Azure container technologies

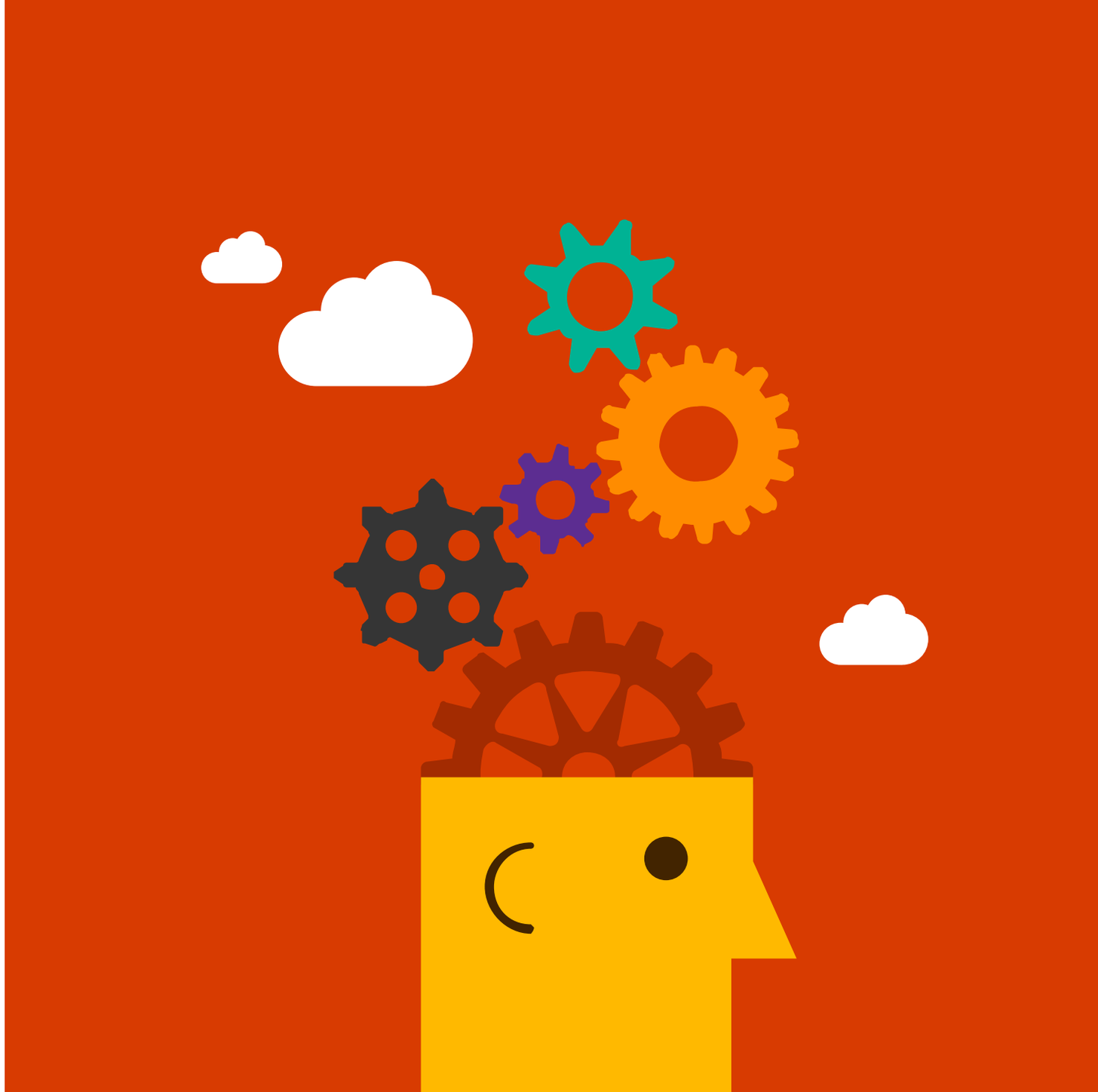
Product Family Overview

Microsoft

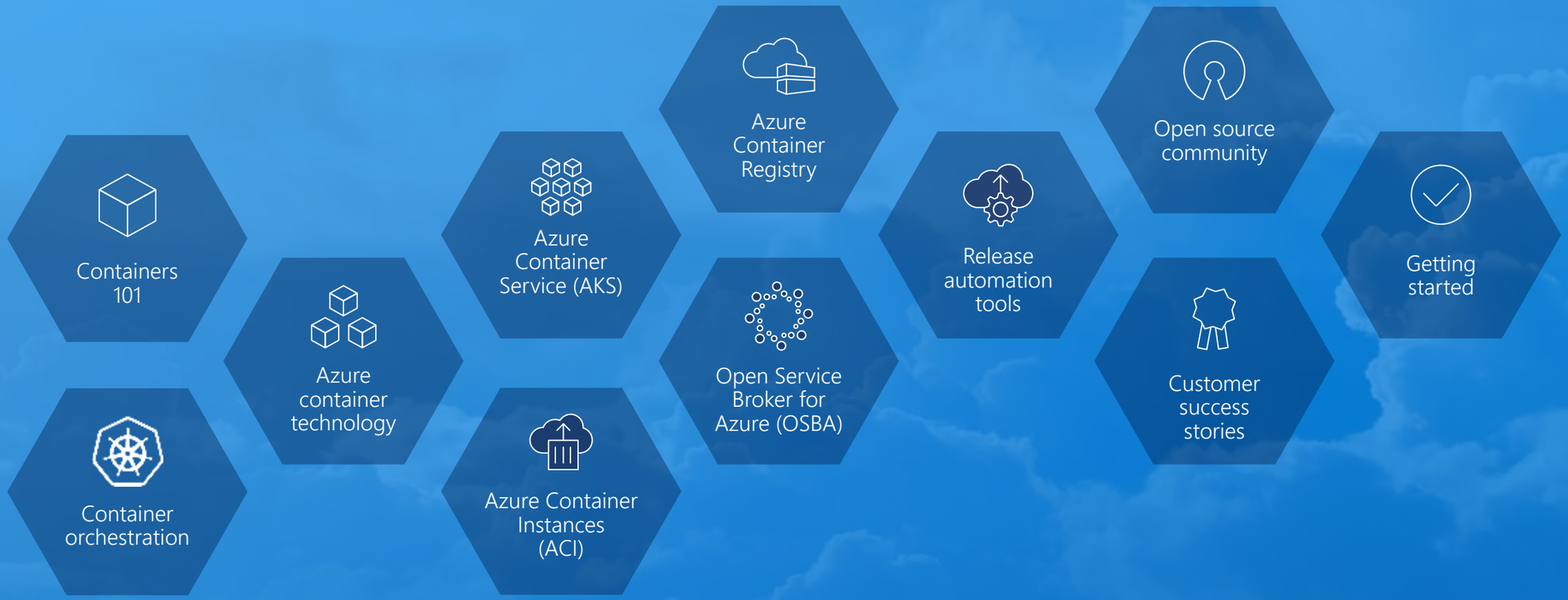
