Microsoft

Azure

Modern Jakarta EE and Microprofile on Azure: Open Liberty, JBoss EAP, and WebLogic

Ed Burns @edburns
Principal Architect
Java on Azure

Sandra Ahlgrimm @edburns
Cloud Advocate, Java Entwicklerin,
DevOps Enthusiast

Our plan for your time investment

- Morning
  - Common set up for the rest of the day
  - Azure services used and not used in this workshop
  - JBoss EAP on Azure App Service workshop
- Afternoon
  - Open Liberty on Azure Kubernetes Service
  - WebLogic Server on Azure Kubernetes Service

Professional Biography

Client

- NCSA Mosaic (1994)
- SGI Cosmo Web Authoring
- Sun Netscape 6 OJI

Server

- J2EE JSF (2002)
- Oracle Java EE
  - Servlet, JSF, Bean Validation, etc.
- Microsoft Azure Cloud (2019)

Books

Professional Biography

Cloud Advocate II

- Docker expert
- Visual Studio Code and GitHub Codespaces specialist
- DevOps and CI/CD practitioner

Software Developer and Scrummaster

- Spring Boot
- DC/OS
- Microservices

Homepage

https://ahlgrimm.dev/

Sandra Ahlgrimm

Common set up for the rest of the day

https://aka.ms/javaland-javaee

How Microsoft delivers its Azure offerings

Meet developers where they are

Cross language

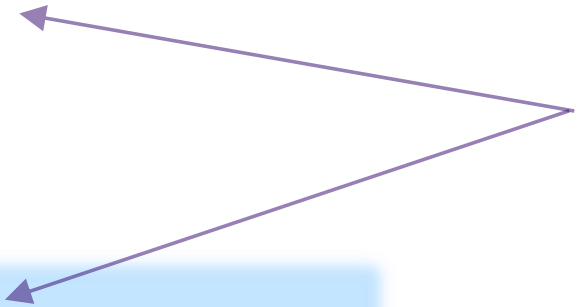
- Azure CLI
- Azure PowerShell
- Azure SDK
- REST APIs
- Infrastructure as code
  - Terraform
  - ARM/Bicep
- Azure Portal

Java specific

- IDE extensions
  - Visual Studio Code
  - IntelliJ Idea
- Maven/Gradle plugins
- Spring starters

How Microsoft explains its Azure offerings

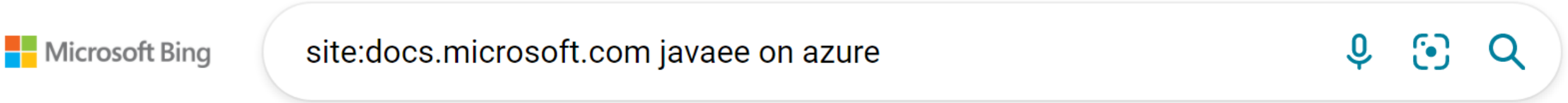
- Brand name: marketing focused
  - Can and does change over time
- Resource provider name: used by ARM
  - Does not change over time
- azure.microsoft.com content
  - Tells why to use it
  - IT manager focused
  - Useful, but also glossy brochureware
- docs.microsoft.com content
  - Tells how to use it to get things done
  - Developer focused content
  - Authoritative source
- MS Learn content



Mess of aka.ms links

How Microsoft explains its Azure offerings

1. Use **`site:docs.microsoft.com`** in search engine
   A. Look for the **Overview** pages



2. Use the template reference https://aka.ms/azref
3. Search for a learn module https://learn.microsoft.com/
4. ~~Look at the javadocs~~

