



# Microsoft Ignite

Learn.  
Connect.  
Explore.





# Container Fest Pre-day



# Agenda

Time	Session	Presenter / Location
9:00 – 9:10	Welcome	
9:10 – 9:40	What and why of containers	Jason Hansen
9:40 – 11:10	Workshop: Containers 101 with Azure App Services	Andrew Westgarth
11:10 – 11:20	Break	
11:20 – 12:00	Introduction to container orchestration	Jason Hansen
12:00 – 1:00	Lunch	
1:00 – 2:30	Workshop: Zero to hero with Kubernetes on Azure	Jason Hansen
2:30 – 2:40	Break	
2:40 – 4:10	Workshop: Service Fabric Mesh	Mikkel Mork Hegnhoj
4:10 – 4:20	Break	
4:20 – 5:00	Panel discussion and Q&A	

# Panel discussion and Q&A

We want to answer your questions!

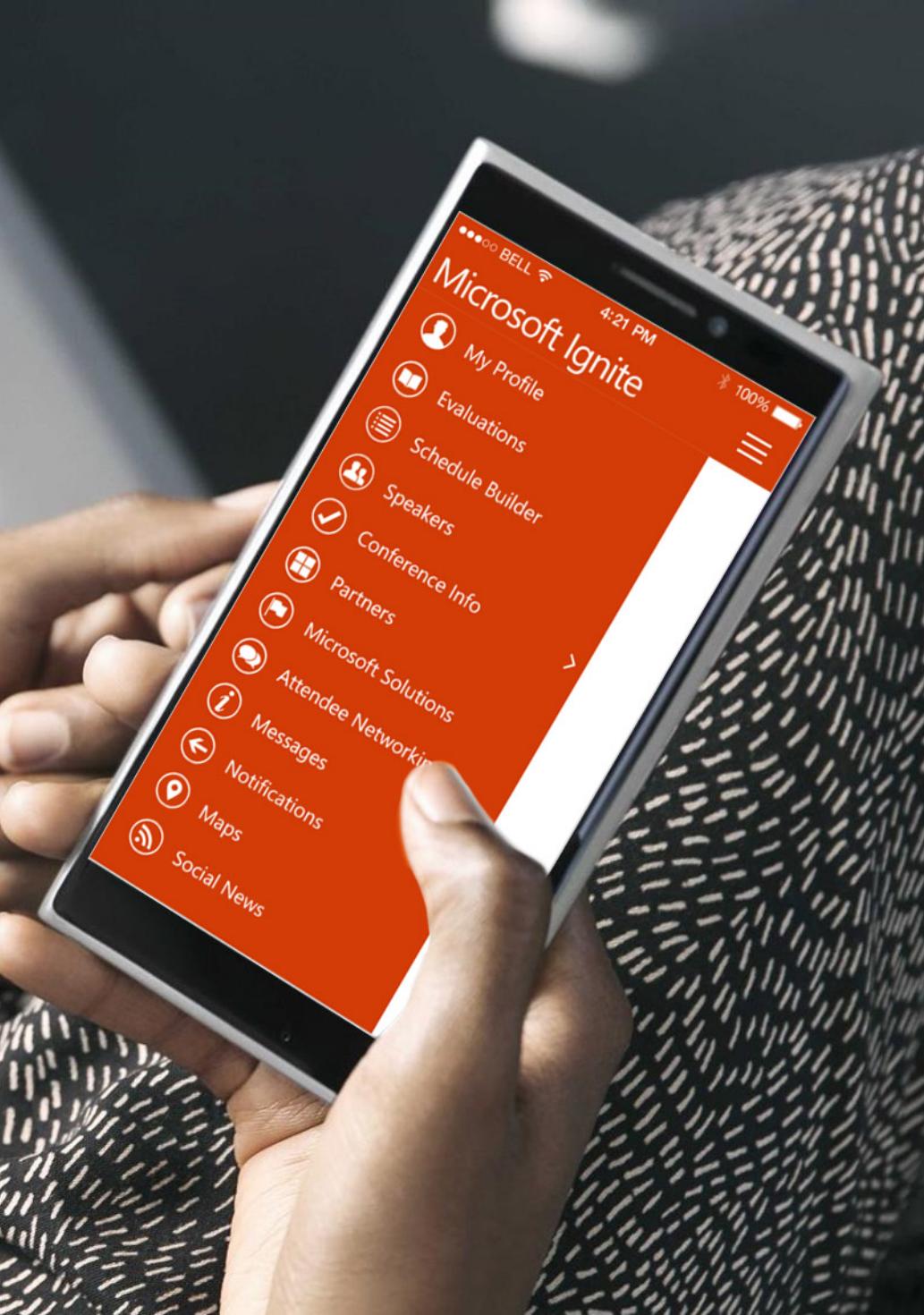
Submit your questions or topics for our afternoon panel at

**<https://aka.ms/containerfest/panel>**

# Azure Pass

If you do not have access to an Azure subscription grab an Azure Pass here:

**<https://aka.ms/IgnitePreDayPasses>**



Please evaluate this session  
Your feedback is important to us!

Please evaluate this session through  
MyEvaluations on the mobile app  
or website.

Download the app:  
<https://aka.ms/ignite.mobileApp>

Go to the website:  
<https://myignite.techcommunity.microsoft.com/evaluations>

# A few questions for you

By show of hands, how many today are...

# A few questions for you

By show of hands, how many today are...

- **New to container technology?**

# A few questions for you

By show of hands, how many today are...

- New to container technology?
- **Evaluating technology and solutions?**

# A few questions for you

By show of hands, how many today are...

- New to container technology?
- Evaluating technology and solutions?
- **Running in production?**

# A few questions for you

By show of hands, how many today are...

**Writing applications for Windows?**

# A few questions for you

By show of hands, how many today are...

Writing applications for Windows?

**Writing applications for Linux?**

# A few questions for you

By show of hands, how many today are...

**Writing applications in .NET?**

# A few questions for you

By show of hands, how many today are...

Writing applications in .NET?

**Writing applications in Java?**

# A few questions for you

By show of hands, how many today are...

Writing applications in .NET?

Writing applications in Java?

**Writing applications in Python, Node, Ruby, PHP?**

# A few questions for you

By show of hands, how many today are...

Writing applications in .NET?

Writing applications in Java?

Writing applications in Python, Node, Ruby, PHP?

**Writing applications in Javascript?**

# A few questions for you

By show of hands, how many today are...

Writing applications in .NET?

Writing applications in Java?

Writing applications in Python, Node, Ruby, PHP?

Writing applications in Javascript?

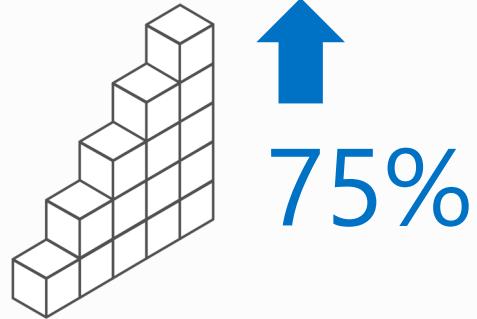
**Writing applications in Go?**

# Containers **momentum**

"By 2020, more than **50%** of enterprises will run **mission-critical, containerized cloud-native applications** in production, up from less than 5% today."

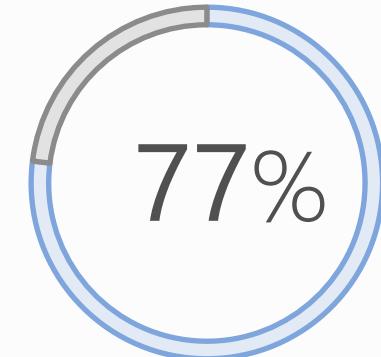
Gartner®

The average size of a container deployment has grown 75% in one year.<sup>1</sup>



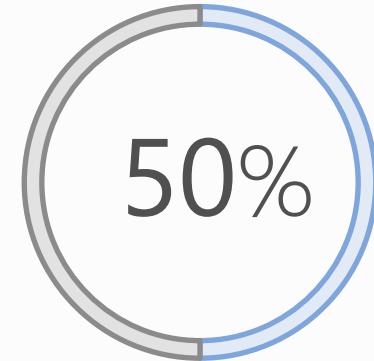
Half of container environments are orchestrated.<sup>1</sup>

77% of companies<sup>2</sup> who use containers orchestrators choose Kubernetes.



Larger companies are leading the adoption.<sup>1</sup>

Nearly 50% of organizations<sup>1</sup> running 1000 or more hosts have adopted containers.



1. Datadog [report](#): 8 Surprising Facts About Real Docker Adoption

2. CNCF [survey](#): cloud-native-technologies-scaling-production-applications

# Helping you achieve more



Connect medical devices to the cloud

Challenge: Needed to **develop applications faster**. Making the shift to platform provider. Deconstructing value add services into **microservices**

Success:

**Speed** of product development significantly increased

Improve mobile service delivery of online shopping cart

Challenge: Finding a solution to support service delivery locally, **on-premises, and in the cloud**. Fixing shopping cart API flaw to maintain zero-downtime policy.

