

微软技术大会

Microsoft Ignite China

基于Azure的DevOps数据中心

- 使用TFS和Docker Swarm搭建持续交付流水线

徐磊

英捷创软首席架构师

Microsoft Regional Director & DevOps MVP

简介

• 徐磊

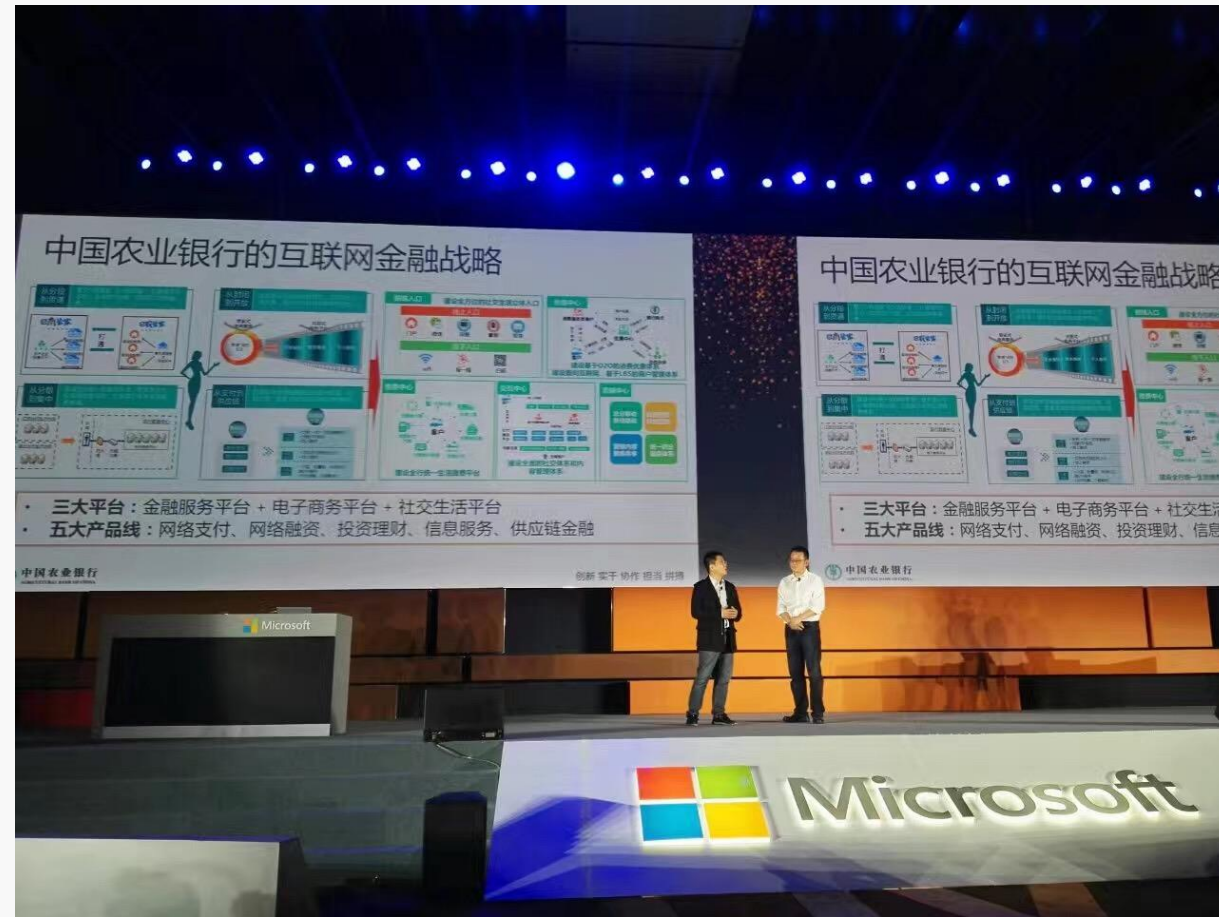
- 英捷创软 CEO 首席架构师
- 资深DevOps顾问和解决方案专家
- Microsoft Regional Director & DevOps MVP
- 认证 ScrumMaster 和 敏捷教练
- 超过10年的软件研发项目管理经验，曾任SSW中国研发中心总经理

项目经验

- 高级ALM/DevOps顾问，中国农业银行，互联网金融项目
- 敏捷教练，兴业银行，海外网银敏捷试点项目
- 高级管理顾问，中国移动南方基地，互联网研发基地外包管理改进
- 高级ALM/DevOps顾问，上海通用汽车有限公司，软件研发过程改进
- 高级ALM/DevOps顾问，上海汇众汽车，软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问，中国人民保险公司软件研发中心，软件研发过程改进和ALM平台落地
- 高级ALM/DevOps顾问，斯伦贝谢中国研发中心，敏捷开发与ALM落地
- 高级ALM/DevOps顾问，京东商城，PMO管理系统与ALM系统集成
- 高级ALM/DevOps顾问，华为2012实验室（深圳），rDelta系统选型
- 高级ALM/DevOps顾问，百威英博，软件外包管理系统ALM落地实施