Azure for architects and team leaders

Azure Architecture Center

https://docs.microsoft.com/en-us/azure/architecture/

Azure Spring Cloud Reference Architecture

https://aka.ms/azspringrefarch

Azure for architects and team leaders

Azure pricing calculator

https://azure.microsoft.com/en-us/pricing/calculator/

Engagement from Microsoft

Cloud Solution Architects

Customer Success Team

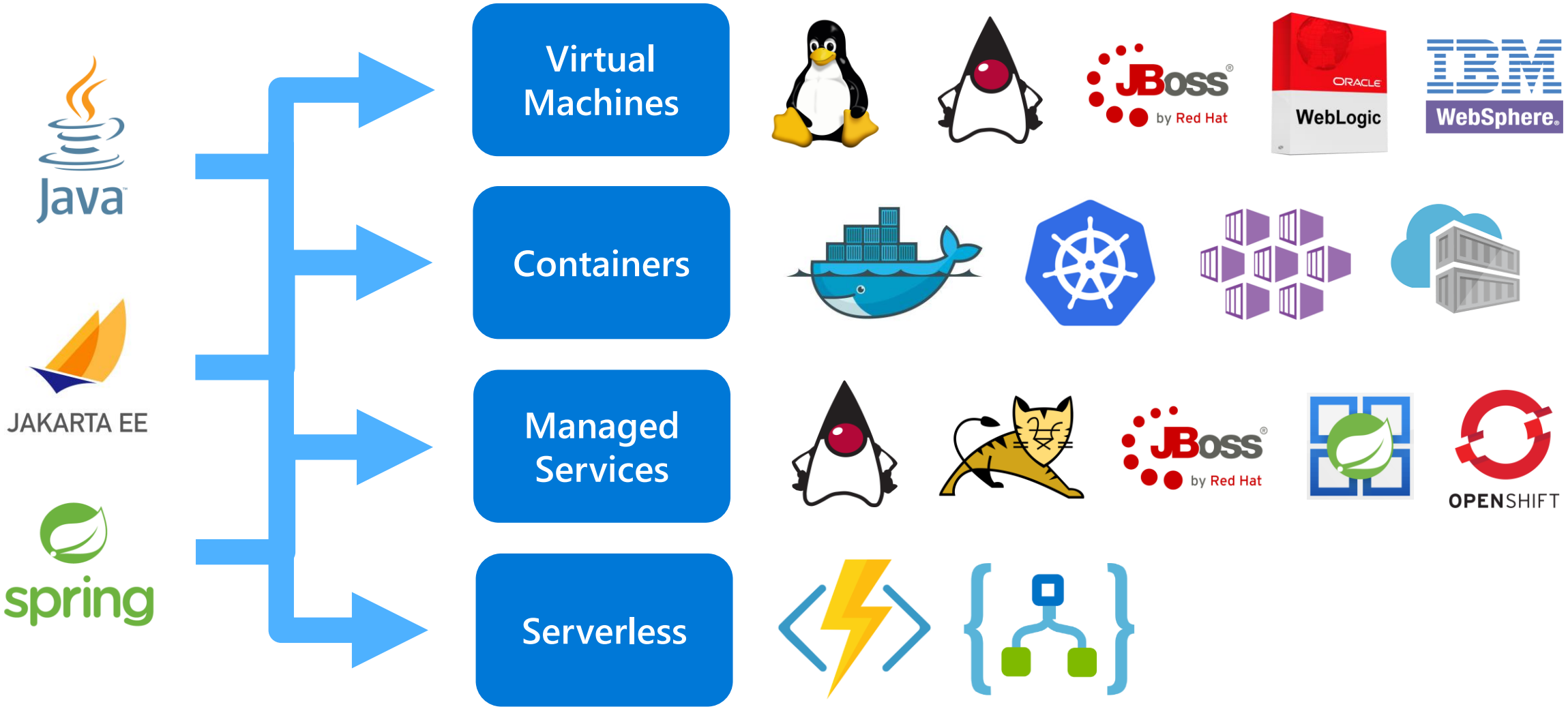
Azure certification

https://docs.microsoft.com/en-us/learn/certifications/azure-fundamentals/

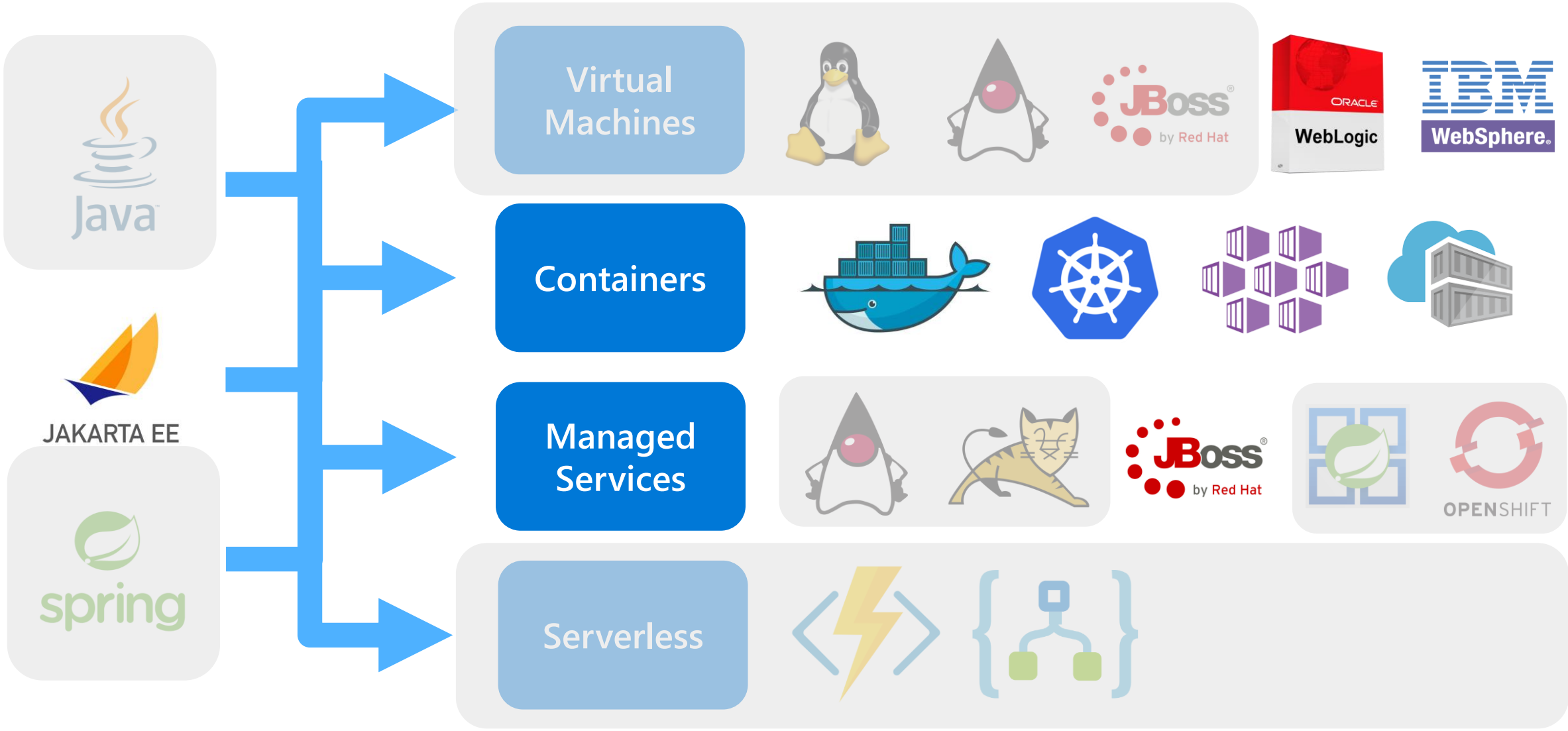
Azure services used in this workshop

- Azure App Service
  - JBoss EAP stack
- Azure Kubernetes Service
  - IBM Liberty
  - Oracle WebLogic
- Azure Database for PostgreSQL

Abundance of Choice



Abundance of Choice



JBoss EAP on Azure App Service

https://github.com/Azure-Samples/workshop-migrate-jboss-on-app-service

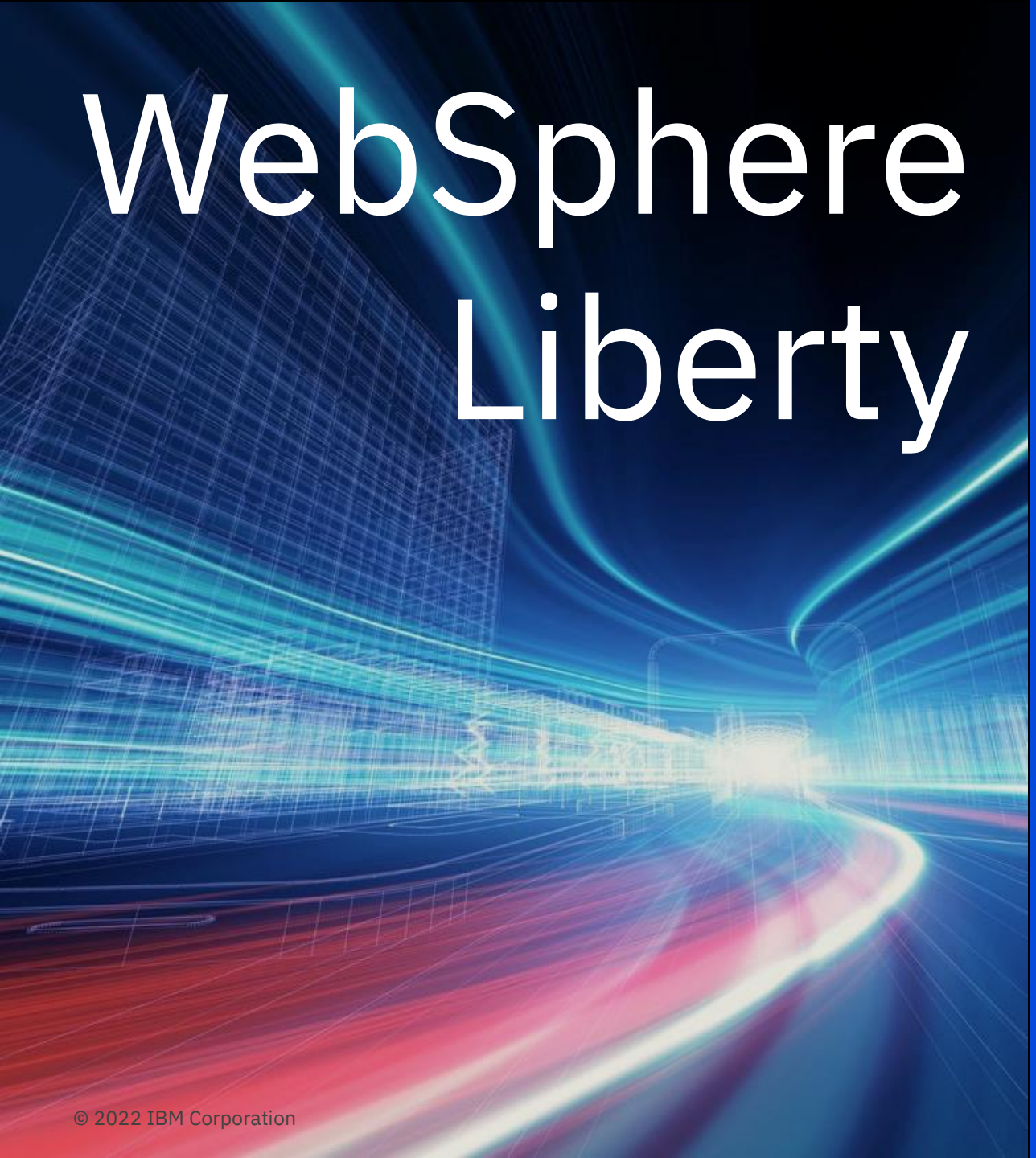
Open Liberty on AKS

https://aka.ms/javaland-javaee

Select Open Liberty on AKS link

Execute steps in **Deploy the minimum viable cluster with the Portal**

Instructor lead slides during deployment



WebSphere Liberty

World-leading application runtime for new cloud-native and modernized workloads. Lightweight, efficient and simple to use enabling businesses to reduce costs and increase agility.

EJ

IBM WebSphere Liberty

1 50% increase in developer productivity

2 40% increase in IT admin productivity

3 195% ROI Payback 8 months

Cloud-Native Development

- Simple rapid inner-loop developer experience in any IDE
- Optimized for Containers and Kubernetes
- Optimized for Continuous Integration, Continuous Delivery

Operational Resource Optimization

- Reduce costs with world-leading performance for microservices and monoliths
- Auto-tuning for continuous optimal performance in any environment
- Simple operator-based management in Kubernetes

Application Modernization

- API & configuration compatibility for reduced effort and risk
- HA clustering in Containers and VMs
- Transformation Advisor & Mono2Micro tools help plan and execute move to container and microservices

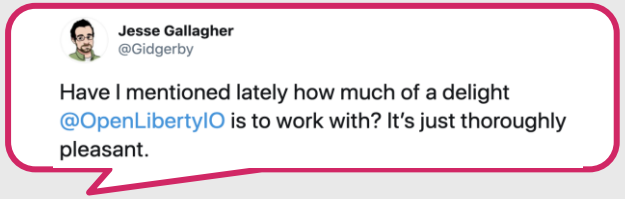
https://ibm.biz/WSHE-TEI

EJ

WebSphere Liberty

Rapid inner-loop developer experience – Dev Mode

- ✓ Immediate code and config change feedback without a rebuild
- ✓ Re-run Integration Tests without a rebuild
- ✓ In any IDE or editor, even Vi!
- ✓ Including in Containers for dev-prod parity



"I just love Liberty."

Developer in Top Insurance Company, Norway

WebSphere Liberty

Containers and Kubernetes Optimized

- ✓ Pre-built containers for UBI & Ubuntu
- ✓ Production-ready, pre-optimized for performance, supported on any Kubernetes & OpenShift
- ✓ Available on Docker Hub and IBM Container Registry
- ✓ Optimized to each application – no excess baggage
- ✓ Designed for Observability & Kubernetes Lifecycle

