Delivering Microservices

Mark Heath
CLOUD ARCHITECT

@mark_heath www.markheath.net

Overview

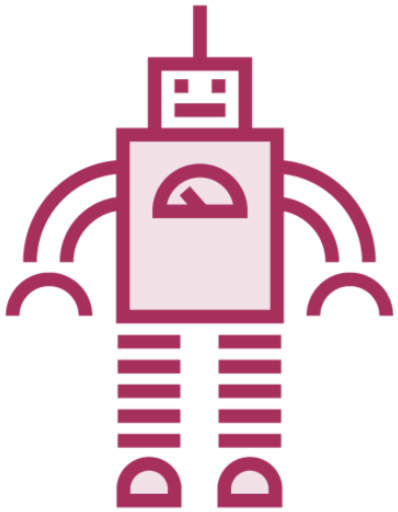
Deploying microservices

- Automation
- Pipelines
- Deployment environments

Monitoring microservices

- Health checks
- Aggregating logs

Automated Deployment



Manual deployment process

- Follow a step by step guide
- Doesn't work for microservices

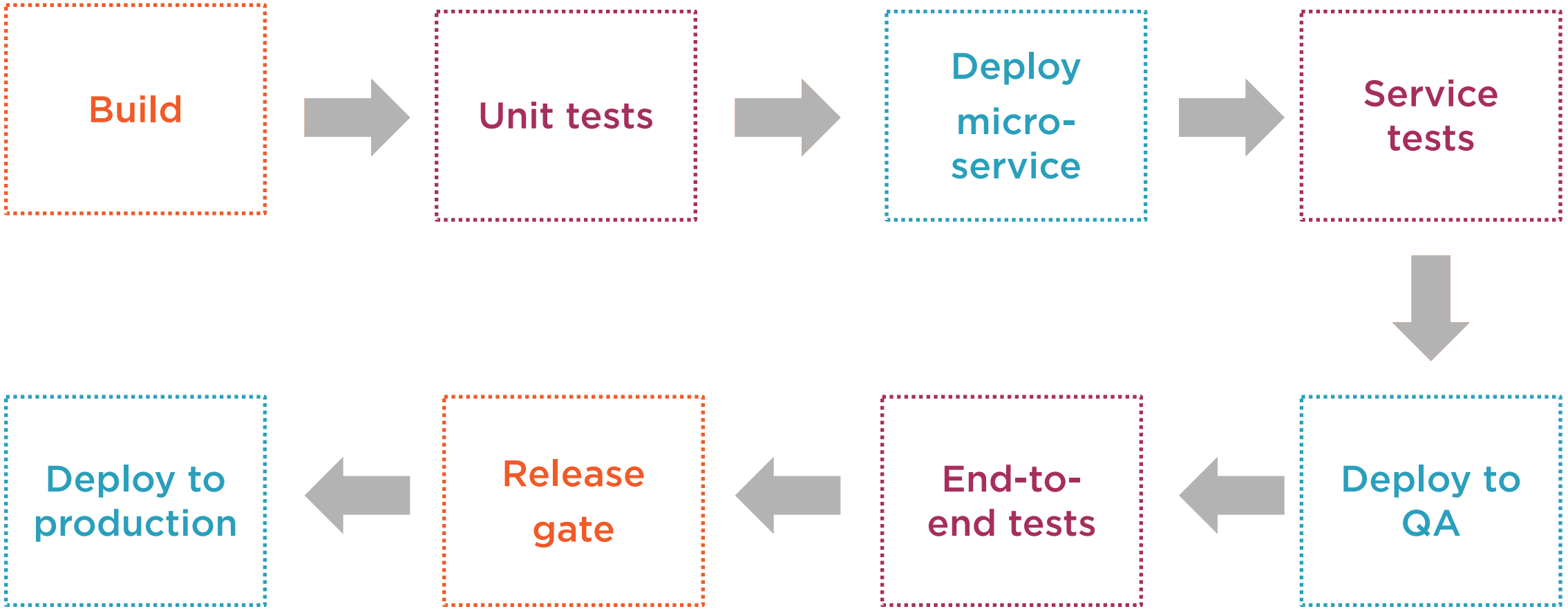
Automated deployment

- Reliable and repeatable
- Frequent deployments

Continuous integration

- Build and run tests

Release Pipelines



Deployment Environments

Development

Debug code

QA

End-to-end tests

Manual testing

Pen testing

Performance testing

Production

Per-customer

Per-region

Parameterize your deployments

```
# qa.yaml
Name: QA
Region: West Europe
VMSize: Medium
MinInstances: 2
MaxInstances: 5
```

