

我们与Azure DevOps的距离

郭家齐

Microsoft Azure MVP

STUDY4
Study For Love

2019
Global Azure
BOOTCAMP

关于我

- Kingston Technology 信息处 经理
- Microsoft Azure MVP
- 台湾技术社群 讲师
- 2018 DOIS DevOps 深圳站 讲师
- 2018 DevOps Day 台北站 讲师
- 2018 Agile Tour 讲师
- 2018 Insider Dev Tour 台北站 讲师
- 2018 .NET Core Conf 台湾站 讲师
- 2017 Microsoft Global Tech Summit 北京站 讲师
- 2017 Agile Tour 讲师
- Global Azure Bootcamp 台湾站 讲师



*IoT Solution,
Azure Stream analysis,
Azure DataLake,
Azure Data factory,
Azure Machine Learning,
Azure Blob
Azure Function
Azure Container
Azure Application Insights
IoT Hub*

DevOps是什么？

“

DevOps是结合人、过程和工具，
并能够持续向最终用户交付价值
的方式。

”

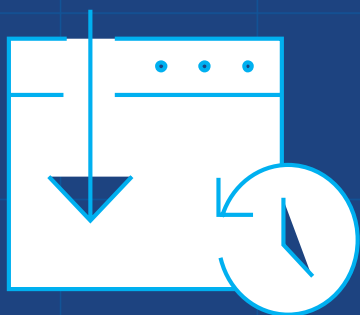


DevOps 四大元素



需要哪些核心技术来更快地部署和学习？

- DevOps将人员、流程和技术结合在一起，使软件交付自动化，为用户提供持续的价值。
- 使用Azure DevOps，可以更快、更可靠地交付软件，无论您的IT部门有多大，或者您使用的开发工具是什么



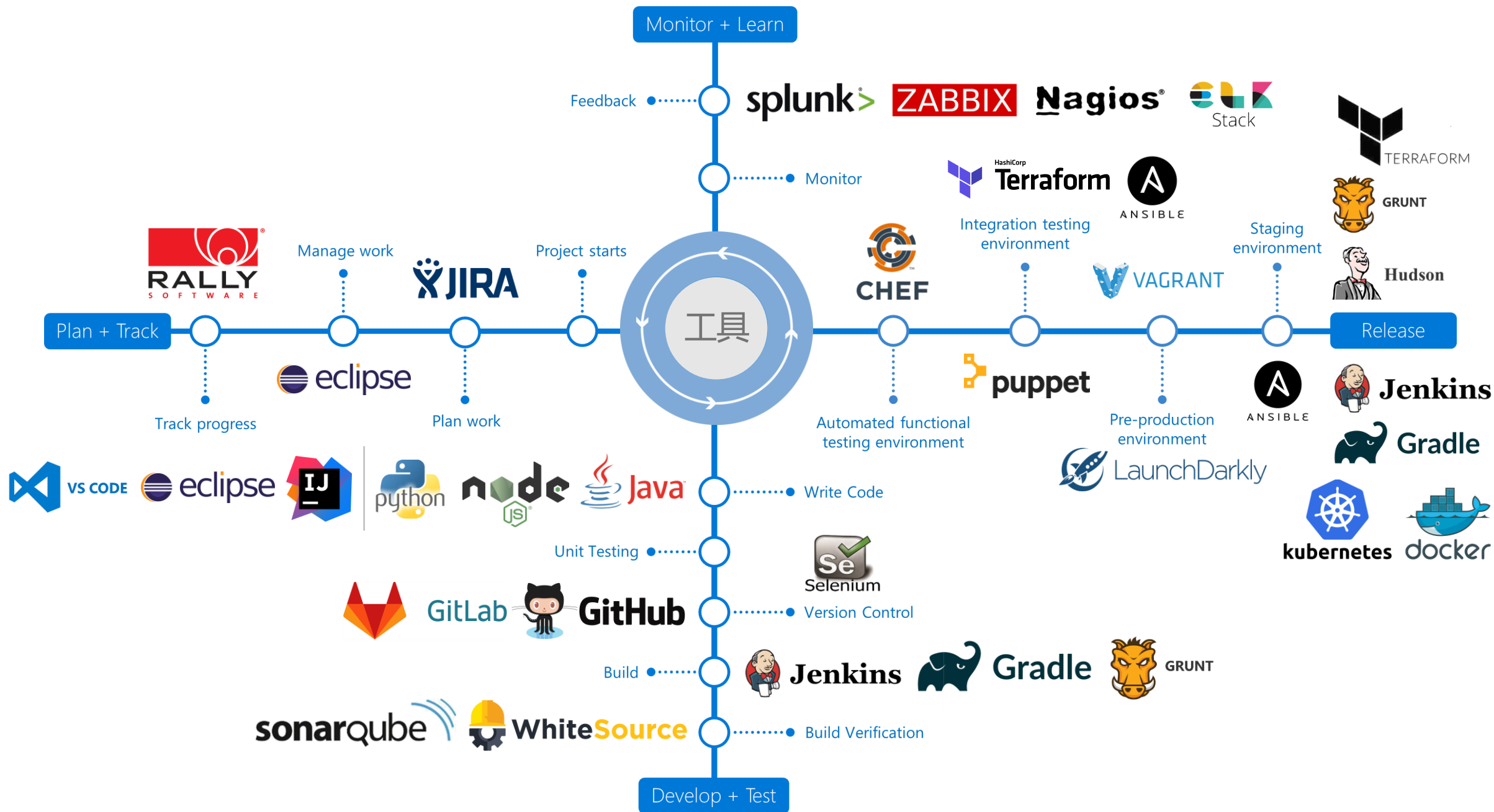
持续集成



持续交付



持续学习 & 监控



Azure DevOps

- Team Foundation Server
 - Azure DevOps Server (线下版本)
- Visual Studio Team Service
 - Azure DevOps Service (云版本)

