A Kubernetes Story

Erica Barone
Software Engineer

@_ericabarone

Jessica Tibaldi
Technology Solutions Professional

@_jetiba

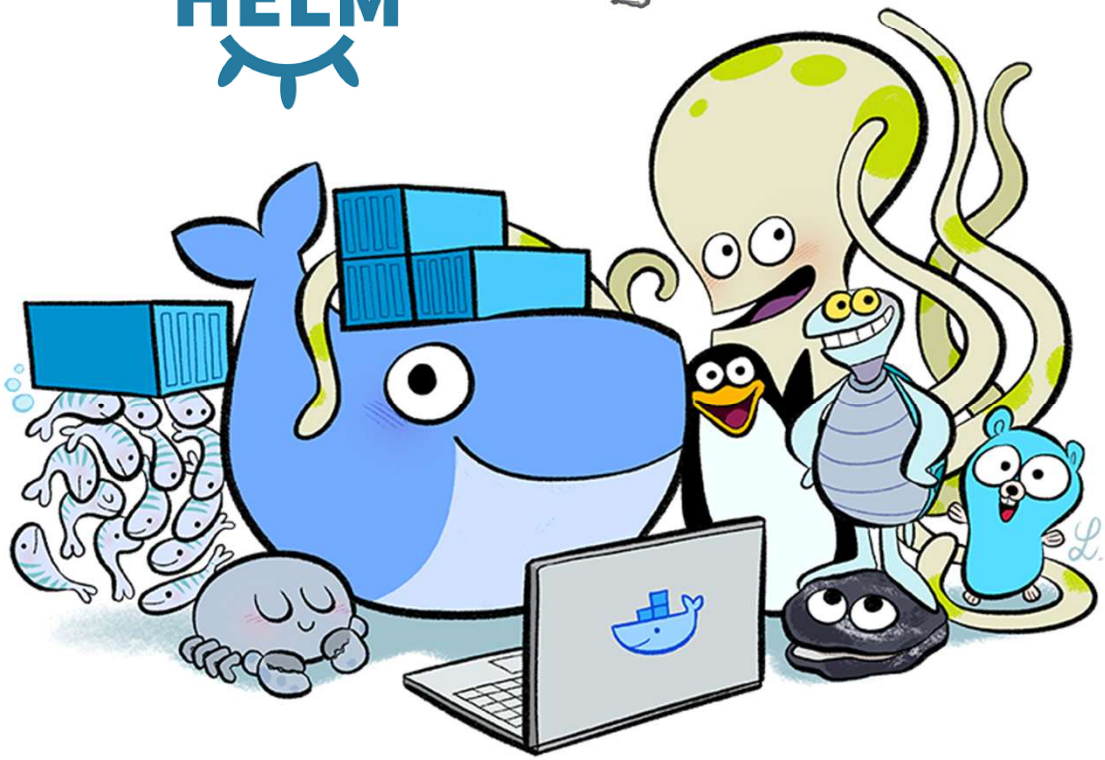
CommunityDays.it

Once upon a time...

