Running Containerized Applications

Ivan Gavryliuk
SOFTWARE ARCHITECT

@aloneguid http://isolineltd.com

Overview

Overview of Service Fabric container support

Reasons to use containers

What you need to develop for containers

Examples of containerized workloads

Why Containers

Other Courses

Getting Started with Docker

by Nigel Poulton

Docker for Web Developers

by Dan Wahlin

Docker Deep Dive

by Nigel Poulton

Lift and Shift

If it’s not broken don’t touch it

Cloud is attractive

Package in a container

Shift to the cloud

Cloud portability

Run anywhere

- Azure
- AWS
- Google Cloud
- Other clouds
- On premise

Wide choice of orchestrators

Isolation and Security

Virtualize resources

- Filesystem
- Memory
- Disk
- Network

Sandbox applications

Service Fabric supports all container features

- But it’s not only an orchestrator

Containers and Issues

Storage

- Have to use external storage
- Or external volumes
  - Slow
  - Data sharing issues

Operational cost

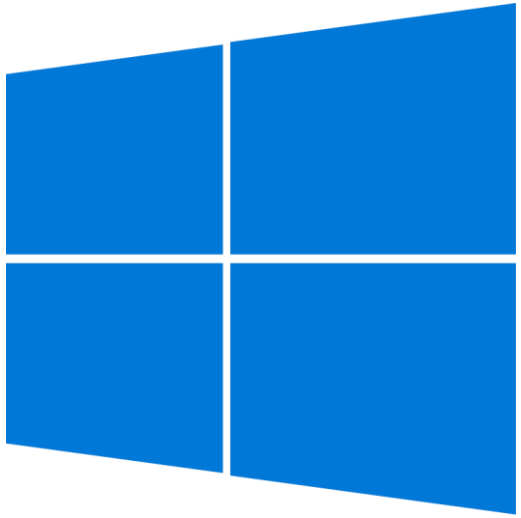
- Manage your own infrastructure
- Dependency on internal team
- Cloud native services is an alternative

High but not highest density

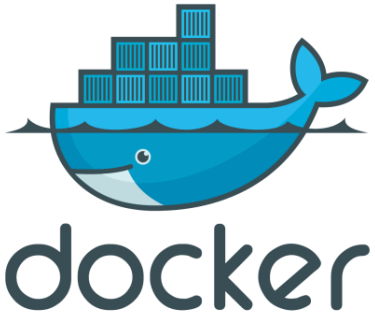
Containers have pros and cons. Carefully consider them in your design.