Success:



**10** seconds to deploy API  
vs 20 minutes

Consolidate data from disparate systems to understand customers

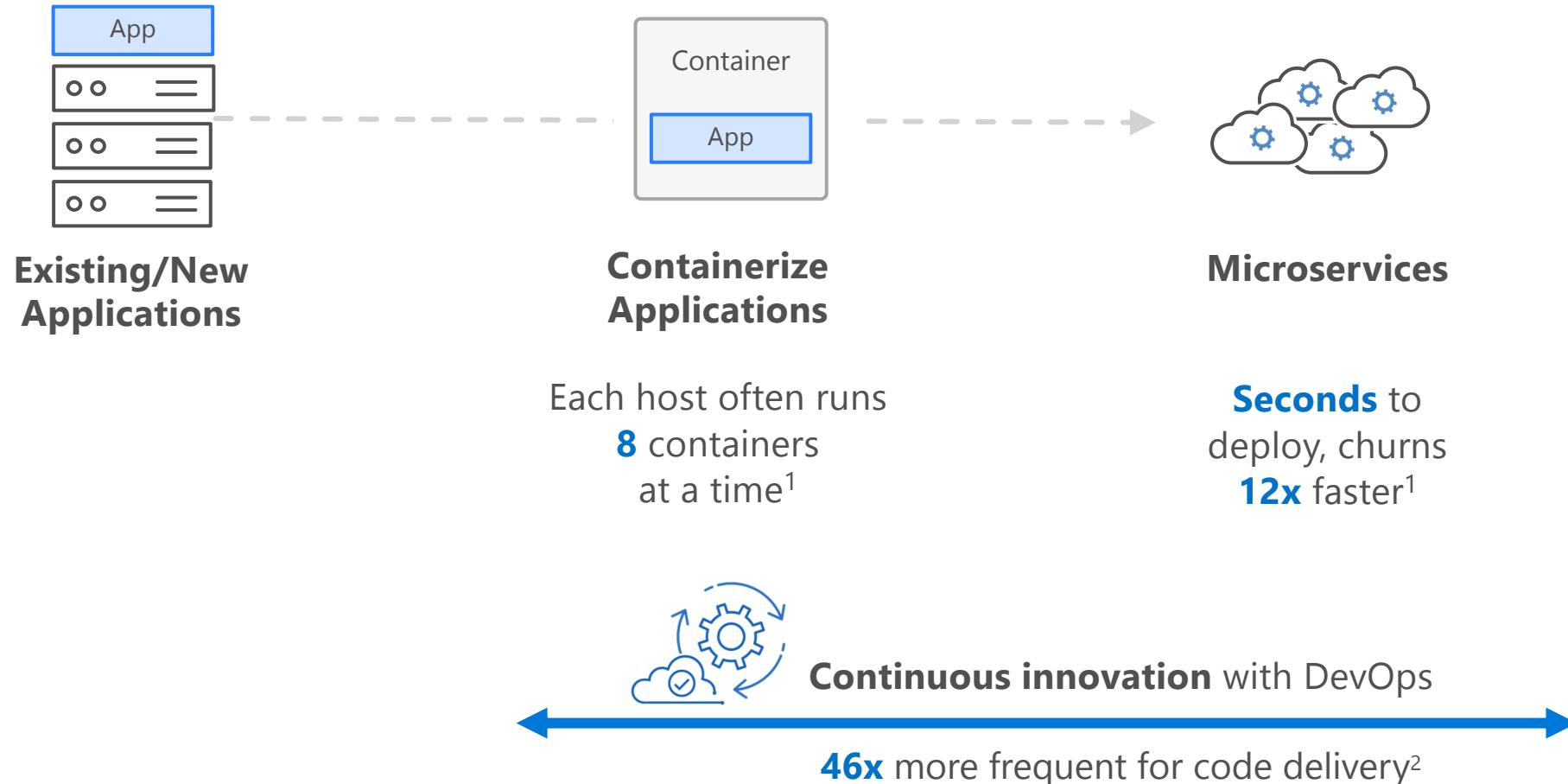
Challenge: Consolidating data from **57 different systems**. Needed to move on-premises workloads to the cloud for better customer management and analysis.

Success:



**50%** time reduction in data management and analysis

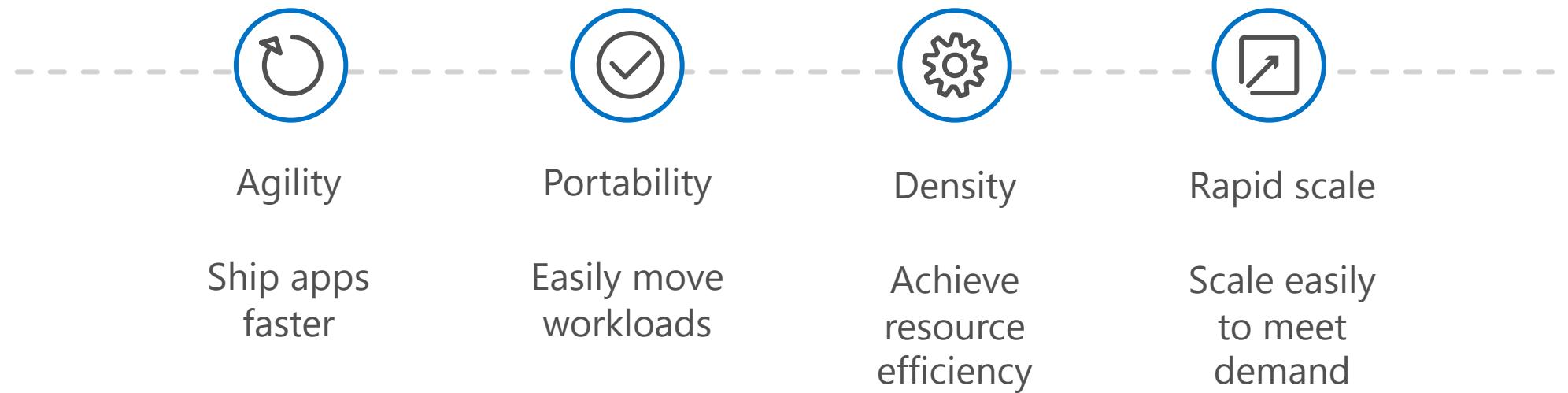
# From traditional systems to portfolio of modern apps



Source:

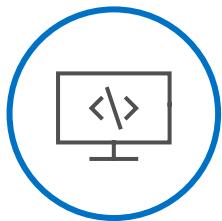
1: Datadog [Report](#): 8 Surprising Facts About Real Docker Adoption; 2: 2017 state of DevOps [Report](#)

# The **benefits** of using containers



# Productivity

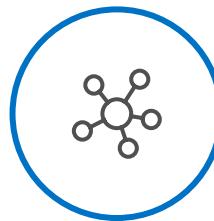
Accelerate containerized application development



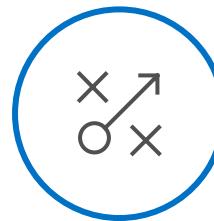
Automatically  
containerize and  
scaffold any  
applications  
directly from IDE



Auto-build  
to a secure  
container  
registry



Rapidly iterate,  
test and debug  
microservices

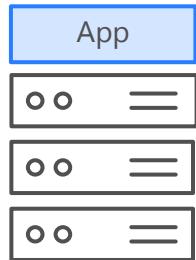


A few clicks  
to receive a  
full CI/CD  
pipeline



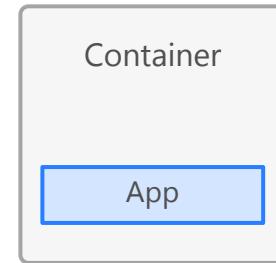
Built-in monitoring  
and logging to get  
full visibility of  
container health  
and app telemetry

# What is a **container**?



## Virtual machines

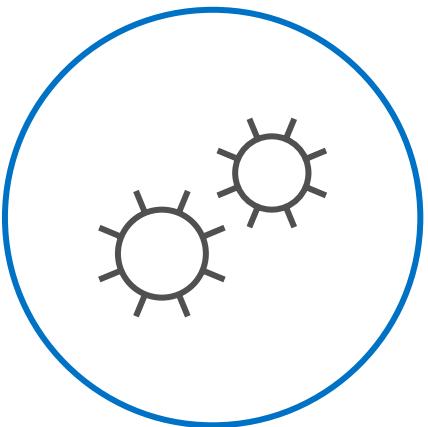
- Virtualize the hardware
- Application and OS dependencies coupled
- VMs as unit of scale



## Containers

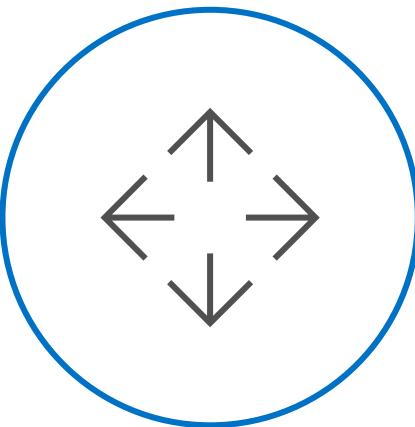
- Process isolation, feels like OS virtualization
- Package your application and dependencies into an artifact
- Decouple application from OS
- Container as a unit of scale

# Containers as the App Packaging Format



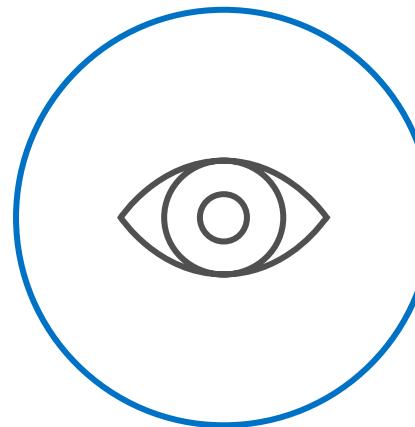
## Build

Create immutable artifact.