Seven of Nine

KUBERNETES

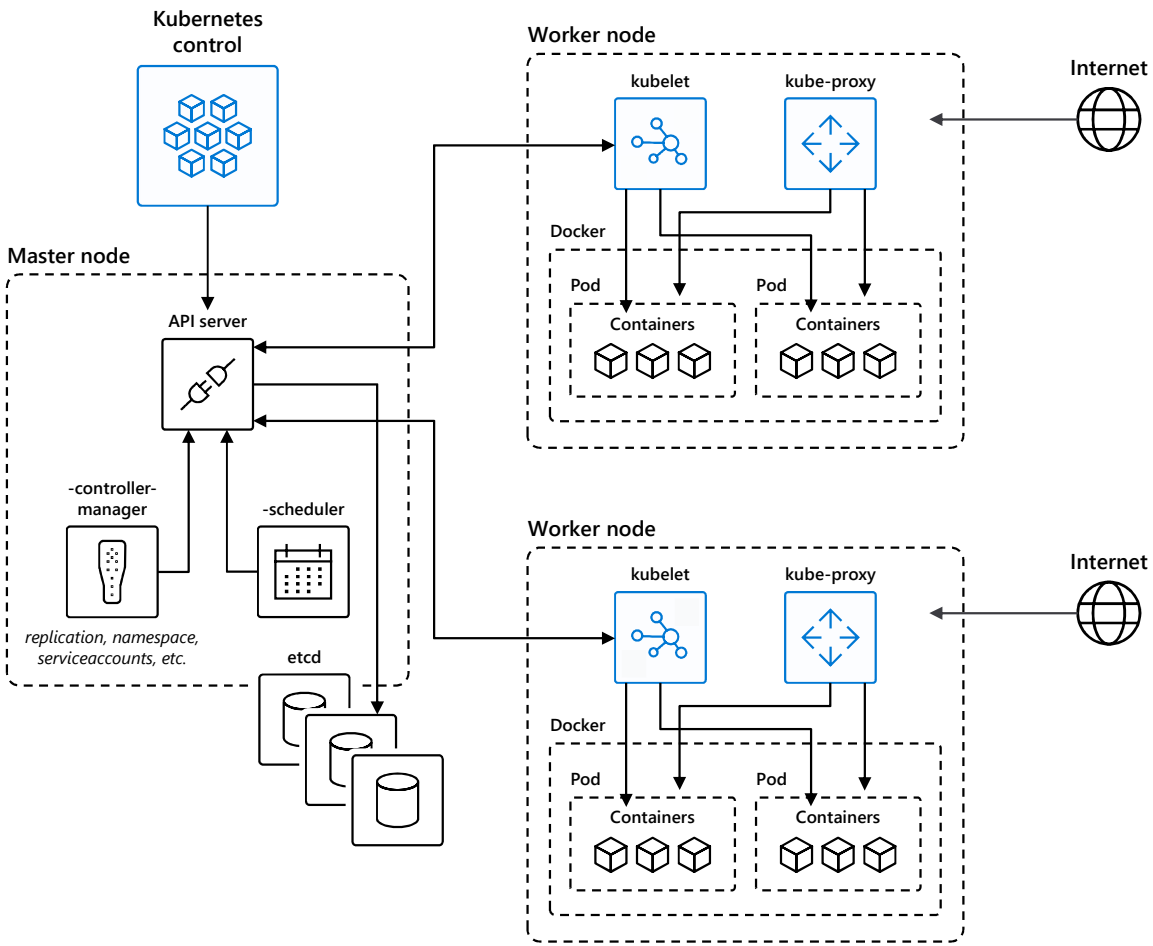
HELM



CommunityDays.it

How Kubernetes works

1. Kubernetes users communicate with API server and apply desired state
2. Master nodes actively enforce desired state on worker nodes
3. Worker nodes support communication between containers
4. Worker nodes support communication from the Internet

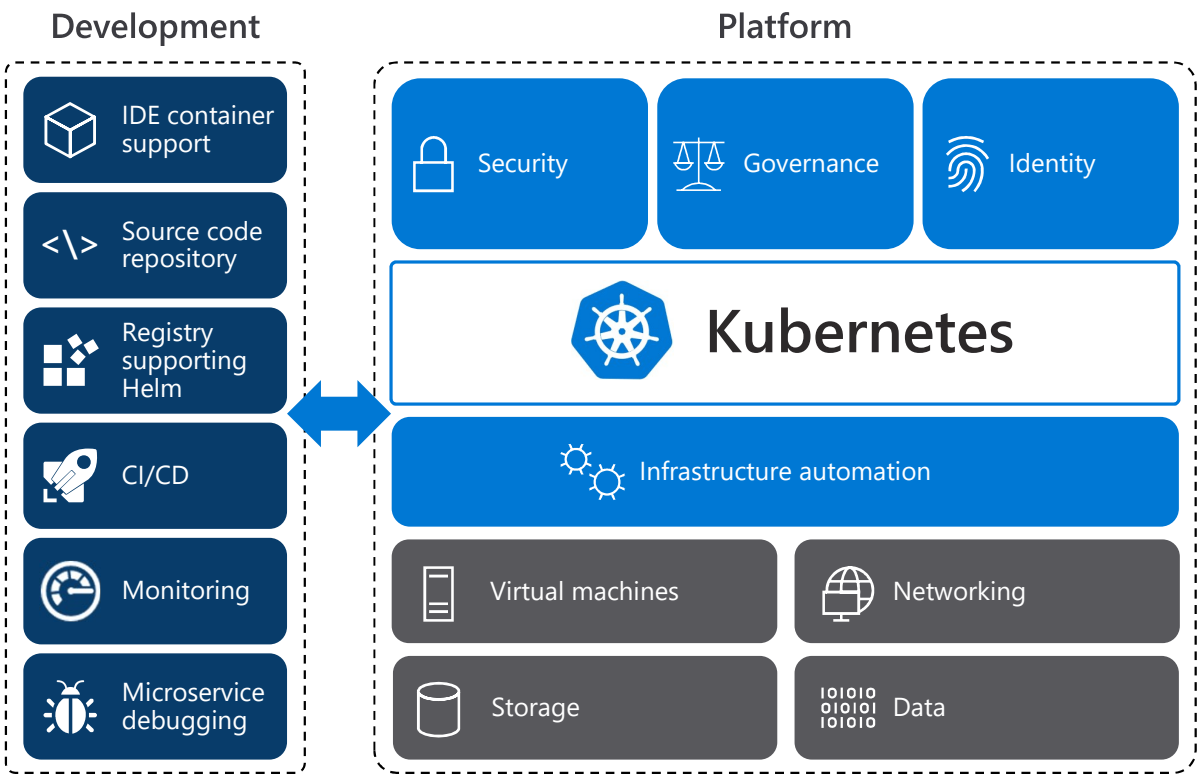


Kubernetes on its own is not enough

Save time from infrastructure management and roll out updates faster without compromising security