Configuration Files

deploy orderingService 1.0.4 qa.yaml

kubectl apply –f myconfig.yaml

Cloud Infrastructure Deployment

Terraform

Azure ARM templates

Hierarchical configuration

Base template

+

Environment-specific override

Artifact Registry

Store build artifacts

Deploy latest or previous version

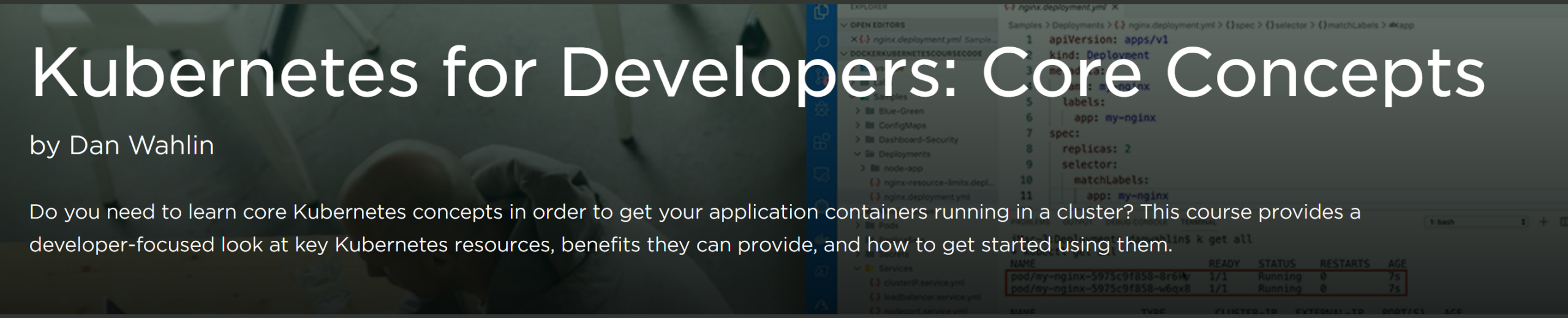
Docker container images

- eshoponcontainers/orderingservice:1.3.1

Deploy to Kubernetes

- YAML manifest files
- Desired state

Learn More About Kubernetes

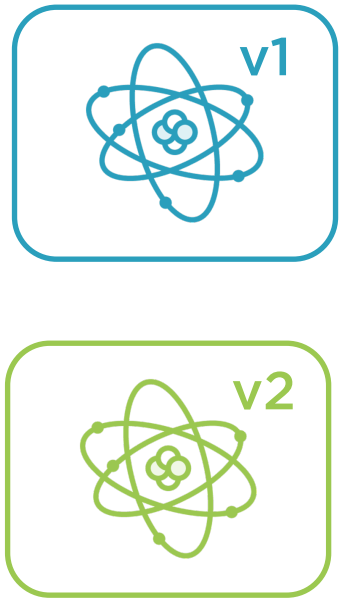


https://app.pluralsight.com/library/courses/kubernetes-developers-core-concepts

Upgrading one microservice should not require all microservices to be upgraded

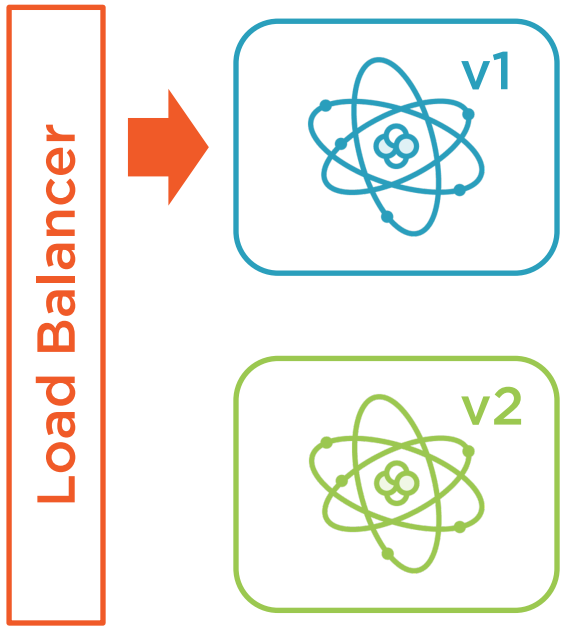
Upgrade Strategies

Stop v1, start v2



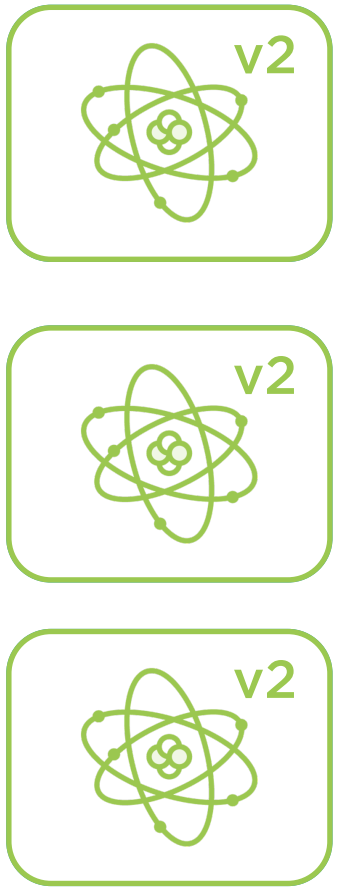
Short period of unavailability

Blue green swap



Minimal downtime

Rolling upgrade



Monitoring Microservices

Many things to monitor