Azure DevOps 五大集成



Azure Boards

使用敏捷式工具与所有成员一起规划、追踪及讨论工作，更快传递价值给您的使用者



Azure Pipelines

适用于任何语言、平台和云端的CI/ CD建置、测试及部署。联机到GitHub或任何其他Git提供者并持续部署。



Azure Repos

取得无限的云端托管私人Git存放库，并透过提取要求和进阶档案管理来共同作业以建置更佳代码。



Azure Test Plans

使用手动与探索测试工具放心测试及交付。

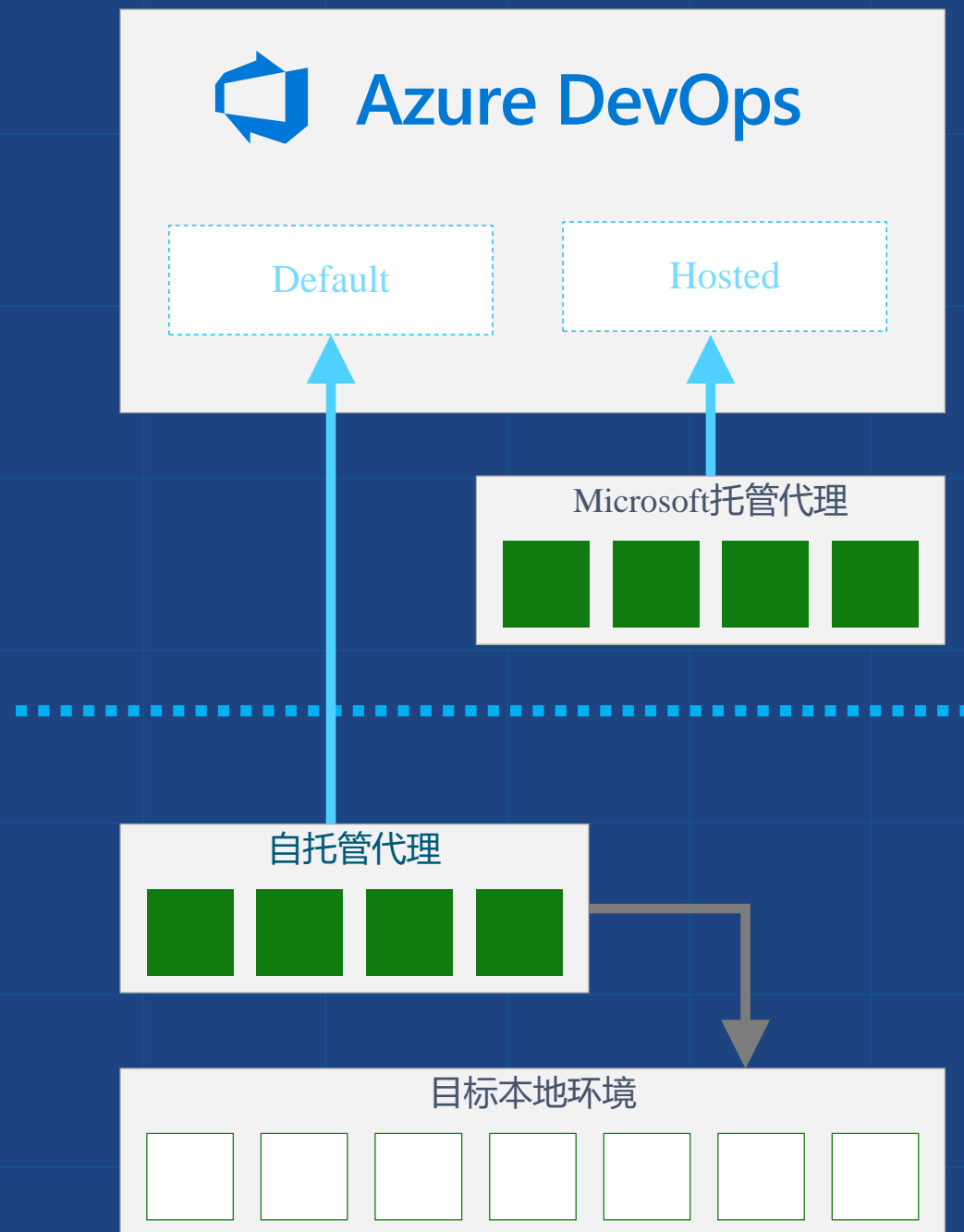


Azure Artifacts

建立、装载套件并与您的成员共享，只要单击即可将成品新增至您的CI/ CD管线。

云到端的集成

- 代理引擎
 - 可以被建立在云或是端
 - 从云部署到地端服务器
 - 使用部署群组，动态调配部署目标类型
- 实践Azure DevOps部署到企业内部服务器



本日要点

- 建立订制化任务群组
- 善用变量群组
- 管线即代码
- 容器解决方案
- 扩充移动端
- Azure DevOps 結合Microsoft Teams

本日要点

- 建立订制化任务群组
- 善用变量群组
- 管线即代码
- 容器解决方案
- 扩充移动端
- Azure DevOps 結合 Microsoft Teams

Azure 管线

用于Linux、Windows和MacOS的云托管管道，
开放源代码的时间不受限制。



任何语言、任何平台、任何云

构建、测试和部署NoDE.JS、Python、Java、PHP、露比、C/C++、.NET、Android和iOS应用程序。在Linux、MacOS和Windows上并行运行。部署到Azure、AWS、GCP或企业内部



可扩展的

探索和实现广泛的社区构建、测试和部署任务，以及从slack到sonarcloud的数百个扩展。支持Yaml、报告等



Containers & Kubernetes

轻松构建容器并将其推送到容器注册表，如Docker Hub和Azure容器注册表。将容器部署到各个主机或kubernetes



开放源代码的同类最佳产品

确保每个开源项目的快速持续集成/持续交付（CI/CD）管道。在Linux、MacOS和Windows上为所有具有最多10个免费并行作业的开源项目获得无限的构建时间

The screenshot displays the Azure DevOps web interface. The top navigation bar shows the project name 'AdventureWorks Mobile' and the pipeline path 'Pipelines / Builds / 10382'. The left sidebar contains a menu with 'Overview', 'Pipelines', 'Builds', 'Releases', 'Library', and 'Deployment groups'. The main content area shows a pipeline run for 'Enabling feature flags for Preview Attachment and...' with a status of 'Running'. Below this, a table lists the jobs: 'Windows Job' (Running, 1m 53s), 'Linux Job' (Running, 3m 29s), and 'macOS Job' (Running, 3m 07s). The 'Linux Job' is selected, and its details are shown on the right. The 'Linux Job' is running on a 'Hosted Linux' agent. The job steps are: 'Prepare job', 'Initialize job', 'Get sources', 'Cmdline', 'Nodetool', and 'Install dependencies'. The 'Install dependencies' step is currently running, and its logs are displayed at the bottom of the screen. The logs show the execution of 'yarn install v1.7.0' and 'node build/npm/preinstall.js', followed by 'npm run compile' and 'tsc -p tsconfig.build.json'. The logs indicate that the build is successful and completed in 4.89s.