Unlock the agility for containerized applications using:

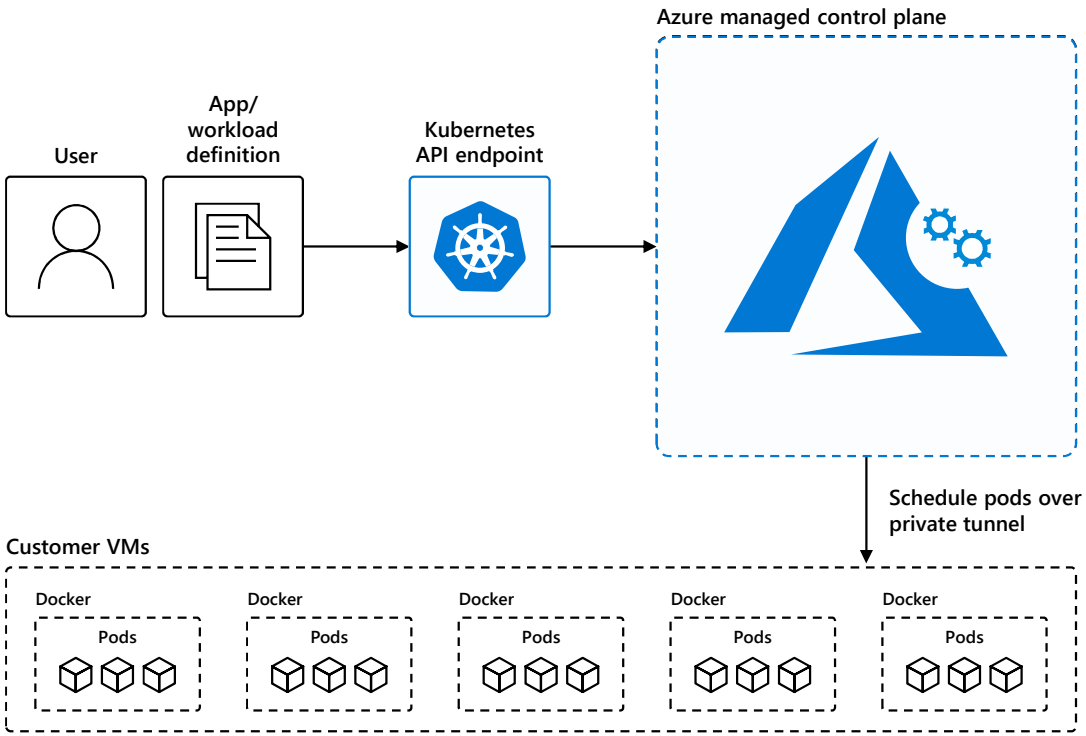
- **Infrastructure automation** that simplifies provisioning, patching, and upgrading
- Tools for **containerized app development and CI/CD workflows**
- Services that support **security, governance, and identity and access management**



