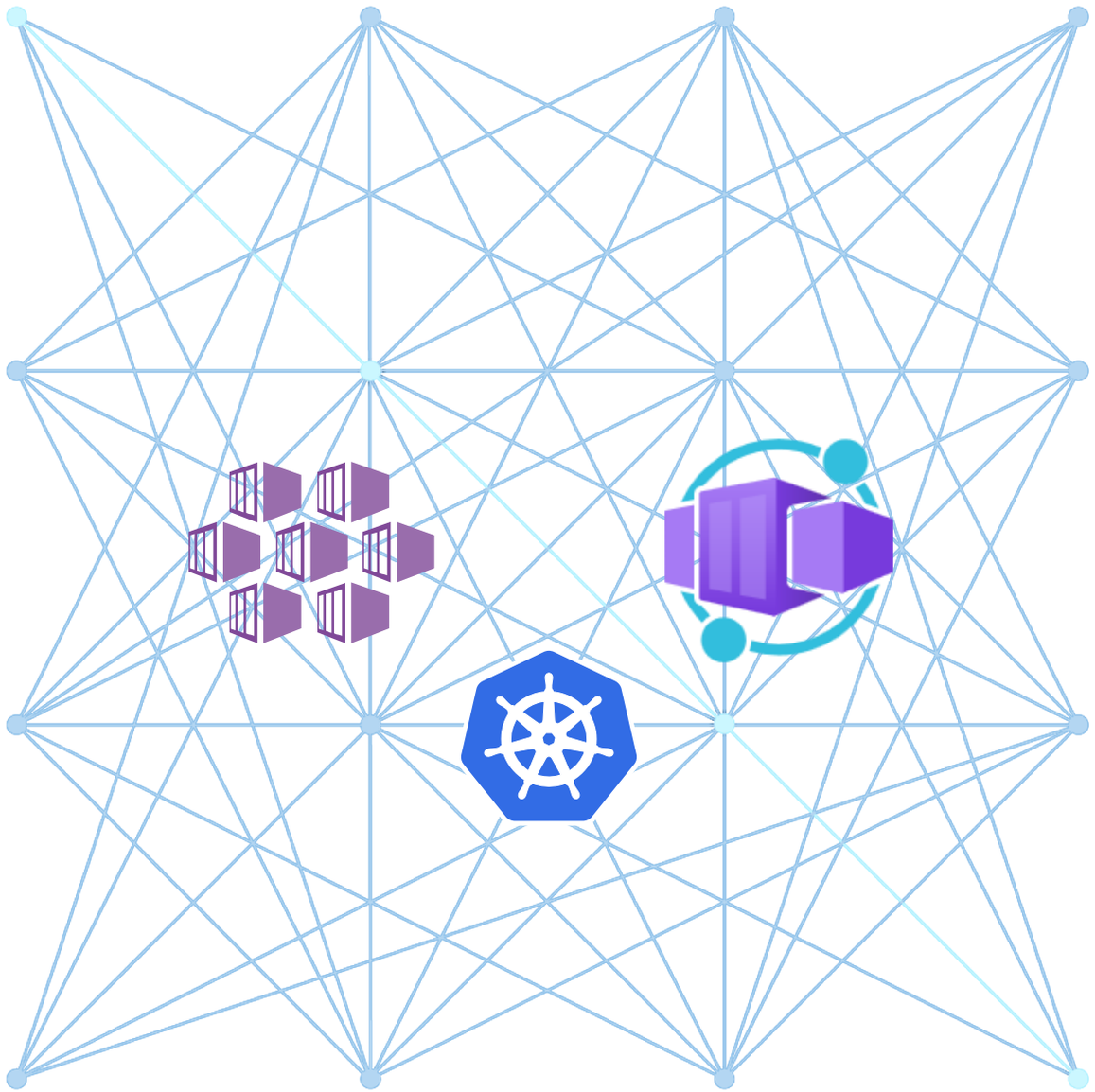
Microsoft Azure

AKS / ACA Deployments

Dan Radu
24 Aug 2022



AKS / ACA Deployments – capabilities assessed

1.

Container image management

Public / private image registries

2.

Infra & IaC

Infrastructure & IaC supported features

3.