## Push

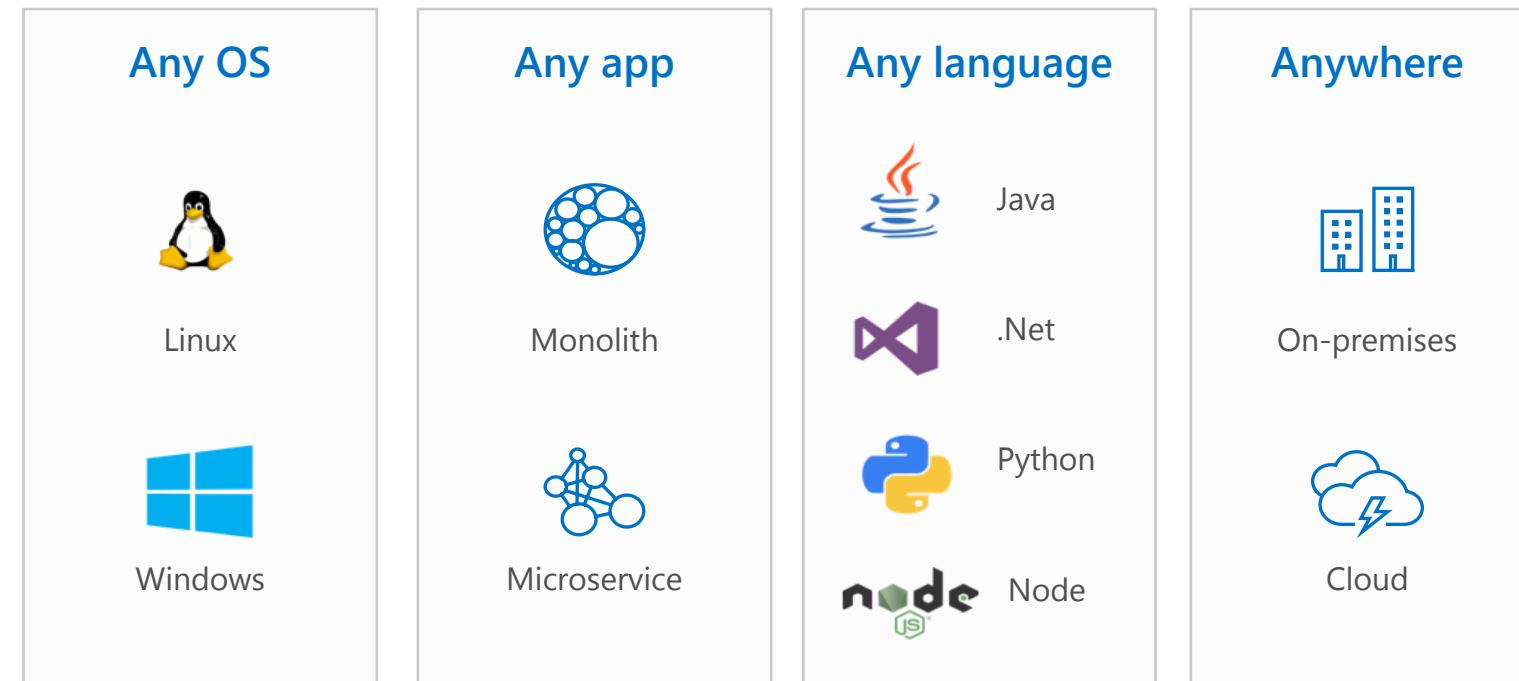
Deliver artifact to a container registry



## Deploy

Deploy containerized applications to your environment

# The benefits of using containers



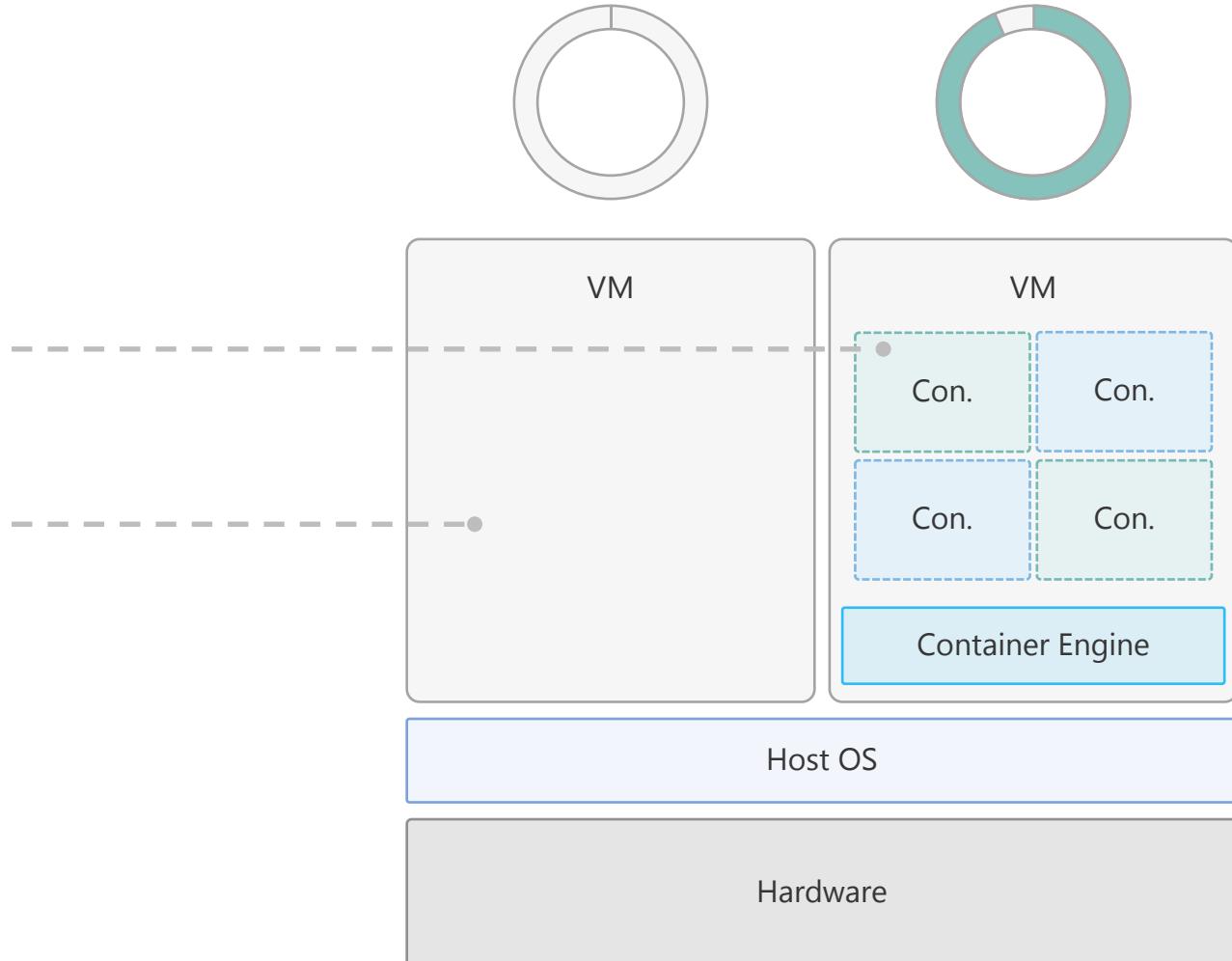
# The container **advantage**

## Containerized environment

Migrate containers and their dependencies to underutilized VMs for improved density and isolation

Decommission unused resources for efficiency gains and cost savings

Container is lighter weight and faster to scale dynamically



# Containers in Azure



App Service



Service Fabric



Kubernetes Service



Container Instance



Ecosystem

Deploy web apps or APIs using containers in a PaaS environment

Modernize .NET applications to microservices using Windows Server containers

Scale and orchestrate Linux containers using Kubernetes

Elastically burst from your Azure Kubernetes Service (AKS) cluster

Bring your Partner solutions that run great on Azure



Azure Container Registry



Docker Hub

----- Choice of developer tools and clients -----



Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service



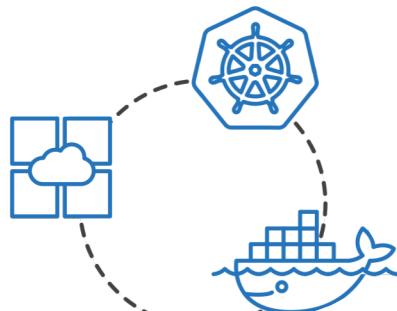
Azure Kubernetes Service (AKS)