Manage Kubernetes with ease

Infrastructure automation

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



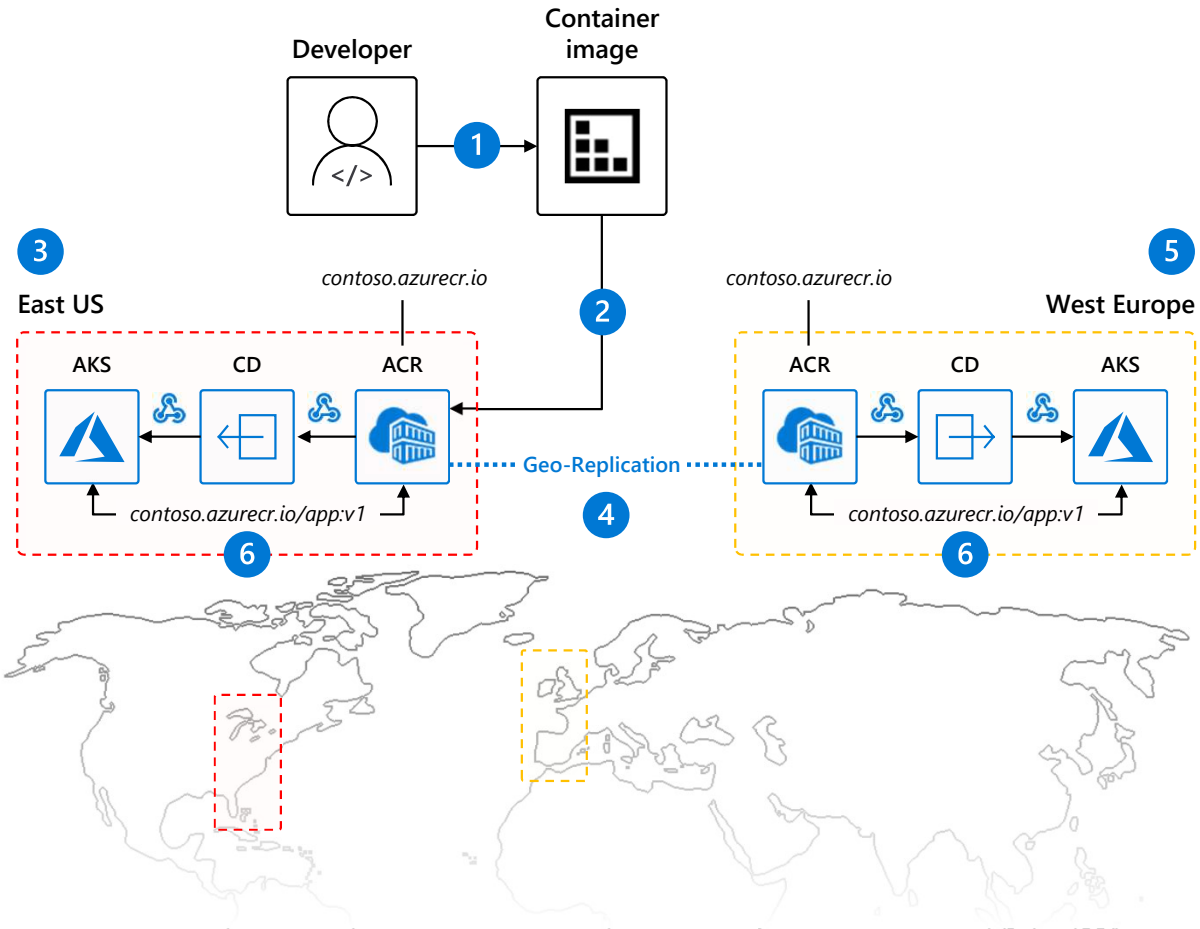
Demo

Create a new AKS cluster

CommunityDays.it

Azure Container Registry

- Private, globally distributed images registry on Azure
- Access control via Azure AD
- Open Source Docker CLI
- ACR Task



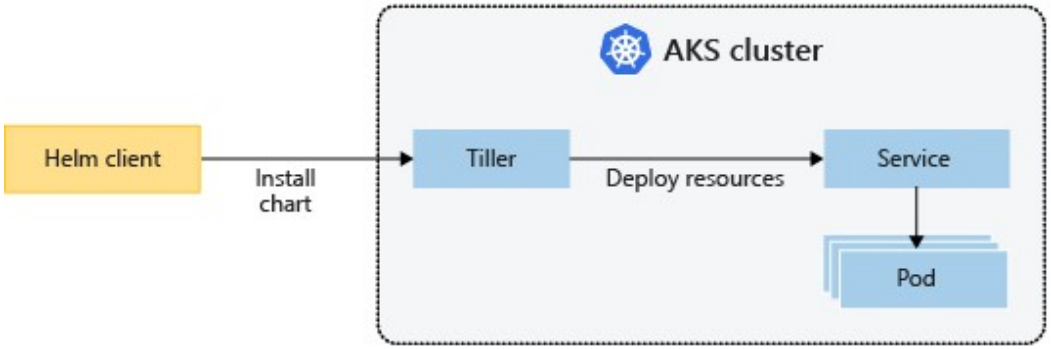
Demo

Azure Container Registry – sample image pull

CommunityDays.it

Helm

- Open-source managing tool
- Charts and Values
- Thanks to the server component, Tiller, installed in the cluster, the communication between the local machine (client) and the cluster is secured during deployments.
  - TLS/SSL used
  - RBAC to control the level of access that the Tiller has to the cluster
- Tiller maintains the status of charts deployment remotely