Azure Containers

Product Family Overview

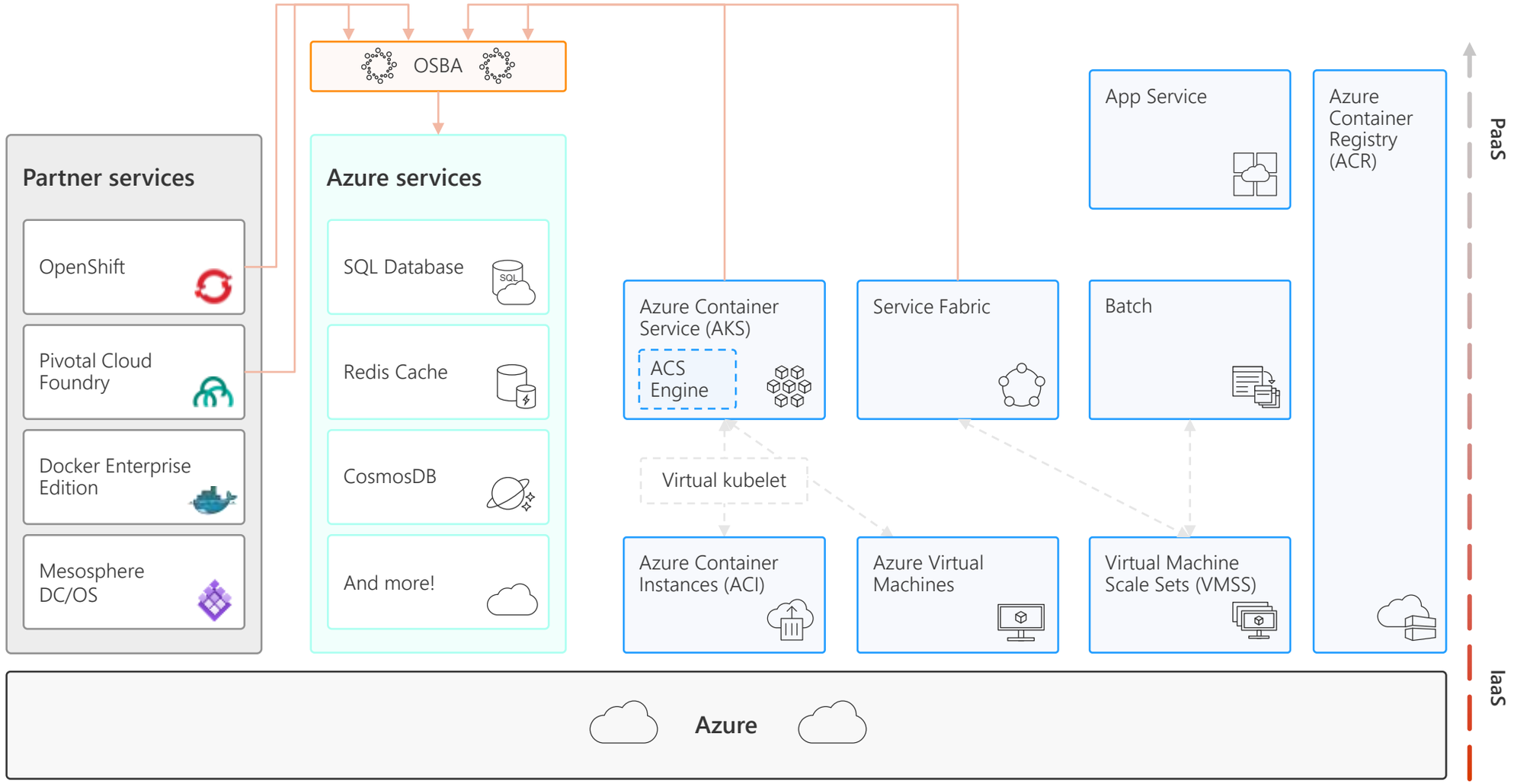
Gustav Kaleta
Global Black Belt
App Dev GBB
Microsoft
gkaleta@microsoft.com



