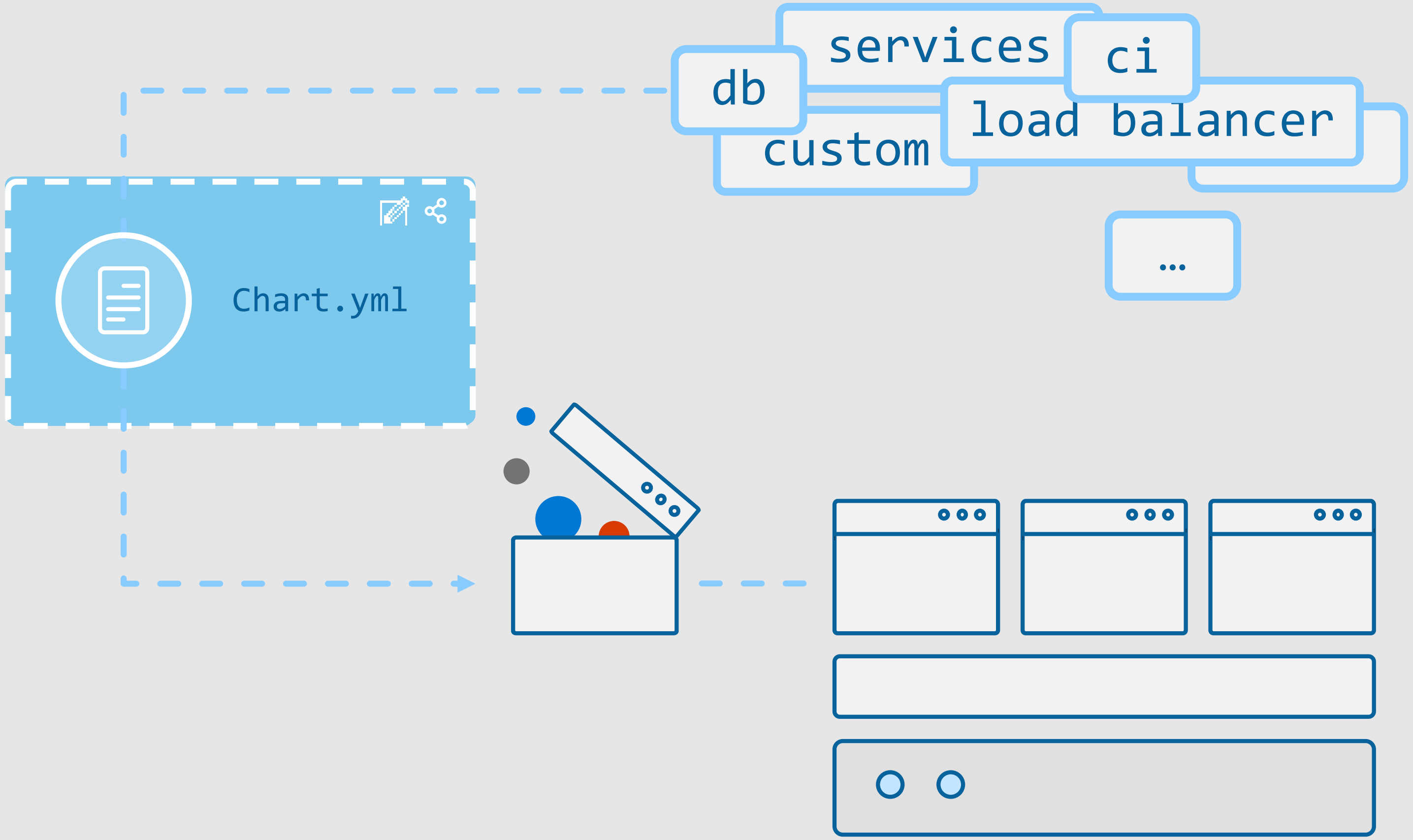
Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Helm

Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application



Demo – Draft and Helm

Introducing Azure DevOps

Azure Boards

Kanban Boards, Backlogs, Dashboards, and Reporting

Azure Pipelines

CI/CD platform, FREE for open source projects

Azure Repos

Unlimited, Cloud-Hosted Private Git Repos

Azure Test Plans

Manual and Exploratory Testing Tools

Azure Artifacts

Package Management for Maven, npm, and NuGet

https://azure.com/devops

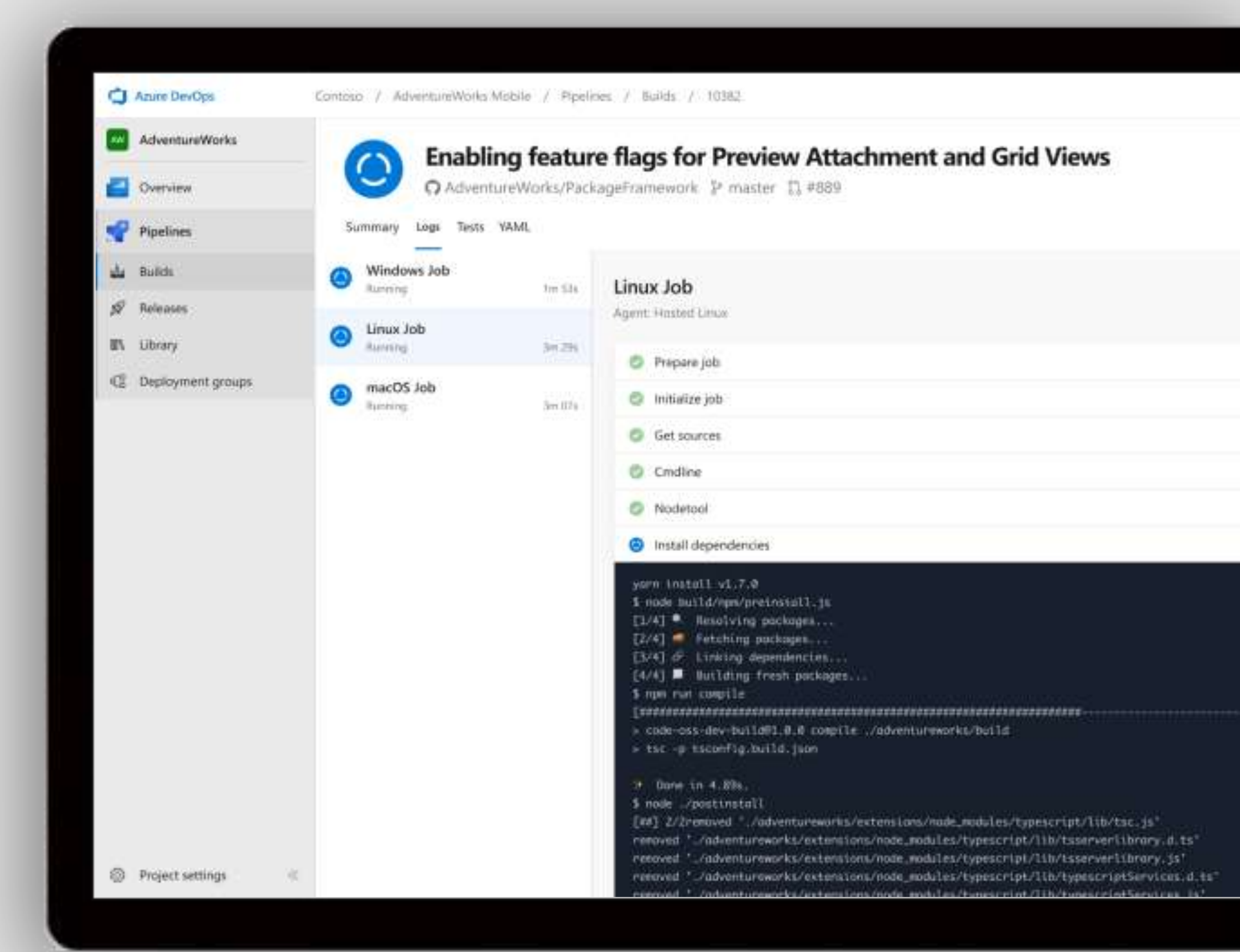
Azure DevOps – Get started for FREE

Open source

- Unlimited public Git repos
- Work item tracking and Kanban boards
- 10 FREE parallel jobs and unlimited build minutes for CI/CD