农行互联网金融DevOps转型



内容

DevOps与持续交付，为什么？

DevOps的商业价值

容器对持续交付的意义

容器化你的开发环境

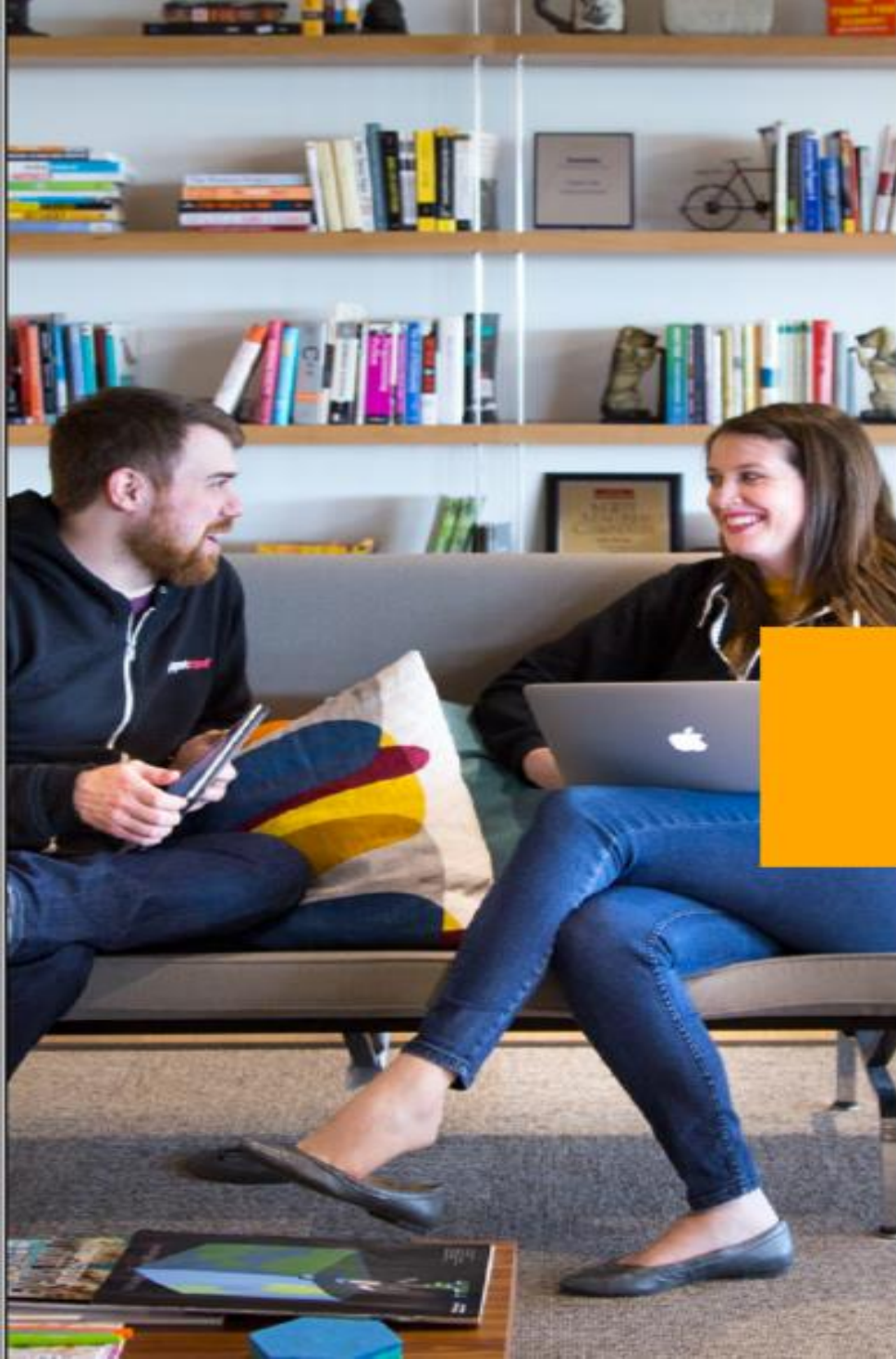
使用Visual Studio / Visual Studio Code 进行容器化开发和调试

容器打包和发布到运行在Azure中的容器化主机

容器化你的持续交付管道（测试-生产环境）

使用ARM模板搭建运行在Azure容器服务的Docker Swarm集群

使用TFS搭建持续交付管道



2016

State of DevOps Report

Presented by:



puppet +

DORA
DEVOPS RESEARCH & ASSESSMENT

Sponsored by:



Hewlett Packard
Enterprise

ThoughtWorks

splunk>



Technologies

Atlassian

Automic

REVOLUTION

DevOps的商业价值

- 速度

- 提升部署速度 200x，加快问题修复速度 24x

- 质量

- 降低变更错误率3x，降低线上问题修复时间 2500x，减少22%的意外问题

- 安全性

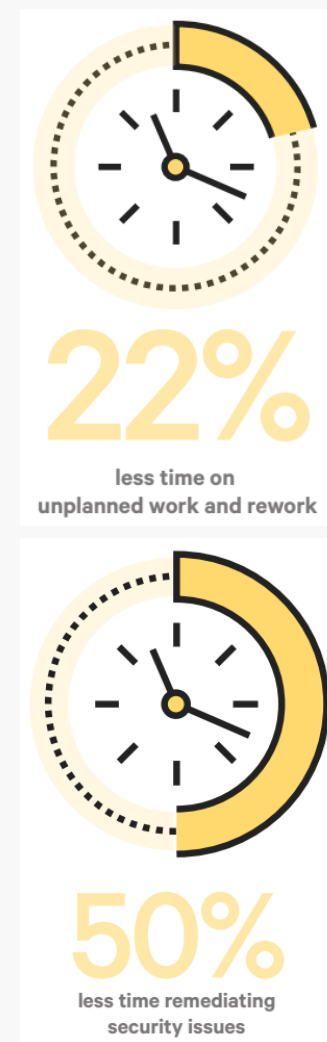
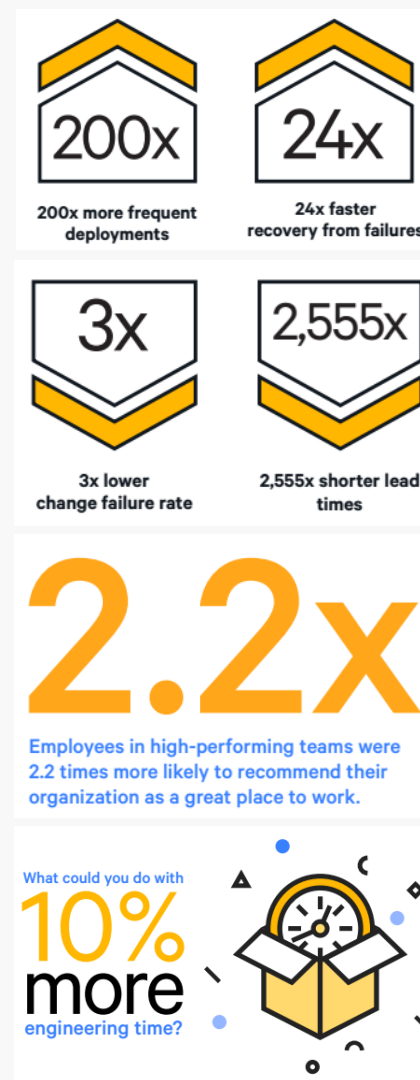
- 减少一半的安全性问题