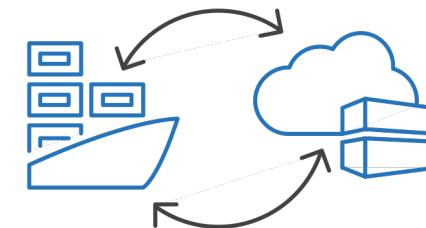
Service Fabric

# Azure Container Registry (ACR)

Manage a Docker private registry as a first-class Azure resource



Manage images for all  
types of containers



Use familiar, open-  
source Docker CLI tools

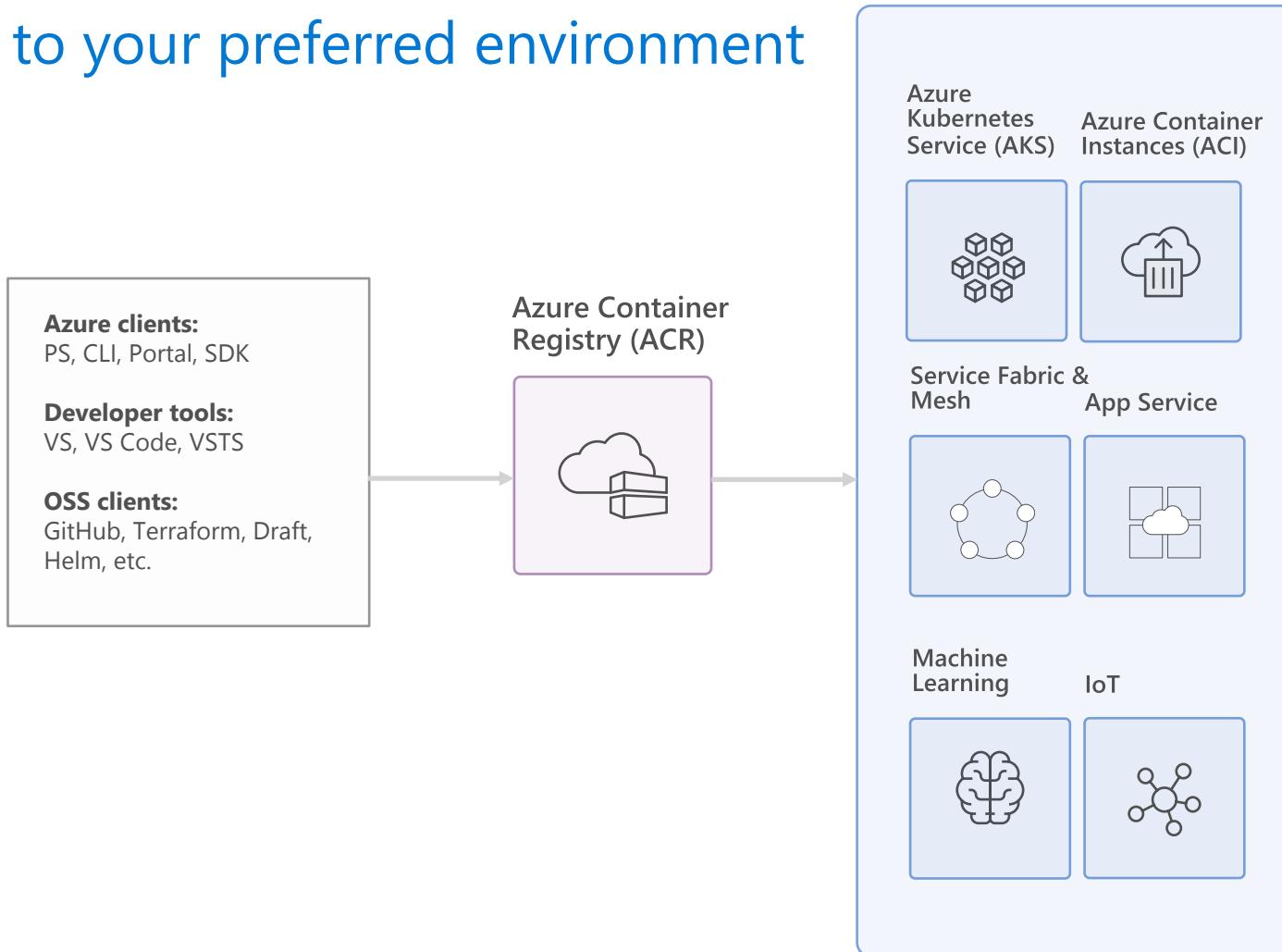


Azure Container Registry  
geo-replication



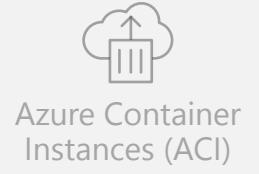
# Containers as the App Packaging Format

Deploy to your preferred environment

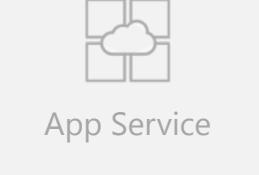




Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service

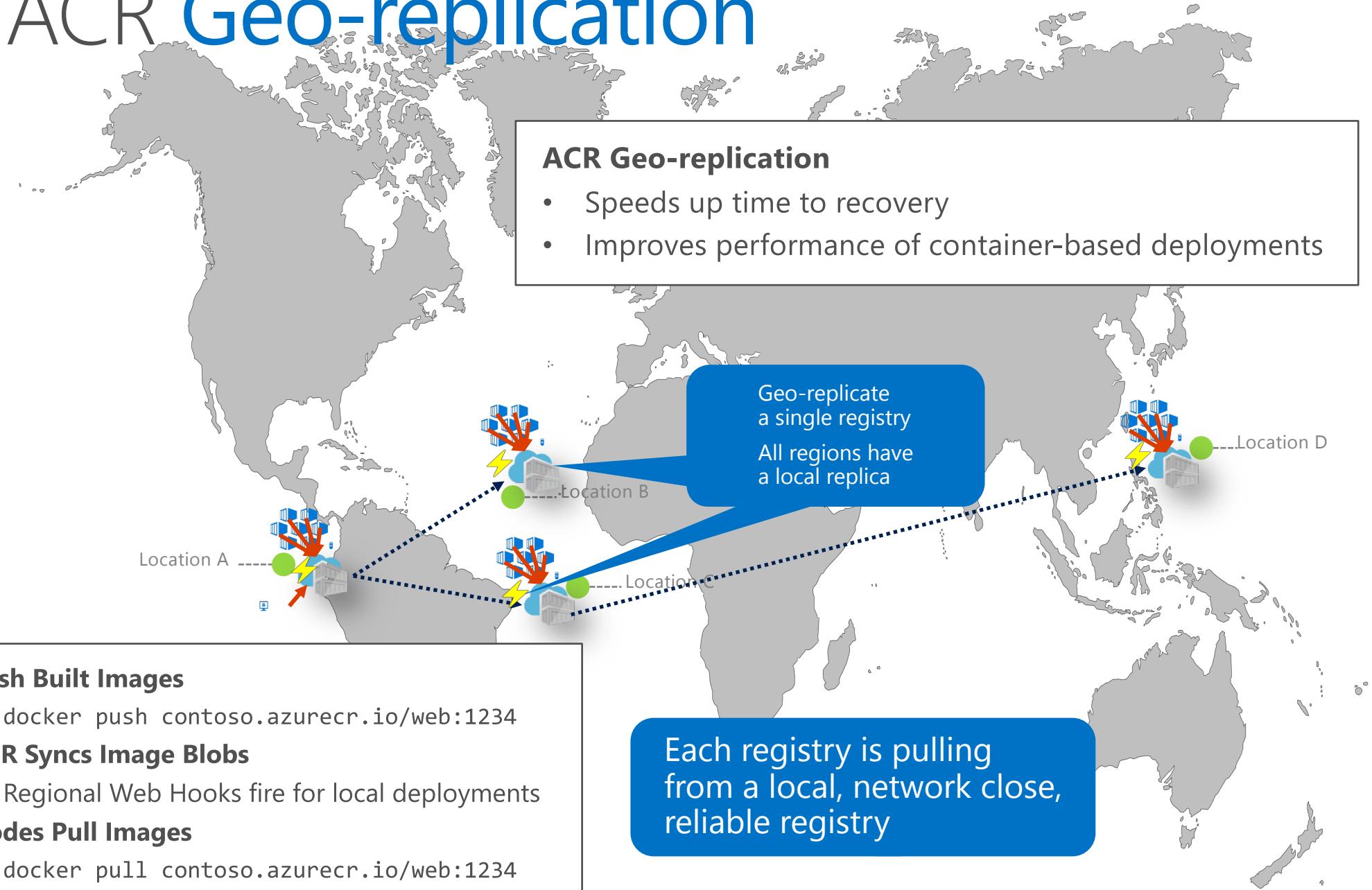


Azure Kubernetes Service (AKS)



Service Fabric

# ACR Geo-replication





Azure Container  
Registry (ACR)



Azure Container  
Instances (ACI)



App Service



Azure Kubernetes  
Service (AKS)

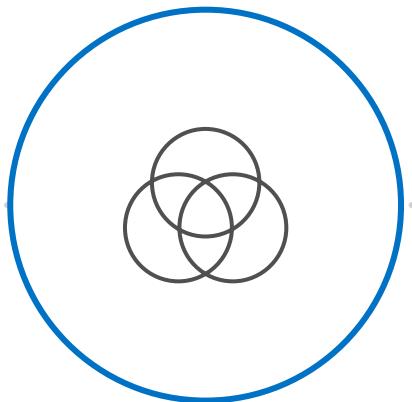


Service Fabric

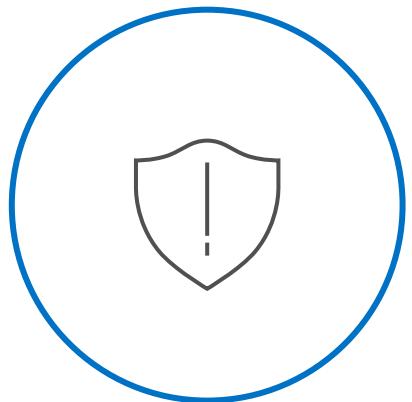
# Built-in security with ACR



Authenticate using Azure  
Active Directory Identity



Integrate Your OS &  
Framework Patching



Quarantine pattern ensures  
secure images



Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service

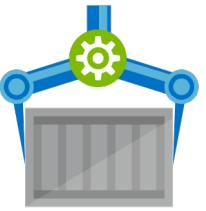


Azure Kubernetes Service (AKS)



Service Fabric

# ACR Build



Native Container Build Service in the cloud

Follows  **docker build** semantics

```
docker build -t helloworld:v1 .
```

```
az acr build -t helloworld{{.Build.ID}} .
```

Trigger based builds (git commits, base image updates)

```
az acr build-task create
  --image      helloworld{{.Build.ID}}
  --name       myBuildTask
  --registry   jengademos
  --context    https://github.com/me/helloworld
  --branch     master
  --git-access-token $PAT
```



Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service



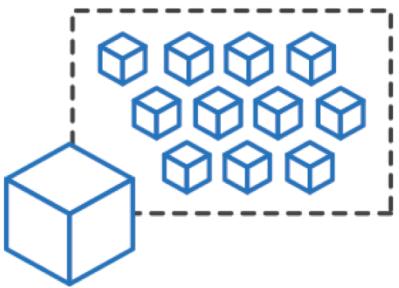
Azure Kubernetes Service (AKS)