Private Projects (up to 5 users)

- Unlimited private Git Repos
- Work item tracking and Kanban boards
- 1 job with 1,800 minutes per month for CI/CD



Enabling feature flags for Preview Attachment and Grid Views

AdventureWorks/PackageFramework master #889

Windows Job Running 1m 53s

Linux Job Running 3m 29s

macOS Job Running 3m 07s

Linux Job

Agent: Hosted Linux

- Prepare job
- Initialize job
- Get sources
- Cmdline
- Nodetool
- Install dependencies

```
yarn install v1.7.0
$ node build/npm/preinstall.js
[1/4] 🔍  Resolving packages...
[2/4] 🚚  Fetching packages...
[3/4] 🔗  Linking dependencies...
[4/4] ⬜  Building fresh packages...
$ npm run compile
[##########################################################################----
> code-oss-dev-build@1.0.0 compile ./adventureworks/build
> tsc -p tsconfig.build.json

✨  Done in 4.89s.
$ node ./postinstall
[##] 2/2removed './adventureworks/extensions/node_modules/typescript/lib/tsc.js'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.js'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.js'
```

https://azure.com/devops

DevOps at Microsoft

Azure DevOps is the toolchain of choice for Microsoft internal engineering with over 90,000 internal users

➔ https://aka.ms/DevOpsAtMicrosoft

372k
Pull Requests per month

4.4m
Builds per month

5m
Work items viewed per day

2m
Git commits per month

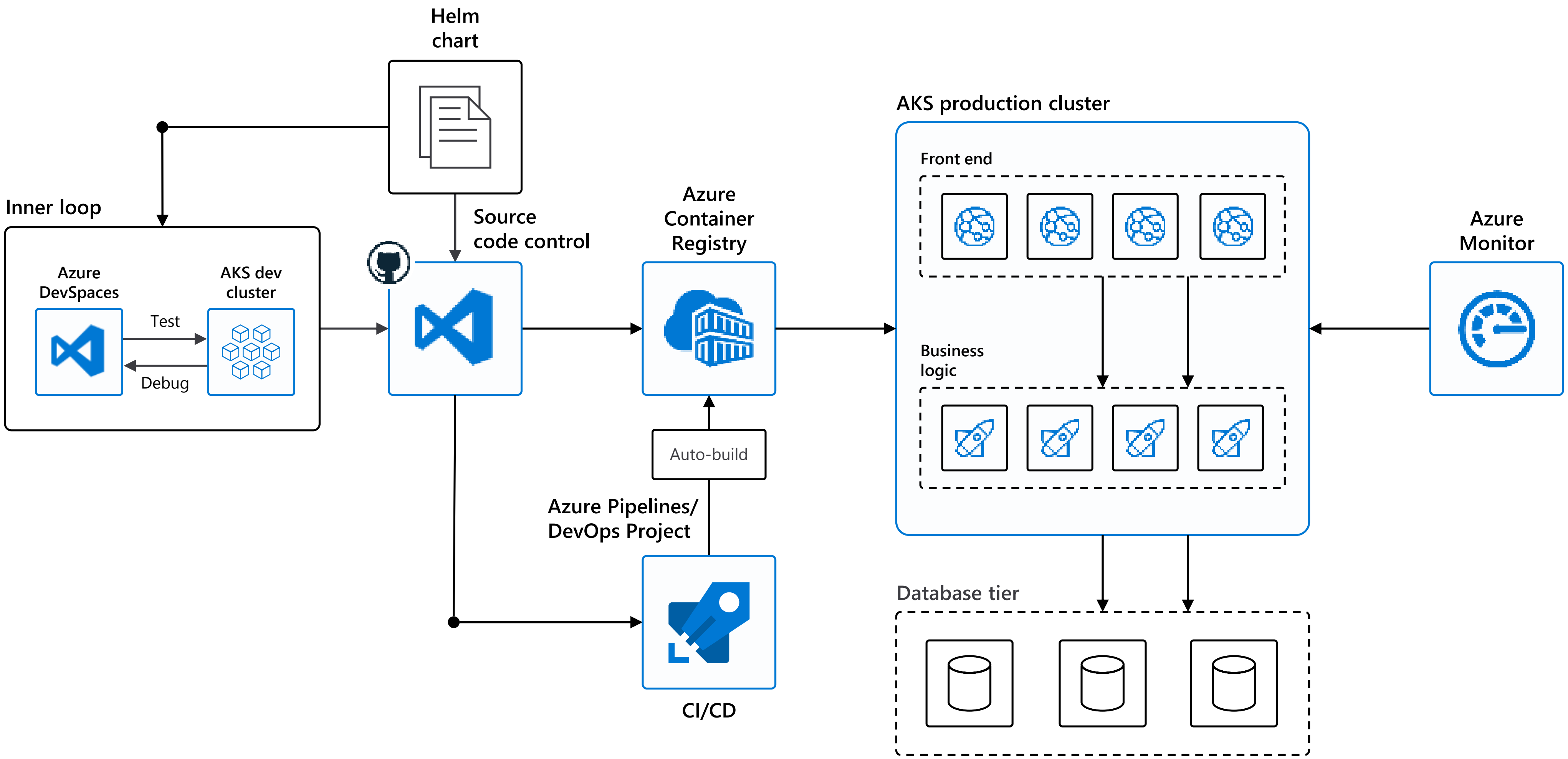
500m
Test executions per day

500k
Work items updated per day

78,000
Deployments per day

Data: Internal Microsoft engineering system activity, August 2018

End to end experience



Demo – Azure DevOps Project

Azure makes Kubernetes easy

Accelerate containerized application development

Task	The Old Way	With Azure
Build a containerized app and deploy to Kubernetes	Build the app Write a Dockerfile Build the container image Push the container to a registry Write Kubernetes manifests/Helm chart Deploy to Kubernetes	`draft init` to configure your environment `draft create` to auto-create Dockerfile/Helm chart `draft up` to deploy to Kubernetes