- 员工满意度

- 更高的工作热情和主动性，更愿意向其他人推荐自己的企业和产品

- ROI投入产出比

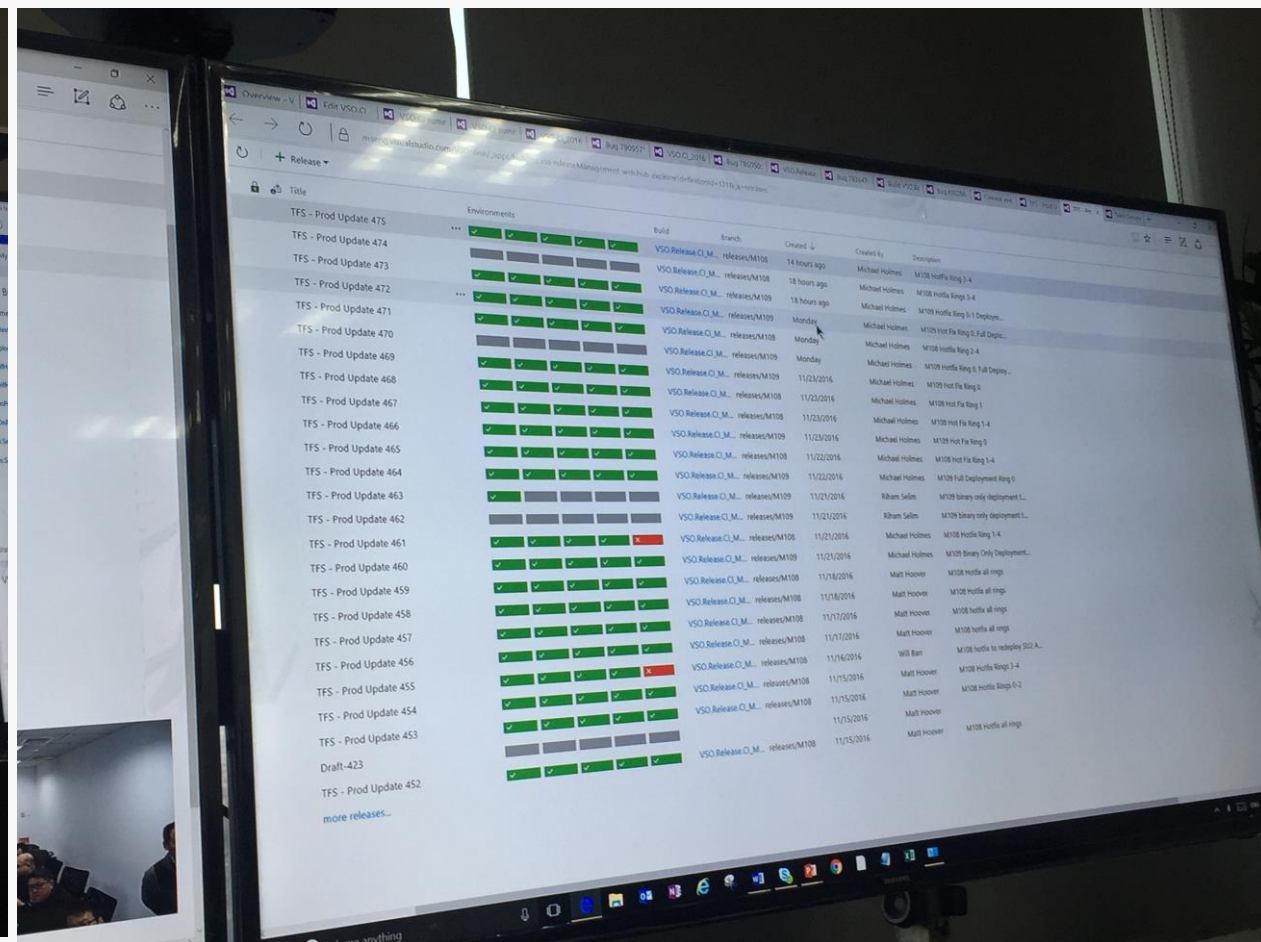
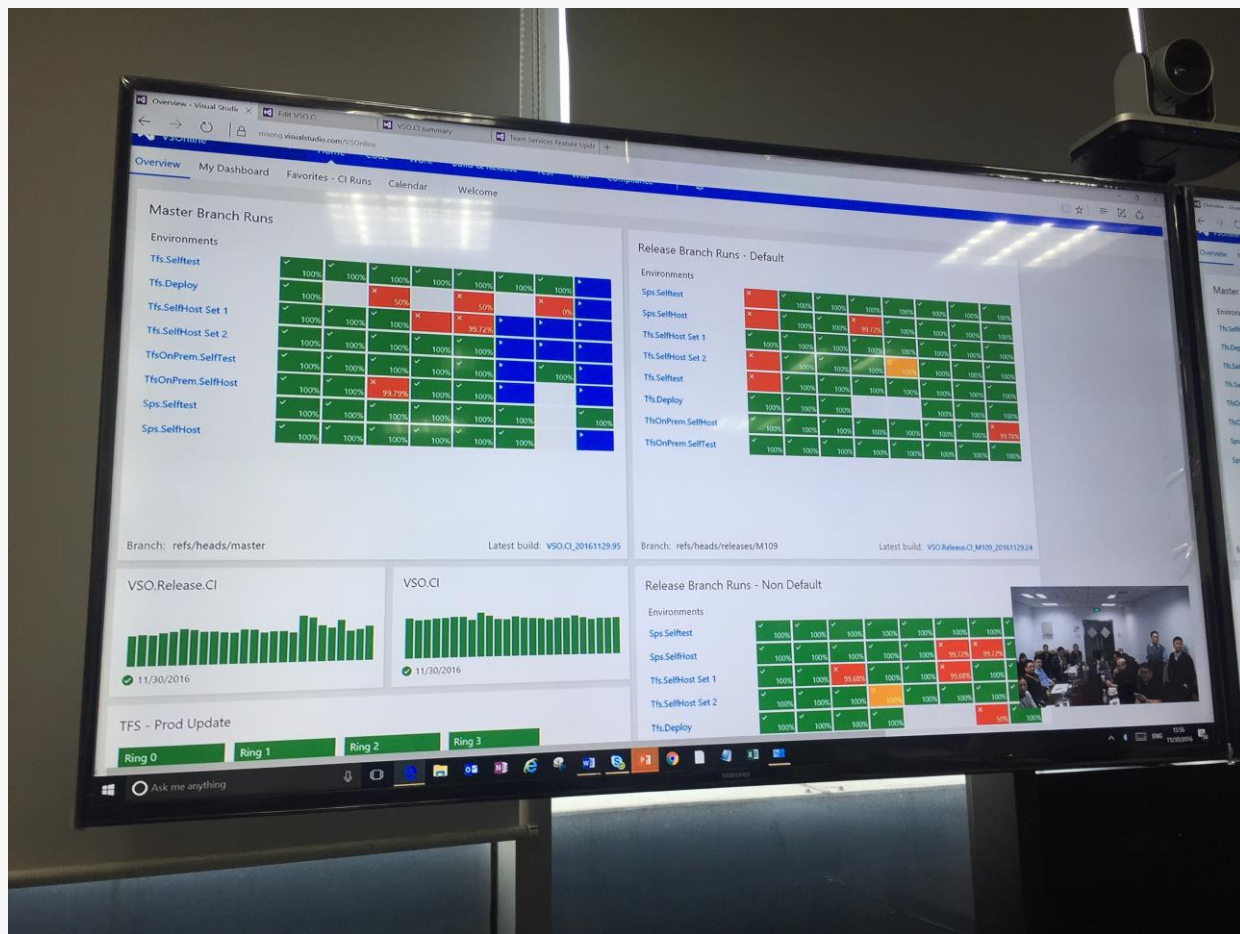
- 减少重复性工作，减少宕机时间，让员工有更多的时间投入更有意义的工作，如新功能开发