Demo

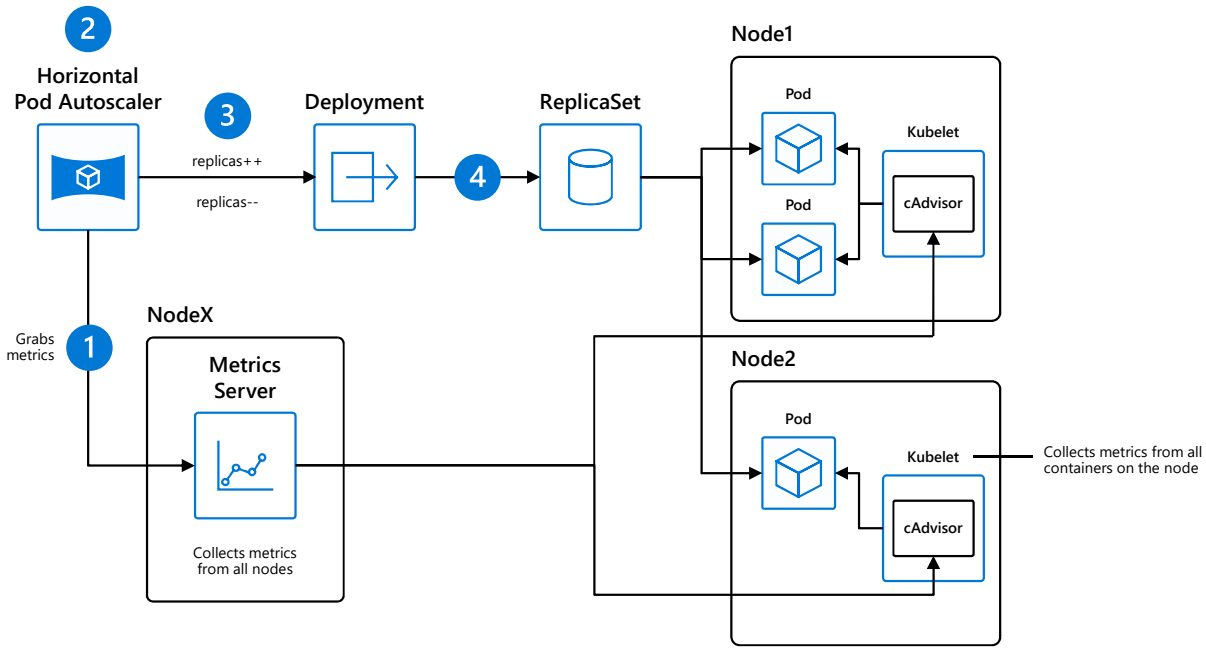
Helm Charts and Values

CommunityDays.it

Horizontal Pod Autoscaler

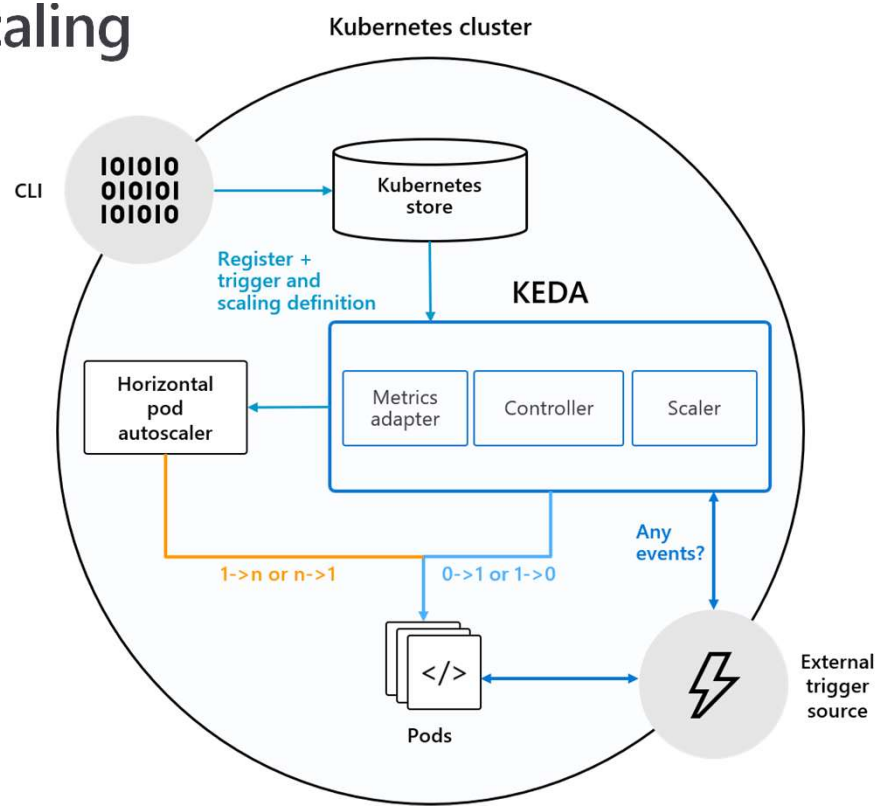
The horizontal pod autoscaler (HPA) uses the Metrics Server in a Kubernetes cluster to monitor the resource demand of pods. If a service needs more resources, the number of pods is automatically increased to meet the demand.

1. HPA obtains resource metrics and compares them to user-specified threshold
2. HPA evaluates whether user specified threshold is met or not
3. HPA increases/decreases the replicas based on the specified threshold
4. The Deployment controller adjusts the deployment based on increase/decrease in replicas



Kubernetes-based Event Driven Autoscaling

- Developed in partnership with Red Hat (experimental)
- Allows fine grained autoscaling (including to/from zero) for event driven Kubernetes workload
- 2 components
  - Custom Metrics Server
  - Scaler Agent
- Open-source project on Github https://github.com/kedacore/keda