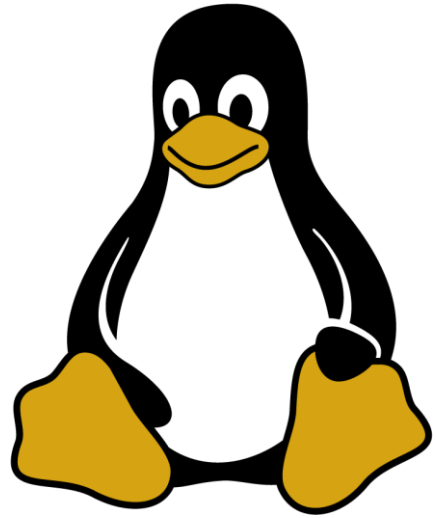
Supported Container Scenarios

OS



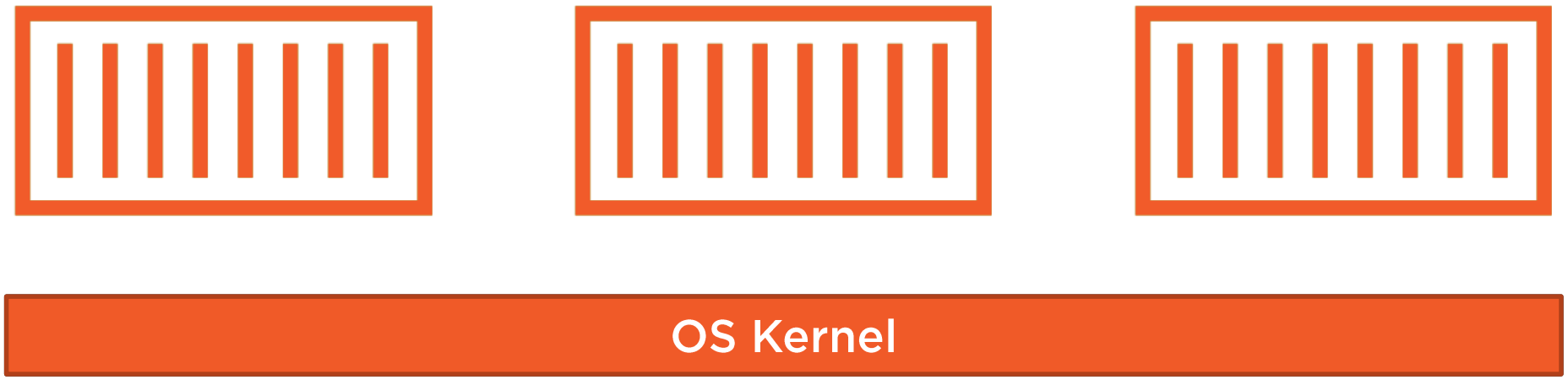
Windows





Linux

Security



Hyper-V Containers

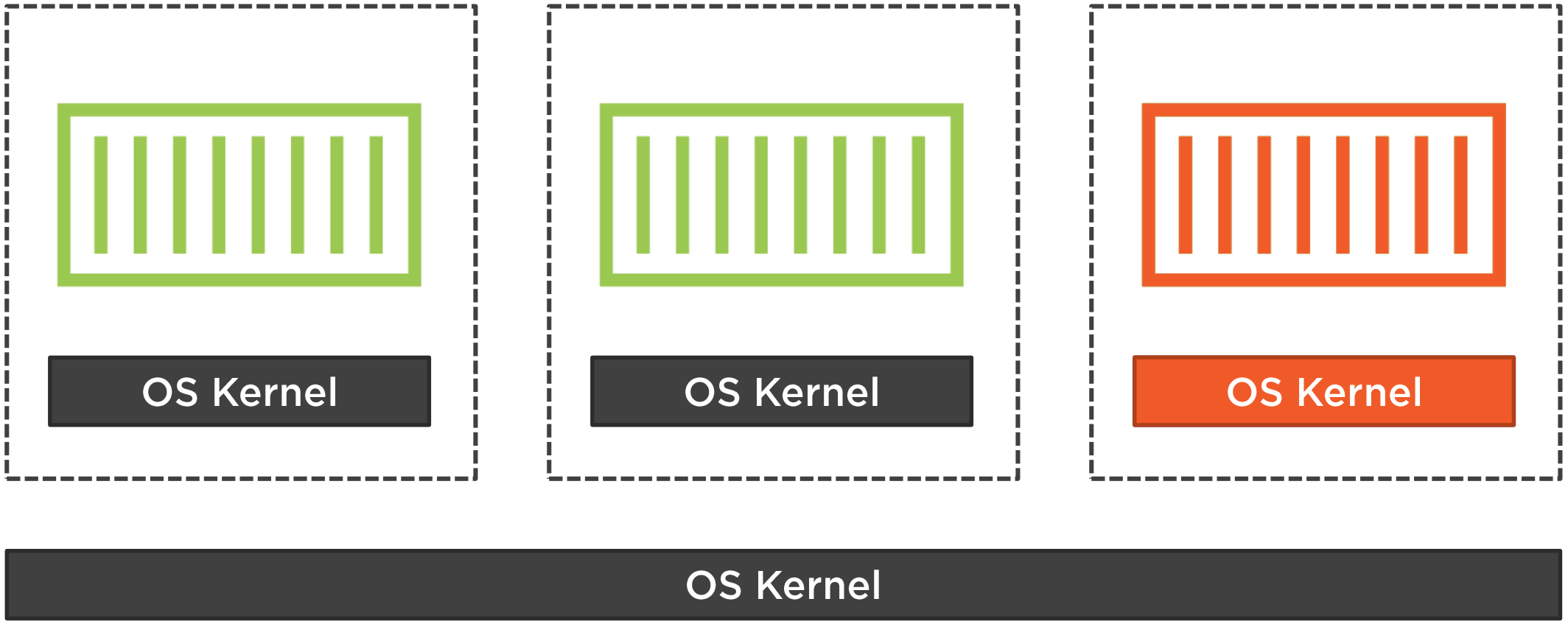


Image deployment and activation

Resource governance

Repository authentication

Container-to-container discovery