IBM Cloud Container Registry

docker hub

OPEN CONTAINER INITIATIVE

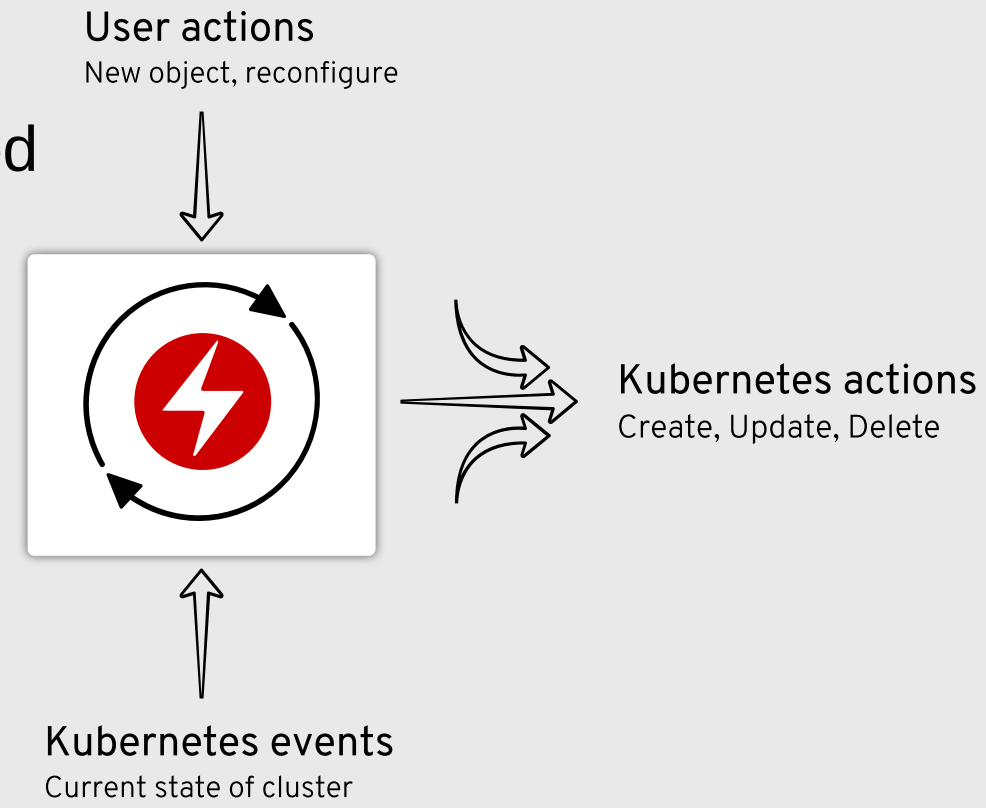
kubernetes

OPENSHIFT

EJ

Embracing Operators

- Extends Kubernetes functionality
- Makes use of Custom Resource Definitions (CRDs)
- Holds the knowledge of how an application needs to be packaged, deployed and managed



Without an Operator

- Day 1 steep learning curve
- Inconsistency across the enterprise and GitOps
- **Lack of day 2 operations and synchronization**

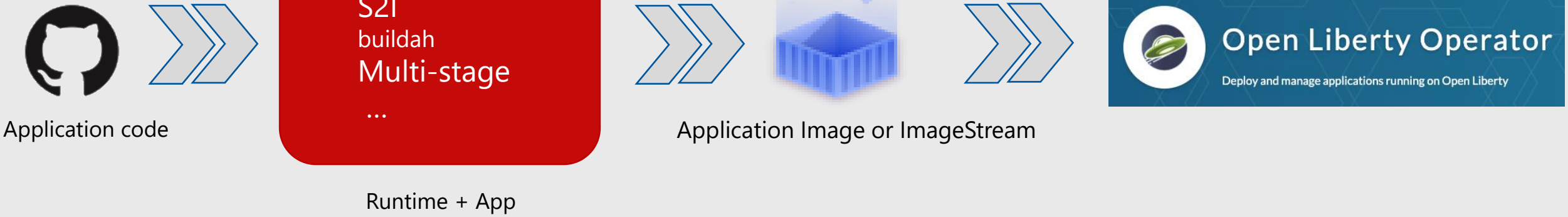
With an Operator

IBM

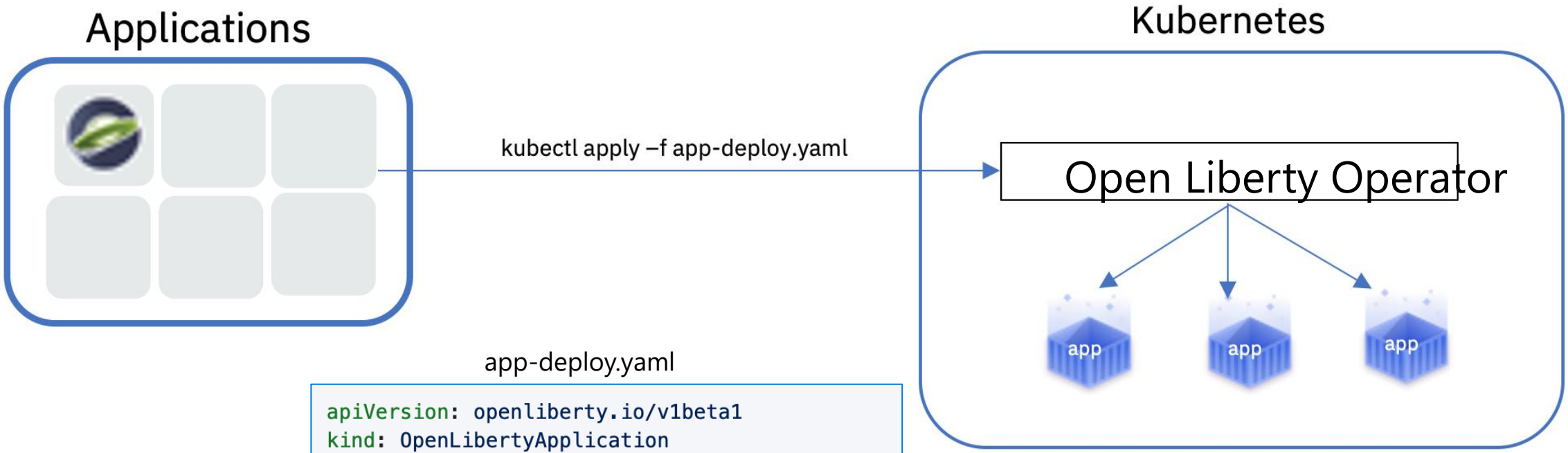
OpenLibertyApplication

- Enables:
  auto-synchronization of runtime resources, day-2 operations
- Improves:
  usage, maintenance, consistency

Input: Application image or ImageStream



Open Liberty Operator



app-deploy.yaml

```yaml
apiVersion: openliberty.io/v1beta1
kind: OpenLibertyApplication
metadata:
  name: my-liberty-app
spec:
  applicationImage: quay.io/my-repo/my-app:1.0
  service:
    type: ClusterIP
    port: 9080
  expose: true
  storage:
    size: 2Gi
    mountPath: "/logs"
```

Open Liberty Operator

OperatorHub: Maturity level 5

- https://github.com/OpenLiberty/open-liberty-operator
  https://operatorhub.io/operator/open-liberty

Basic Install

Automated application provisioning and configuration management

Seamless Upgrades

Patch and minor version upgrades supported

Full Lifecycle

App lifecycle, storage lifecycle (backup, failure recovery)

