

Windows Azure 云服务生命周期

姓名
职位
公司



Agenda



部署

最小化下线时间

调试

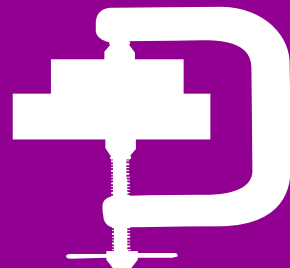
部署



2个独立的环境



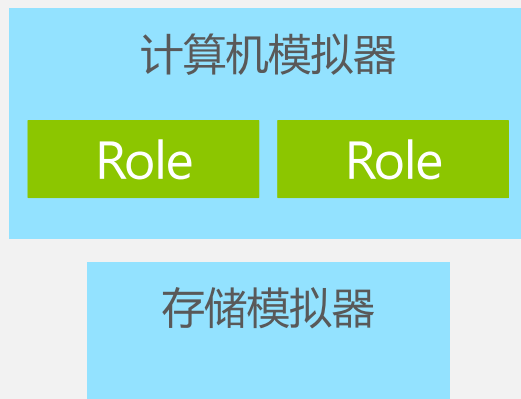
Production



Staging

服务部署的步骤

第一阶段:
本地开发和测试



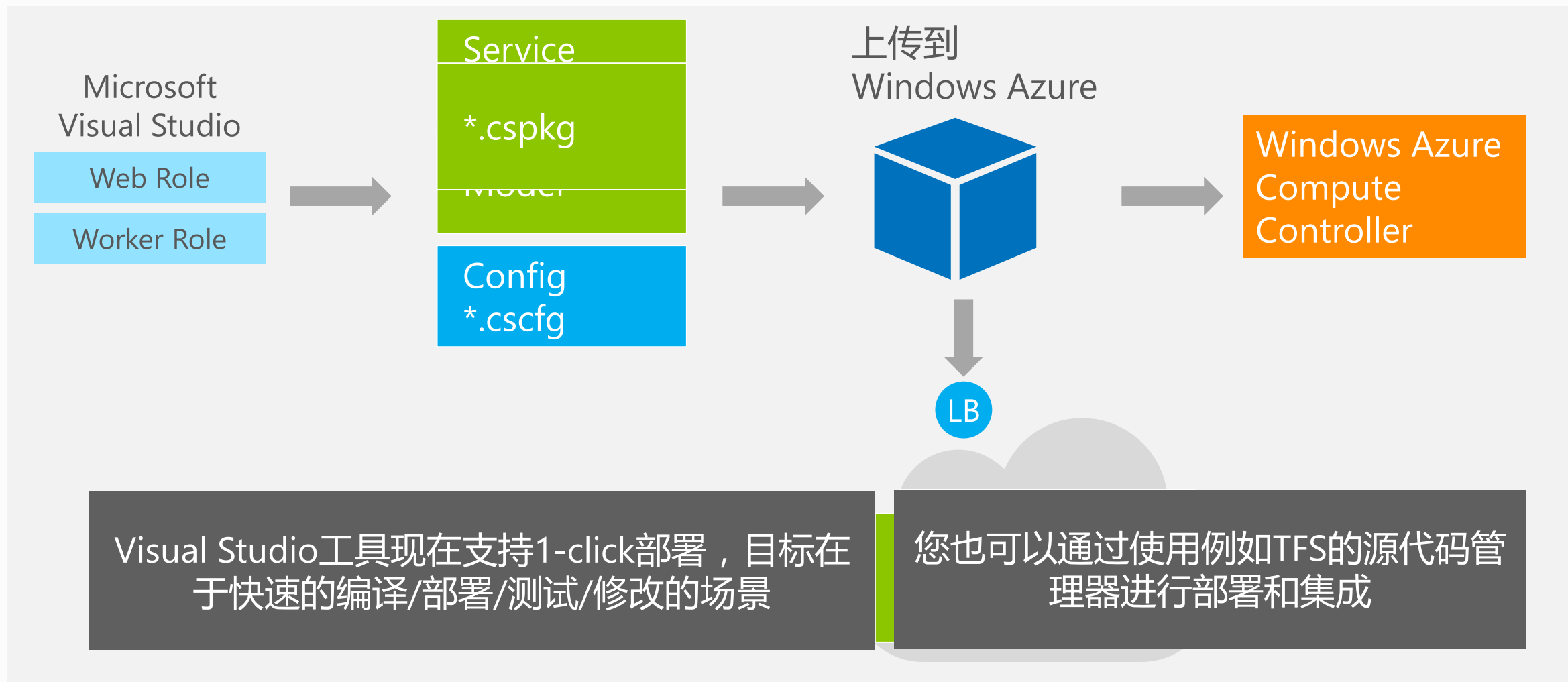
第四阶段:
使用混合模式,
测试云端的数据



第三阶段:
在Staging模式下测试



打包和部署

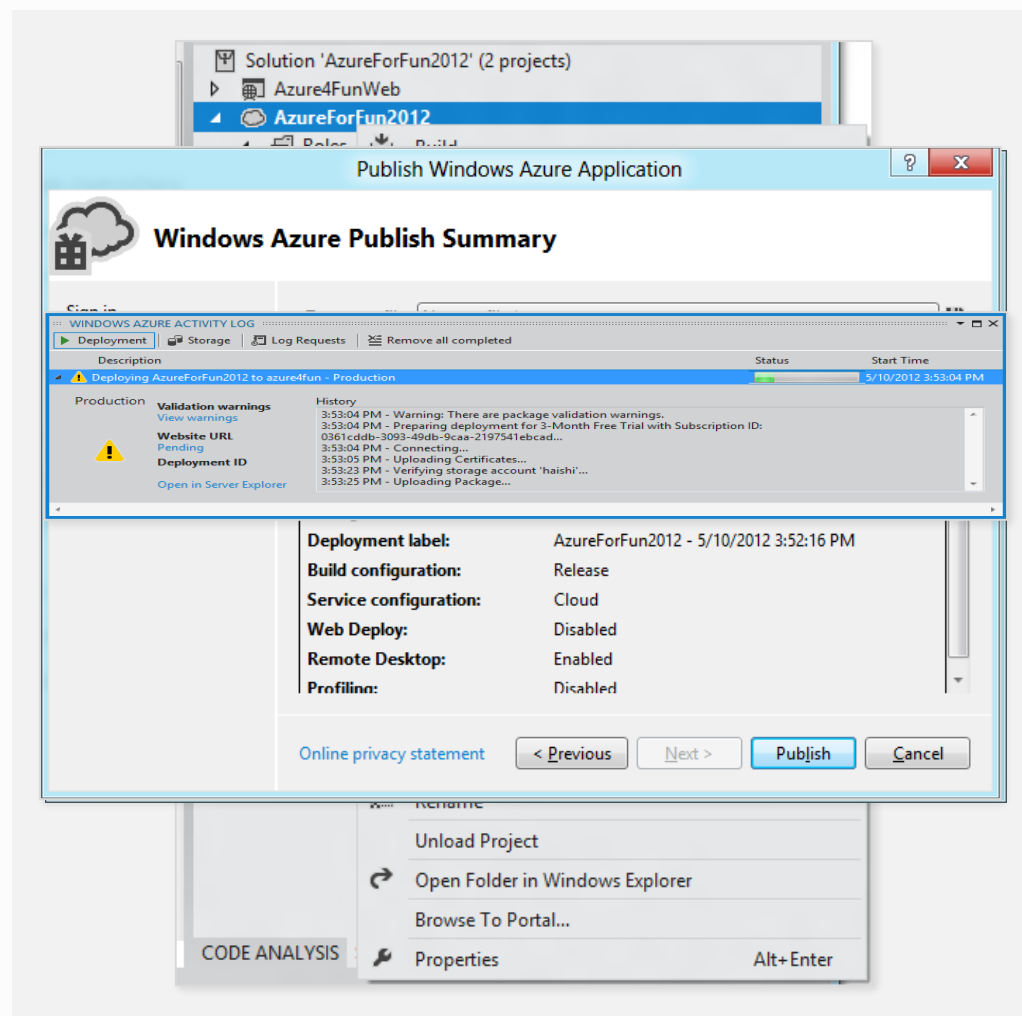


从Visual Studio进行部署

发布到Production 或Staging
只打包或者部署到Windows Azure
需要3件事情

Windows Azure subscription
Windows Azure hosted service
Windows Azure storage account

使用x509 证书认证



地理位置和地缘组



地缘化的

依赖的资源
保证地理位置在一个数据中心
也能够设置子区域（推荐）



具体地理位置

清楚地基于服务定义子区域



非地缘化的

Windows Azure选择子区域
(不推荐)



地缘组和地理位置设置是无法改变的

客户操作系统版本

Windows Azure运行在Base OS上

Base OS定期更新
最好定义操作系统版本
默认获得最新的版本-使用版本"*"获得自动更新
使用osVersion 属性
在云的配置信息中定义版本

```
<ServiceConfiguration serviceName="CloudService1" osVersion="WA-GUEST-OS-1.2_201003-01">
```

发布在此:

<http://msdn.microsoft.com/ee924680>

| 发布 | 更新 |
|---------------------------|-------------------------------|
| WA-GUEST-OS-1.0_200912-01 | 补丁 |
| WA-GUEST-OS-1.1_201001-01 | 补丁 |
| WA-GUEST-OS-1.2_201003-01 | 补丁 URL重写和压缩 .NET 4.0 RC |
| WA-GUEST-OS-1.3_201004-01 | 不懂, .NET 4, URL 重写 2 |