建立订制化任务群组

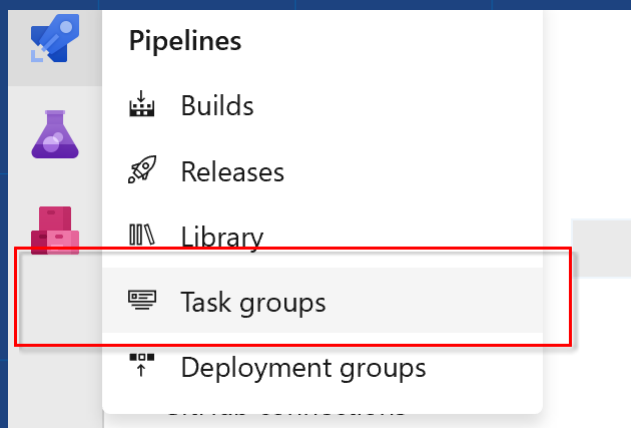


为什么要用订制化任务群组

- 重复任务流程设定时间
- 管线流程标准化
- 易于管理

订制化任务群组

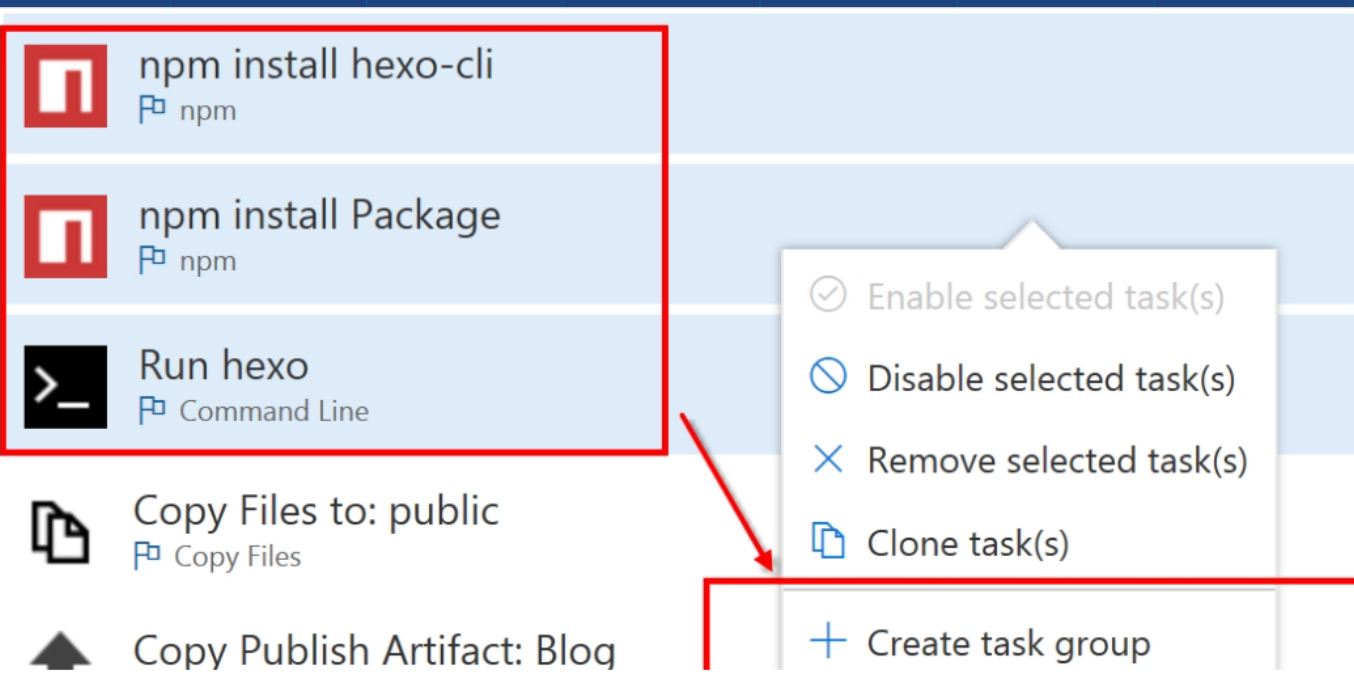
- 建立订制化任务方式
 - 直接到Task Groups建立








- 从管线中选取多个项目建立群组

订制化任务群组

圈选项目

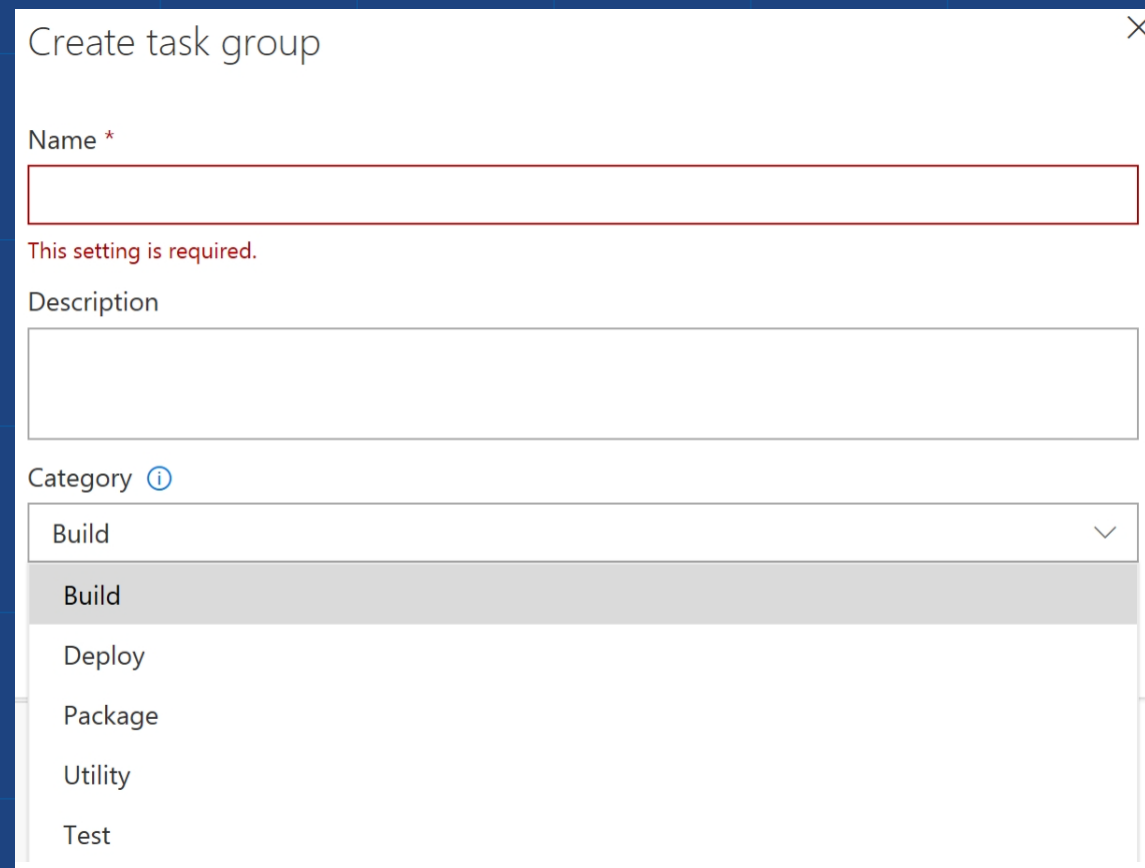


A screenshot of a task list interface. A red rectangular box highlights the first three tasks: 'npm install hexo-cli' (npm icon), 'npm install Package' (npm icon), and 'Run hexo' (Command Line icon). A context menu is open over the 'Run hexo' task, showing options: 'Enable selected task(s)', 'Disable selected task(s)', 'Remove selected task(s)', 'Clone task(s)', and 'Create task group'. A red arrow points from the 'Create task group' option in the context menu to the 'Create task group' option at the bottom of the task list.

-  npm install hexo-cli
npm
-  npm install Package
npm
-  Run hexo
Command Line
-  Copy Files to: public
Copy Files
-  Copy Publish Artifact: Blog

- ☒ Enable selected task(s)
- ☐ Disable selected task(s)
- ☐ Remove selected task(s)
- ☐ Clone task(s)
- ☐ Create task group

建立



A screenshot of the 'Create task group' dialog box. It contains a 'Name' field with a red border and a red asterisk, a 'Description' field, and a 'Category' dropdown menu. The 'Category' dropdown is open, showing a list of categories: 'Build', 'Build', 'Deploy', 'Package', 'Utility', and 'Test'. The 'Build' category is selected and highlighted.