数据来源：“PuppetLab 2016 State of DevOps Report”中汇集了过去5年中超过2,500名来自全球的技术专业人士的调研结果。

<https://puppet.com/blog/2016-state-of-devops-survey-here>

微软DevOps转型



容器的价值

一次构建，多处运行

- **干净的，环境独立的，可迁移**的运行平台
- 在多次部署中不必担心依赖，包含环境相关配置
- 隔离性，同时运行不同版本的库和依赖环境，而不用担心他们互相影响
- 自动化测试，集成，打包过程；全部可以通过**简单的脚本**实现
- 降低与不同应用运行平台的兼容性问题
- 享受VM所提供的隔离性，快照等能力，同时又**不被笨重的VM所拖累**

配置一次，运行任何应用

- 让应用生命周期管理变得更加高效，**统一可复制**
- 提升开发人员的**代码质量**
- 消除开发，测试，生产和客户定制化环境的差异性
- 为不同职能/技能的人员**各司其职**提供了条件
- 大大**提升CI/CD的可靠性**，速度和可复制性
- 应为容器非常轻量，VM所存在的性能，成本，部署和可迁移性问题都迎刃而解

持续交付实施框架



持续交付实施框架 – 容器的定位/价值



内容

DevOps与持续交付，为什么？

DevOps的商业价值

容器对持续交付的意义

容器化你的开发环境

使用Visual Studio / Visual Studio Code 进行容器化开发和调试

容器打包和发布到运行在Azure中的容器化主机

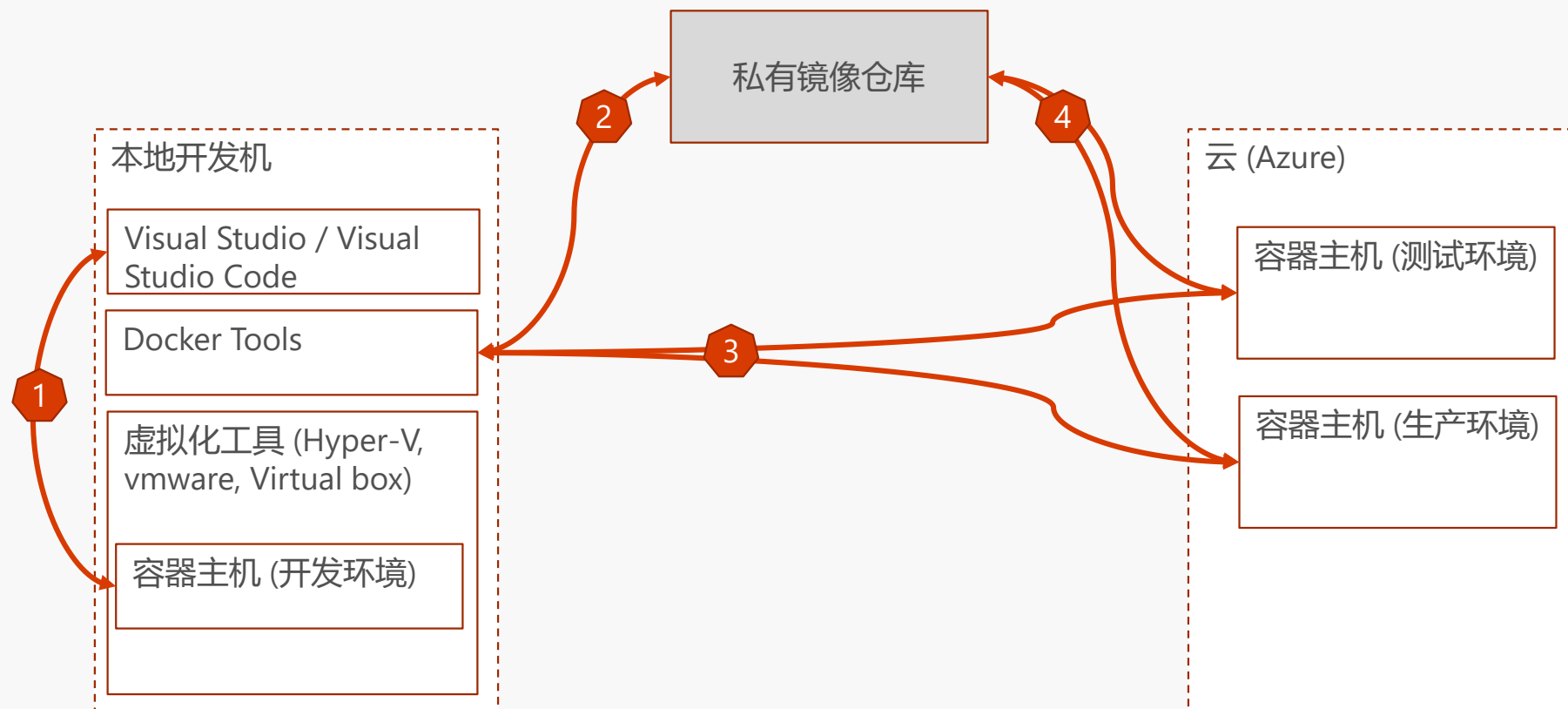
容器化你的持续交付管道（测试-生产环境）

使用ARM模板搭建运行在Azure容器服务的Docker Swarm集群