Demo

Autoscaling from zero to the infinity ☺ with KEDA

CommunityDays.it

KEDA ScaledObject yml sample

```yaml
apiVersion: keda.k8s.io/v1alpha1
kind: ScaledObject
metadata:
  name: {scaled-object-name}
  labels:
    deploymentName: {deployment-name} # must be in the same namespace as the ScaledObject
spec:
  scaleTargetRef:
    deploymentName: {deployment-name} # must be in the same namespace as the ScaledObject
    containerName: azure-functions-container  #Optional. Default: deployment.spec.template
  pollingInterval: 30  # Optional. Default: 30 seconds
  cooldownPeriod:  300 # Optional. Default: 300 seconds
  minReplicaCount: 0   # Optional. Default: 0
  maxReplicaCount: 100 # Optional. Default: 100
  triggers:
  # {list of triggers to activate the deployment}
```

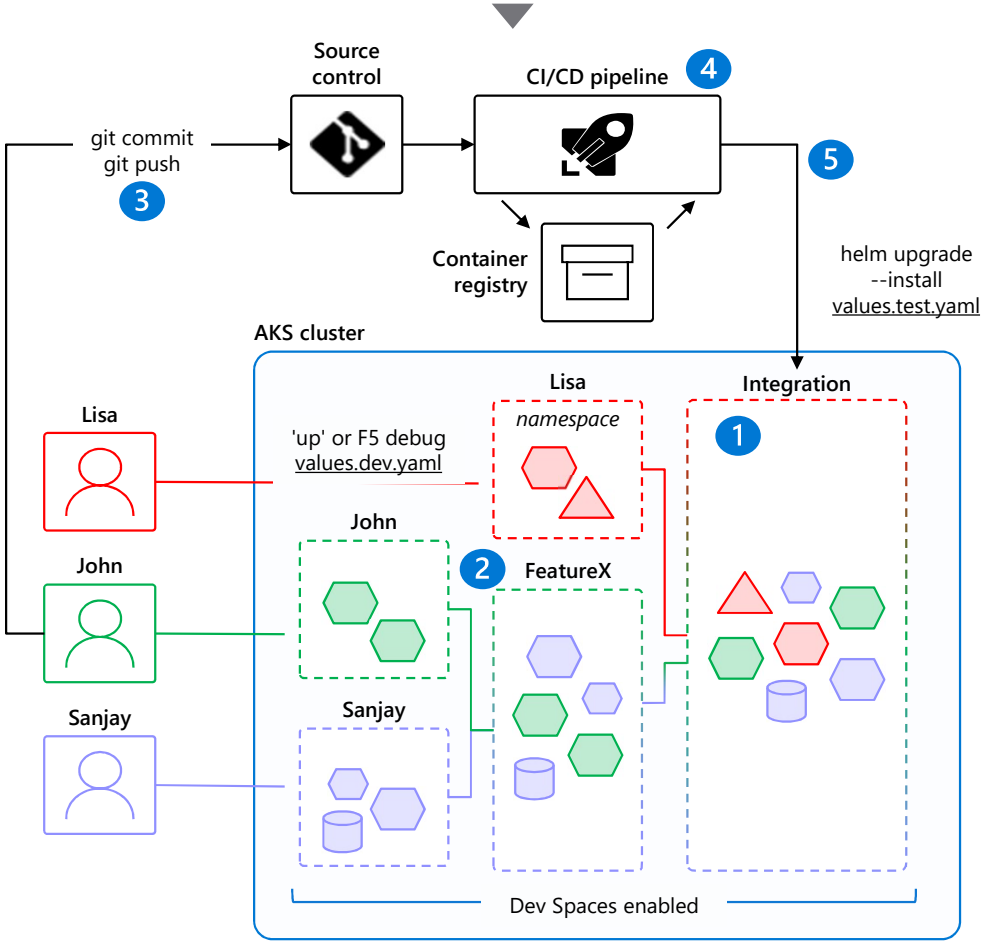
sample

```yaml
triggers:
- type: azure-queue
  metadata:
    queueName: functionsqueue
    queueLength: '5' # Optional. Queue length target for HPA. Default: 5 messages
    connection: STORAGE_CONNECTIONSTRING_ENV_NAME
```