Application configuration

Container security credentials

Different networking modes

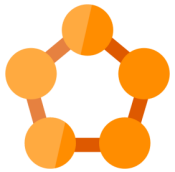
And more

Preparing Environment

Prerequisites

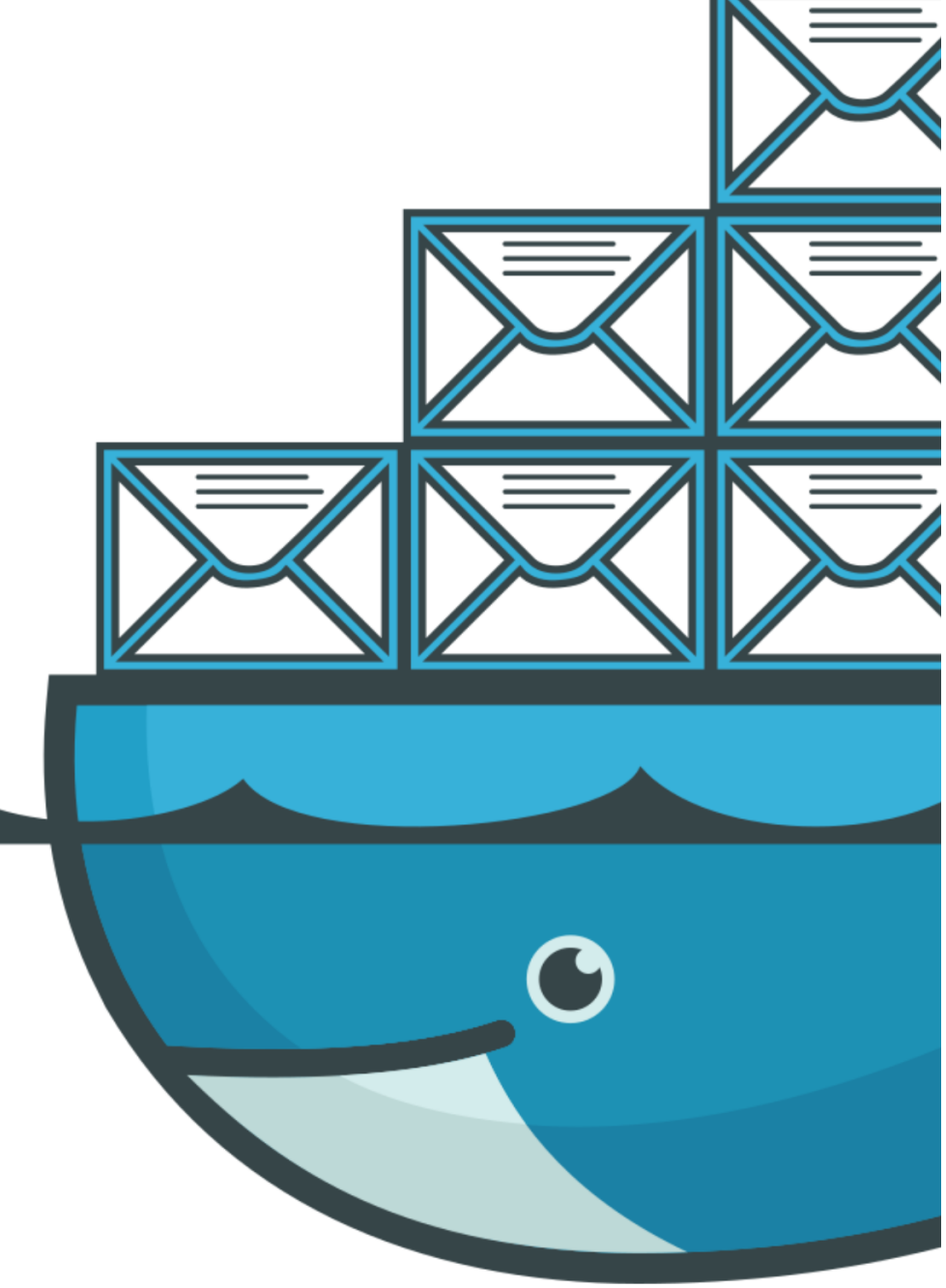
Docker Desktop

Docker Hub account



Service Fabric cluster

Containerize a Go Application