Deep Insights

Metrics, alerts, log processing and workload analysis

Auto Pilot

Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

EB

WebLogic Server on AKS

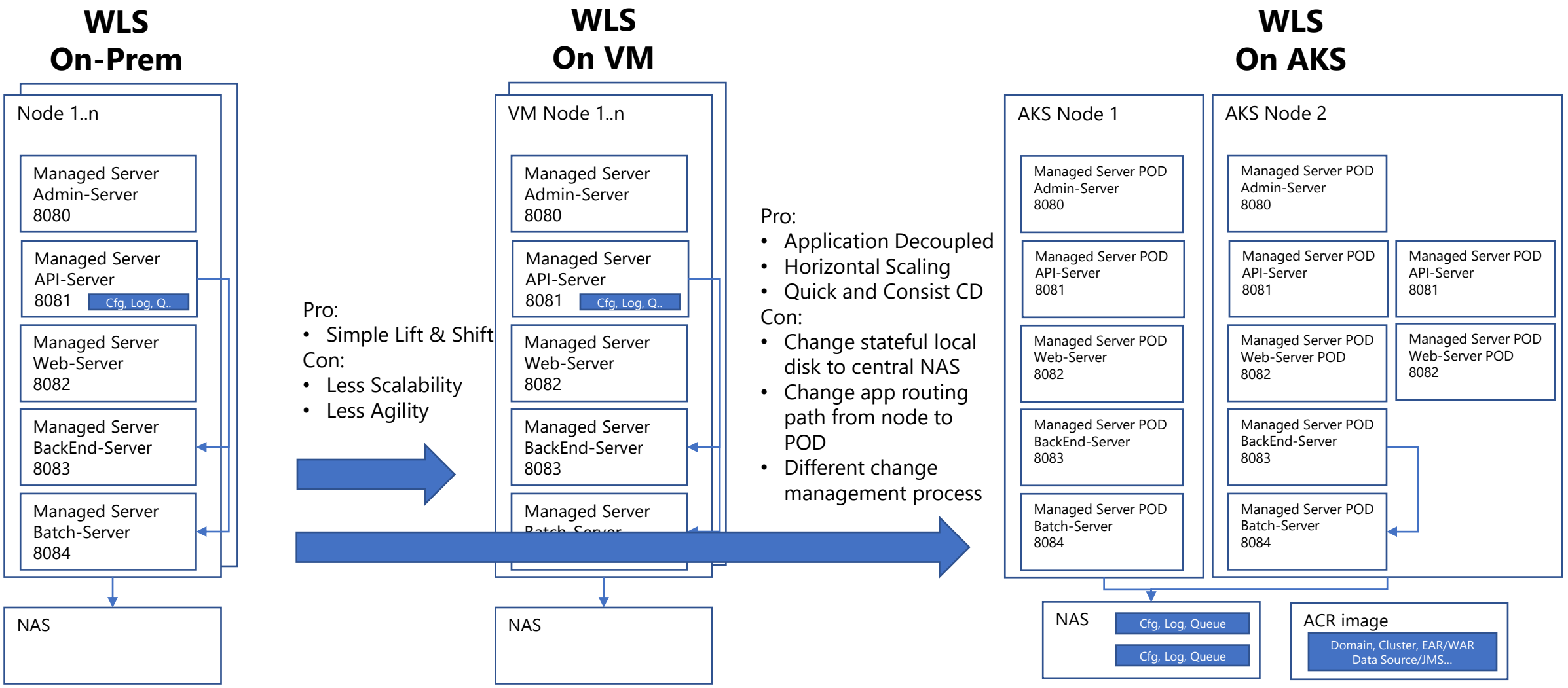
https://aka.ms/javaland-javaee

Select SebLogic Server on AKS link

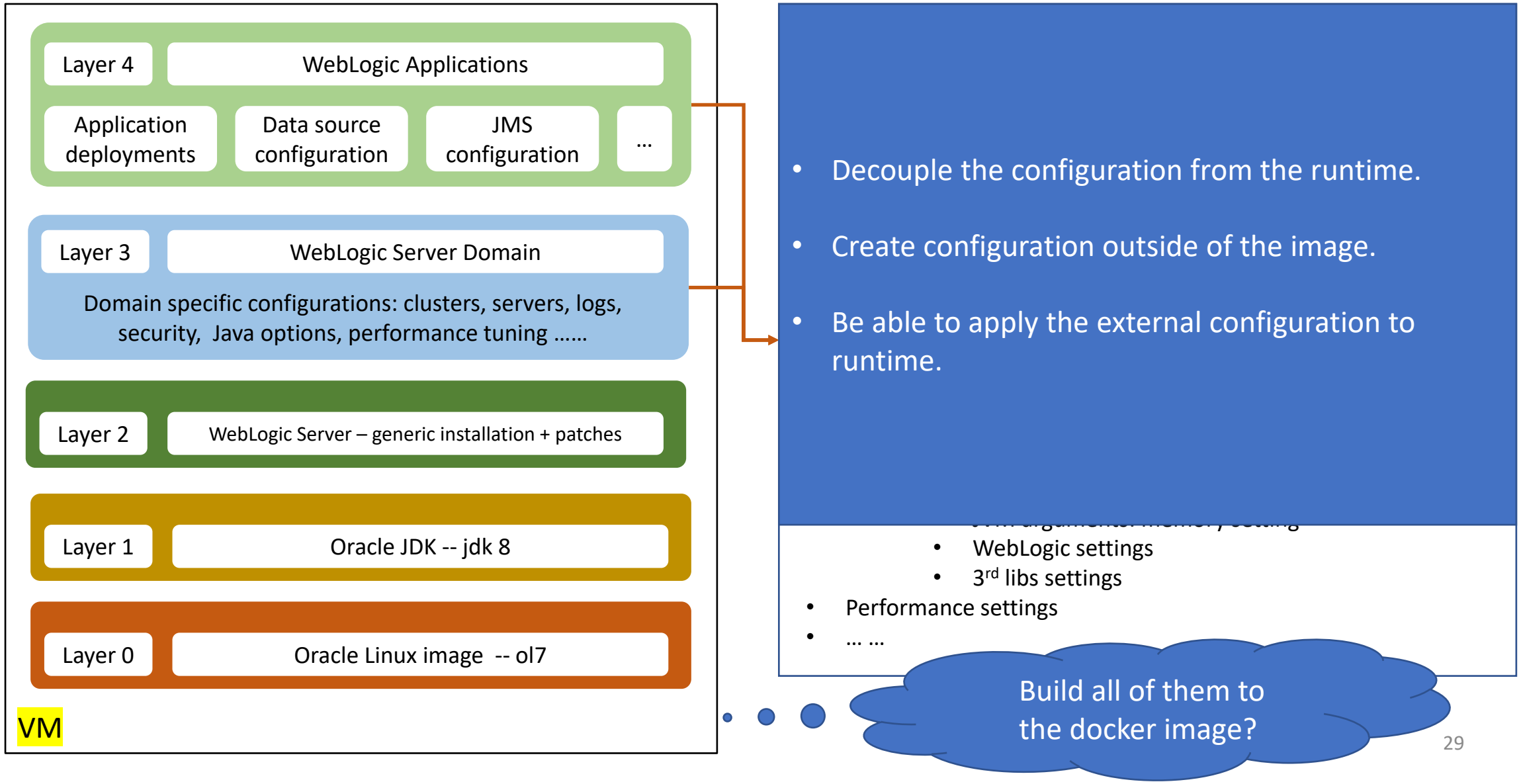
Execute steps up to and including **Perform the deployment with GitHub Actions Infrastructure as Code**

Instructor lead slides during deployment

Migrating WLS to AKS



Assets in VM

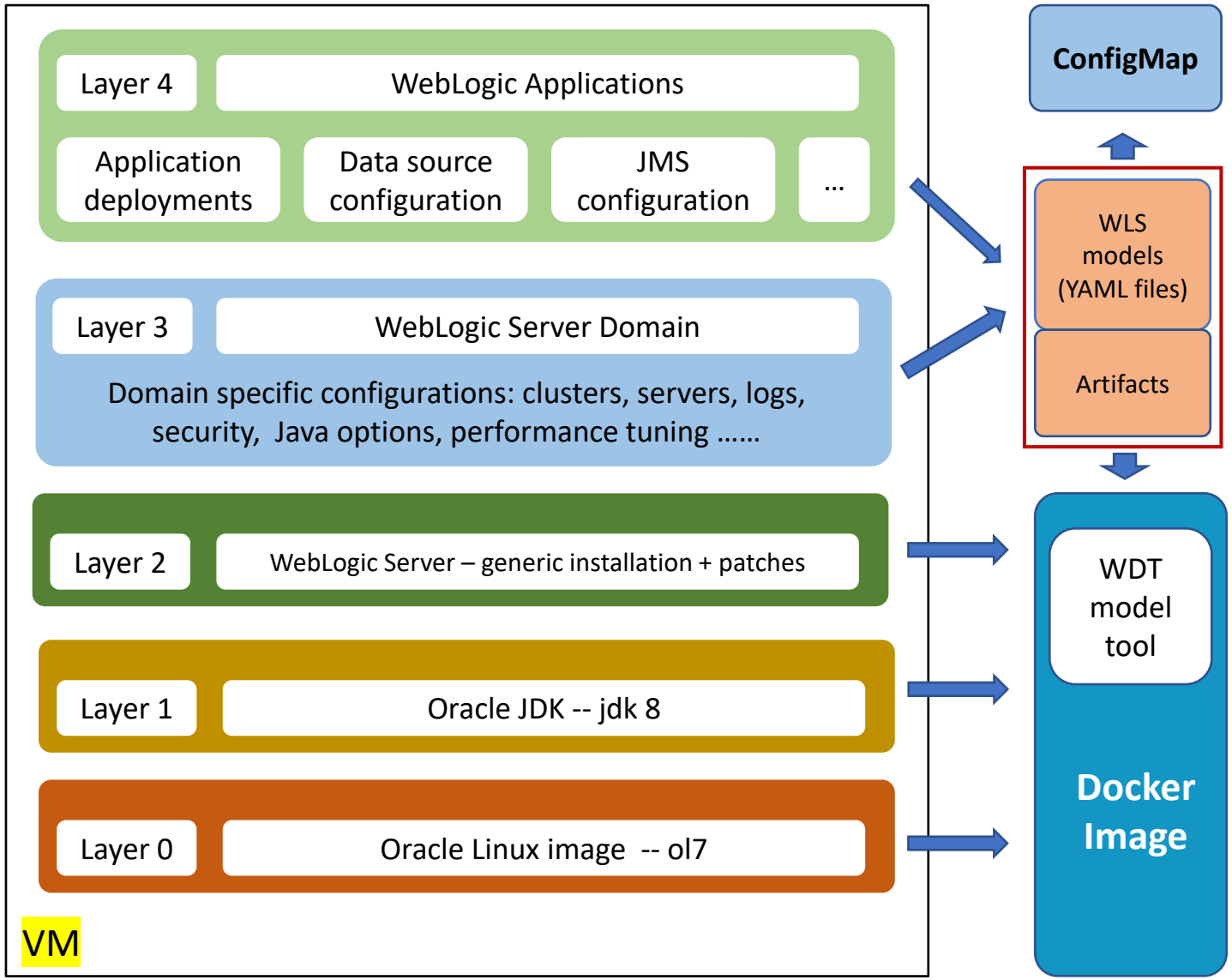

Layer 4
WebLogic Applications
Application deployments
Data source configuration
JMS configuration
...
Layer 3
WebLogic Server Domain
Domain specific configurations: clusters, servers, logs, security, Java options, performance tuning ……
Layer 2
WebLogic Server – generic installation + patches
Layer 1
Oracle JDK -- jdk 8
Layer 0
Oracle Linux image -- ol7
VM
• Decouple the configuration from the runtime.
• Create configuration outside of the image.
• Be able to apply the external configuration to runtime.
• WebLogic settings
• 3rd libs settings
• Performance settings
• … …
Build all of them to the docker image?