Windows Azure服务管理API

基于REST

使用X509证书作为客户端认证

托管服务

查看，创建
删除部署，
交换部署，
修改配置，
扩展，
更新部署
遍历更改域

存储账户

列出账户
获取账户属性
获取账户Key
重新生成Key

证书

列出
增加
获取
删除

地缘组

列出
获得属性

系统信息

获取OS版本



Windows Azure Tools for Visual Studio

CSPack

CSRun

CSManage

Powershell Cmdlets



Visual Studio 工具

项目模板

用户界面扩展

存储浏览器

服务器浏览器

集成的部署

智能跟踪调试

剖析支持



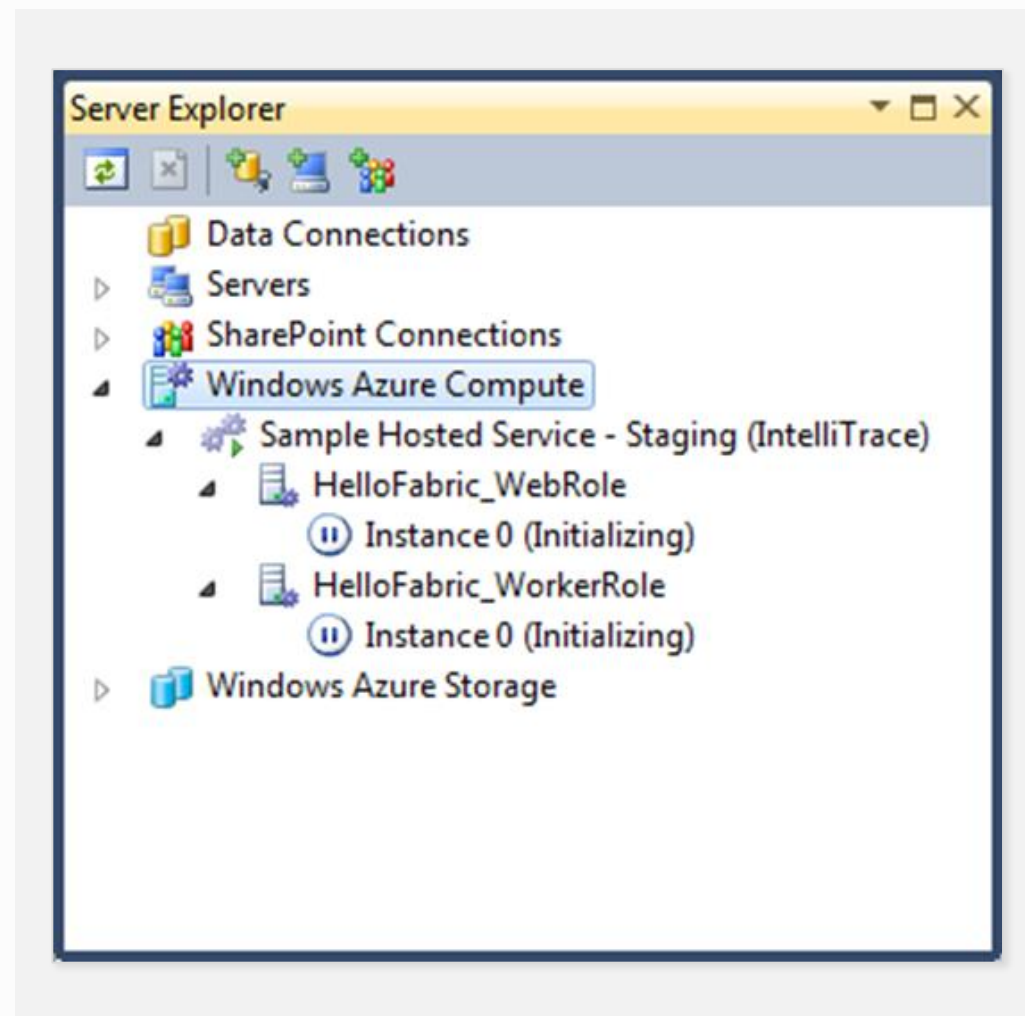
Windows Azure 服务器管理器

从Visual Studio查看和监视部署

不能启动或停止实例

使用x509证书认证

定时查询Windows Azure自动更新



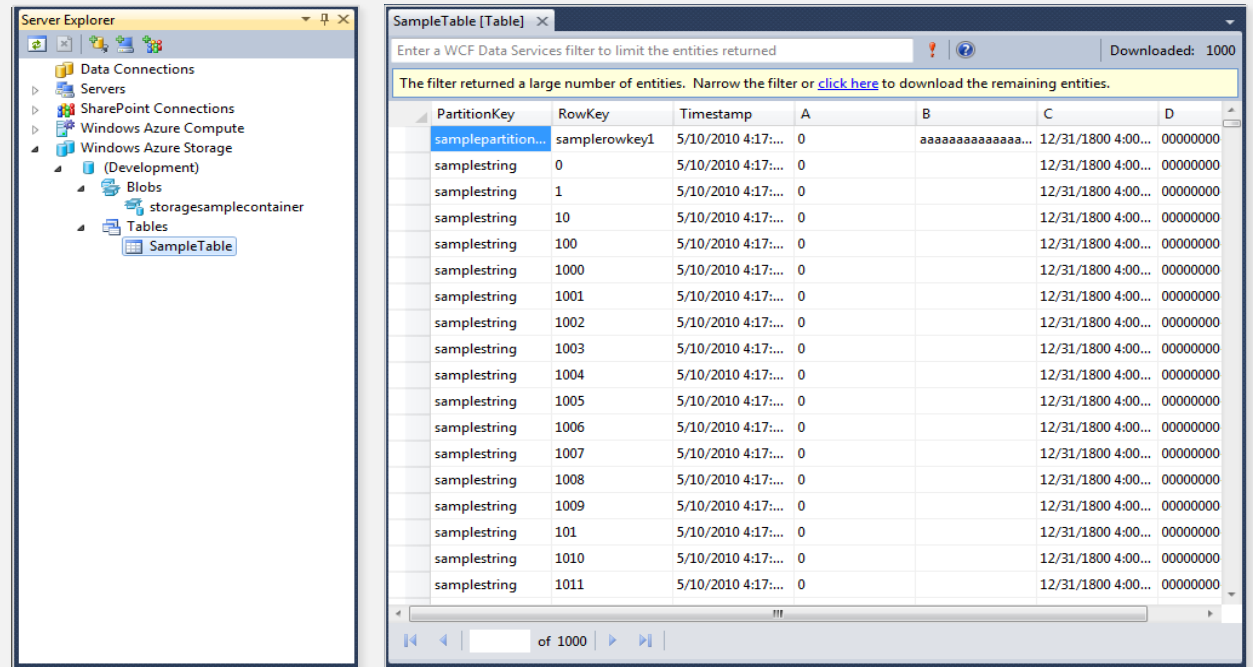
Windows Azure 存储管理器

Visual Studio 服务器管理器的扩展

查看Windows Azure 存储的数据

异步下载

本地缓存Blob数据



CSPack.exe

Windows Azure SDK的一部分

命令行工具;
被VS.NET工具调用

为部署打包服务

确定服务包的名称&位置

```
cspack HelloFabric\ServiceDefinition.csdef  
    /role:HelloFabric_WebRole; HelloFabric_WebRole  
    /out:HelloFabric.cspkg
```

给云部署打包

```
cspack HelloCloudService.csdef  
    /role:HelloCloudServiceWebRole; HelloCloudService  
    /generateConfigurationFile:HelloCloudService.cscfg  
    /out:HelloCloudServicePackage  
    /copyOnly
```

给Dev Fabric部署打包

CSRun.exe

命令行工具;
被VS.NET工具调用

将服务部署到Fabric

管理运行中的服务

转储已部署实例的日志

控制fabric的状态

控制开发存储服务

```
cshrun myservice.csx myservice.cscfg /launchbrowser
```

运行+打开浏览器

```
cshrun /devfabric: shutdown  
cshrun /devfabric: clean
```

通知& 清楚持久的
状态
例如.本地存储

```
cshrun /devstore: start
```

启动开发存储

X.509 证书

服务管理使用证书认证

自行签名

任何Key \geq 2048 bits的有效X.509 v3证书



```
makecert -r -pe -a sha1 -n CN=AzureMgmt -ss My -sky exchange "AzureMgmt.cer"
```

创建一个新的自我签名的证书，写到当前目录下 "AzureMgmt.cer" 文件中，并且存储到
CurrentUser\My certificate store.

准备使用管理API

创建/ 获取证书

注意ValidFrom,
服务器上的本地实践可能不同

准备Windows Azure 账户

上传证书到Windows Azure 账户
每个账户最多关联10张证书



准备使用管理API

获取指纹Id


获取订阅Id


生成服务

通过服务名称或;
使用管理API遍历服务



管理证书


Add Certificate


Delete Certificate

Certificates

Deployment Health

Affinity Groups

Management Certificates

Hosted Services (5)

Storage Accounts (5)

User Management

VM Images

CDN

Choose Columns

Filter management certificates

| Name | Type | Status | Subscription ID | Thumbprint | Valid from | Valid to |