- Many processes on many hosts

Centralized logs and telemetry

- System health dashboard

What Should We Monitor?

Host Metrics

CPU percentage

Memory usage

Configure alerts

Application Metrics

HTTP requests

Errors (401, 500)

Queue length

Health checks

Logs

Aggregated

Standardize log output

Kibana

Application Insights

Take advantage of the built-in monitoring and observability capabilities of your microservices hosting platform

Demo

Logging and health checks in eShopOnContainers

Summary

Automate your deployments

Create a release pipeline

- Deploy to different environments
- Run automated tests

Upgrade microservices

- Rolling updates
- Blue-green swap

Summary

Observability

- Host and application metrics
- Centralized logs
- Health check endpoints

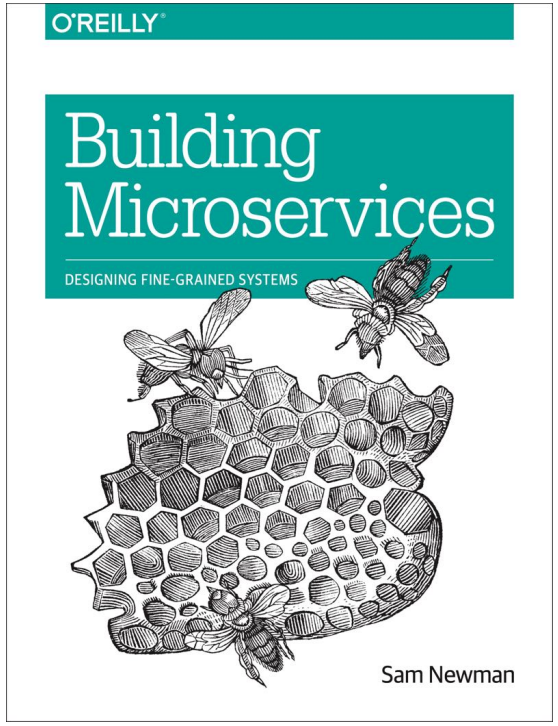
Final Thoughts

Microservices give you options

Every context is different

Keep learning!

You won't get it right first time!



Sam Newman
"Building Microservices"