Build a containerized app and deploy to Kubernetes	Set up a local dev environment using Minikube Determine the transitive closure of your dependencies Identify behavior of dependencies for key test cases Stub out dependent services with expected behavior Make local changes, check-in, and hope things work Validate with application logs	Use DevSpaces Do breakpoint debugging in your IDE
Expose web apps to the internet with a DNS entry	Deploy an ingress controller Create a load-balanced IP for it Add an ingress resource to your deployment Acquire a custom domain Create a DNS A-record for your service	Turn HTTP application routing on in your cluster Add an ingress resource to your deployment

Azure makes Kubernetes easy

Set up CI/CD in a few clicks

Task	The Old Way	With Azure
Set up a CI/CD pipeline and deploy to Kubernetes	Create git repo Create a build pipeline Create a container registry Create a Kubernetes cluster Configure build pipeline to push to container registry Configure build pipeline to deploy to Kubernetes	Create an Azure DevOps project with AKS as a target
Make container images available for deployment worldwide	Create a container registry in every region Configure build pipeline with multiple endpoints Loop through all regions and push following build	Create an Azure Container Registry with geo-replication Push your image to a single endpoint
Track health with consolidated cluster and application logs	Choose a logging solution Deploy log stack in your cluster or provision a service Configure and deploy a logging agent onto all nodes	Checkbox “container monitoring” in the Azure portal

5 Kubernetes Best Practices

- Build small containers
- Application architecture
  - **Use Namespaces**
  - **Use helm charts**
  - **RBAC**
- Implement Health checks
- Set requests and limits
- Be mindful of your services
  - **Map external services**
  - **Don’t rely on load balancers**

Microsoft ♡

Questions?