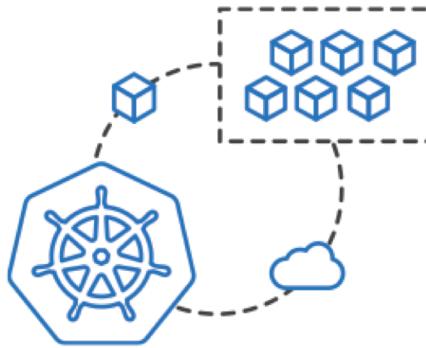
Service Fabric

# Azure Container Instances (ACI)

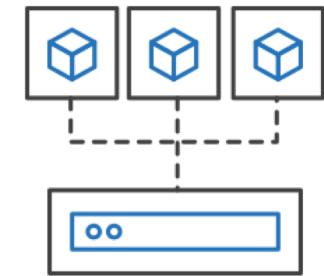
Easily run containers on Azure without managing servers



Run containers  
without managing  
servers



Increase agility  
with containers on  
demand



Secure applications  
with hypervisor  
isolation



Azure Container Registry (ACR)



Azure Container Instances (ACI)



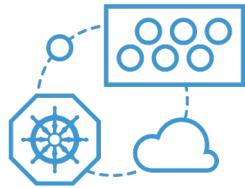
App Service



Azure Kubernetes Service (AKS)



Service Fabric



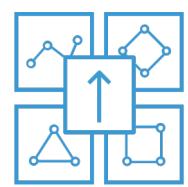
Elastic bursting with AKS

Azure Container Service (AKS) can use the Virtual Kubelet to provision pods inside ACI that start in seconds. Then ACI provides fast, isolated compute to meet traffic that comes in spikes, without the need to manage servers.



Event-driven applications with Azure Logic Apps

Combine ACI with the ACI Logic Apps connector, Azure queues, and Azure Functions to build robust infrastructure that can elastically scale out containers on demand.



Data processing jobs

Use Azure Container Instances for data processing where source data is ingested, processed, and placed in a durable store such as Azure Blob storage. Achieve significant cost savings through per-second billing.



Azure Container Registry (ACR)



Azure Container Instances (ACI)



**App Service**



Azure Kubernetes Service (AKS)



Service Fabric

# App Service

Easily deploy and run container-based web apps at scale

Accelerated outer loop



Tight integration w/ Docker Hub, Azure Container Registry



Built-in CI/CD w/  
Deployment Slots



Intelligent diagnostics &  
troubleshooting, remote debugging

Fully managed platform



Automatic scaling  
and load balancing



High availability  
w/ auto-patching



Backup &  
recovery

Flexibility & choices



From CLI, portal, or  
ARM template



Single Docker image, multi  
container w/ Docker Compose

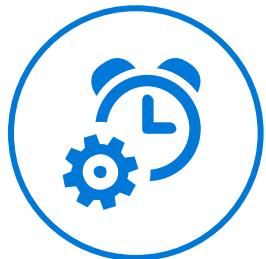


IntelliJ, , Jenkins, Maven  
Visual Studio family



# Azure App Service

Quickly build, deploy and scale powerful cloud applications without worrying about infrastructure



## High productivity

Accelerate development using existing language skills, familiar frameworks, built-in CI/CD, and intelligent diagnostic capabilities



## Fully managed

Focus on application logic and let Azure take care of mundane tasks like capacity provisioning, OS patching, scaling, load balancing and domain management

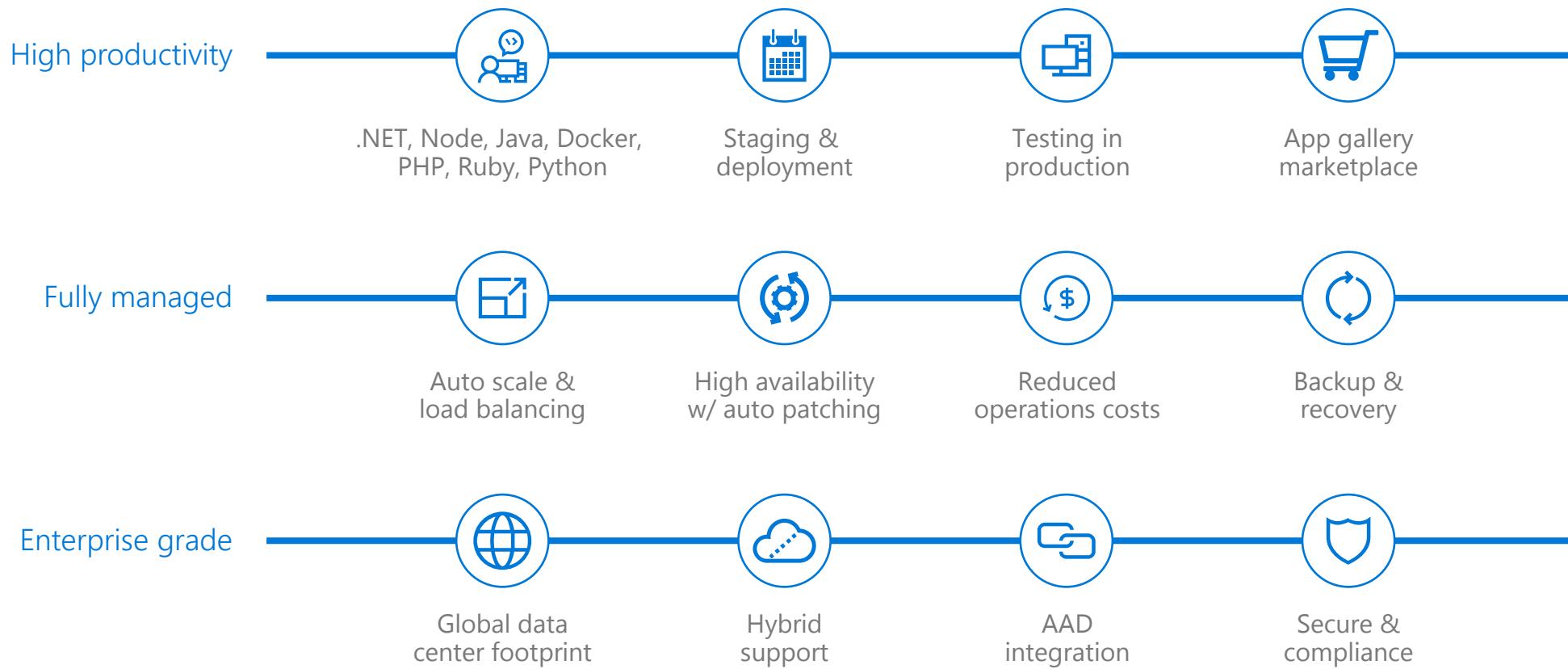


## Enterprise grade

Meet rigorous performance, security and compliance requirements through a choice of hosting options, robust regulation support and MSI integration

# Azure App Service

Quickly build, deploy and scale powerful cloud applications without worrying about infrastructure



550K active customers

1.7M apps & sites hosted

>20B requests per day



ADRIATIC SAILING



Ahresty



Flexible is better.<sup>®</sup>



B E Y M E N





Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service



Azure Kubernetes Service (AKS)



Service Fabric

# Azure Kubernetes Service (AKS)

Simplify the deployment, management, and operations of Kubernetes



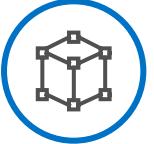
Deploy and manage Kubernetes with ease



Scale and run applications with confidence



Secure your Kubernetes environment



Accelerate containerized application development



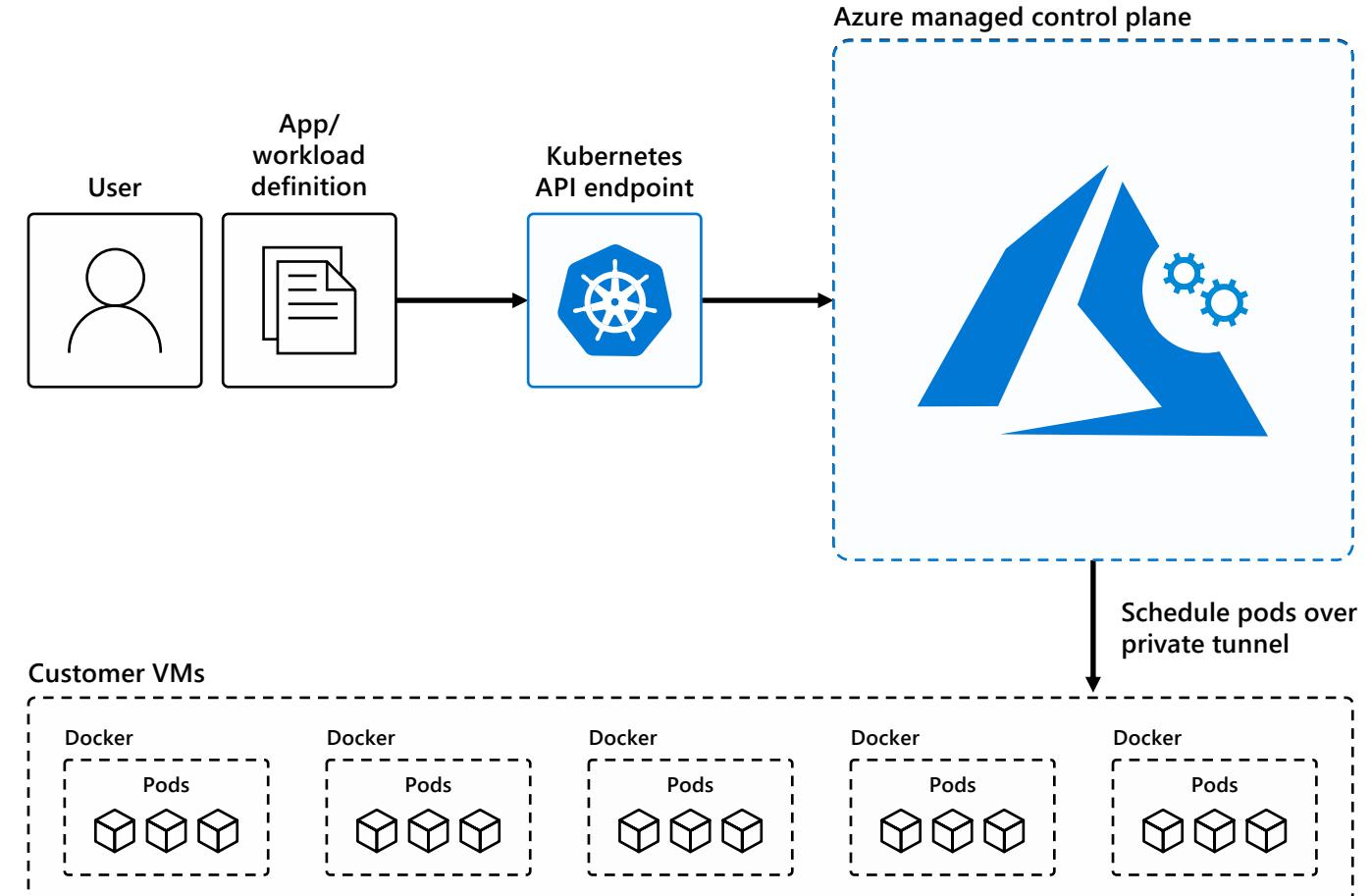
Work how you want with open-source tools & APIs