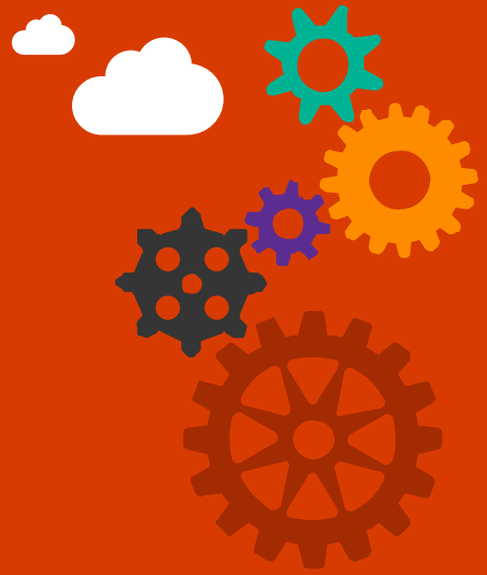
Agenda


Containers 101
Container orchestration
Azure container technology
Azure Container Service (AKS)
Azure Container Instances (ACI)
Azure Container Registry
Open Service Broker for Azure (OSBA)
Release automation tools
Open source community
Customer success stories
Getting started

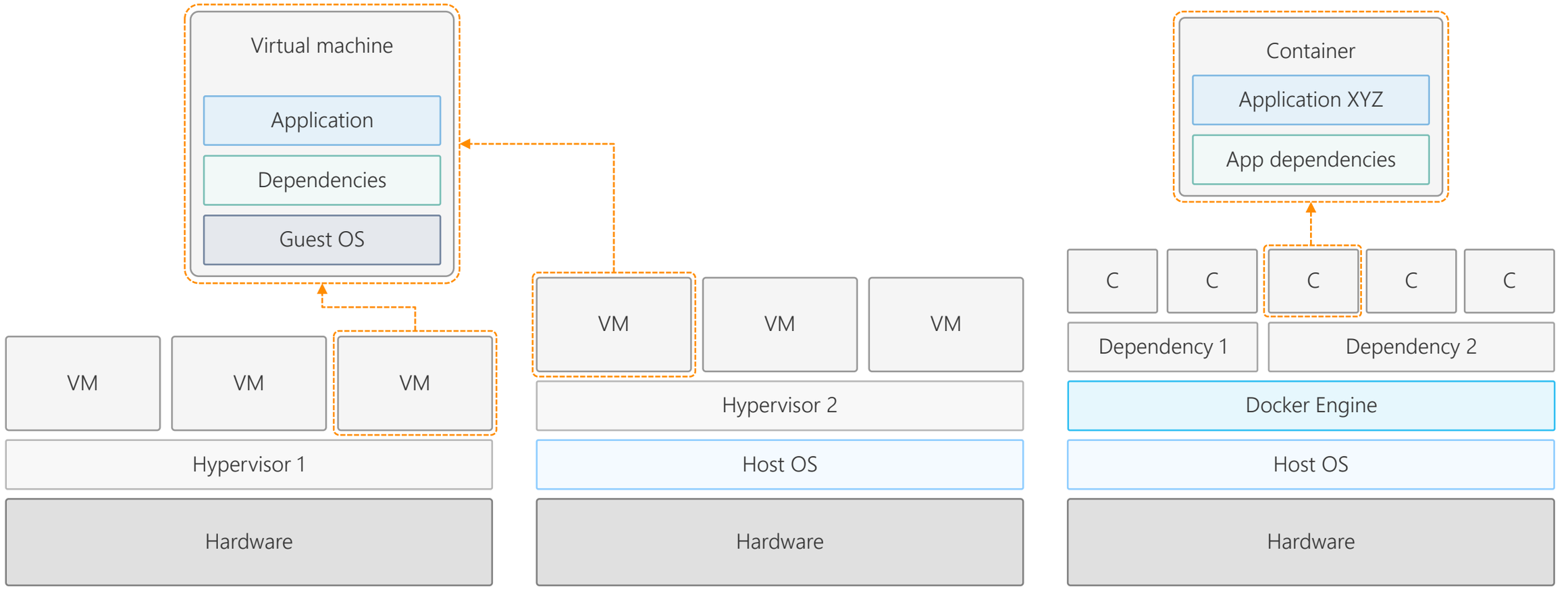
Azure container **ecosystem**



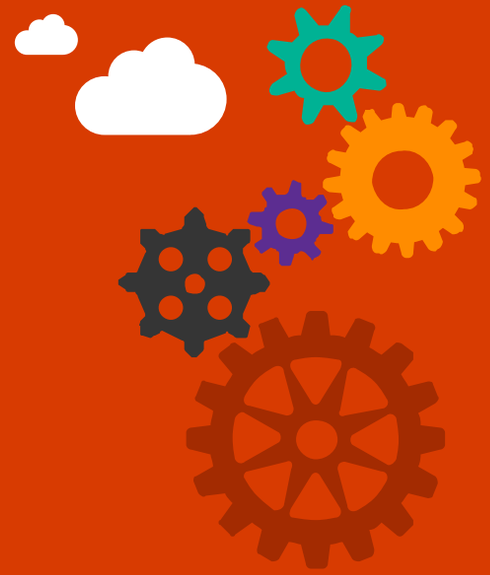
VM vs Container



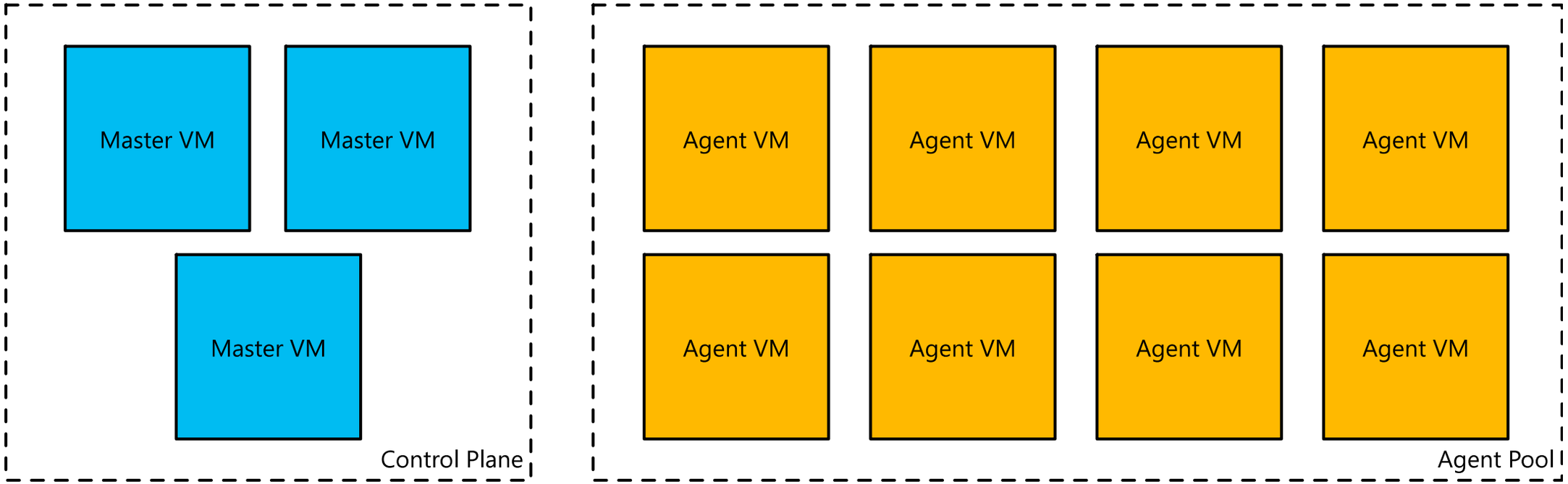
Virtualization versus **containerization**



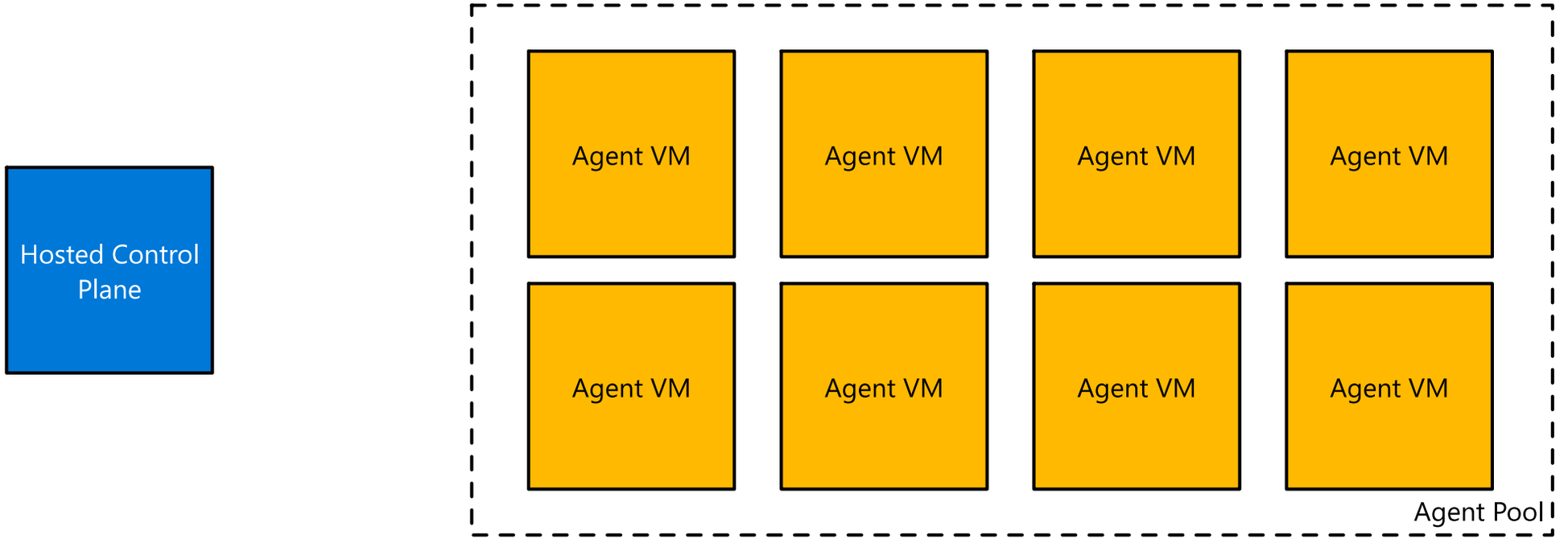
Azure Container Service (AKS)