Secure your Kubernetes environment

Control access through AAD and RBAC

Safeguard keys and secrets with Key Vault

Secure network communications with VNET and CNI

Compliant Kubernetes service with certifications covering SOC, HIPAA, and PCI

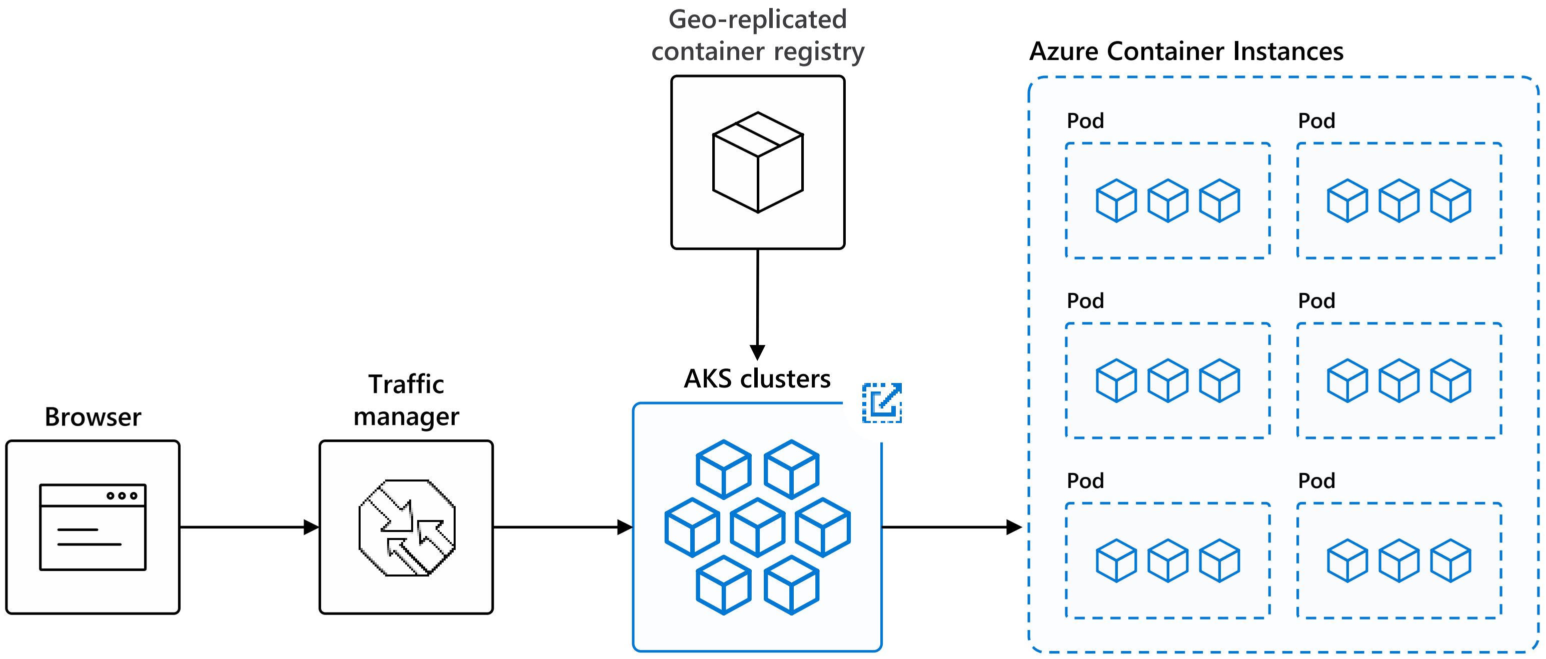
Scale and run with confidence

Built-in
auto scaling