Application deployments

Options for application deployments

4.

DevOps

Deployment automation options via Azure DevOps and GitHub

5.

Scaling

Manual / automated scaling

6.

Workloads

Supported workload types

AKS / ACA Deployments – capabilities assessed (cont.)

7.

Ecosystem

Tooling, OSS, integration

8.

Networking

Networking support

9.

Security & Identity

Vulnerability scanning, AuthN / AuthZ

10.

Monitoring

Monitor infrastructure & workloads

11.

Development

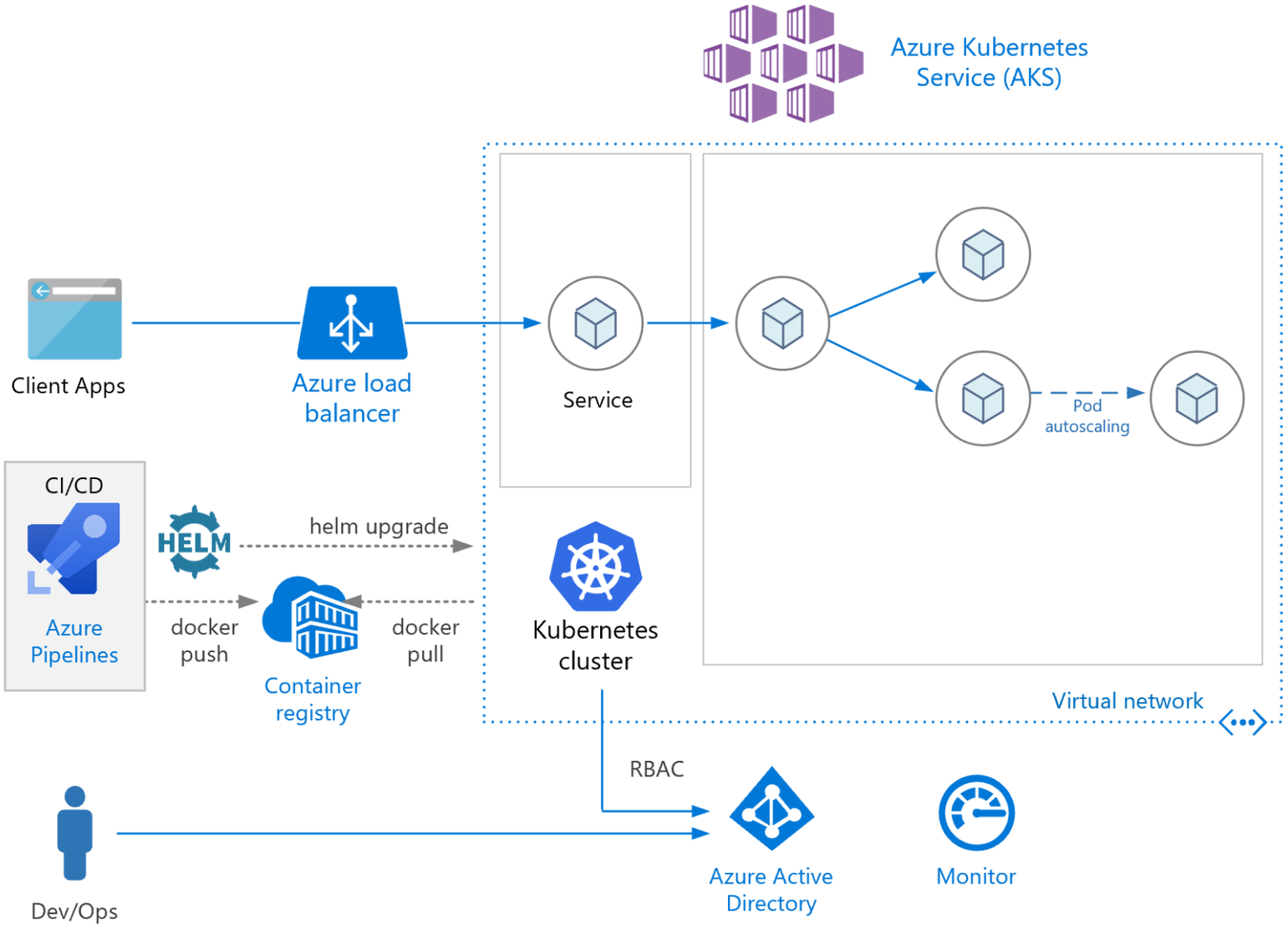
Developer experience, tooling

12.