|-----------------------------------|---------------|---------|-----------------------------------|-------------------------|----------------------|-----------|
| Field: | Subscription | Active | a772b688-93c3-45bc-aebe-c04e18... | | | |
| Windows Azure Tools Management... | Management... | Created | a772b688-93c3-45bc-aebe-c04e18... | 59149E6126D464DE6CA4... | 4/27/2011 5:11:06... | 4/27/2012 |

订阅Id和服务名称

The screenshot displays the Windows Azure Management Portal interface. On the left, a navigation pane lists various management categories: Deployment Health, Affinity Groups, Management Certificates, Hosted Services (5), Storage Accounts (5), User Management, VM Images, and CDN. The 'Hosted Services (5)' category is selected, and the 'WoodgroveMortgage' service is highlighted with a red circle.

The main content area shows a table of hosted services. The table has columns for Name, Type, and Status. The 'WoodgroveMortgage' service is listed with a status of 'Created'. A red circle highlights the 'WoodgroveMortgage' service name in the table.

Below the table, there is a section for 'Certificates' and 'Deployments'. The 'Deployments' section shows two deployments: 'MortgageApp - 11/14/2011' and 'MortgageApp - 11/22/2011', both with a status of 'Ready'.

| Name | Type | Status |
|-----------------------------------|--------------|--------|
| Field: | Subscription | Active |
| Windows Azure Tools Management... | Created | |

| Name | Type | Status |
|--------------------------|----------------|---------|
| Field: | Subscription | Active |
| DeployDebuggerDemo | Hosted Service | Created |
| MortgageRatesService | Hosted Service | Created |
| Server&TwitterFeed | Hosted Service | Created |
| WoodgroveMortgage | Hosted Service | Created |
| MortgageApp - 11/14/2011 | Deployment | Ready |
| WoodgroveMortgagWE | Hosted Service | Created |
| MortgageApp - 11/22/2011 | Deployment | Ready |

CSManage.exe

<http://tinyurl.com/azuresamples>

挂起一个命名的
Staging部署

```
csmanage /update-deployment /hosted-service :<service>  
/slot:staging /status:suspended
```

删除一个命名的
staging部署

```
csmanage /delete-deployment /hosted-service :<service>  
/slot:staging
```