使用TFS搭建持续交付管道

个人开发者或小团队方案





1. 编码并使用本地容器环境进行调试
2. 使用本地Docker Tools构建容器镜像，并推送到私有仓库
3. 使用Docker Machine在Azure中创建容器主机
4. 部署容器到Azure中的容器主机



ASP.NET Core 跨平台开发

- 使用C#编写可以运行于Windows / Mac / Linux上跨平台应用
- 可以使用Windows / Mac / Linux 作为开发环境
- 兼容使用IIS, Azure Web应用或Docker容器的方式进行部署

ASP.NET 4.6 and ASP.NET Core 1.0

ASP.NET 4.6		ASP.NET Core 1.0	
.NET Framework 4.6 		.NET Core 1.0   	
.NET framework libraries		.NET core libraries	
Compilers and runtime components (.NET Compiler Platform: Roslyn, C#, VB, F# Languages, RyuJIT, SIMD)			



<https://www.asp.net/core>

Docker for Windows 容器化你的开发环境

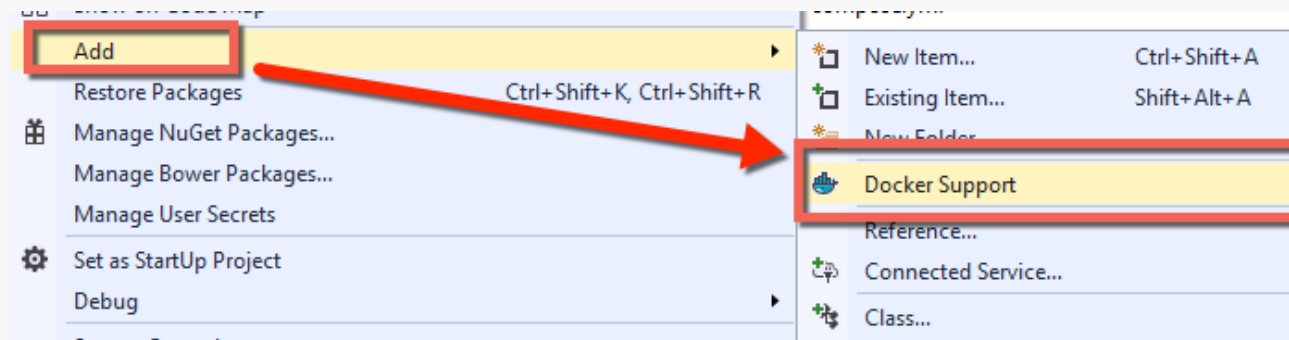
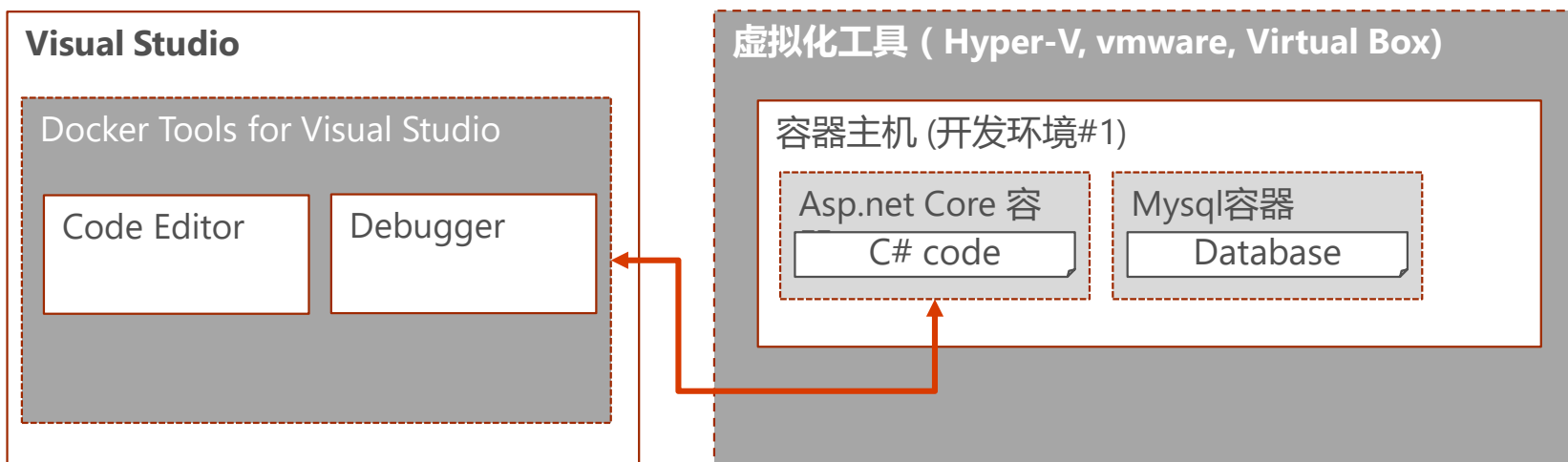
- 创建完全隔离的开发环境容器
- 使用Docker Tools for Visual Studio完成一键编译，打包镜像，部署镜像并启动调试
- 通过Visual Studio设置断点，单步调试运行在容器中的代码