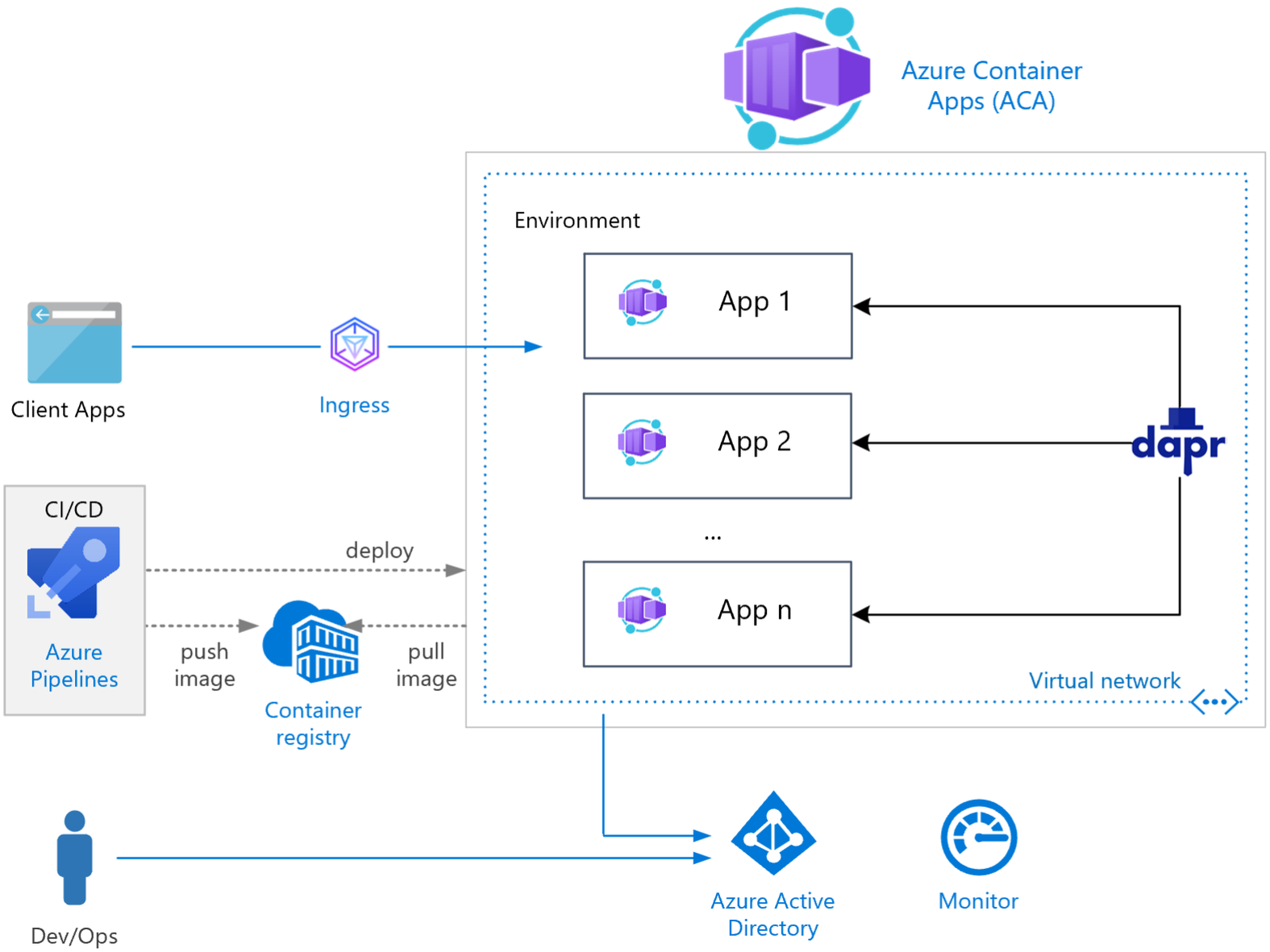
Cost

Cost options

Application Architecture AKS



Application Architecture ACA



Microsoft Azure

Container image management

Deploy images into Azure Container Registry

Manage repositories, namespaces and tags

Scan images

Both AKS & ACA can use public or private container registries

Microsoft Azure

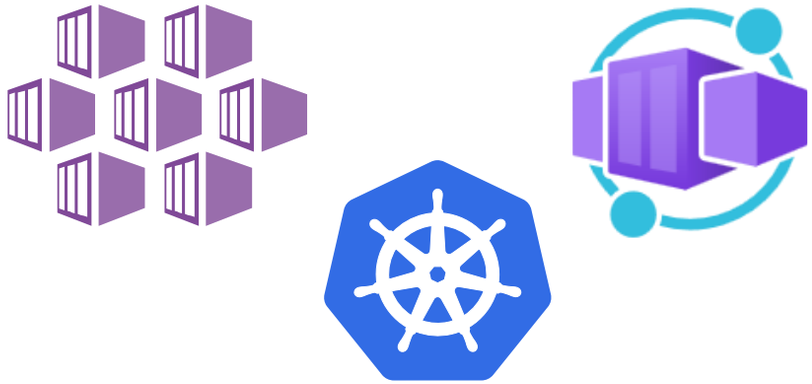
Infra & IaC

- Options for VMs or VMSS
- Node pools
- Control plane managed by Azure
- Data plane managed by service consumer
- Linux / Windows container images
- GPU support
- Availability zones supported, K8S resources require config
- ARM, Bicep, Terraform, az CLI, Az PowerShell support

- Serverless (K8S Infrastructure fully abstracted)
- Control / Data plane abstracted from Azure
- No access to API server (NO kubectl)
- Linux only
- No GPU support
- Simplified zone redundancy and abstracted from end user
- ARM, Bicep, az CLI, Az PowerShell

Microsoft Azure

Application deployments