CommunityDays.it

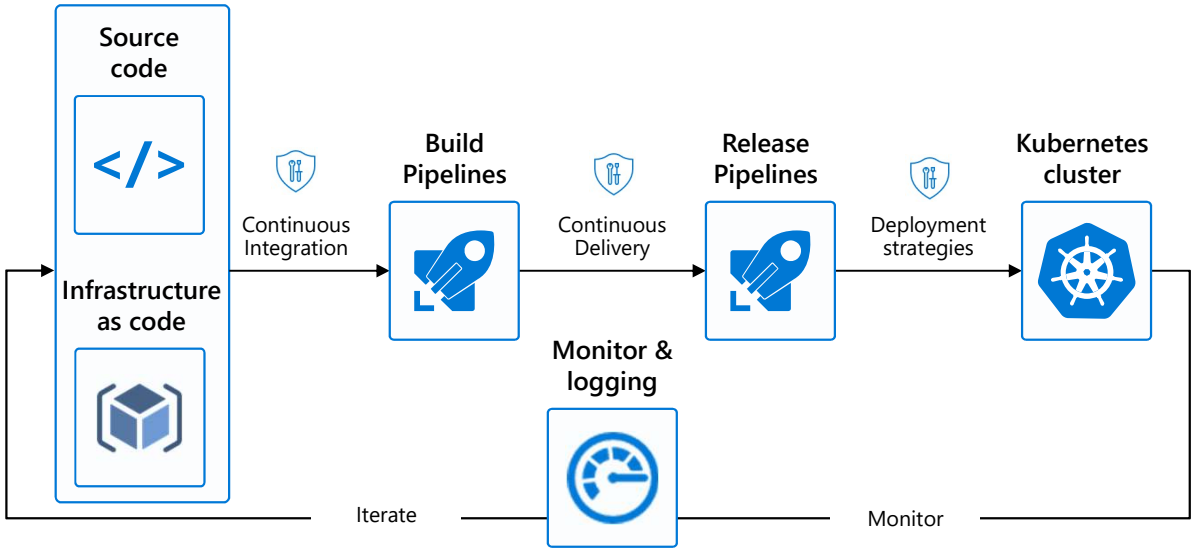
Azure Dev Spaces

- Minimize local machine configuration
- Using namespaces to separate development environment
- Shared dev spaces for features in development
- Test in a production-like environment without the need to redeploy the entire microservices solution



Azure Pipelines

- Component of **Azure DevOps**
- Allows CI/CD scenarios
- **Build pipelines** (YAML editor) are triggered from changes in the source code and push Docker images in ACR
- **Release pipeline** can manage deployment of infrastructure (via ARM Template or other tools like Terraform) and images created in the build phase



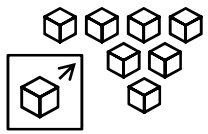
Demo

Using Terraform, Helm and Azure DevOps to deploy AKS cluster and application code **all at once**