Create task group

Name *

This setting is required.

Description

Category ⓘ

- Build
- Build
- Deploy
- Package
- Utility
- Test

任务群组版本控制

- 任何项目都需要版本管理
- 避免多个管线使用同一个任务群组，随意更动任务群组，影响到管线
- 步骤
 - 修改任务组内容
 - 選擇另存为草稿
 - 发布草稿节选预览
 - 出版预览

Demo

建立多版本任务群组

善用变量群组

```
Program
Console.WriteLine(" ON YOUR MARK,\n");
Console.WriteLine();

// Run at least one time then statement loops the race until either variables tortoise or hare (or both) equal to variable finishline
do
{
    tortoisemethod(); // calls tortoisemethod to determine variable tortoise's value after each loop
    haresmethod(); // calls haresmethod to determine variable hare's value after each loop
    showrace(); // prints the race

    // Slow down race
    // This is a brute force delay in the code
    // allows users to view the race at a readable speed
    for (int slowdown = 0; slowdown < 100000000; slowdown++);

} while (tortoise < finishline && hare < finishline);

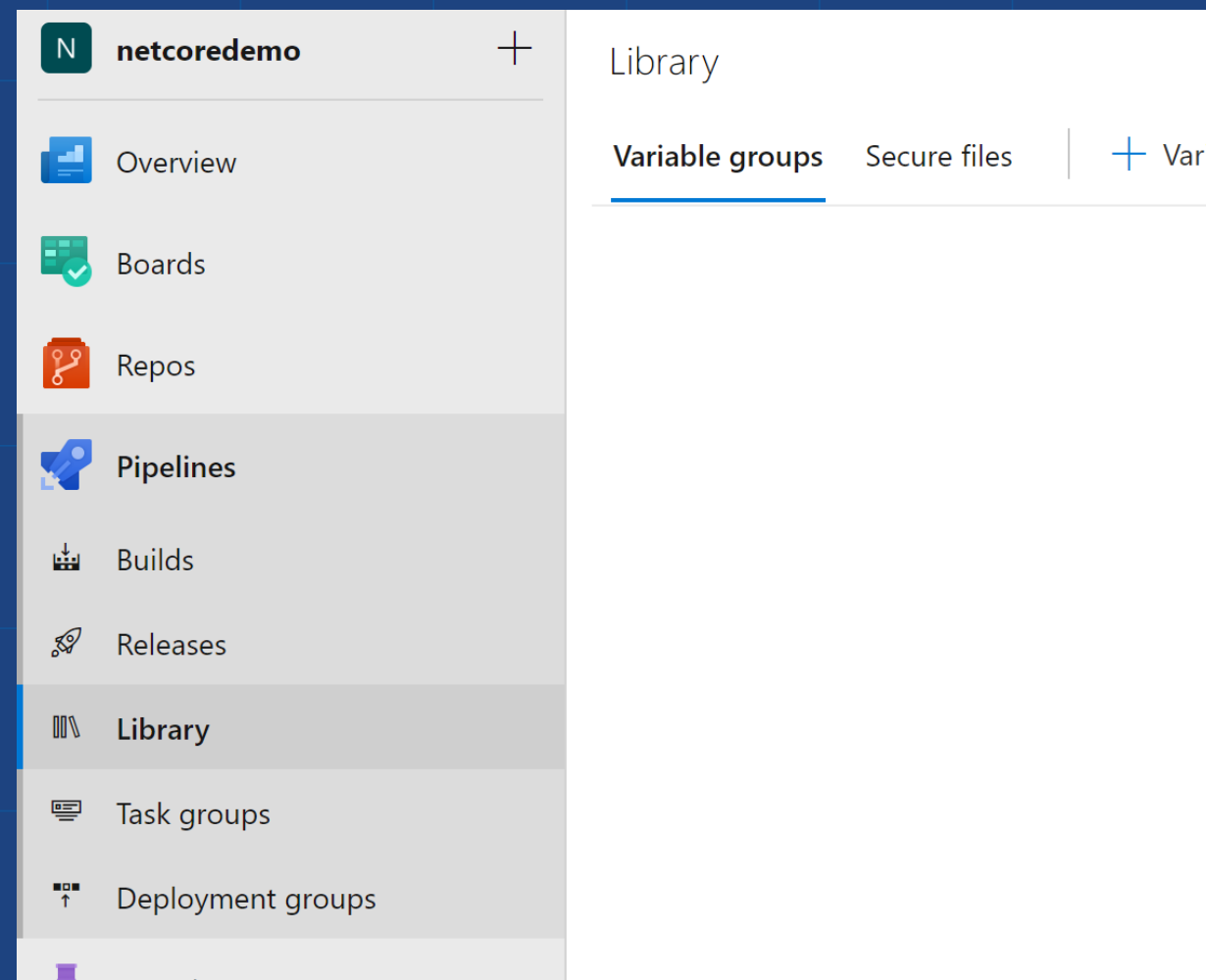
// if and if else statements on lines 37-50 print the conclusion of the race
if (tortoise == hare)
{
    Console.WriteLine("TIE!!!");
}
else if (tortoise < hare)
{
    Console.WriteLine("TORTOISE WINS!!! YAY!!!");
}
else if (hare > tortoise)
{
    Console.WriteLine("HARE WINS!!! YAY!!!");
}
Console.WriteLine();
```

```
ON YOUR MARK,
GET SET,
BANG !!!!!
AND THEY'RE OFF !!!!!

T H T T H H T H T
T H T H T H T H T
TORTOISE WINS!!! YAY!!!
OUCH!!!
Press any key to continue . . .
```