- Applications deployed using K8S manifests – pods, replica sets, deployments
- K8S expertise required
- Many options: single / multi containers, initContainers, env vars, secrets, volume mapping, health probes (full), CPU / memory requests and limits
- DevOps support
- DAPR full support for K8S (dapr CLI, Helm package, AKS extension)

- Applications deployed via portal or template itself
- No K8S knowledge required
- Limited options: single / multi containers, container image / version, env vars, health probes (no “exec”), CPU / memory requests
- DevOps support
- DAPR OOTB support

Microsoft Azure

DevOps

- Deploy infrastructure and workloads using Azure Pipelines / GitHub Actions;

- Manage environments – strategy

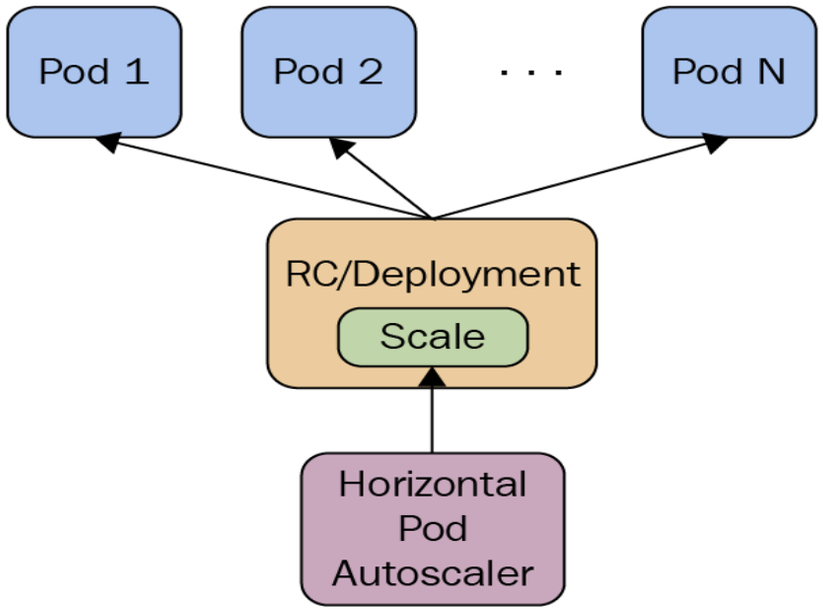
- Potential separate pipelines for infra / applications
- Dedicated K8S tasks – Azure Pipelines
- Helm package deployment
- Rolling update / Canary / Blue-Green strategies for Deployments
- GitOps using Flux, ArgoCD