Kubernetes without AKS -> ACS – K8s



Kubernetes with AKS



Why orchestration

Scheduling

Affinity/anti-affinity

Health monitoring

Failover

Scaling / upgrading

Networking

Service discovery / self healing

Coordinated app upgrades

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 ..
```

AKS Roadmap (GA in Q1/Q2 CY 2018)

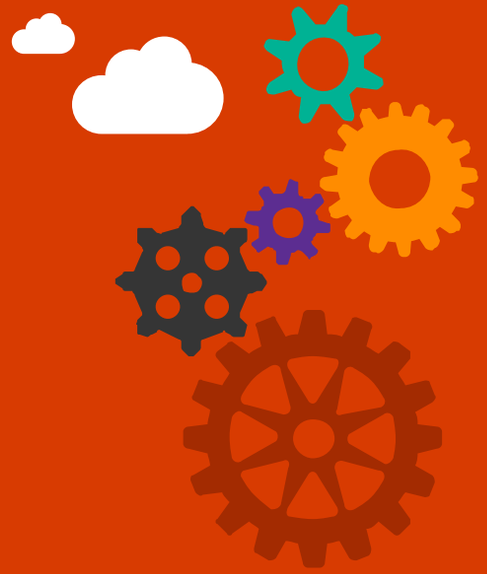
Feature support targeted in Q1/Q2 '18 (subject to change)

- Stable and reliable cluster
- Terraform support
- AAD + Kubernetes RBAC
- Custom VNET

Differentiated features

- Windows support
- AAD integration with 2FA
- Service Broker integration

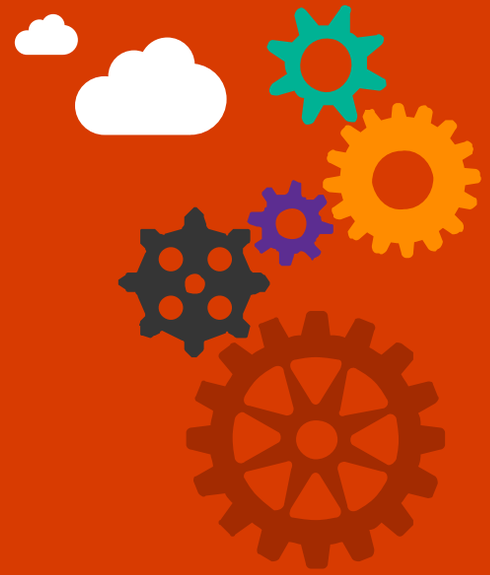
ACS Engine



Why ACS Engine?

- Develop in the open
  - Proving ground and incubation for new features
  - Preview for later product functionality
- Total flexibility
  - Enables custom deployment with all the knobs and dials
- https://github.com/Azure/acs-engine

Azure Container Instances (ACI)



Azure Container Instances

- Start in seconds
- No VM management
- Custom CPU/memory
- Billed per second
- Hypervisor-level isolation
- Linux and Windows containers

Getting Started with ACI

```
$ az container create --name mycontainer --image microsoft/aci-helloworld --
resource-group myResourceGroup --ip-address public
  "ipAddress": {
    "ip": "52.168.86.133",
    "ports": [...]
  },
  "location": "eastus",
  "name": "mycontainer",
  "osType": "Linux",
  "provisioningState": "Succeeded",