变量群组优势

- 保有机密性的变量
- 多个管线共享相同变量
- 高效管理系统设定变量
- 订制化群组管理
 - 读取
 - 使用
 - 管理员








变量群组优势

群组管理

属性管理

变量管理

Library >  CICD.Build.Parameter*


Variable group |  Save  Clone  Security  Help

Properties

Variable group name

Description

☒ Allow access to all pipelines

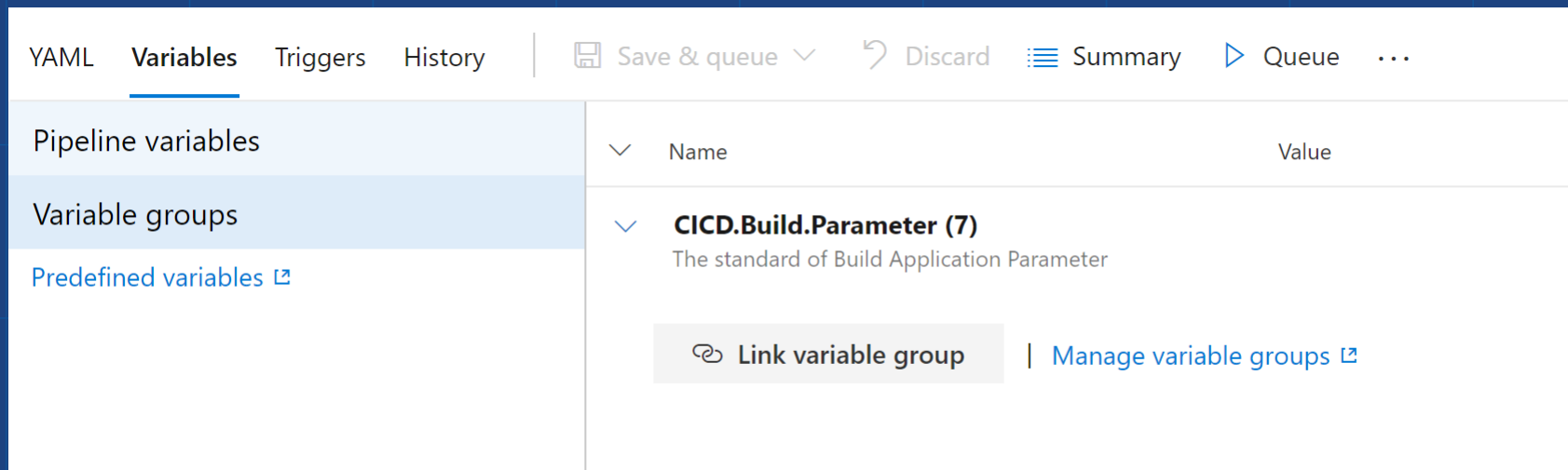
☒ Link secrets from an Azure key vault as variables 

Variables

Name ↑	Value
Build.Platform	any cpu
DEV.Configuration	debug
Docker.Configuration	docker
PRO.Configuration	release

如何使用变量

- 管线中加入要使用的变量群组



- 管线中加入变量名称 \$(变量名称)

```
... #arguments: '/p:OutputPath=$(Build.artifactstagingdirectory)\app'
... arguments: '--self-contained true -o $(Build.artifactstagingdirectory)\app -r win-x64'
... zipAfterPublish: false
... feedsToUse: config
... nugetConfigPath: nuget.config
... configuration: $(DEV.Configuration)
```

在CI注入变量

- 持续集成中，放入变量，作为软件变数
 - 例如: Web.config 数据库联机字符串讯息

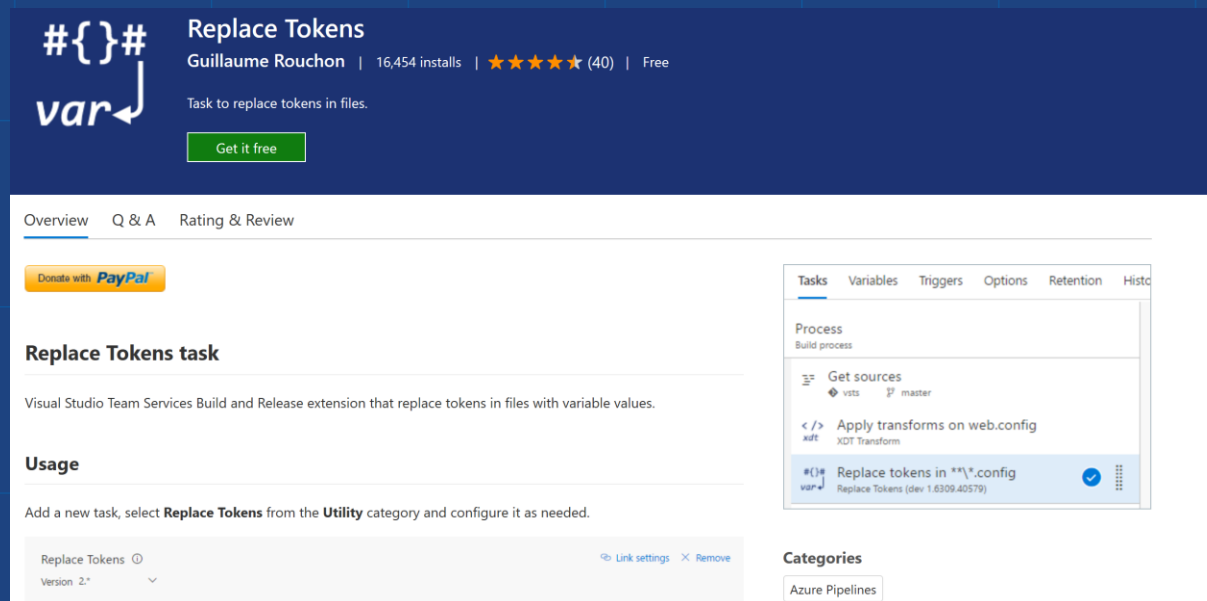
```
<?xml version="1.0" encoding="utf-8"?>

<!-- For more information on using web.config transformation visit http://go.microsoft.com/fwlink/?LinkId=125889 -->

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
  <connectionStrings xdt:Transform="Replace">
    <add name="AIAOIQAS" connectionString="#{QAS_AOI_sa}#" />
  </connectionStrings>
  <appSettings xdt:Transform="Replace">
    <add key="MailServiceAddress" value="#{QAS_MailServiceAddress}#" />
    <add key="ApplicationInsightskey" value="#{QAS_ApplicationInsightsKey}#" />
  </appSettings>
</configuration>
```

在CI注入变量

- 使用Replace Tokens task



- 使用 #{变量名稱}# 格式

- `<add key="MailServiceAddress" value="#{ QAS_MailServiceAddress }#" />`

在CI注入变量

放入需要被注入的档案，
任何档案类型都可以

找不到变量名称，则集
成失败

The screenshot shows a configuration window for injecting variables into a file. The interface is dark-themed with white text. Two sections are highlighted with red rectangles:

- Target files ***: A text input field containing "Web.QAS.config".
- Action ***: A dropdown menu set to "fail".