Create a container image first

Not a part of local development

Create a Dockerfile

- Contract between code and deployment
- Produces a container

Demo

Build with Docker

Test locally

Upload to Docker Registry

Containerize a Legacy .NET Framework Application

Demo

Containerize a legacy .NET Application

More Windows specific

Deploying Containers with Service Fabric

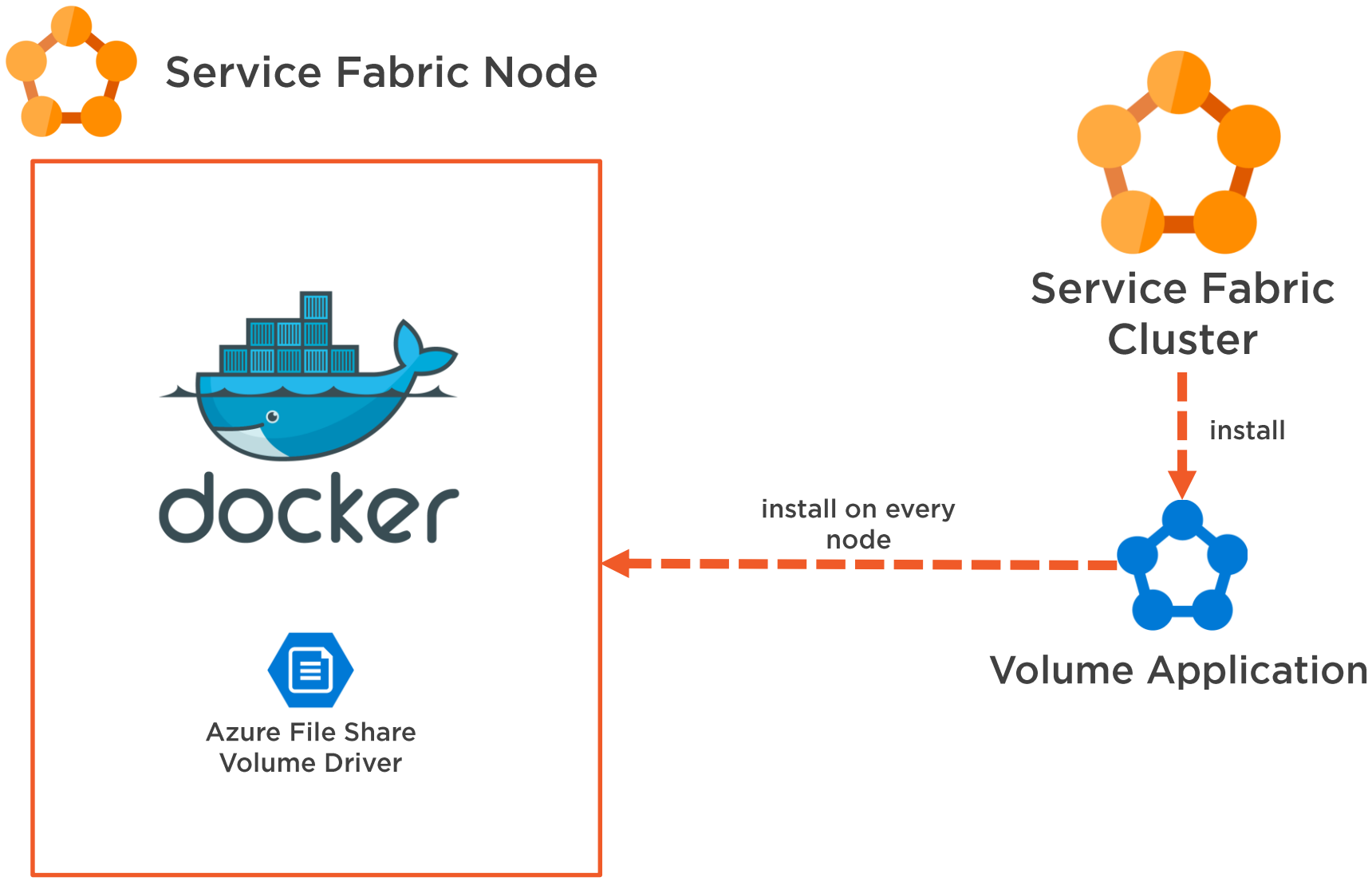
Demo

Deploy two containers

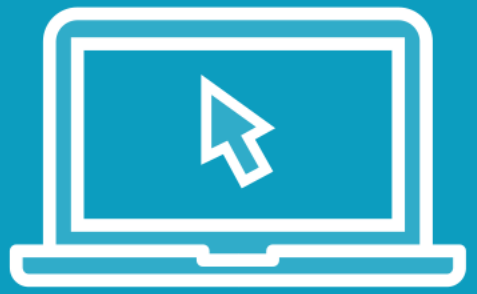
Using Visual Studio

Setting up Azure File Share Driver

Volume Enablement



Demo



Install PowerShell Az module