使用Blob存储中的包
创建一个新的部署到
产品中

```
csmanage /create-deployment /hosted-service:<service>  
/slot:production  
/name:<name>  
/label:<label>  
/package:$(BlobStorageEndpoint)packages/ServicePackage.cspkg  
/config:$(SolutionDir)\ServiceConfiguration.cscfg
```

执行命名的Staging
部署

```
csmanage /update-deployment /hosted-service:<service>  
/slot:staging /status:running
```

PowerShell Cmdlets

封装服务管理API

通过管道，一条命令的输出可以是下一条的输入

```
$cert = Get-Item cert:\CurrentUser\My\D6BE55AC439FEA8CBEBBAFF432BDC0780F1BD00CF  
$sub = "CCCEA07B-1E9A-5133-8476-3818E2165063"  
$servicename = 'myazureservice'  
$package = "c:\publish\MyAzureService.cspkg"  
$config = "c:\publish\ServiceConfiguration.cscfg"
```

设置一些变量，包括
证书指纹

```
Add-PSSnapin WAPPSCmdlets
```

添加Azure管理元

```
Get-HostedService $servicename -Certificate $cert -SubscriptionId $sub |  
  Get-Deployment -Slot Production |  
  Set-DeploymentStatus 'Running' |  
  Get-OperationStatus -WaitToComplete
```

获取服务

获取产品槽

设置状态为运行

等待异步操作完成

```
Get-HostedService $servicename -Certificate $cert -SubscriptionId $sub |  
  Get-Deployment -Slot Production |  
  Set-DeploymentConfiguration {$_.RolesConfiguration["WebUx"].InstanceCount += 1}
```

向外扩展产品，增加
一个实例

自动化你的部署

设置服务

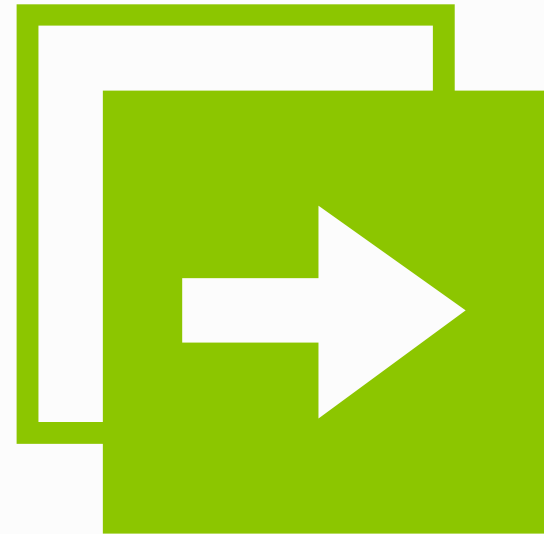
MSBuild

CSPack.exe

上传包到存储

CSManage.exe / PowerShell Cmdlet

部署
执行



自动化你的部署

测试

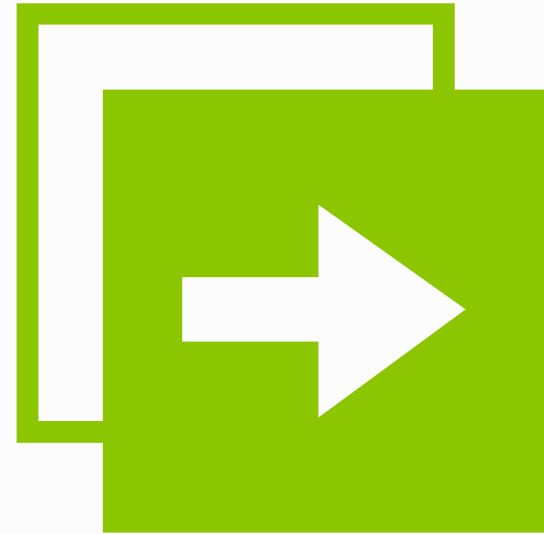
CSManage.exe / PowerShell Cmdlet

挂起

删除

MS Build 脚本举例:

<http://tinyurl.com/4uhal5t>



最小化下线时间

故障和更新域

故障域

代表一组可能一起出现故障的资源
即. 相同的机架，相同的服务器
Fabric将实例分布在故障域中
默认是2

更新域

代表一组一起更新的资源
通过在ServiceDefinition文件中upgradeDomainCount定义
默认是5

Fabric将更新域分布在故障域和角色中



更新的服务模型举例

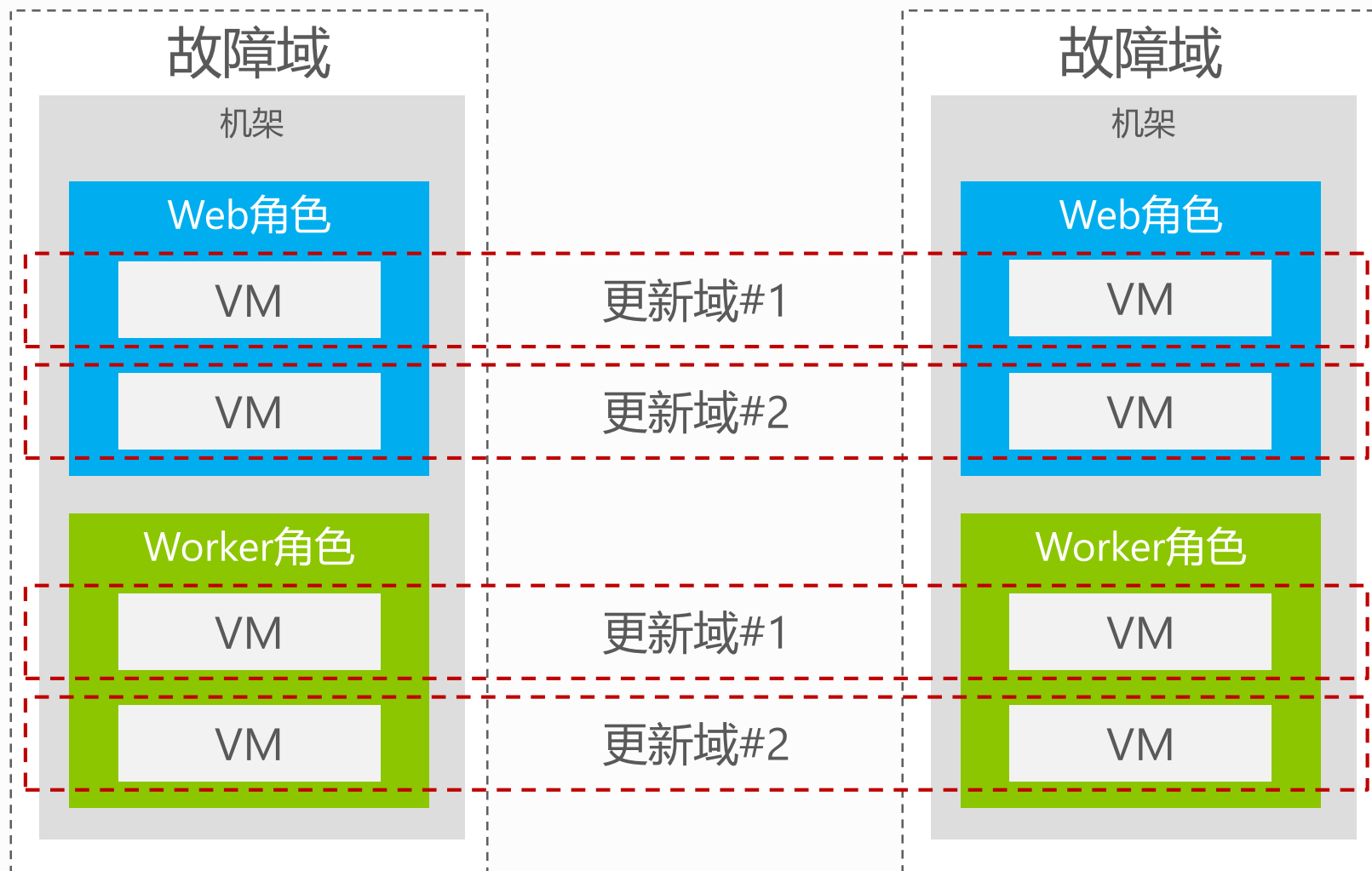


4个实例



4个实例

故障和更新域



虚拟IP交换

在2个槽中交换虚拟IP

Production 变成 Staging
Staging 变成 Production

允许快速交换环境

当服务模型变化，这是必须的

例外: 修改公开的endpoints需要删除部署

没有downtime

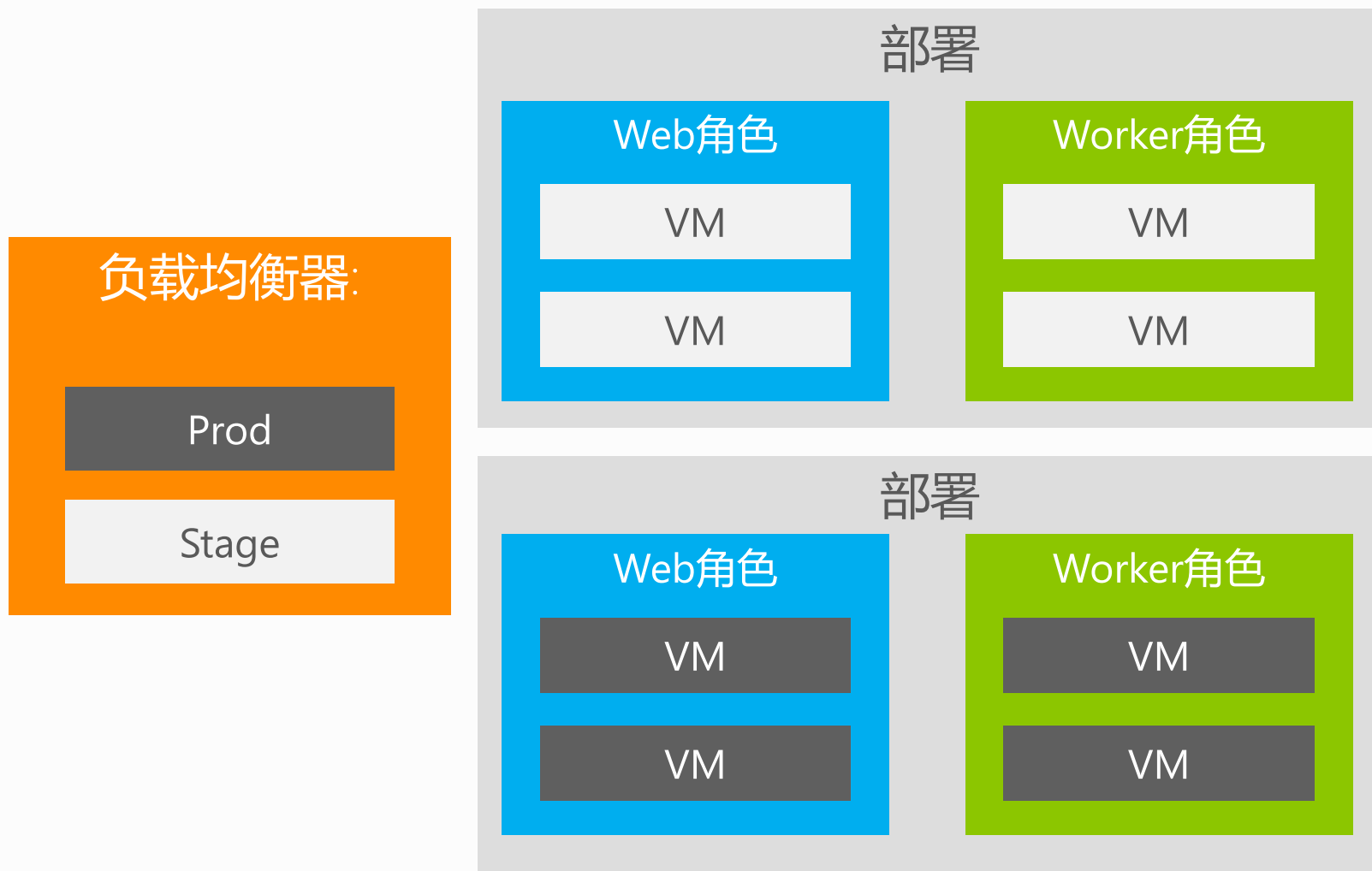
开发网站和服务管理API

维护相同的外部IP



为了最好的用户体验，在交换前，在Staging和Production使用相同数量的实例