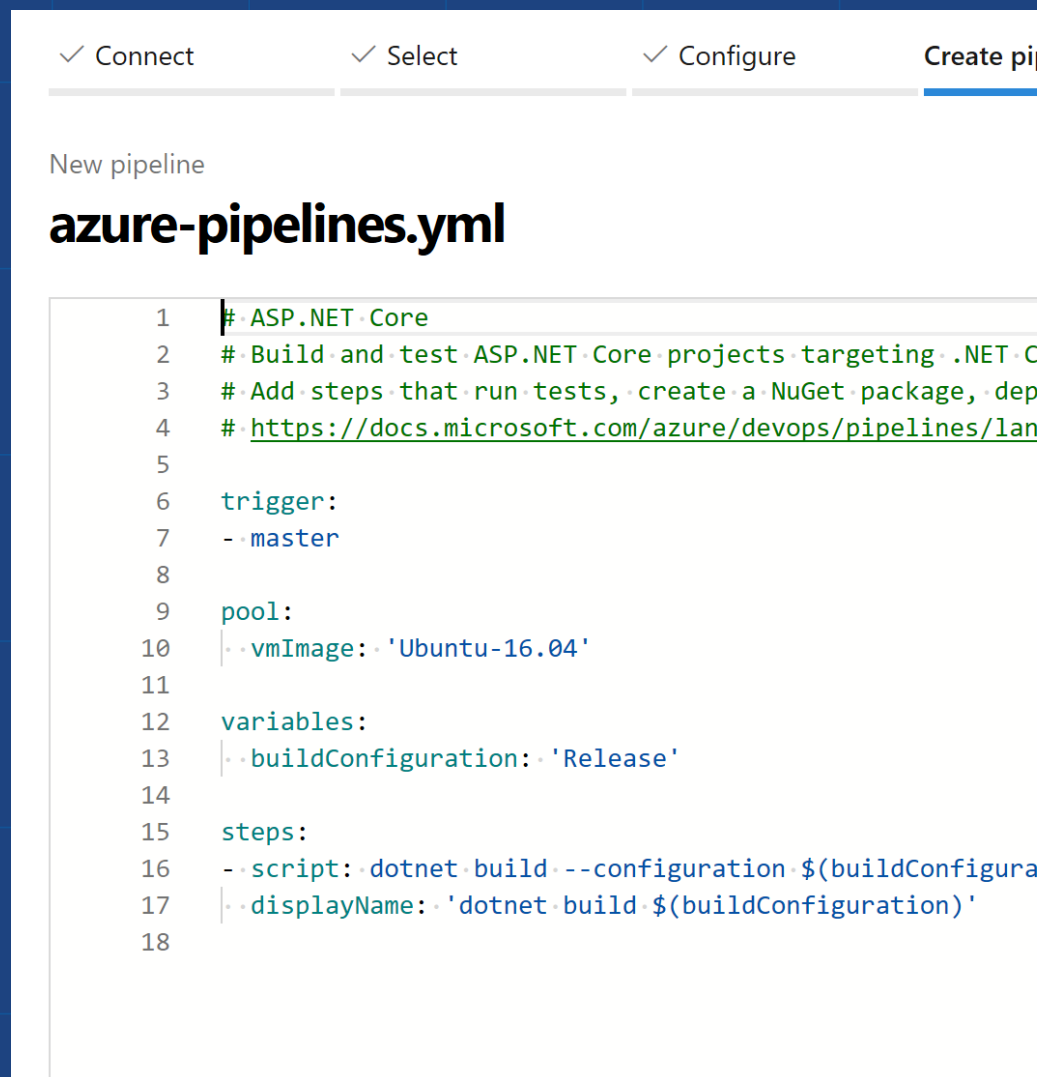
Other visible fields include:

- Display name ***: "Replace tokens in Web.QAS.config"
- Root directory**: An empty text input field with a browse button (three dots).
- Files encoding ***: A dropdown menu set to "auto".
- Write unicode BOM**: A checked checkbox.
- Missing variables**: A section header with an expand/collapse arrow.
- Keep token**: An unchecked checkbox.
- Advanced** and **Control Options**: Section headers with expand/collapse arrows.

管线即代码

管线即代码

- 透过YAML格式，建立持续集成管线
- 管线流程可以纳入版控
- 代码移转容易
- 更符合软件开发流程
- 更高效、更弹性
- Release管线不支援



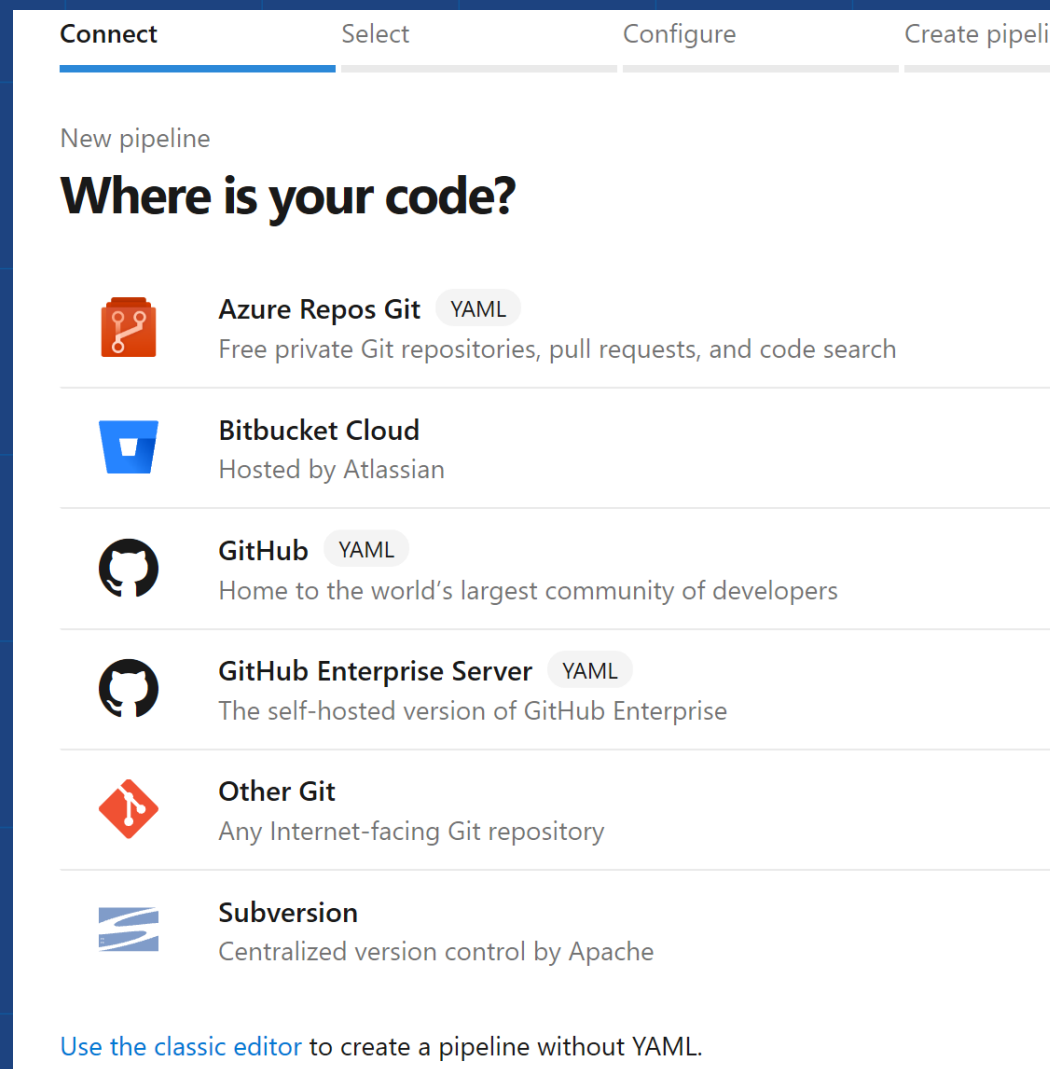
建置方式

- 线上编辑

- 在Azure DevOps直接产生管线代码
- 管线代码档案会直接并入版控

- 线下编辑

- 线下用Visual Studio撰写
- 在Azure DevOps选择管线代码档案



建置步骤

- 连结版本控制

- Azure DevOps Git、Bitbucket Cloud、Github...