$ curl 52.168.86.133
<html>
<head>
  <title>Welcome to Azure Container Instances!</title>
</head>
```

Use Cases for ACI Today

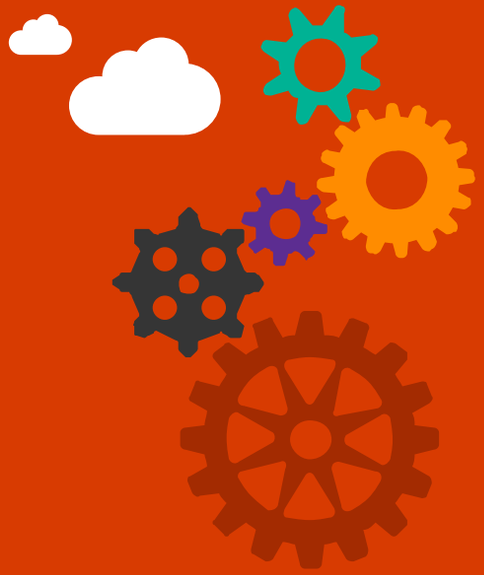
- Lift and shift into containers
- Simple web services
- Compute intensive workloads
- Batch processing

ACI Permanent Limitations

- No support for:
  - Zero-downtime upgrades
  - Maintain > 1 running replicas
  - Batch workflows
  - Service discovery
  - Integrated load balancing
  - Auto-scaling

If you need these features, you probably need an orchestrator

ACI Connector for Kubernetes



Azure Container Service (AKS)

Azure Container Instances (ACI)

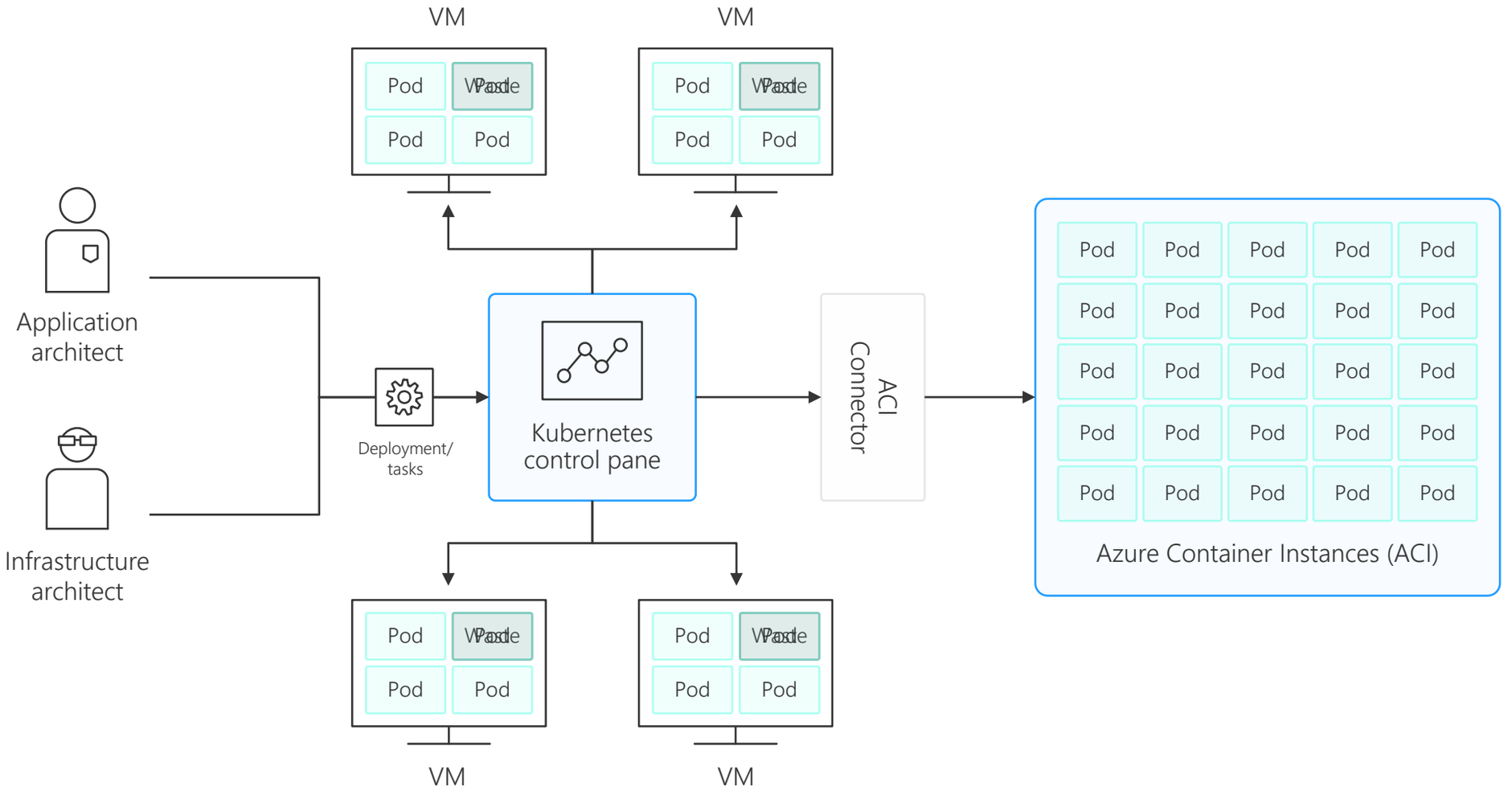
Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

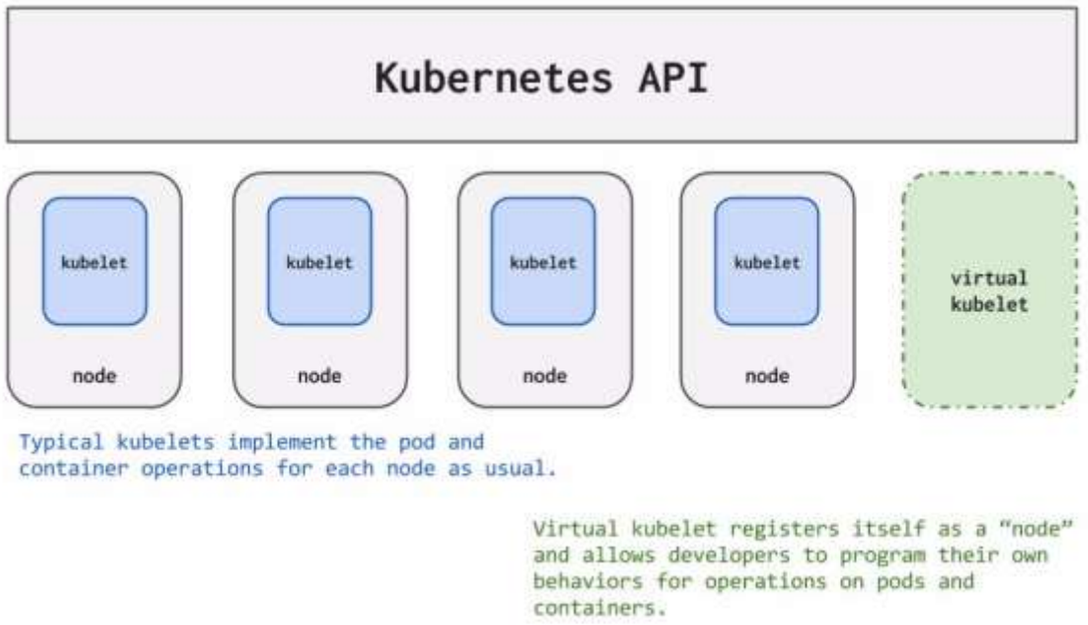
Azure Container Instances (ACI) PREVIEW

Bursting with the ACI Connector

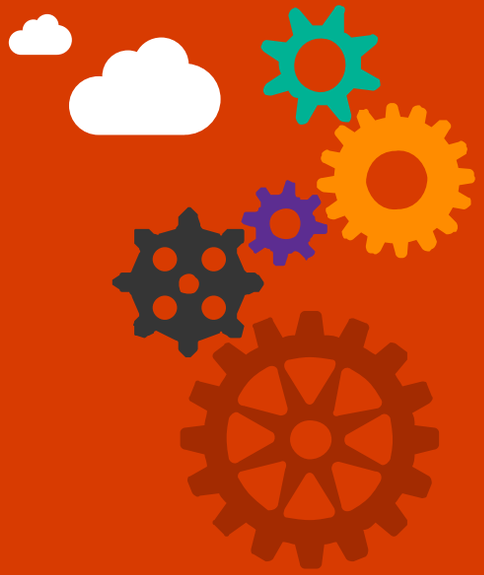


How does it work?

1. Registers as a virtual node with unlimited capacity