Visual
Studio



本地开发机



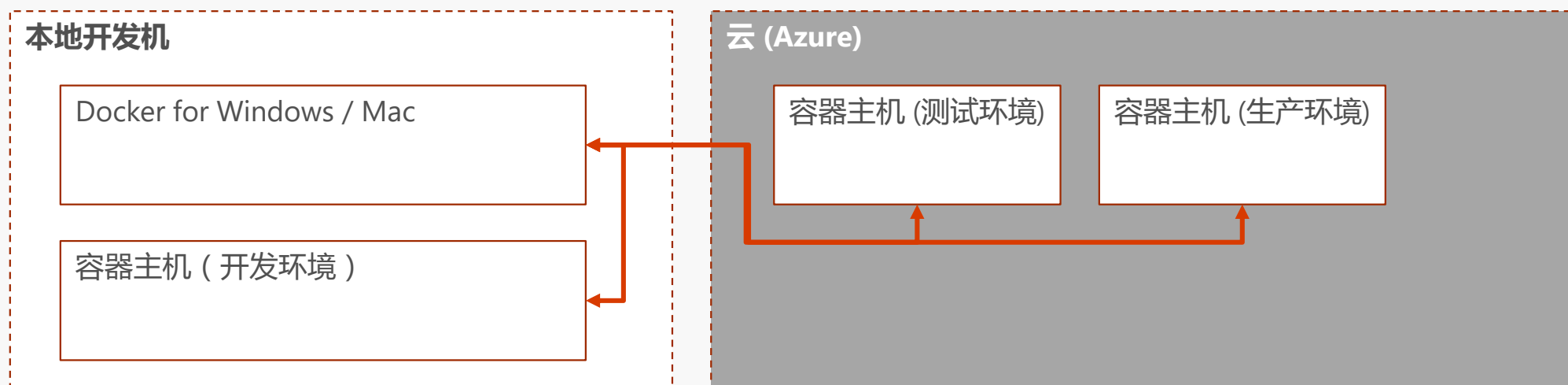
扫码阅读文档



演示

docker-machine --driver azure

- 在本地开发机直接管理Azure数据中心的容器化主机
- 自动完成TLS连接配置 (SSH KEY配置)



```
docker-machine create --driver azure \  
--azure-environment "AzureChinaCloud" \  
--azure-subscription-id {sub-id} \  
--azure-open-port 80 \  
--azure-location "ChinaNorth" \  
{machine-name}
```

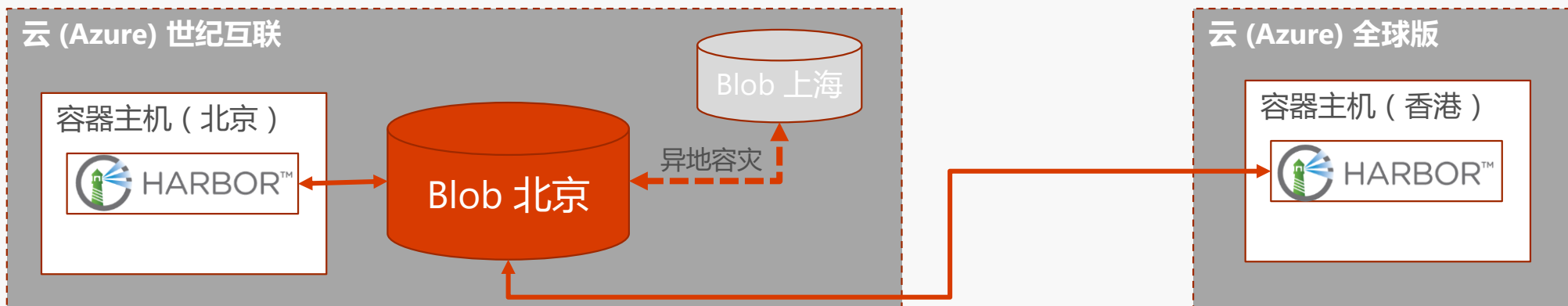
<https://docs.docker.com/machine/overview/>

扫码阅读文档



私有镜像仓库（基于Azure云存储）

- 使用Azure云存储作为仓库的后端数据存储，容器可以随时回收，数据被持久化，并可以使用异地容灾
- 借助Azure全球数据中心之间的高速网络，在香港的容器主机中从 docker hub 拉取镜像并为国内环境提供高速访问



```
docker run -d -p 5000:5000 \
-e REGISTRY_STORAGE=azure \
-e REGISTRY_STORAGE_AZURE_ACCOUNTNAME="{Account Name}" \
-e REGISTRY_STORAGE_AZURE_ACCOUNTKEY="{ access key }" \
-e REGISTRY_STORAGE_AZURE_CONTAINER="{storage container name}" \
-e REGISTRY_STORAGE_AZURE_REALM="core.chinacloudapi.cn" \
--name=registry \
--restart=always \
registry:2
```