- Deploy infrastructure using Azure DevOps pipelines / GitHub Actions (workloads included in template);
- Single environment supports multiple container apps
- No separation required

- No dedicated pipeline task – not needed?
- Blue-Green via ingress traffic split

Microsoft Azure

Scaling



- Rich manual and auto-scaling support
- HPA
- Cluster Austoscaler (leverage VMSS)
- CPU / memory rules OOTB
- Other metrics via KEDA

- Transparent HPA OOTB
- Max replicas limited to 30
- Native KEDA integration – variety of metrics

```
$desiredReplicas = ceil($currentReplicas * ($currentMetricValue / $desiredMetricValue))
```

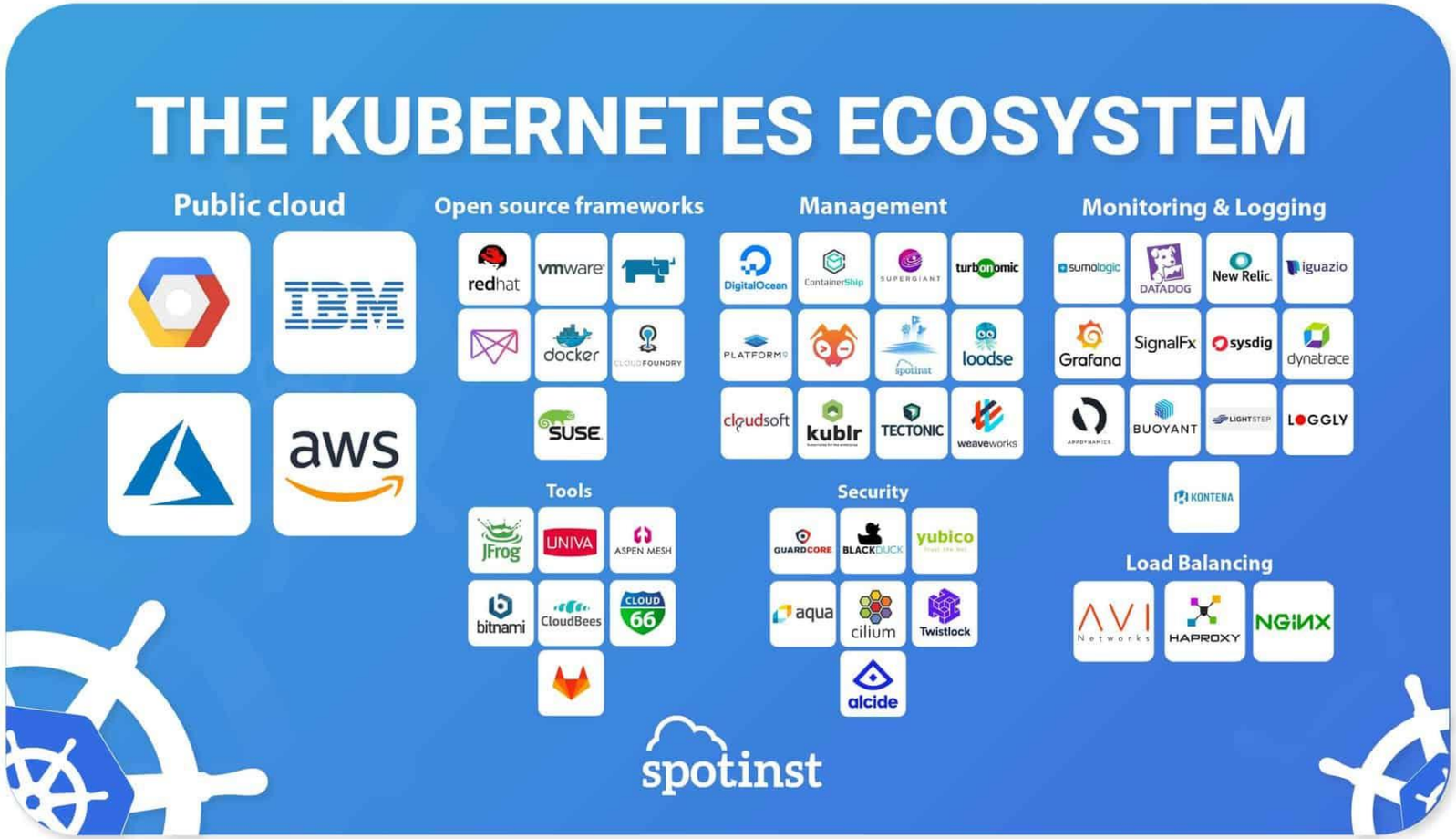
Microsoft Azure

Workloads

- Pods
- Replica sets / controllers
- Deployments
- Stateful sets
- Daemon sets
- Jobs / Cron jobs
- Services / Ingresses
- Config maps, secrets
- PV, PVC, SC
- CRD

- Container apps – defined by container image
- Ingress OOTB
- DAPR integrated

Ecosystem



© SpotInst https://spot.io/blog/kubernetes-ecosystem/

- ACA - new PaaS – not many tools around
- CNCF landscape https://landscape.cncf.io/

Microsoft Azure

Networking

- Native VNET Integration
- Kubenet or Azure CNI - flexibility in IP address management
- Namespace / Pods level isolation using network policies
- Service types: ClusterIP, NodePort, LoadBalancer
- Ingress controllers – default NGINX, others can be used AGIC, Envoy

- Native VNET Integration
- CNI OOTB abstracted from user
- Isolation achieved via Azure container environment
- Ingress backed by Envoy is simplified

Microsoft Azure

Security & Identity

- Secret management OOTB, CSI driver integration with AKV, Hashicorp vault
- No OOTB mTLS support in pods
- Managed identity
- RBAC controls with K8S RBAC or Azure AD RBAC
- Microsoft Defender for Containers (Qualys)
- Runtime protection engines can be used such as Aqua Trivy, Claire, Snyk etc.
- Flexibility in exposing ports for pods, isolation via network policies