VIP 交换更新



就地更新

对一个线上的服务进行滚动更新

利用更新域

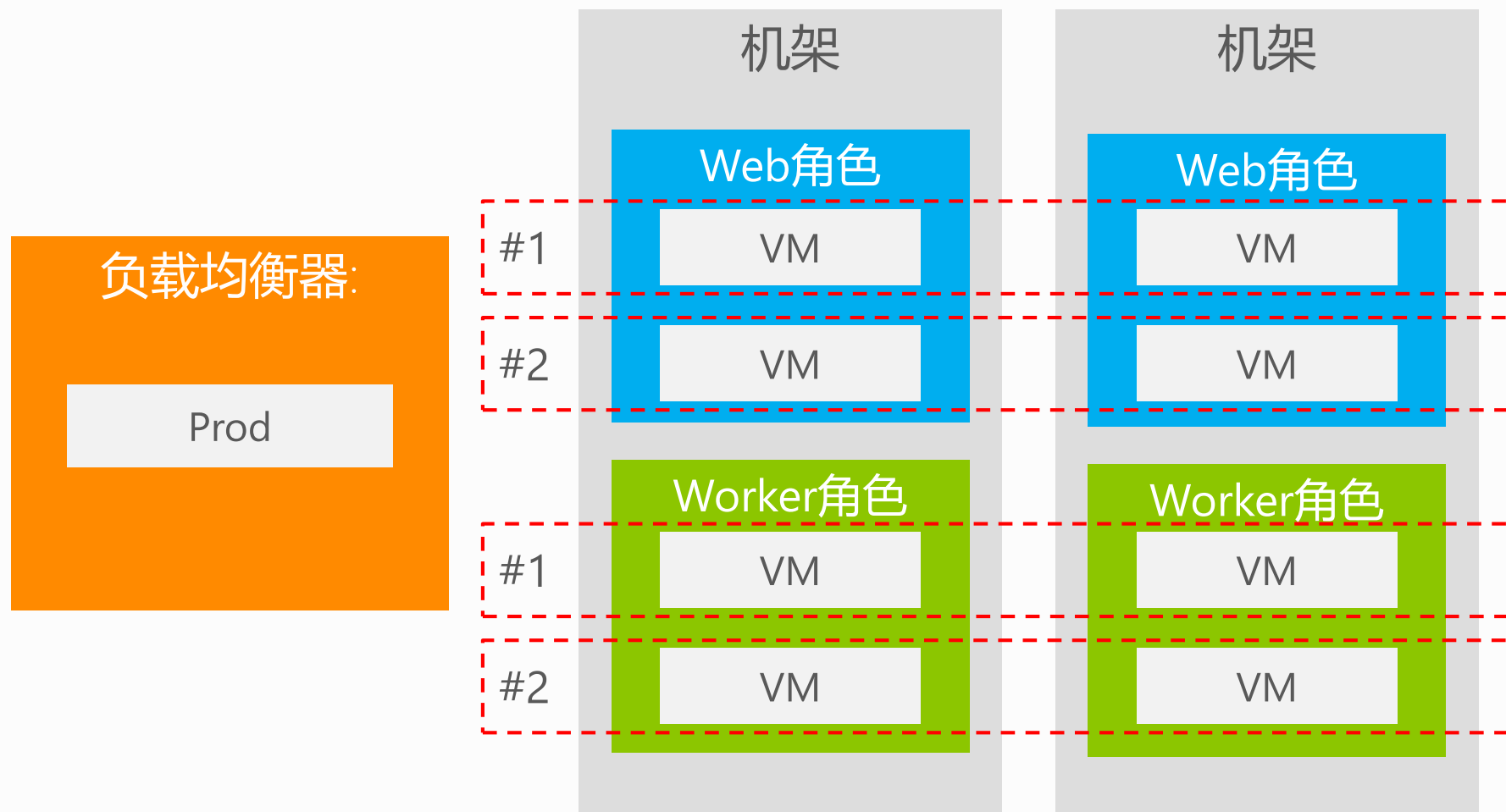
使用管理界面或者服务管理API

自动或者手动
操作系统补丁



总是假设您有旧的和新的版本
同时运行
编写能意识到版本的代码

就地更新



调试



在开发中调试服务

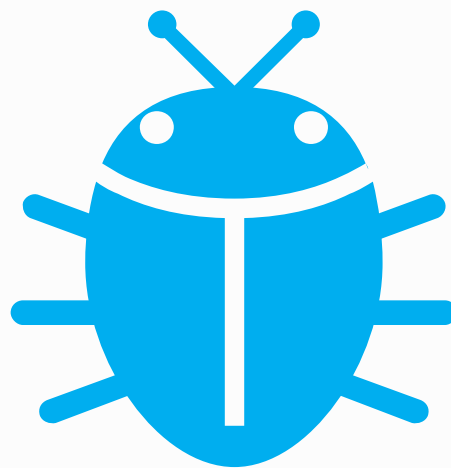
调试仅在开发Fabric中支持

Visual Studio在调试器启动时附加到所有的实例

在调试过程中修改配置

默认托管的调试器

支持本地代码



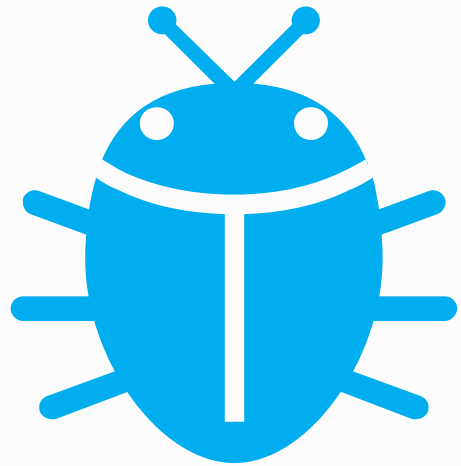
智能跟踪(IntelliTrace)

只用于调试场景

需要.NET 4 和Visual Studio 2010 Ultimate

必须在发布时启用

子进程无法使用智能跟踪

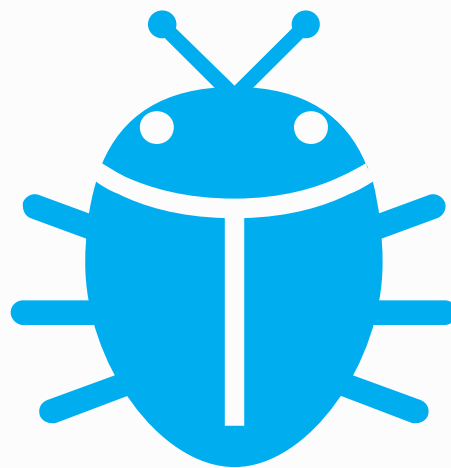


智能跟踪(IntelliTrace)

在VM的文件系统上收集日志