扫码阅读文档



<https://www.azure.cn/home/features/storage/>

内容

DevOps与持续交付，为什么？

DevOps的商业价值

容器对持续交付的意义

容器化你的开发环境

使用Visual Studio / Visual Studio Code 进行容器化开发和调试

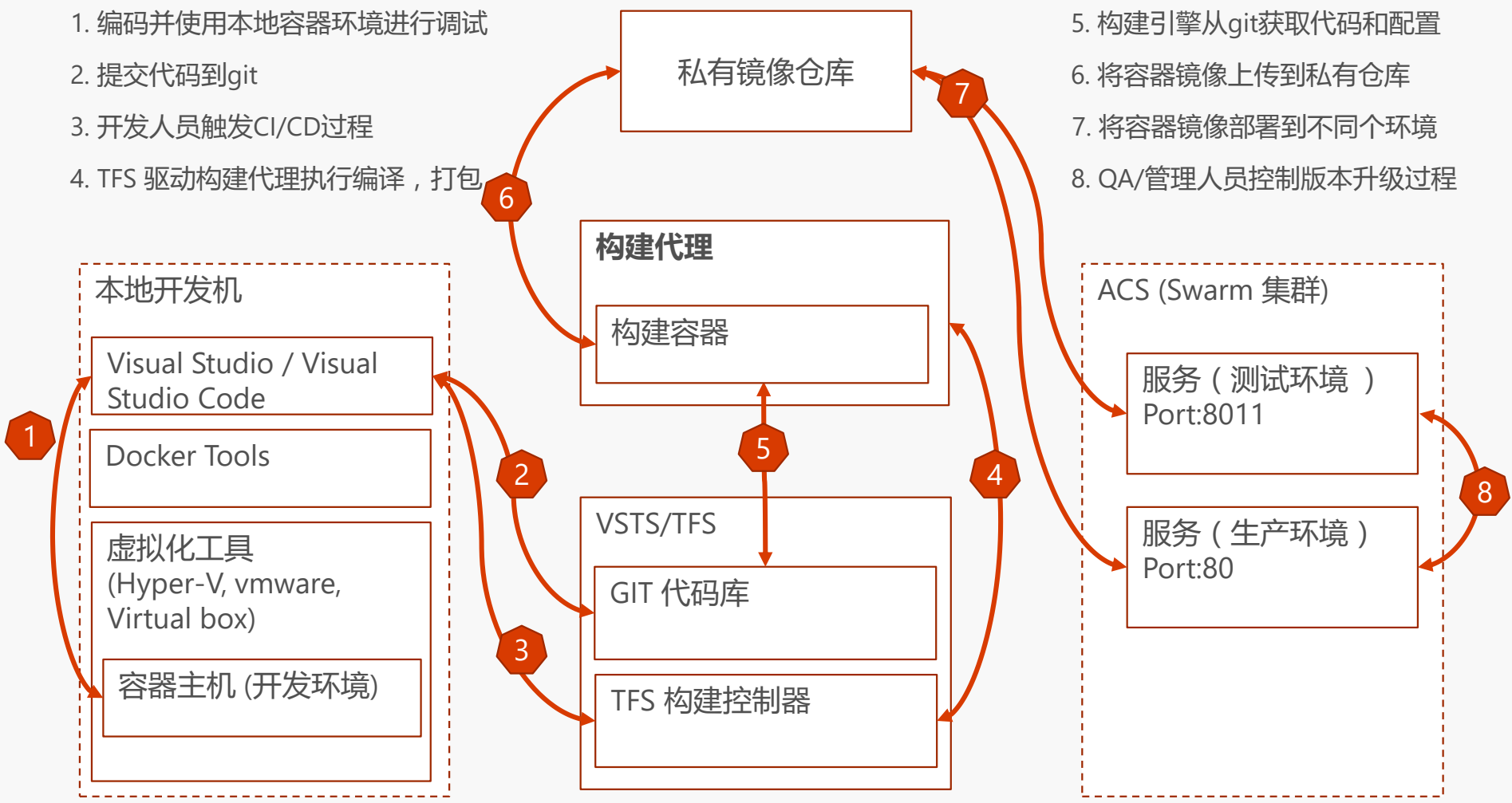
容器打包和发布到运行在Azure中的容器化主机

容器化你的持续交付管道（测试-生产环境）

使用ARM模板搭建运行在Azure容器服务的Docker Swarm集群

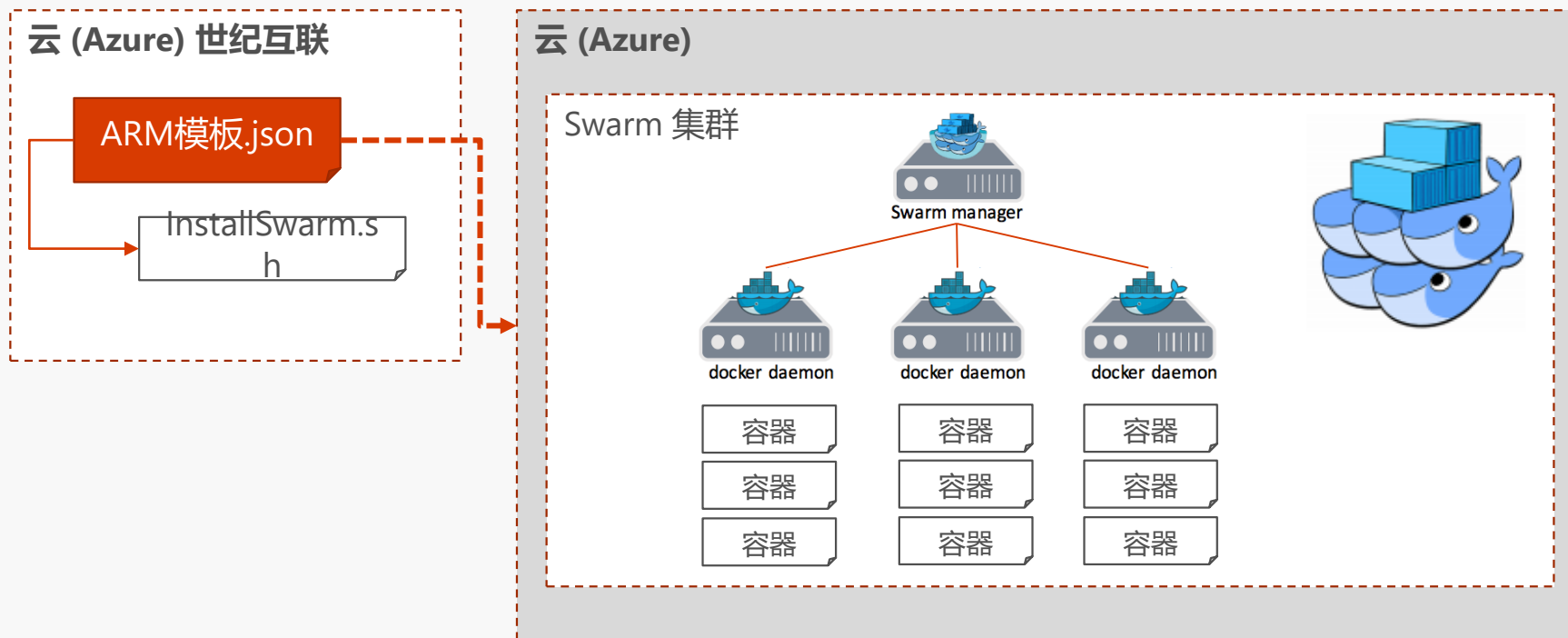
使用TFS搭建持续交付管道

中大型团队方案



Swarm集群(Azure ARM 模板部署)

- 使用Azure Resource Manager 模板一键完成 Docker Swarm 集群部署
- 自动完成TLS连接配置 (SSH KEY)

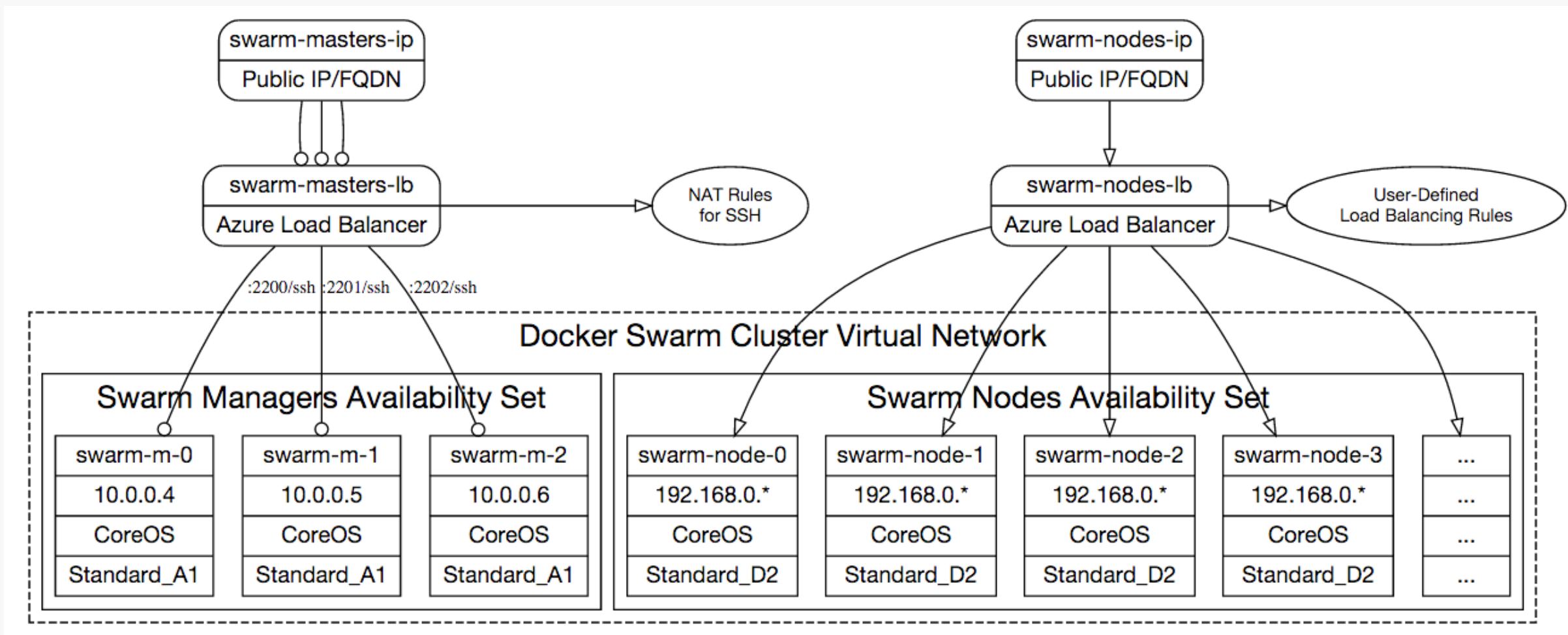


扫码阅读文档



<https://github.com/ups216/DockerSwarm>

Swarm集群架构



容器化的DevOps 发布管道

- 使用VSTS/TFS作为持续集成(CI)和持续部署(CD)工具
- 驱动运行于Azure中的容器化主机中的Build Agent完成镜像构建，推送及部署

Visual Studio Team Services / Team Foundation Server

GIT 代码库

持续集成

持续部署

本地开发机



云 (Azure)

容器化主机

Build Agent

Asp.net Core 容
C# code

Java SDK 容器
JAVA code

Node JS 容器
javascript

私有镜像仓库

Asp.net Image

Node Image

Java Image

Docker Swarm 集群



服务

Node #1

Node #2

服务

Node #1

Node #2

Node #3

Node #4

演示

演讲者姓名
演讲者职务

内容

DevOps与持续交付，为什么？

DevOps的商业价值

容器对持续交付的意义

容器化你的开发环境

使用Visual Studio / Visual Studio Code 进行容器化开发和调试

容器打包和发布到运行在Azure中的容器化主机

容器化你的持续交付管道（测试-生产环境）

使用ARM模板搭建运行在Azure容器服务的Docker Swarm集群

使用TFS搭建持续交付管道