WebLogic domain model in image (1/2)

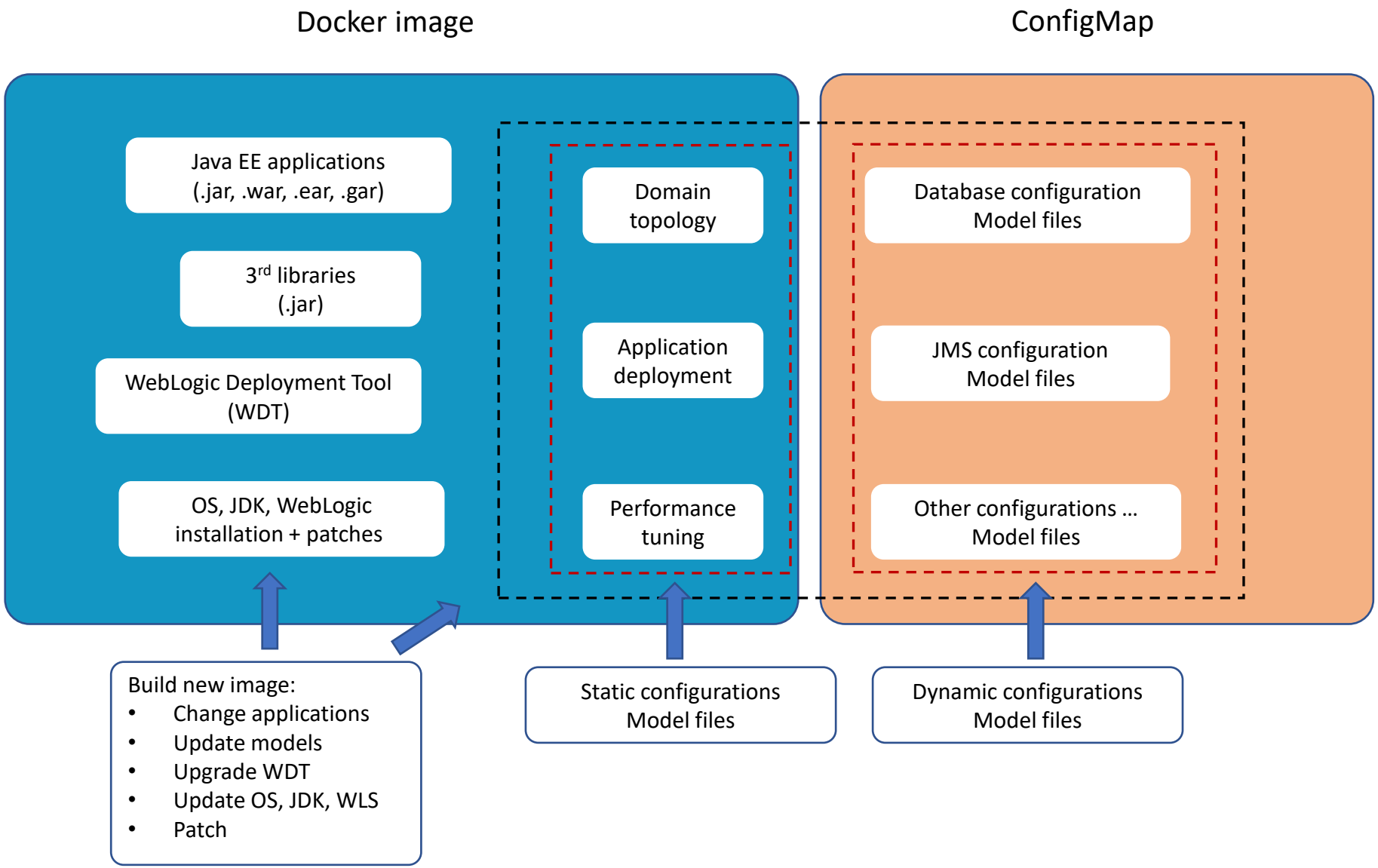
WebLogic domain model:

- Models are defined with YAML files
- WDT model tools to apply the models to WLS runtime

- Model files in image or ConfigMap



WebLogic domain model in image (2/2)



Docker image sample

WebLogic Image Tool

```
$ bash imagetool.sh cache addInstaller \
  --type wdt \
  --version latest \
2 --path weblogic-deploy.zip

$ bash imagetool.sh update \
  --tag model-in-image:WLS-v1 \
1 --fromImage container-registry.oracle.com/middleware/weblogic:12.2.1.4 \
  --wdtModel      ./model.yaml \
3 --wdtVariables  ./model.properties \
  --wdtArchive    ./archive.zip \
  --wdtModelOnly \
  --wdtDomainType WLS \
  --chown oracle:oracle
```

1. WebLogic standard image: OS + JDK + WLS
2. WebLogic Server Deployment Tooling
3. Artifacts and WLS domain models

```
├── applications
│   └── ejb-server-stateless.ear
└── domainLibraries
    ├── postgresql-42.2.8.jar
    └── mssql-jdbc-7.4.1.jre8.jar
```

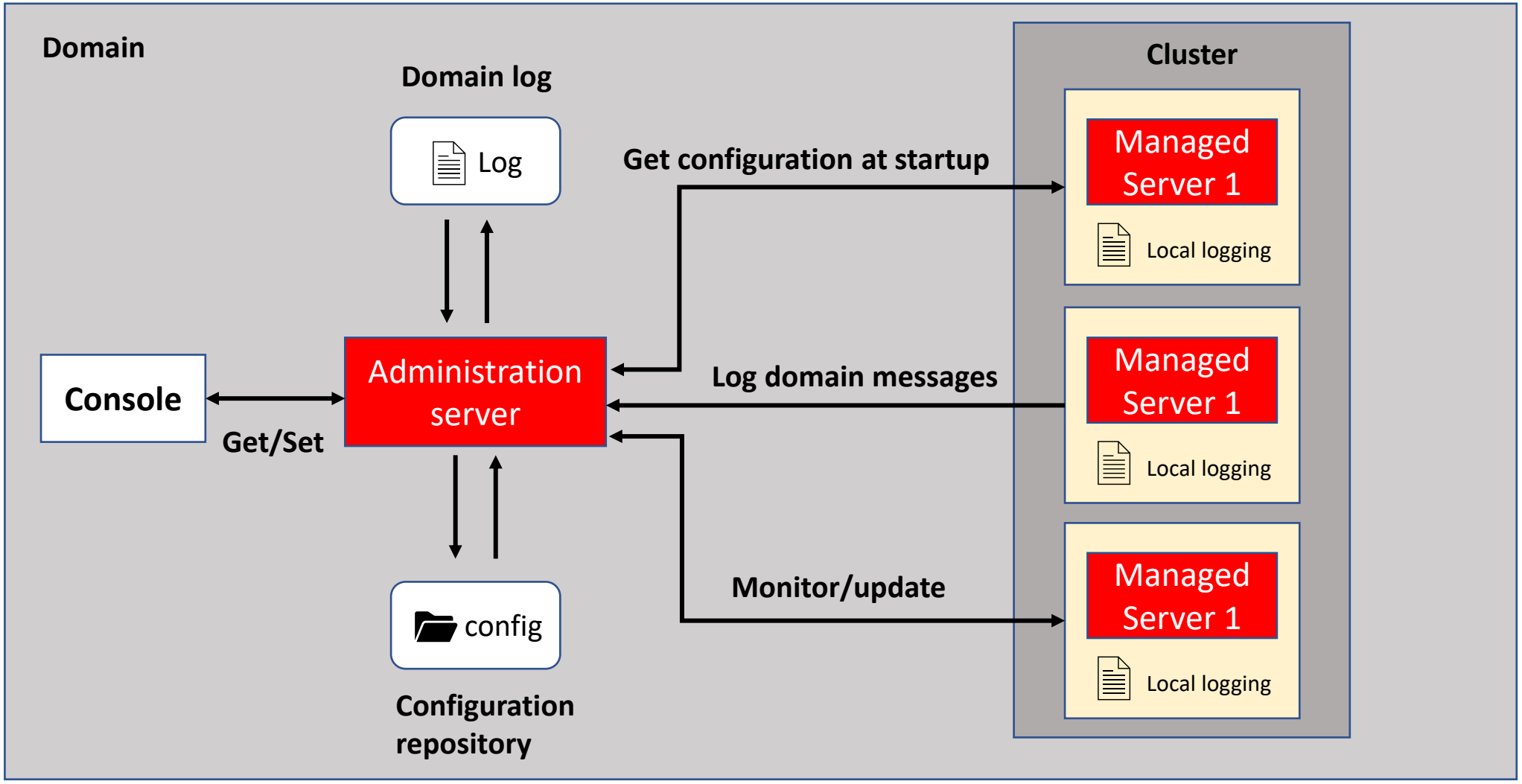
archive.zip

```
domainInfo:
  AdminUserName: "@@SECRET:__weblogic-credentials__:username@@"          1
  AdminPassword: "@@SECRET:__weblogic-credentials__:password@@"
  ServerStartMode: "prod"
  domainLibraries: [
   'wlsdeploy/domainLibraries/postgresql-42.2.8.jar',
   'wlsdeploy/domainLibraries/mssql-jdbc-7.4.1.jre8.jar']

topology:
  Name: "@@ENV:CUSTOM_DOMAIN_NAME@@"
  ProductionModeEnabled: true
  AdminServerName: "admin-server"
  Cluster:
    "cluster-1":
      DynamicServers:
        ServerTemplate: "cluster-1-template"                             2
        ServerNamePrefix: "@@ENV:MANAGED_SERVER_PREFIX@@"
        DynamicClusterSize: "@@PROP:CLUSTER_SIZE@@"
        MaxDynamicClusterSize: "@@PROP:CLUSTER_SIZE@@"
        MinDynamicClusterSize: "0"
        CalculatedListenPorts: false
  Server:
    "admin-server":
      ListenPort: 7001
  ServerTemplate:
    "cluster-1-template":
      Cluster: "cluster-1"
      ListenPort: 8001
  SecurityConfiguration:
    NodeManagerUsername: "@@SECRET:__weblogic-credentials__:username@@"  1
    NodeManagerPasswordEncrypted: "@@SECRET:__weblogic-credentials__:password@@"

appDeployments:
  Application:
    ejb-server:
      SourcePath:wlsdeploy/applications/ejb-server/cluster_ejb_stateful_session.ear
      ModuleType: ear
      Target: 'cluster-1'
```