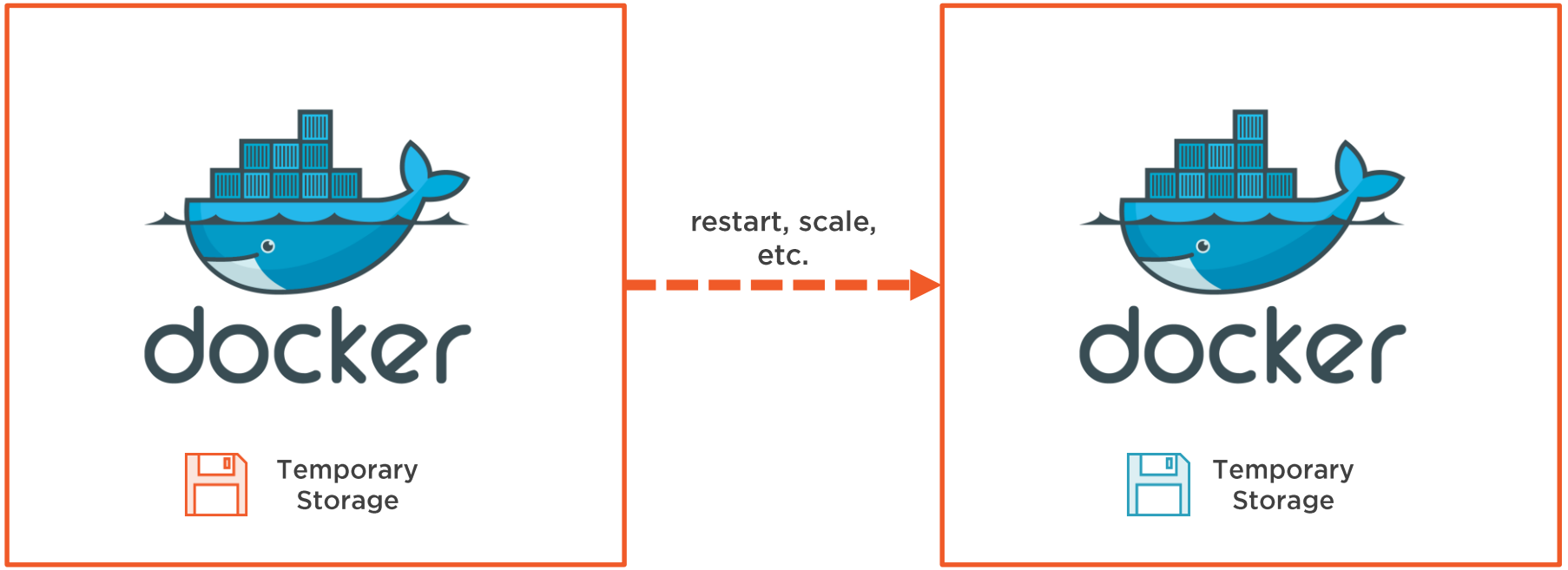
Create cluster with required parameters

Install Docker volume driver

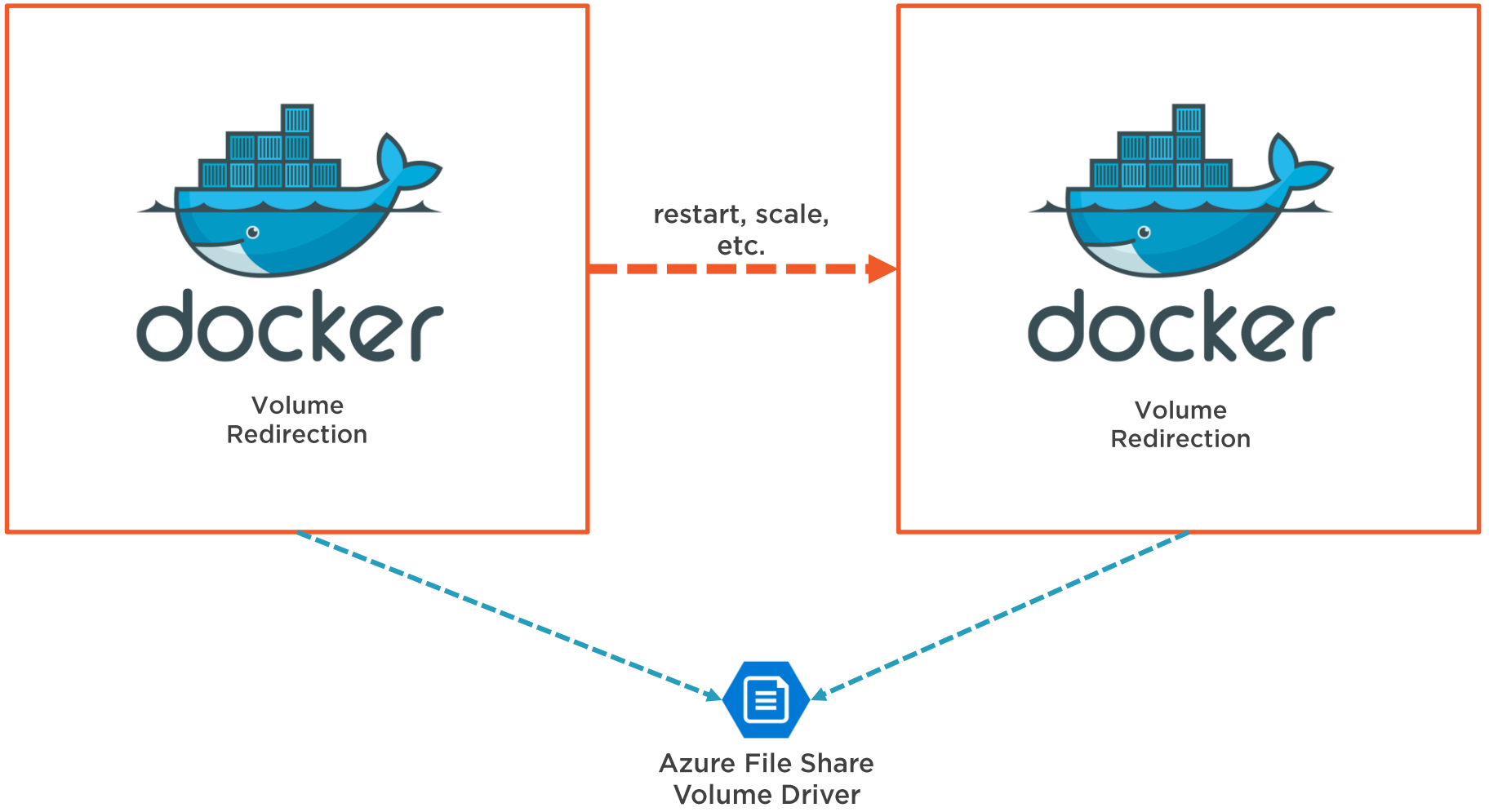
Create Azure File Share

Storing State

Volume Issue



Solution


docker
Volume
Redirection
restart, scale,
etc.
docker
Volume
Redirection
Azure File Share
Volume Driver