- Namespace isolation
- Role, ClusterRole, RoleBinding, ClusterRoleBinding, Service Accounts, Users, Groups

- Secret management limited to Key Value pairs
- No native integration to AKV
- OOTB support for mTLS with dapr integration
- Managed identity* (access to ACR still requires admin enabled)
- Azure RBAC for simplified security controls
- No support for runtime protection (defender for container apps in the future)
- HTTPS endpoint by default

Microsoft Azure

Monitoring

- Native Azure monitor integration
- Many monitoring tools: ELK stack, Prometheus / Grafana
- Logs / Metrics can be shipped to third party tools (Splunk, Sumologic)
- kubectl / API Server

- Native Azure monitor integration
- No support for log / metric shipping capability to third party tools

Microsoft Azure

Development

- K8S / YAML expertise required for K8S manifests
- Typically, an app requires service / ingress, deployment, persistence objects to work in K8S
- Troubleshooting via kubectl / containers debugging
- Bridge to K8S with VS Code / VS

- K8S / YAML understanding is not required for developing apps in ACA
- ACA requires only container images along with required configurations for env vars, DAPR, KEDA
- Dev focus on what's important

Microsoft Azure

Cost

- No cost for control plane, only the nodes attached
- Standard Node based costing (including System Node pool)
- For multi-tenant app cost overview 3rd party tools: kubecost, cloudhealth

- Cost based on:
  - Resource consumption (vCPU-seconds, GiB-seconds)
  - HTTP requests
- Concept of idle state, 0 replicas

Resources

Code
https://github.com/ivee-tech/geekready2022
Docker documentation
https://docs.docker.com/
ACR documentation
https://docs.microsoft.com/en-us/azure/container-registry/
AKS documentation
https://docs.microsoft.com/en-us/azure/aks/
K8S documentation
https://kubernetes.io/docs/home/
ACA documentation
https://docs.microsoft.com/en-us/azure/container-apps/

Microsoft Azure

Thank you.