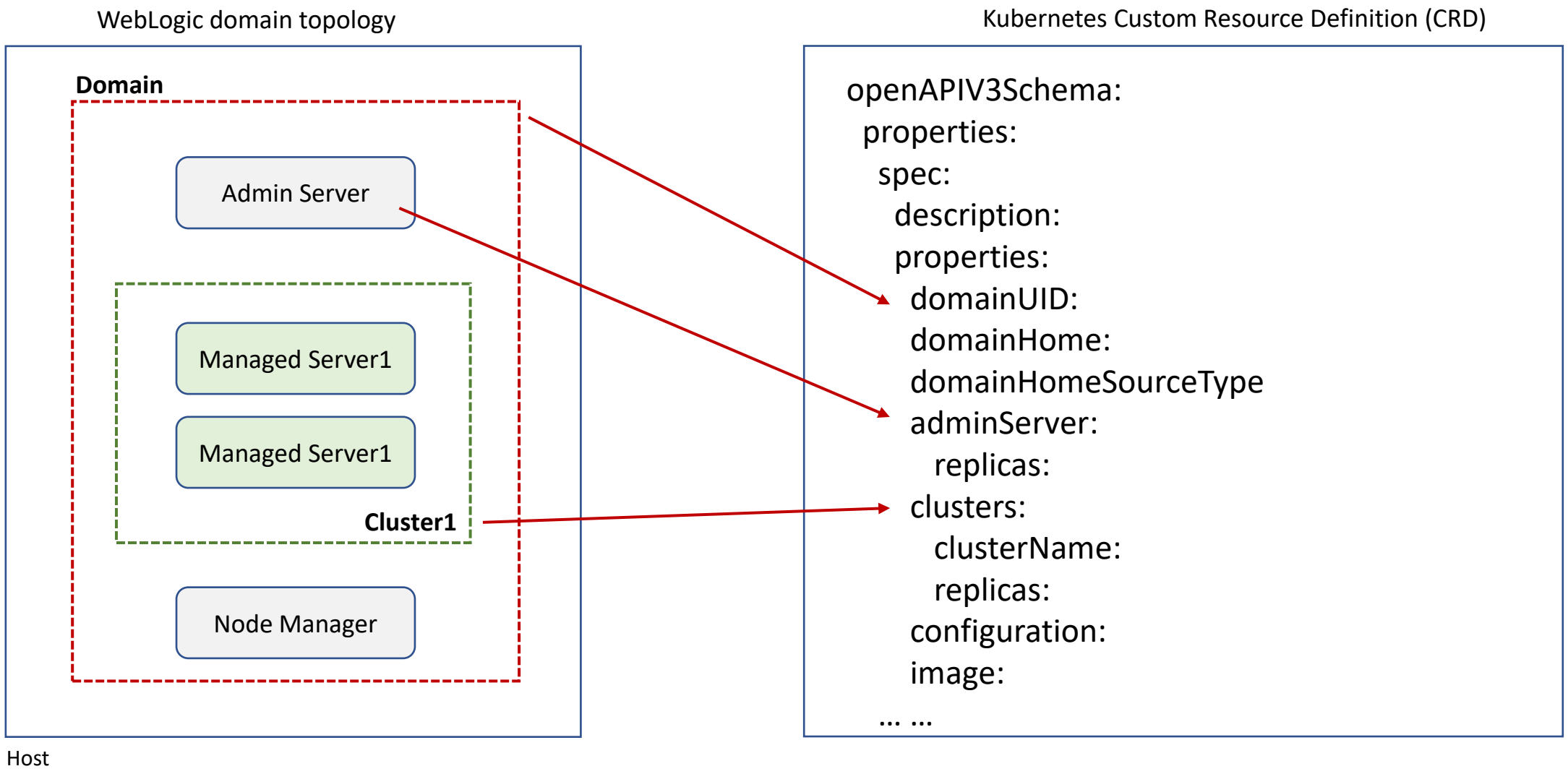
model.yaml

1. Run time: configured with Kubernetes Secrets
2. Run time: configured with ENV variables

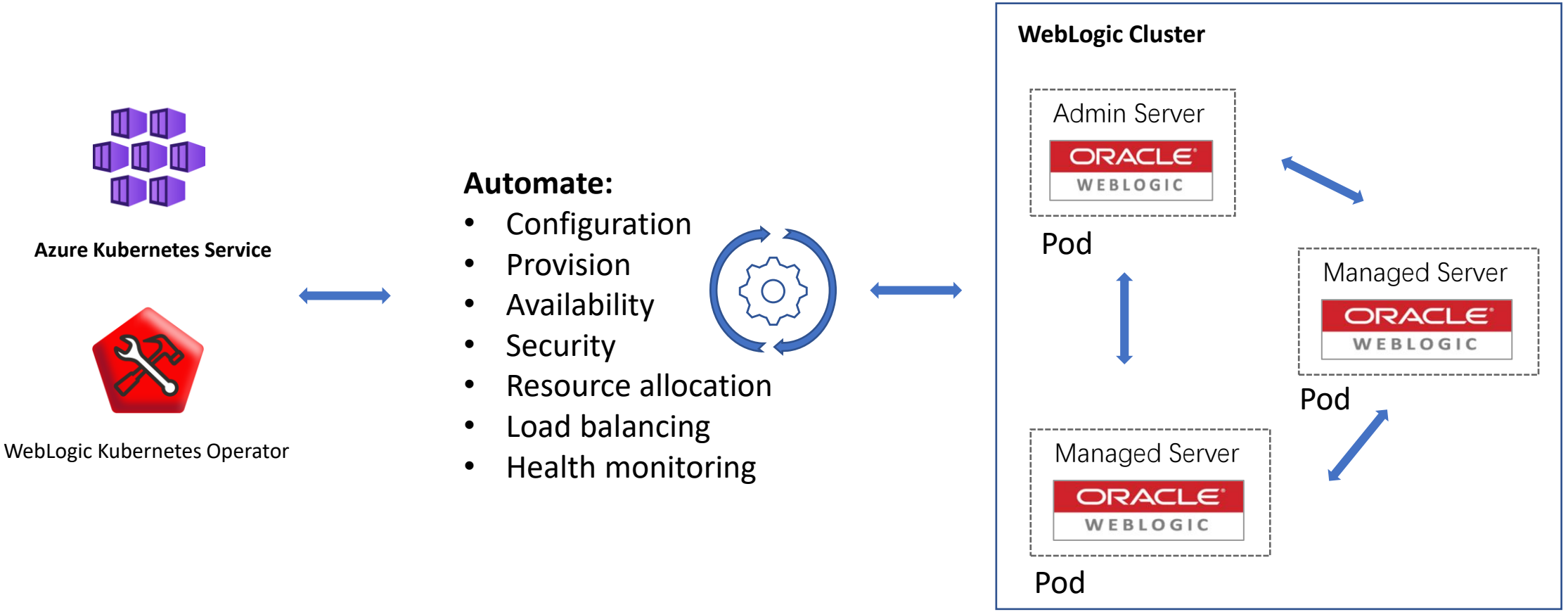
WLS domain


Domain
Domain log
Log
Get configuration at startup
Cluster
Managed Server 1
Local logging
Managed Server 1
Local logging
Managed Server 1
Local logging
Console
Get/Set
Administration server
Log domain messages
Monitor/update
config
Configuration repository