Set up CI/CD in a few clicks

# How managed Kubernetes on Azure works

- Automated upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge



# From infrastructure to **innovation**

**Managed Kubernetes  
empowers you to do more**

Focus on your containers  
and code, not the plumbing  
of them.

Responsibilities	DIY with Kubernetes	Managed Kubernetes on Azure
Containerization		
Application iteration, debugging		
CI/CD		
Cluster hosting		
Cluster upgrade		
Patching		
Scaling		
Monitoring and logging		

Customer  
 Microsoft

# Work how you want with opensource tools and APIs

	Development	DevOps	Monitoring	Networking	Storage	Security
Take advantage of services and tools in the Kubernetes ecosystem	 	 <b>Jenkins</b>  <b>Terraform</b>  <b>BRIGADE</b>  <b>CODESHIP</b>	 <b>Prometheus</b>    <b>OPENTRACING</b>  <b>DATADOG</b> 	 <b>CNI</b> Networking 	 	    RBAC
OR, Leverage growing Azure support	 <b>VS Code</b>	 <b>VSTS</b>  <b>ARM</b>	 <b>Azure Monitor</b>	 <b>Azure VNET</b>	 <b>Azure Storage</b>	 <b>Azure Container Registry</b>  <b>AAD</b>  <b>Key Vault</b>

# Microsoft leads **open source** communities



Two members of the  
Kubernetes  
steering committee



Member of the  
technical board of the  
Cloud Native  
Compute Foundation



Board member of the  
Linux Foundation



Several leads or co-  
leads of Kubernetes  
SIGs (special  
interest groups)



# Microsoft builds open source projects



Helm – The de-facto package manager for kubernetes (<https://helm.sh>),  
Top level CNCF project



Draft – A rapid-development environment for new kubernetes developers (<https://draft.sh>)



Brigade – Easy to use javascript based workflow definition for kubernetes (<https://brigade.sh>)



Kubernetes + VS-Code (<https://github.com/Azure/vscode-kubernetes-tools>)





Azure Container Registry (ACR)



Azure Container Instances (ACI)



App Service



Azure Kubernetes Service (AKS)



Service Fabric

# Azure Service Fabric

A microservices platform for business critical applications



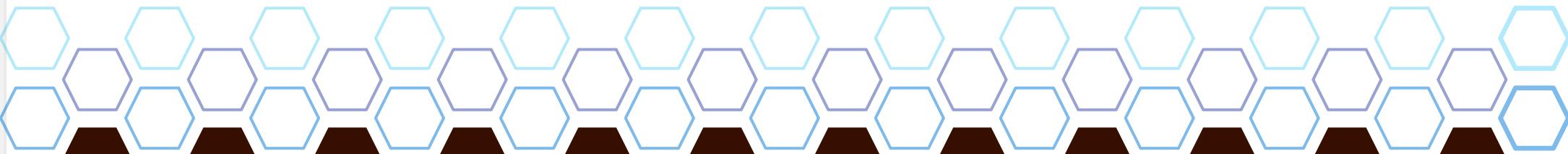
Build



Deploy



Operate



Programming  
Models



Dev & Ops  
Tooling



Orchestration



Lifecycle  
Management



Health &  
Monitoring



Always On  
Availability



Auto  
Scaling



Dev machine



Any cloud



On-premises  
infrastructure



Azure

Use any language, any framework

.NET & .NET Core



Build Java



Reliable Services and  
Reliable Actors .NET/Java  
libraries



ASP.NET Core



spring

ASP.NET Core



Go



C++



.NET

Integrate with any CI/CD and Test Environment



Deploy

Azure DevOps



Jenkins

Maven™

Gradle



Octopus Deploy



Azure DevTest Labs

Integrate with any monitoring solution



Operate

Azure Log Analytics



Azure Application  
Insights



dynatrace



DATADOG



elastic



New Relic®



APPDYNAMICS

# Build: data-aware microservices



Programming  
Models



Dev & Ops  
Tooling



Orchestration



Lifecycle  
Management



Health &  
Monitoring



Always On  
Availability



Auto  
Scaling



**Reliable Actors**

Use familiar tools: Visual Studio + Team Services for .NET or Jenkins + Yeomen for Java



**Reliable Services**

Manage state reliability without a database, lowering latency



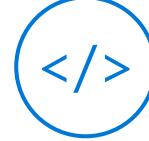
**Guest Executables**

Run existing code and orchestrate life cycle using service fabric



**Containers**

Orchestrate your Windows Server or Linux containers reliably at scale



**.NET or Java ...**

Built-in ASP.NET core integration; work with VS and VSTS or Eclipse and Jenkins

# Deploy: any code on any OS



Programming  
Models



Dev & Ops  
Tooling



Orchestration



Lifecycle  
Management



Health &  
Monitoring



Always On  
Availability



Auto  
Scaling



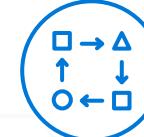
CI/CD

Maximize uptime and scalability  
with isolated compute threads  
running concurrently



Docker Compose

Orchestrate existing  
container applications  
natively



Automate

Deploy or remove applications  
using PowerShell, CLI, Visual  
Studio, and other APIs



Rolling upgrades

Upgrade non-disruptively and  
roll-back in case of failures,  
automate with PowerShell



Monitor and diagnose

Generate, aggregate, and analyze  
events with built-in tooling and  
integration with Azure services

# Operate: on any cloud at any scale



Programming Models



Dev & Ops Tooling



Orchestration



Lifecycle Management



Health & Monitoring



Always On Availability

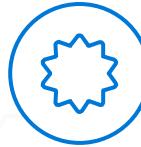


Auto Scaling



**Use familiar tools**

Such as Splunk, OMS, ELK, or AppInsights to gain deep insights or monitor application health



**Use controlled chaos**

Test graceful and ungraceful failure scenarios



**Recover gracefully**

Recover from node or service failure gracefully; replicate data automatically



**Secure at scale**

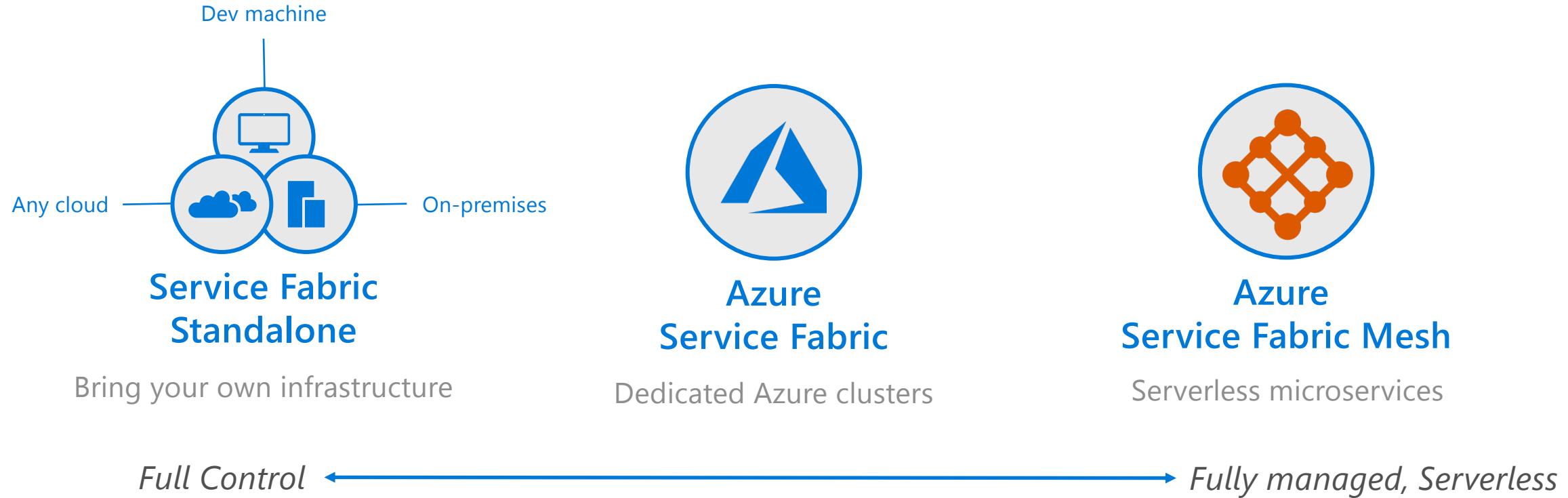
Secure node-to-node communication and user access using built-in capabilities