- 选择代码项目

- 设定模板

- 撰写管线YAML代码

New pipeline

Configure your pipeline



ASP.NET Core recommended

Build and test ASP.NET Core projects targeting .NET Core.



ASP.NET

Build and test ASP.NET projects.



ASP.NET Core (.NET Framework)

Build and test ASP.NET Core projects targeting the full .NET Framework.



Universal Windows Platform

Build a Universal Windows Platform project using Visual Studio.



Xamarin.Android

Build a Xamarin.Android project.



Xamarin.iOS

Build a Xamarin.iOS project.



.NET Desktop

Build and run tests for .NET Desktop or Windows classic desktop solutions.



Starter pipeline

Start with a minimal pipeline that you can customize to build and deploy your code.



Existing Azure Pipelines YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

YAML代码结构

- Pipeline Job 1
 - Step 1.1
 - Step 1.2
 - ...
- Job 2
 - Step 2.1
 - Step 2.2
 - ...
- ...



- 区块一
 - Pool
 - 要启用VM类型
- 区块二
 - 设置
 - 设定Task流程
- 其他
 - 细节属性

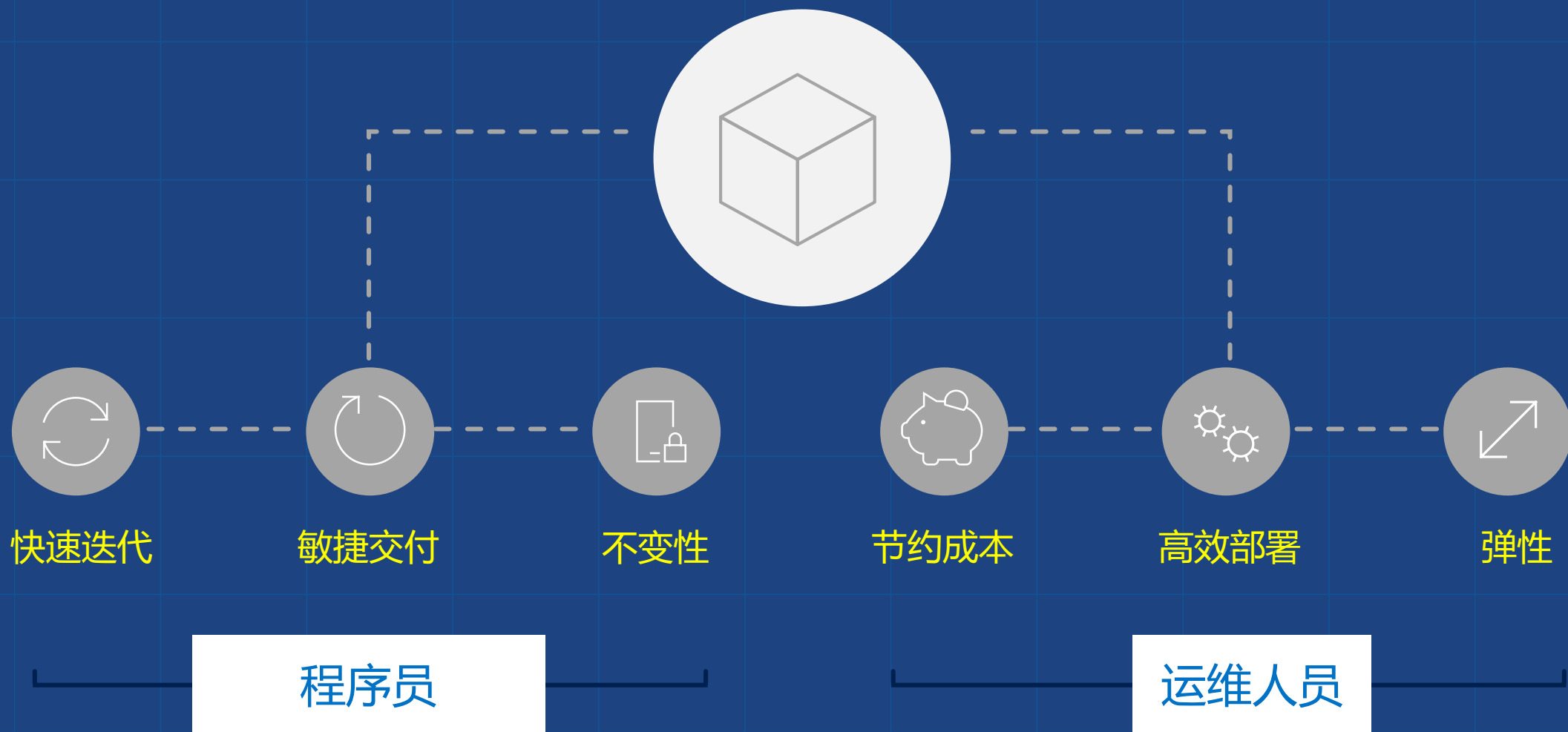
Demo

建立YAML

容器解决方案



DevOps对容器的优势



Azure Container Registry

将Docker私有注册表作为一流的Azure资源进行管理



管理所有类型容器



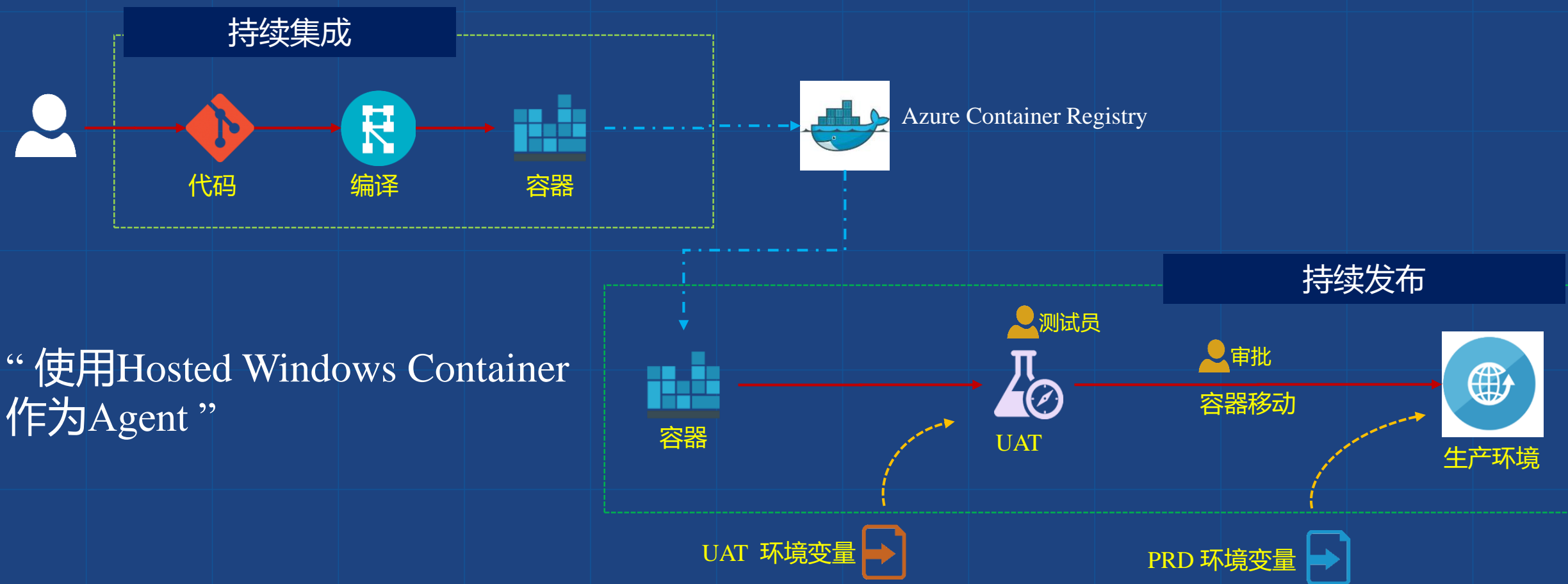