Extending Kubernetes API – the WLS operator

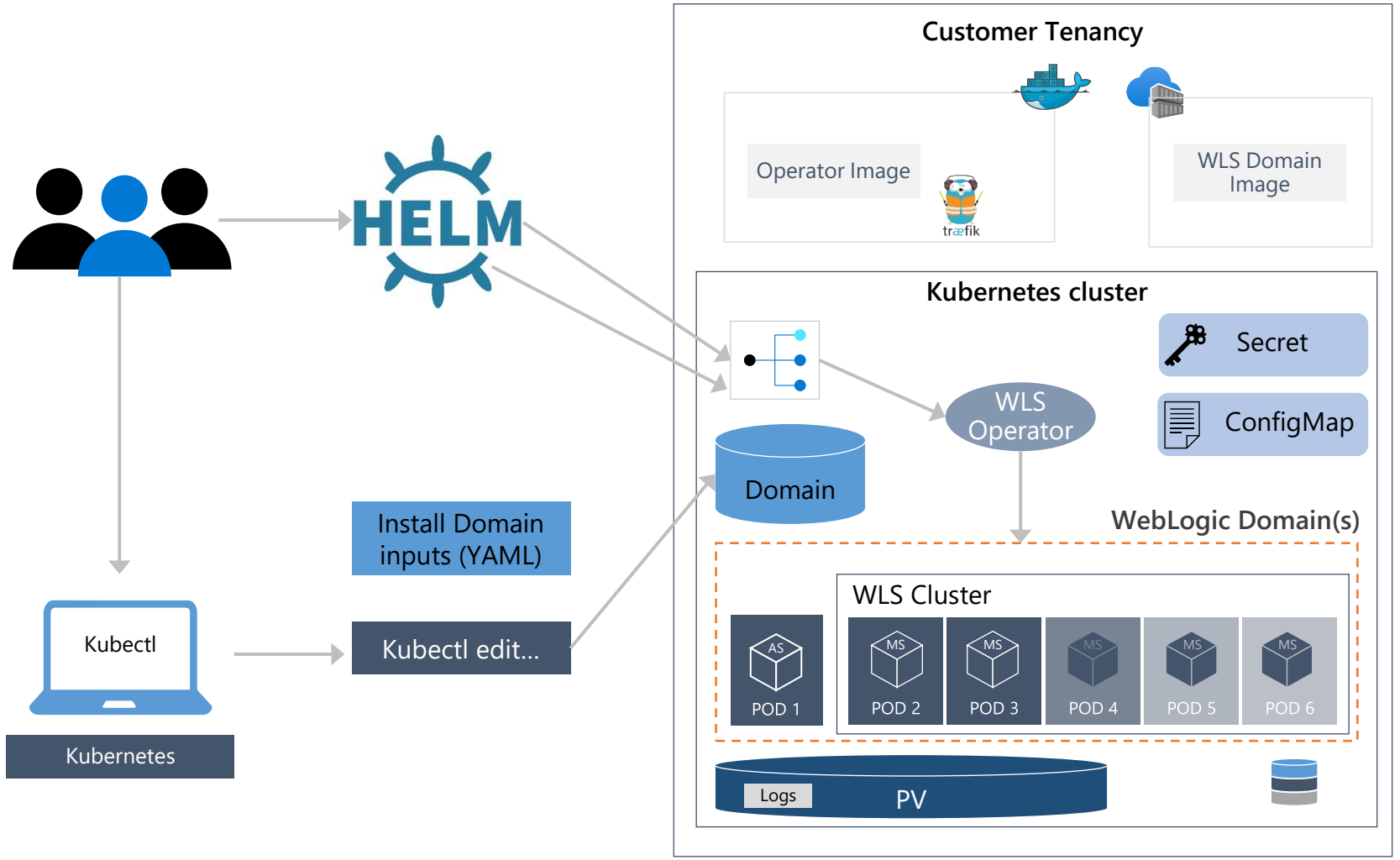


Container Orchestration



Run WebLogic using the WLS operator

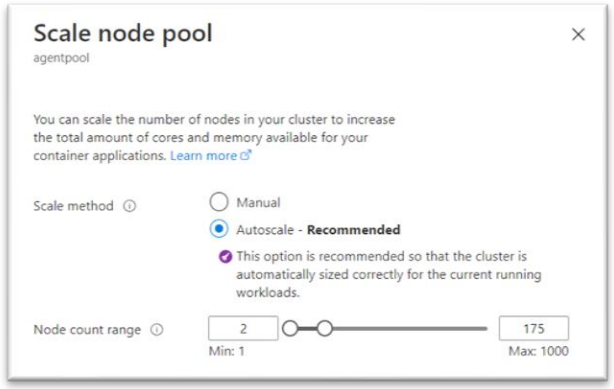
- Build WLS application image and push it to ACR
- Install WLS operator with HELM charts
- Create WLS domain resource definition (YAML)
- Ask the WLS operator to install WLS domain
- The WLS operator creates pods for admin server and managed servers



Scaling

Node horizontal scaling

AKS supports node auto scaling



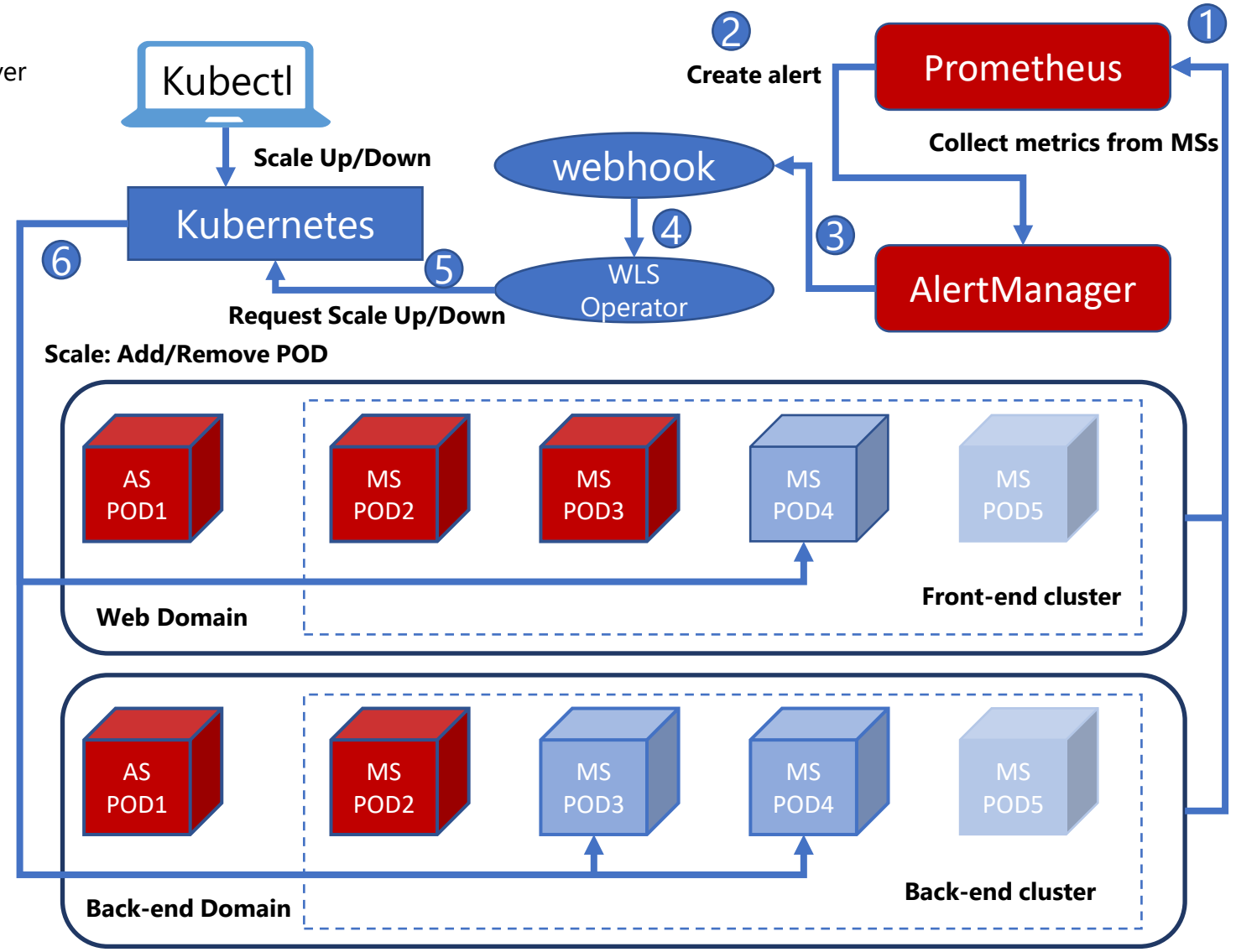
Pod horizontal scaling

- Different parts of the application have different scaling requirement.
- Metrics:
  - WebLogic monitoring exporter
  - Metrics of JVM, request, servlet… …
- Solution:
  - Prometheus
  - AlertManager
  - Webhook
  - WebLogic operator

Pod horizontal autoscaling

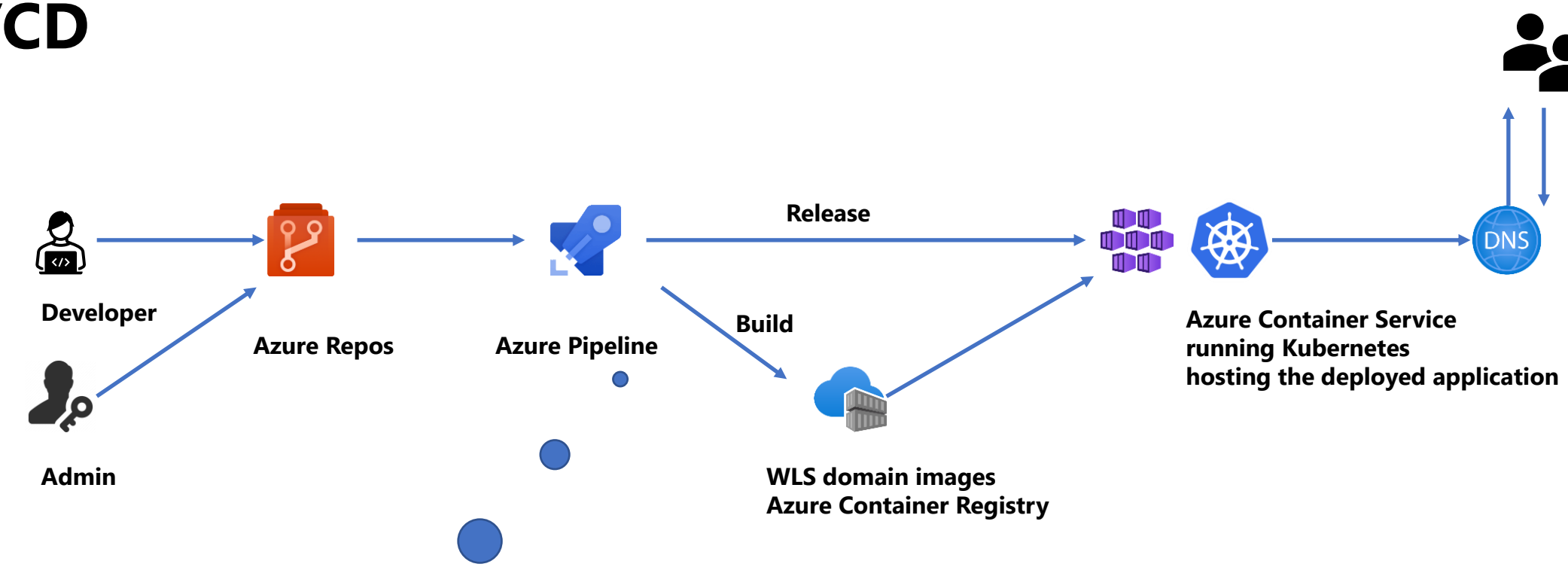
AS: administration server
MS: managed server

Demo video



wls-aks/auto-scaling at main · galiacheng/wls-aks (github.com)