2. Dispatches scheduled pods to ACI instead of VMs



Open Service Broker for Azure (OSBA)



Why Open Service Broker for Azure?

- Container-based data services
  - Challenging operational characteristics
  - No SLA

- Azure data services
  - Strong operational characteristics
  - Guaranteed SLA

OSBA combines containers with Azure data services, providing the best of both worlds

Open Service Broker for Azure

- Easily connect Kubernetes apps to Azure services
  - Azure Database for MySQL
  - Azure Database for PostgreSQL
  - Azure SQL
  - Azure CosmosDB
  - Azure Redis
  - Azure Container Instances
  - Azure Service Bus
  - Azure Storage
  - More to come…
- Built on an open standard
- Integrated with Helm

OPEN SERVICE BROKER API™

Azure

Supported Platforms

- Cloud Foundry
- OpenShift
- Kubernetes (AKS)
- Service Fabric (Coming soon)

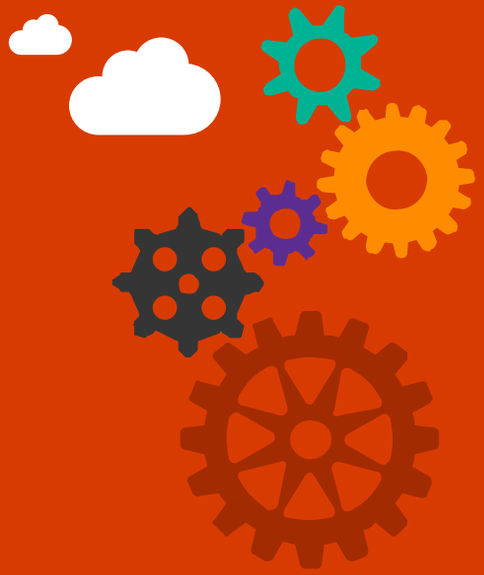
Getting Started with OSBA on Kubernetes