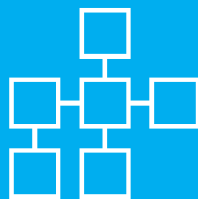
在Visual Studio的服务器管理器中收集日志

在Visual Studio中重现事件

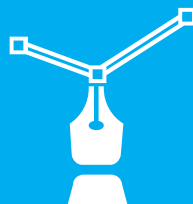


剖析

4种 方法



CPU 采样



注入



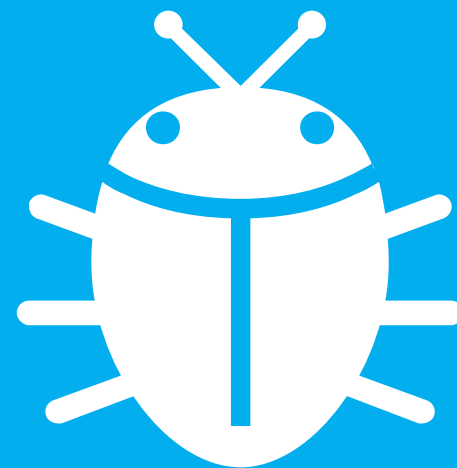
.NET 内存分配



并发

智能跟踪

演示



总结



部署

代码-> 打包-> 上传-> 部署
服务管理 Api
Windows Azure 工具
自动化整个生命周期

最小化Downtime

故障和更新域
就地升级 vs 虚拟IP交换

调试

Development Fabric
智能跟踪



© 2012 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Translated to Chinese Simplified Version by Shanghai Yungoal Info Tech Co., Ltd. [YunGoal](#)