Demo

Mount container folder to a volume

Use Azure File Share volume

Deploy Service Fabric with volumes

Check that volumes are used

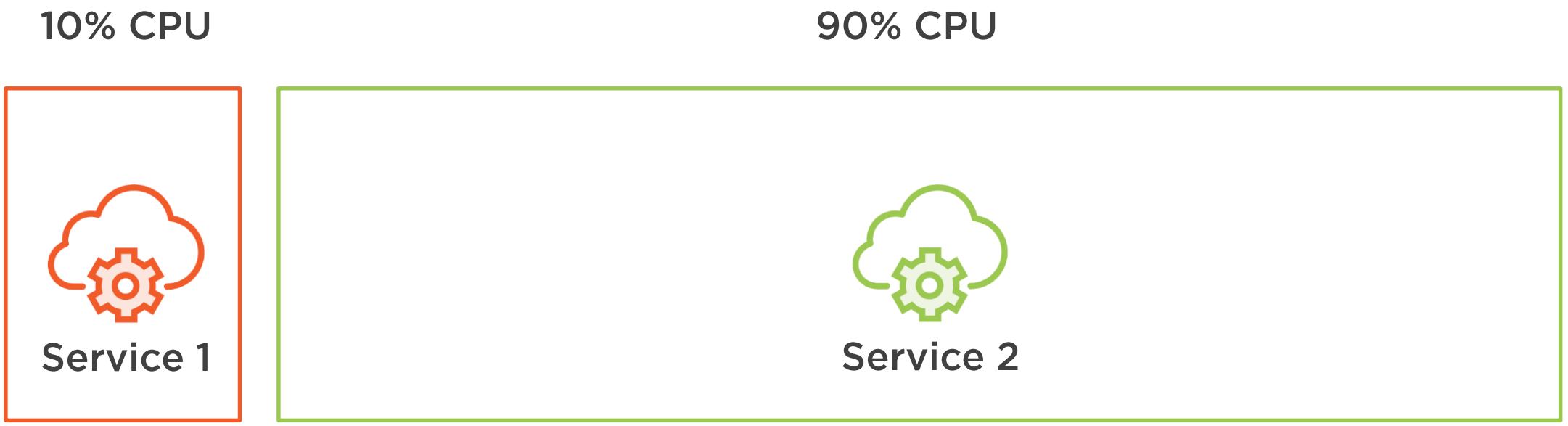
Resource Governance

Noisy Neighbours

50% CPU

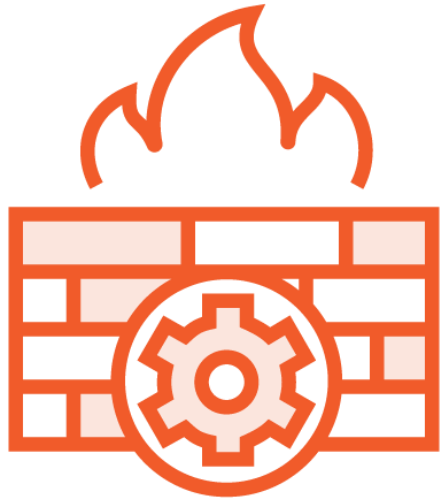
50% CPU

Noisy Neighbours



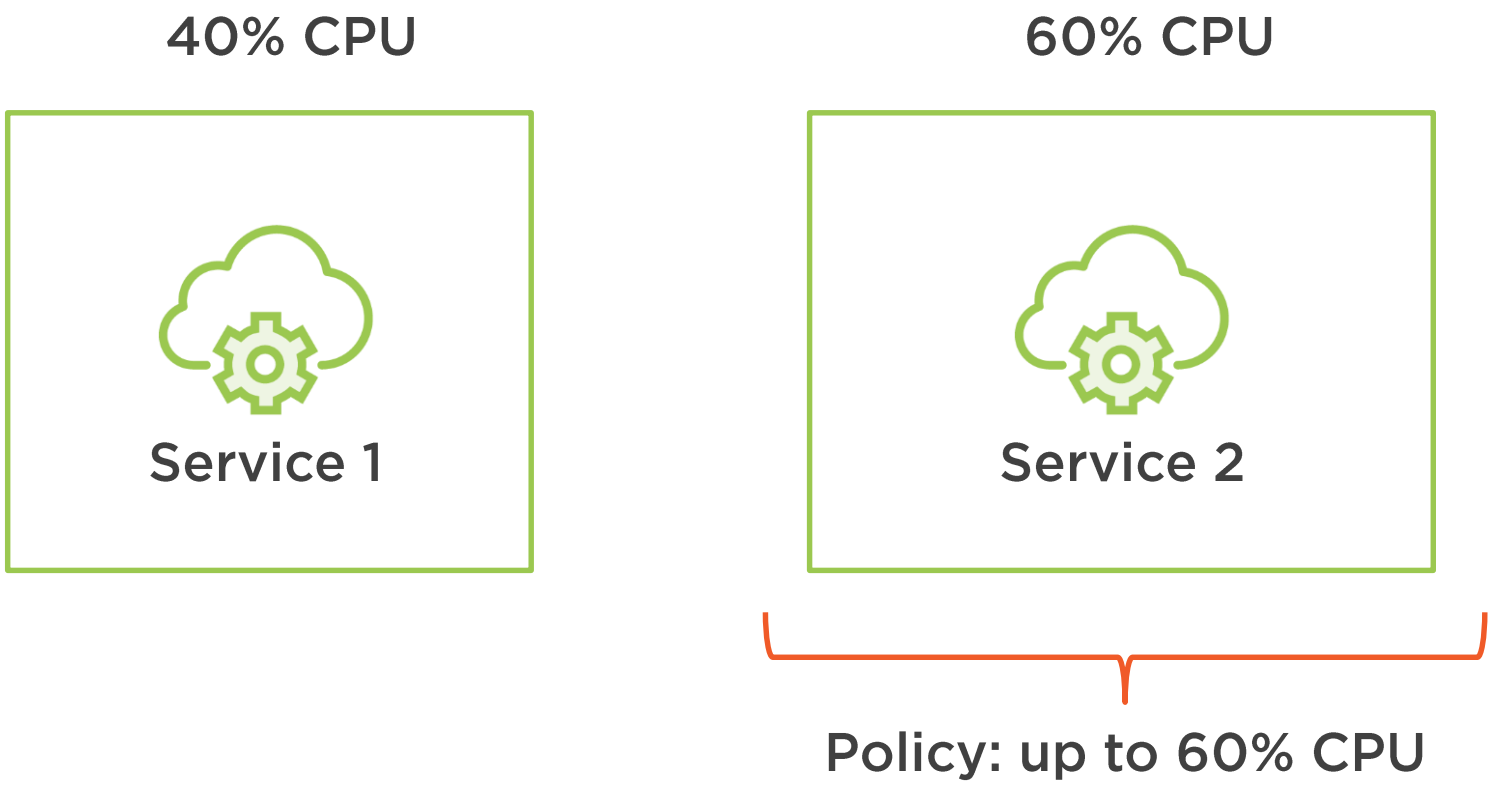
Solutions

Scale out.
Helps to reduce resource consumption.
Won't help misbehaving services.



Limit available resources.
Helps with misbehaving services.

Noisy Neighbours



Service Fabric supports policies for

- Native services on
  - CPU
  - Memory
- Containers
  - CPU
  - Memory (limit, swap, reservations)