使用熟悉的开源Docker
CLI工具



Azure容器注册表跨区域
复制



打造容器CI / CD 管线



导入容器化技术

- 减少程序部署造成的服务停止时间
- 确保测试环境与上线环境设定是标准化
- 降低系统内部的耦合程度
- 快速发布新功能

扩充移动端

Visual Studio App Center



Build

每次推送到Repos.，可以自动编译可安装的App，支持Azure Devops、GitHub或是Git Repos.



Test

在400多种设备上配置运行测试，可以用Xamarin.UITest，Appium，Espresso（Android），替iOS、Android撰写测试



Distribute

用户可以通过电子邮件分发的列表安装应用程序测试，就像他们从应用程序商店下载一样。



Diagnostics

从所有设备收集崩溃数据，根据看到崩溃的用户数量确定优先级，并获取完整的数据与跟踪



Analytics

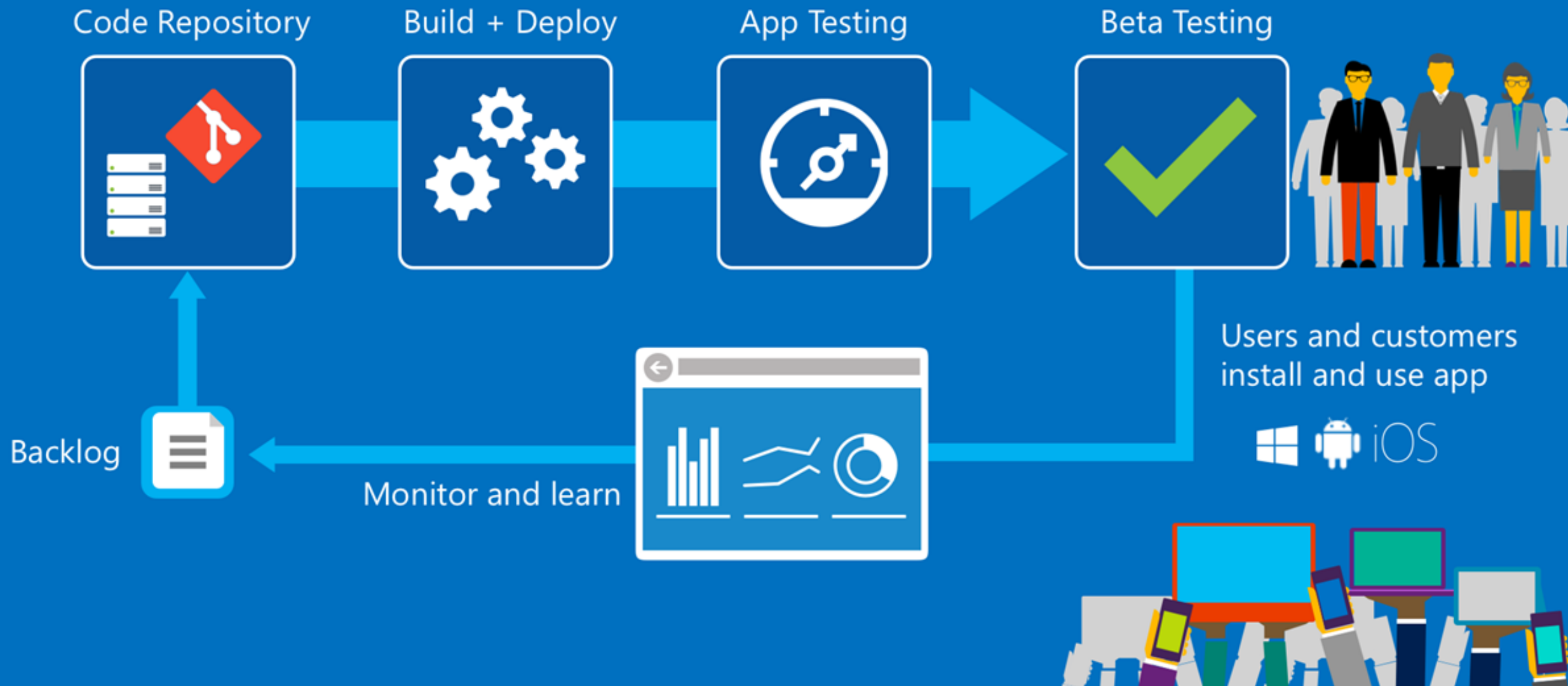
获取有关每日，每周和每月用户数，持续时间，设备等的信息，收集丰富的行为分析数据。



Push Notifications

向特定用户组发送有针对性的消息。

App持续集成与发布



总结

桌面软件的Azure DevOps

- DevOps原则不仅仅适用于Web应用程序、容器和微服务架构。共同关注的领域是通过将CI&CD放到桌面程序，提高客户的价值流。
- 安装技术
 - ClickOnce
 - Squirrel
 - Microsoft Store

Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines



Azure Test Plans



Azure Artifacts



通过一套现代开发服务，更智能地规划、更好地协作和更快地交付



任何开发者，任何平台，任何云。全面支持混合云、内部部署和容器



使用所有的Azure DevOps服务，或者只选择您需要补充现有工作流的内容。



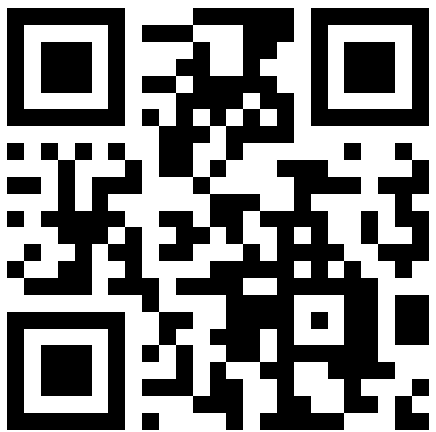
一流的开源软件。在Windows、Linux和MacOS上为公共项目提供免费的无限制的构建时间和最多10个免费的并发作业



小团队免费开始，可扩展以支持最大的企业

End

更多的细节，
请参阅我的博客



欢迎与我联系
我的微信



特别感谢