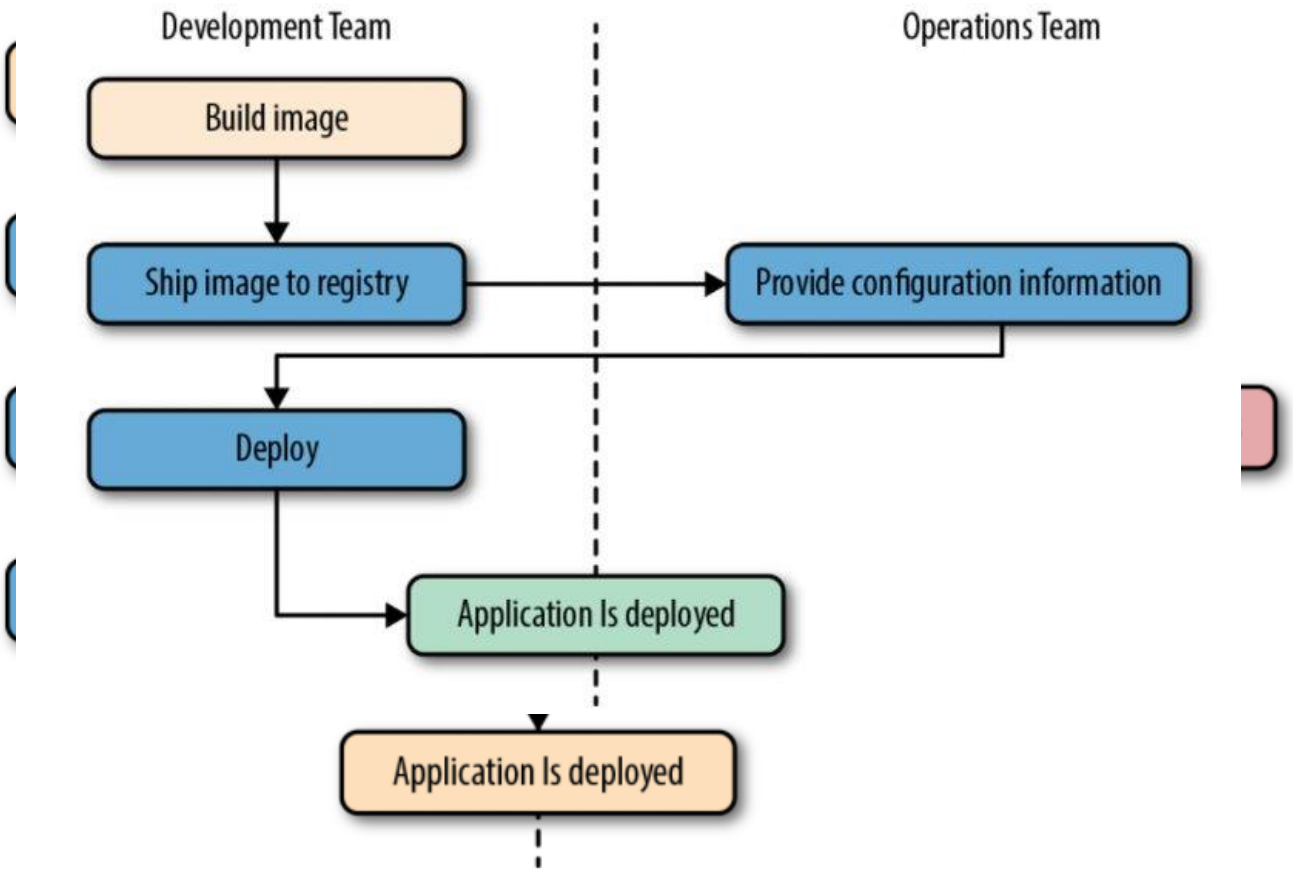
CI/CD



Use a Log4j compliant base image.

Ease the installation of security fixes.

CI/CD



Simplify collaboration between development and operations team.

Observability

Pillars	Layer	Tools: Open source	Tools: Azure Services
Metrics	Infrastructure (AKS)	Prometheus + Grafana	Azure Container Insight
	WebLogic Server	Prometheus + Grafana	
	Java EE Applications	Prometheus + Grafana	
Traces	Java EE Applications	Elastic APM	Azure Application Insight
Logs	Infrastructure (AKS)		Azure Monitor Logs
	WebLogic Server	Fluentd + Elasticsearch + Kibana	Azure Elasticsearch service Azure Kibana Service
	Java EE Applications	Fluentd + Elasticsearch + Kibana	Azure Elasticsearch service Azure Kibana Service
Service mesh	Java EE Applications	Istio	

Integrating with monitoring services

- Azure file share
  - Persistent volume

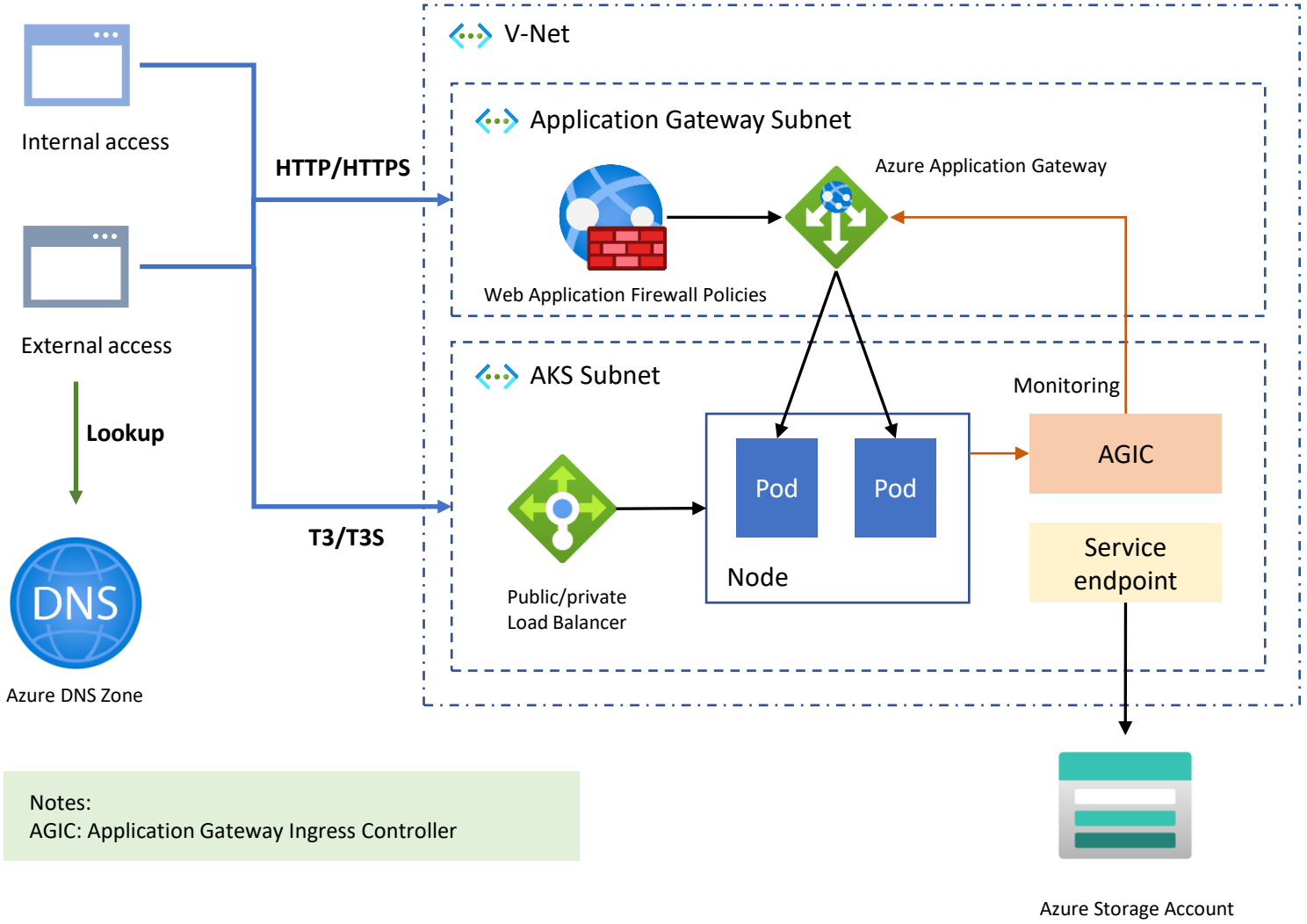
- JVM args
  - -javaagent:/shared/app-insight/applicationinsights-agent-3.1.1.jar
  - -javaagent:/shared/elastic-apm/elastic-apm-agent-1.24.0.jar

- Kubernetes sidecar
  - Fluentd sidecar container to export application logs
  - Istio sidecar container to integrate with Istio

Capture automatically, no code change required.

Networking

- Traffic from layer 7 and layer 4
- TLS/SSL with custom certificates
- External DNS



Notes:
AGIC: Application Gateway Ingress Controller

Summary

- Microsoft very serious about Java
- Many pathways, choices and architectures supported
- Choose from virtual machines, containers, managed services, serverless and anything in between
- Full suite of Java tooling for seamless end-to-end, best of breed experience
- Actively evolving solutions for rapidly growing customer base

Resources

- **Java EE on Azure landing page**
  - https://azure.microsoft.com/en-us/develop/java/ee/
- **Java on Azure docs**
  - https://docs.microsoft.com/en-us/java/azure/
- **Azure Java code samples**
  - https://azure.microsoft.com/en-us/documentation/samples/?term=java

Java on Azure for Wherever You Are in your Cloud Journey