**Scale programmatically**

Use PowerShell, CLI, or APIs to scale programmatically achieving very high densities

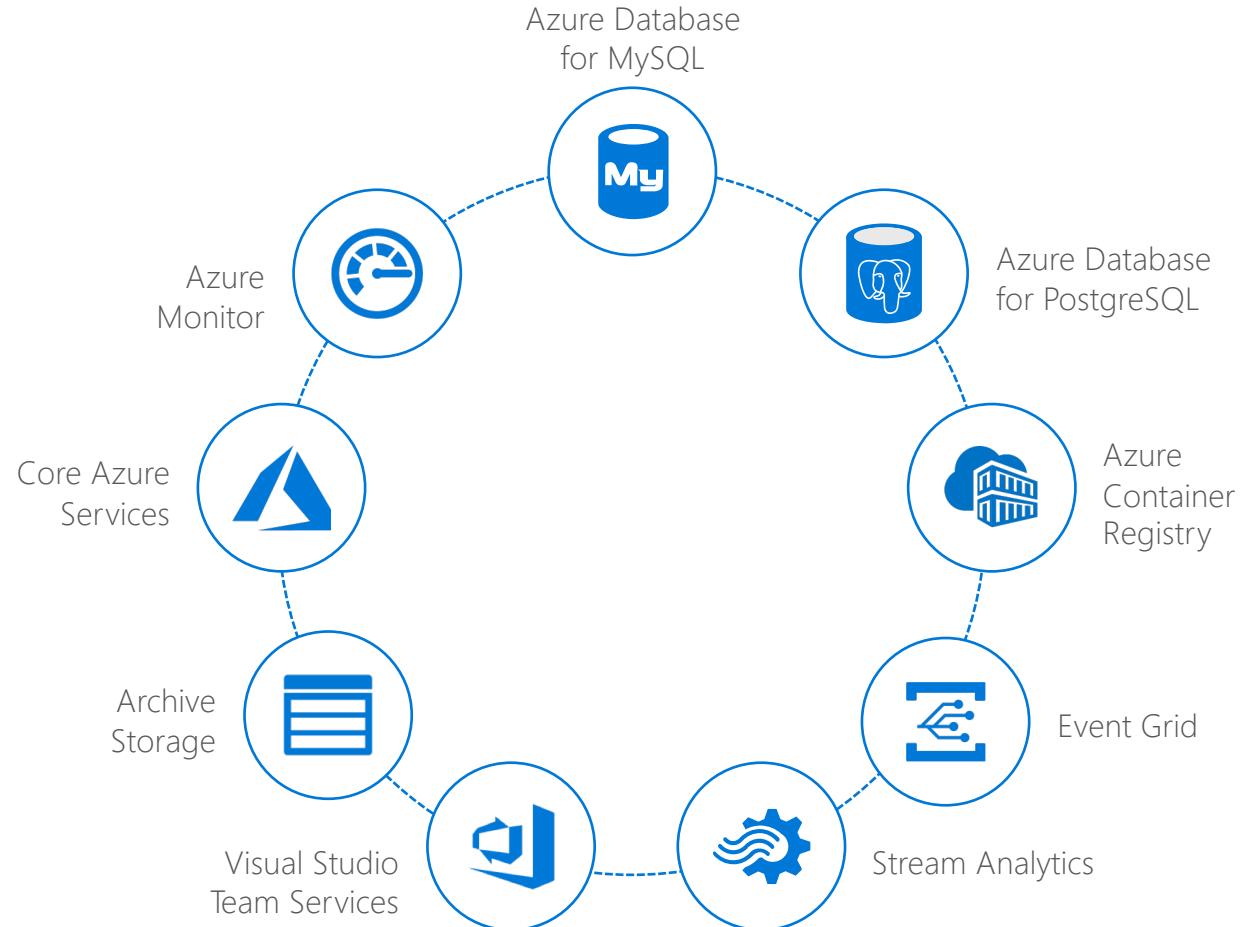
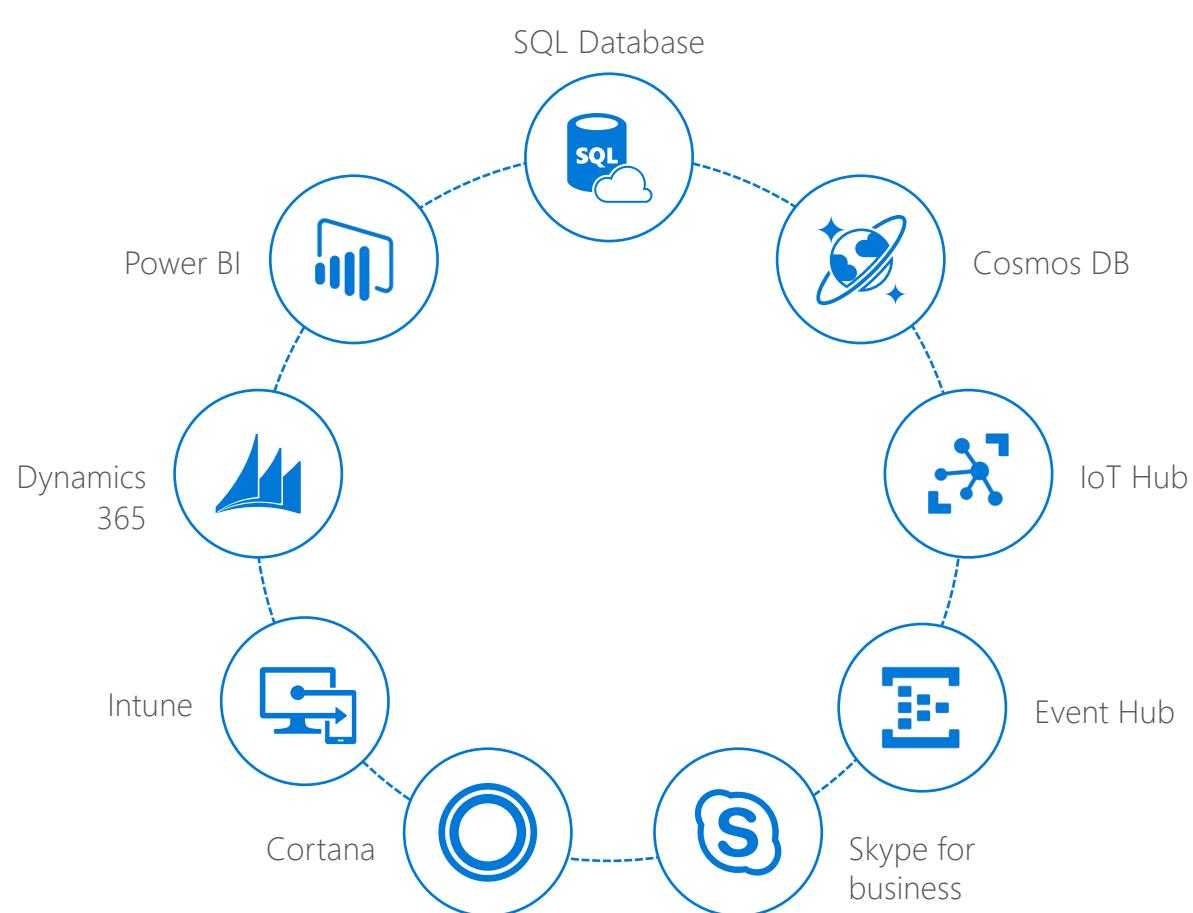
# Azure Service Fabric offerings