```
$ helm repo add azure Azure/helm-charts

$ helm install azure/service-broker

$ helm install azure/wordpress
```

Kubernetes Developer Tools



Helm

The package manager for Kubernetes

Why Helm?

- Manage Complexity
- Easy Updates
- Simple Sharing
- Rollbacks

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

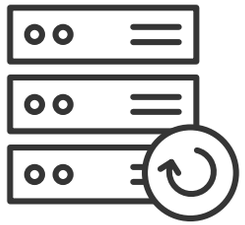
Charts can describe complex apps; provide repeatable app installs, and serve as a single point of authority

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 rollout` to roll back to an older version of a release with ease

Azure Container Service (AKS)

Azure Container Instances (ACI)

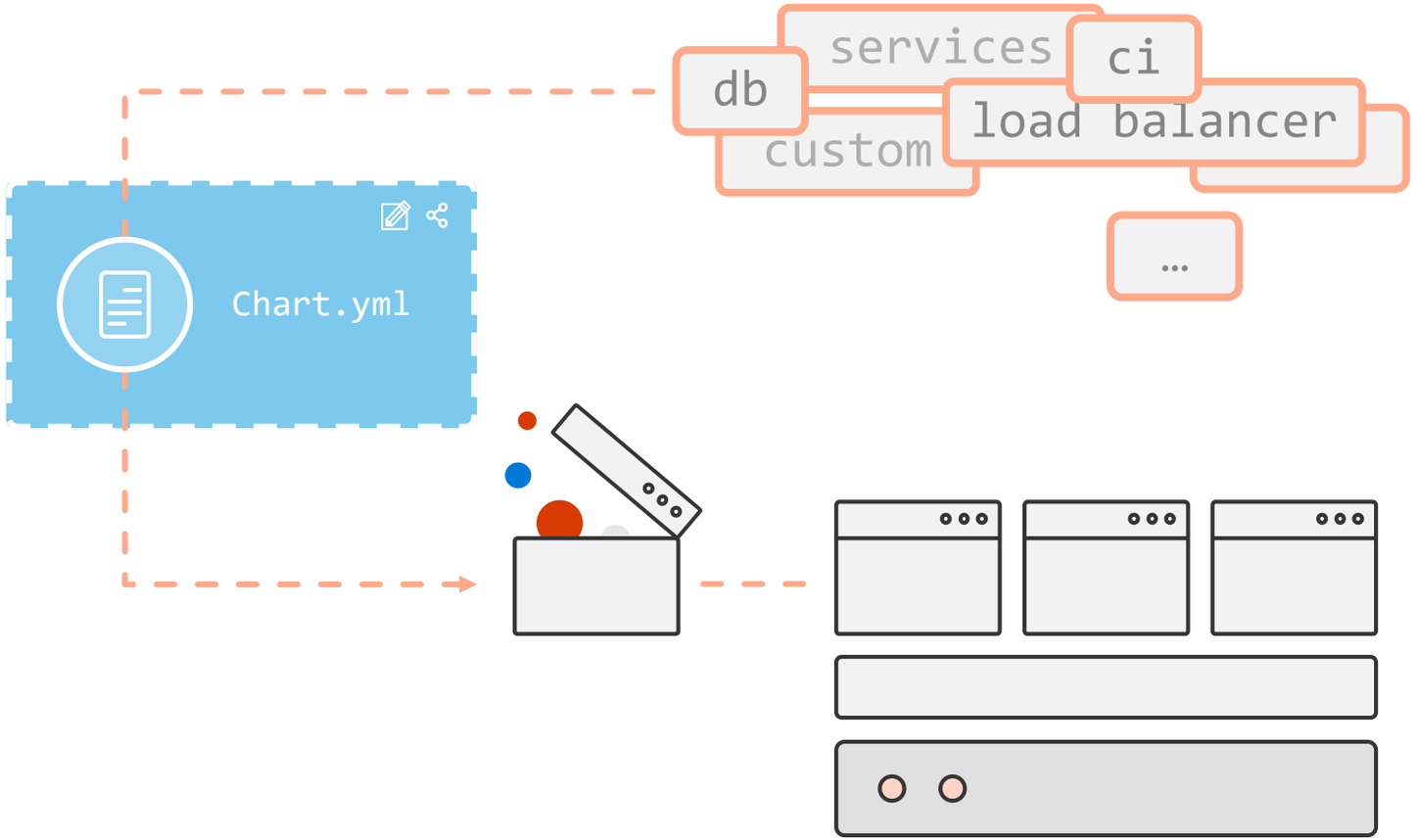
Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Helm

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



Draft

Streamlined Kubernetes Development

Why Draft?

- Develop Faster
- Use Best Practices
- Integrate with CI

Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Release automation workflow

Once developers are up and running—or working on a service that is in a complex system—Draft **ALSO** helps devs ignore artifacts and focus on code

Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

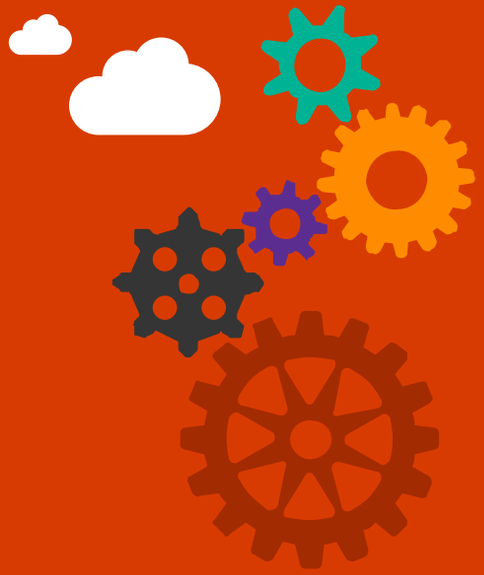
Open Service Broker API (OSBA)

Release Automation Tools

Draft

Draft in action

Thank You

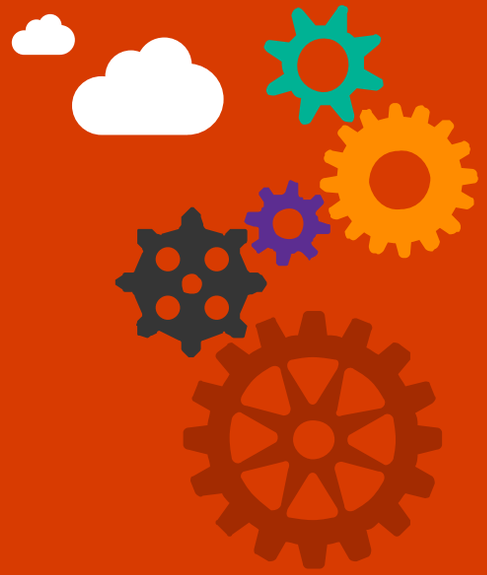


Resources

- Azure Container Service (AKS)
  - https://azure.microsoft.com/en-us/services/container-service/
  - https://docs.microsoft.com/en-us/azure/aks/
- Azure Container Instances (ACI)
  - https://azure.microsoft.com/en-us/services/container-instances/
  - https://docs.microsoft.com/en-us/azure/container-instances/
- ACI Connector for Kubernetes
  - https://github.com/Azure/aci-connector-k8s
- Azure Service Broker
  - https://github.com/Azure/meta-azure-service-broker
- Kubernetes Developer Tools
  - https://helm.sh/
  - https://draft.sh/
  - https://brigade.sh/

Microsoft

3rd Party Container Ecosystem



Tier0 ISV Container Partners

Platform: Pivotal, Red Hat, Docker, Mesosphere
Helm/Monocular: Bitnami
App Dev: HashiCorp, JFrog, CloudBees
Storage: Portworx, Netapp (NFS as a Service)
Security: Twistlock, Aqua
Networking: Bouyant, Tigera, Sysdig