  - IOps limits (read/write)

Summary

- What are containers
- Preferred container scenarios
- Problems and solutions
- Service Fabric container support
- Getting development environment ready
- Containerized applications samples
- Deploying containers to Service Fabric
- Docker volumes
- Resource governance

What's Next

Ivan Gavryliuk
SOFTWARE ARCHITECT

@aloneguid http://isolineltd.com

Related Courses

Getting Started with Docker

Nigel Poulton

Microservices Architecture

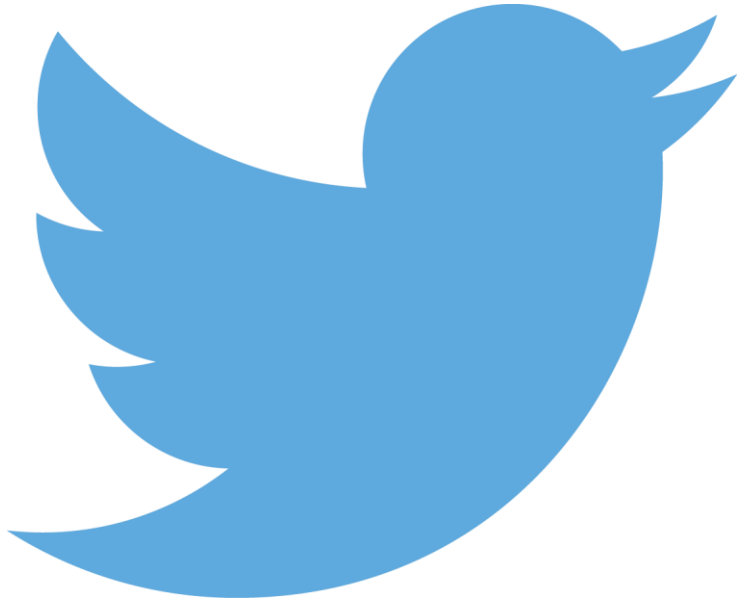
Rag Dhiman

Using Azure Service Fabric in production

Ivan Gavryliuk

Building an application with Azure Service Fabric Mesh

Ivan Gavryliuk



@aloneguid