Service Fabric

# Powering Azure and Microsoft services

Azure Service Fabric is designed for mission-critical services

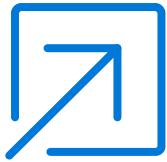


# What applications can you build?



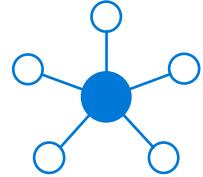
Interactive session  
and media

*Media sharing,  
communication and  
gaming*



Mission-critical  
business SaaS

*B2B solutions in  
the cloud*



IoT data  
processing

*Efficient data  
processing on  
millions of devices*



Low-latency data  
processing



*Data-centric  
applications requiring  
fast response times*



Web and mobile  
backend apps



*Retail and e-  
commerce*



# Customer stories



# Rapidly growing software company attracts customers with seamless cloud demo experience

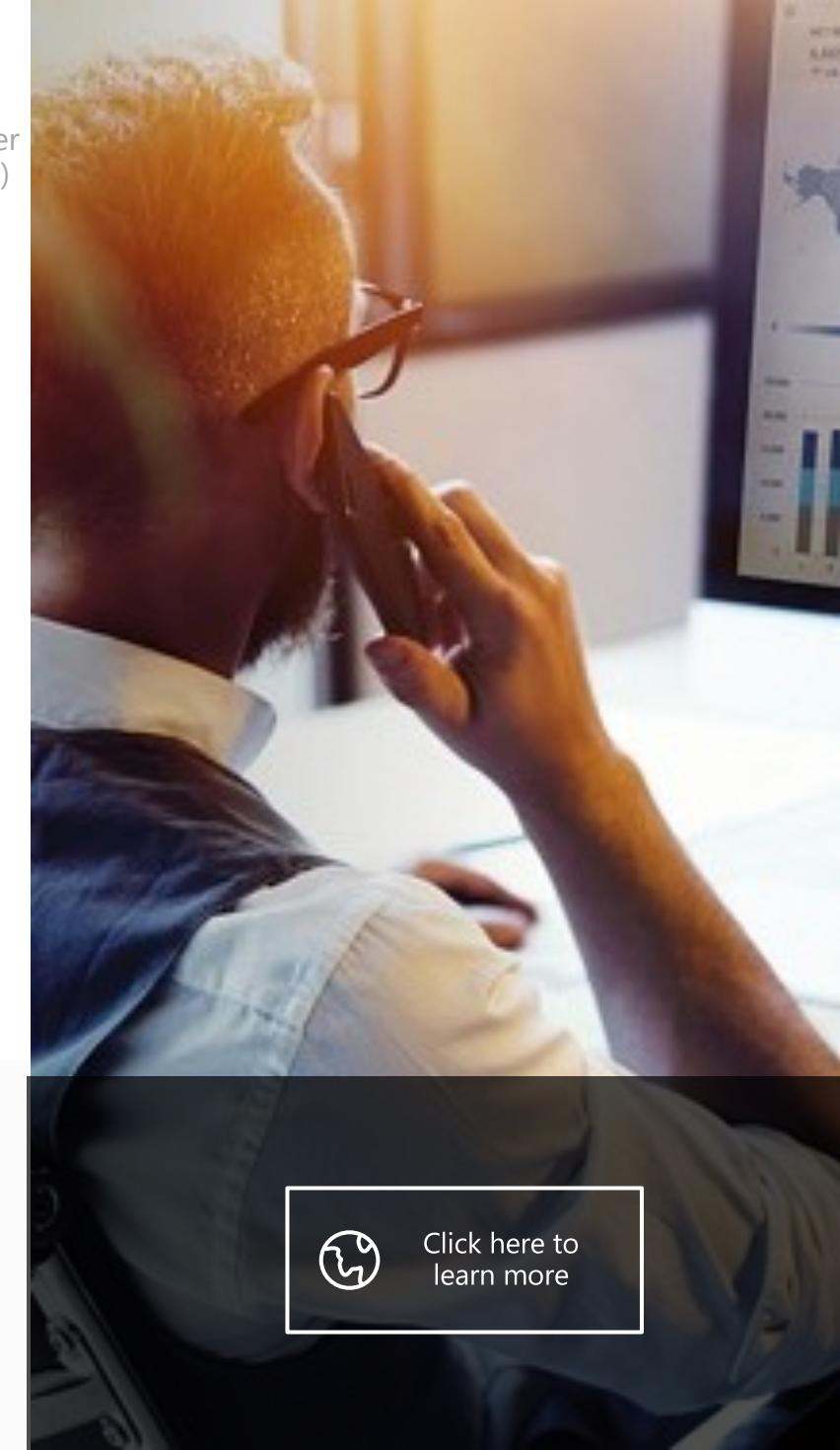
**Challenge:** Jedox needed a more lightweight compute unit than virtual machines to power its website demos and provide a good customer experience.

**Solution:** The company used Microsoft Azure Container Instances to support the Jedox Marketplace and power its demos.

**Outcome:** With ACI, Jedox is able to spin up customer demos on demand, improve provisioning speed, build confidence in the cloud, and lower IT costs.

**“** We are far more responsive to customer needs since adding Azure Container Instances. Our Marketplace demos are powerful marketing tools for Jedox, and we've used Azure to improve the customer experience significantly. **”**

Vladislav Malicevic , Vice President Development and Support, Jedox



Click here to learn more



App Service

# Japanese financial technology company speeds app development, expands market using cloud platform

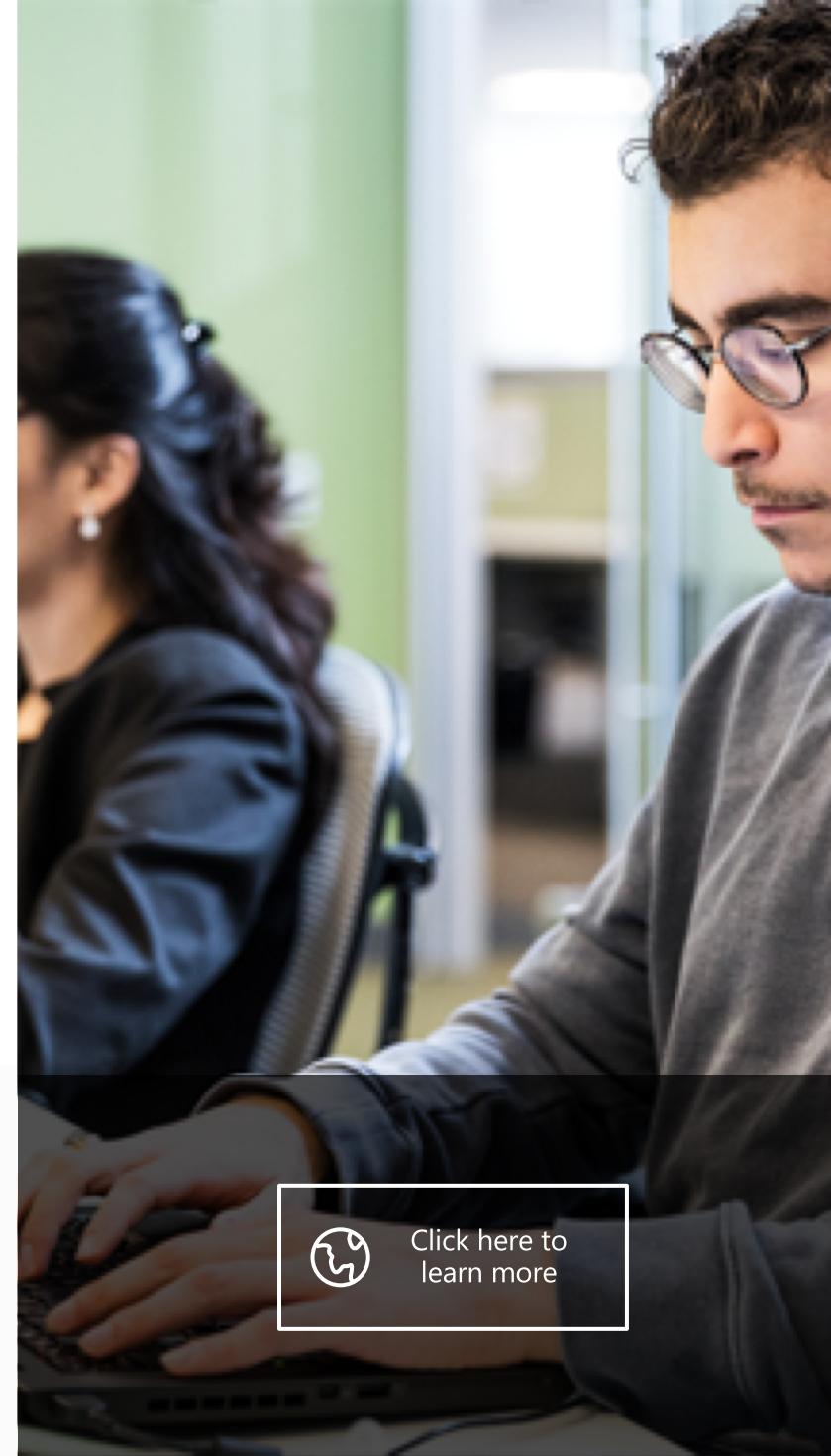
**Challenge:** Crowd Cast experienced growing demands for Staple, their cloud-based expense management service, and wanted to increase its development speed and agility to meet demand from enterprise customers.

**Solution:** The company switched its platform as a service (PaaS) environment from Heroku and Amazon Web Services to Microsoft Azure, transferring its Ruby on Rails development environment using Docker containers and App Service.

**Outcome:** With App Service on Linux, Crowd Cast developers can take advantage of the agility and scalability of the microservices approach, building their app as a small set of services, while reducing the complexity of managing microservices deployments.

**“** We created the system on Azure with technical support from Microsoft. Because we could transfer Ruby on Rails in its existing state, using Docker containers, we encountered no technical problems. **”**

Haley Koike, Development Manager, Crowd Cast



Click here to learn more



# Siemens Health leverages technology to connect medical devices to the cloud through AKS

**Challenge:** Siemens needed to speed up their development process to make the transition from value-added services provider to platform provider.

**Solution:** Siemens adopted Azure Kubernetes Service (AKS) to speed up application development and run their microservices-based apps.

**Outcome:** With AKS, Siemens has driven newfound product development agility. AKS enables them to use an applicant gateway and API management to manage exposure, control, and to meter the access continuously.

“ The managed Azure Kubernetes Service puts us really into a position to not only deploy our business logic in Docker containers, including the orchestration, but it's also really easy through application gateway and API management to manage that exposure and control and meter the access continuously.

Thomas Gossler, Lead Architect - Digital Ecosystem Platform, Siemens



## Zeiss creates smart devices by connecting field devices with back-end systems

**Benefits:**

- Run Windows containers with existing code & new microservices together
- Securely expose business applications using REST APIs
- Improved agility makes it easier for developers to update container applications and services
- Scalable microservices-based platform for stateless/stateful workloads



“ With Service Fabric we rely on a robust and scalable platform which host our digital integration scenarios – stateful integrations in Reliable Services and stateless integrations in containers can be hosted side by side on one platform.

”

Kai Walter, Lead IT Solution Architect ZEISS Group



# Next Games powers its global augmented reality game in Microsoft Azure

## Challenge:

Next Games required a lot of back-end processing power—processing power that was sited close to players, wherever they wanted to play in the world. The company also needed database and compute services that could handle massive amounts of geo-distributed data.

## Solution:

Next Games adopted a microservices-based development and hosting environment that relies on Azure Service Fabric, Azure Functions, and Azure Cosmos DB to provide very fast, very low-overhead support of geo-distributed data services.

## Outcome:

Azure Service Fabric handles the passage of messages between players, and Azure Cosmos DB provides a geo-distributed database that locates player data geographically close to the player and coordinates player matchups.

**“**In addition to all the new services, Microsoft is constantly improving existing Azure services ... This continual improvement enhances our games and makes our lives easier. **”**

Kalle Hiitola, Chief Technology Officer and Cofounder, Next Games