CommunityDays.it

Top scenarios for Kubernetes on Azure

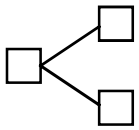
Lift and shift to containers



Cost saving

minimizing refactoring your app

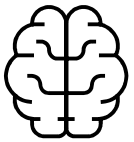
Microservices



Agility

Faster application development

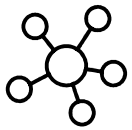
Machine learning



Performance

Low latency processing

IoT



Portability

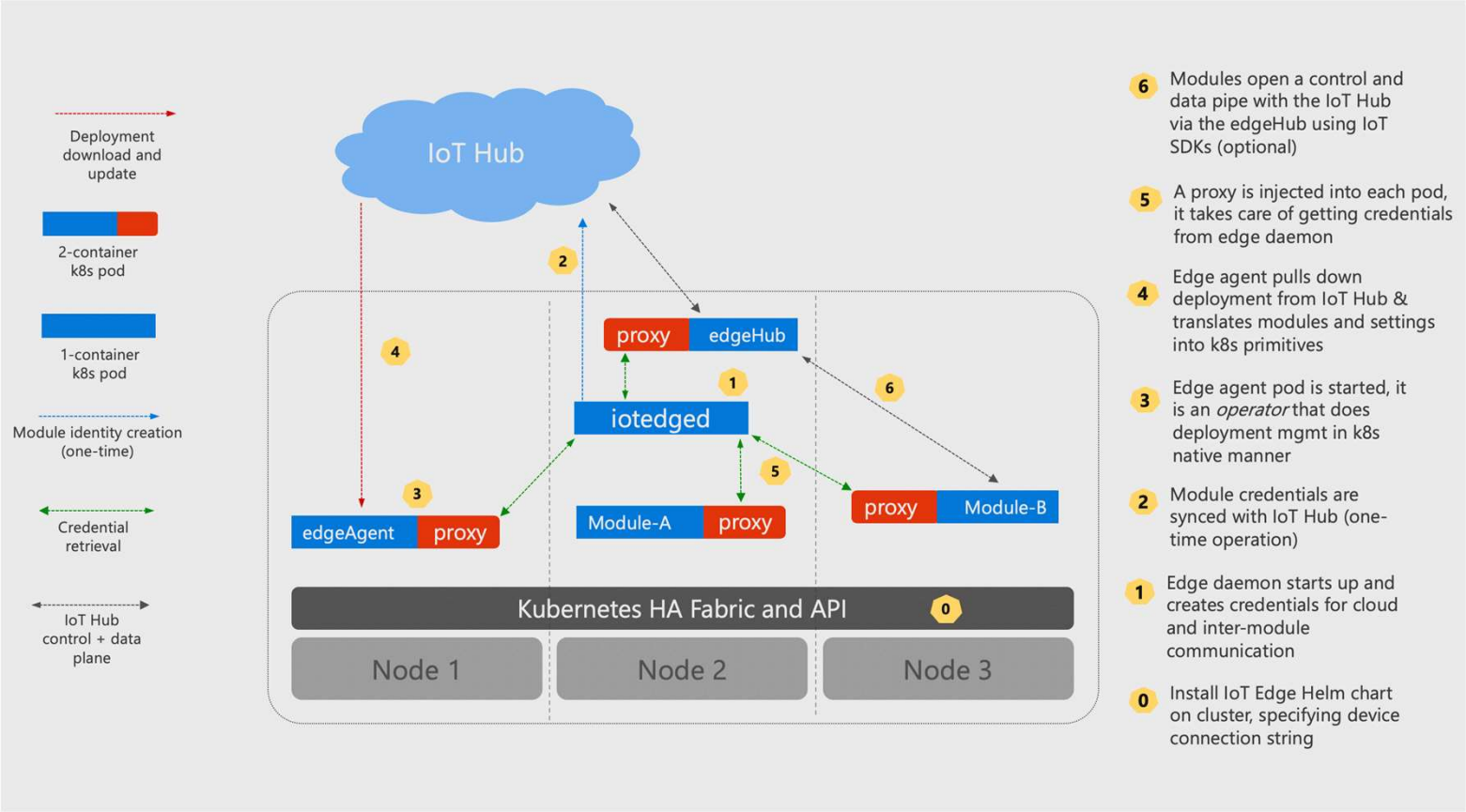
Build once, run anywhere

DevSecOps

Security

Deliver code faster and securely at scale

IoT Edge on Kubernetes (Preview)



© Microsoft Corporation

Demo

Deploy IoT Edge on a Kubernetes cluster

CommunityDays.it

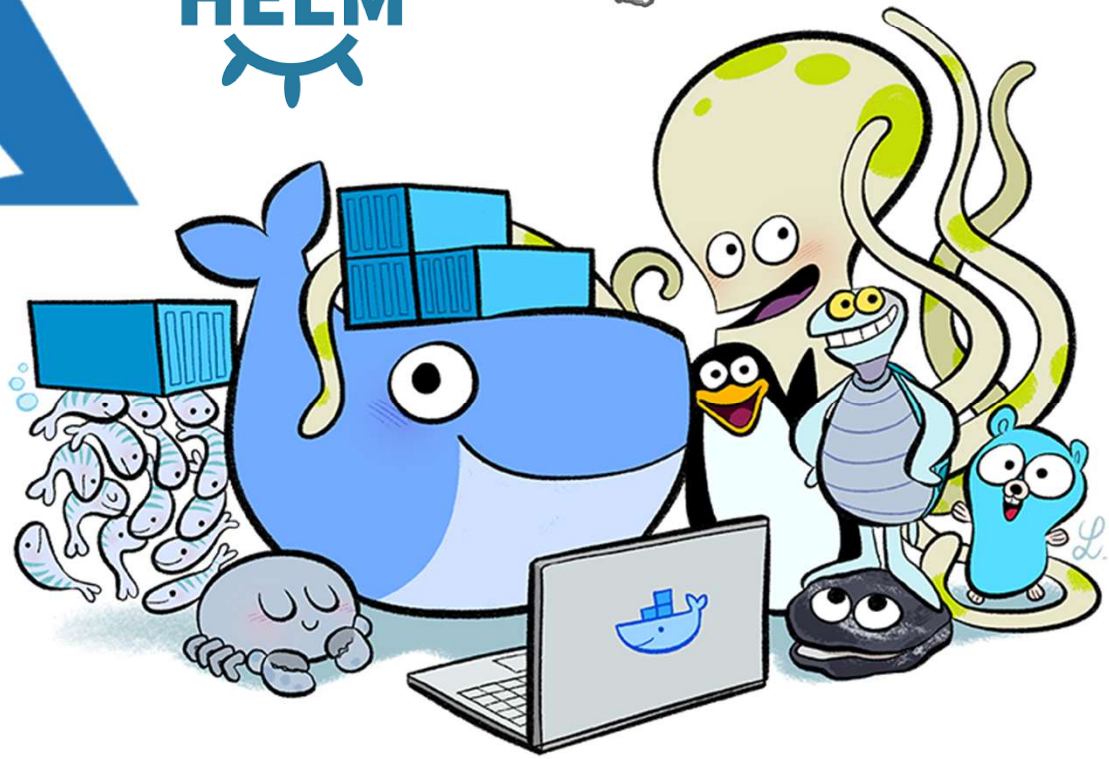
DEMO

```
helm init
helm repo add edgek8s https://edgek8s.blob.core.windows.net/helm/
helm repo update

helm install --name k8s-edge1 --set "deviceConnectionString=<your-device-connection-string>" edgek8s/edge-kubernetes

kubectl get pods -n msiot-<your-iot-hub>-<your-device-id>
```

CommunityDays.it

And they lived happily ever after ☺

Seven of Nine



CommunityDays.it

Grazie!

- Il materiale sarà online nei prossimi giorni su http://www.communitydays.it

CommunityDays.it