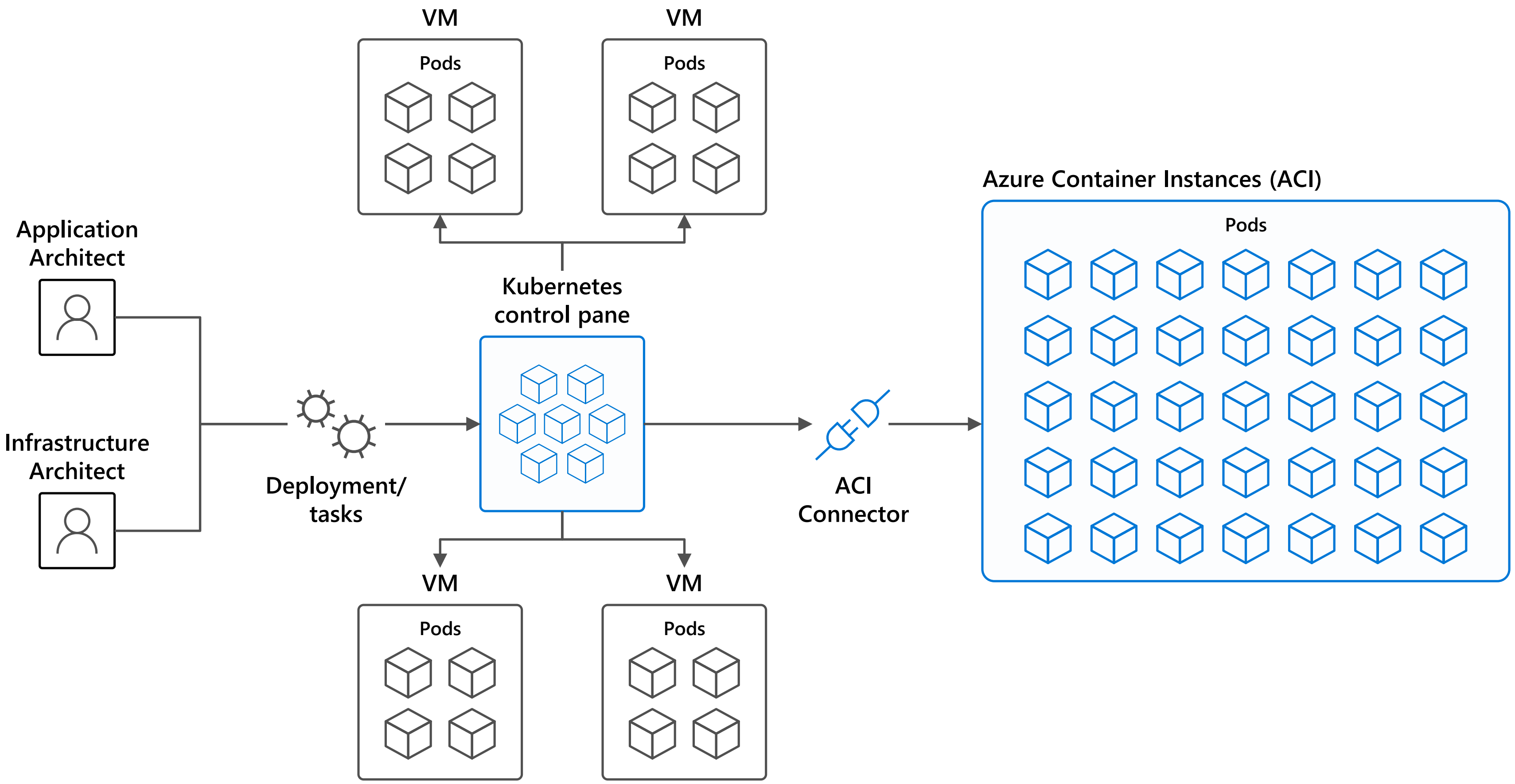
Global
data center

Elastically burst
using ACI

Geo-replicated
container registry



Bursting with the ACI Connector/ Virtual Kubelet


VM
Pods
VM
Pods
Application Architect
Infrastructure Architect
Deployment/ tasks
Kubernetes control pane
ACI Connector
Azure Container Instances (ACI)
Pods
VM
Pods
VM
Pods