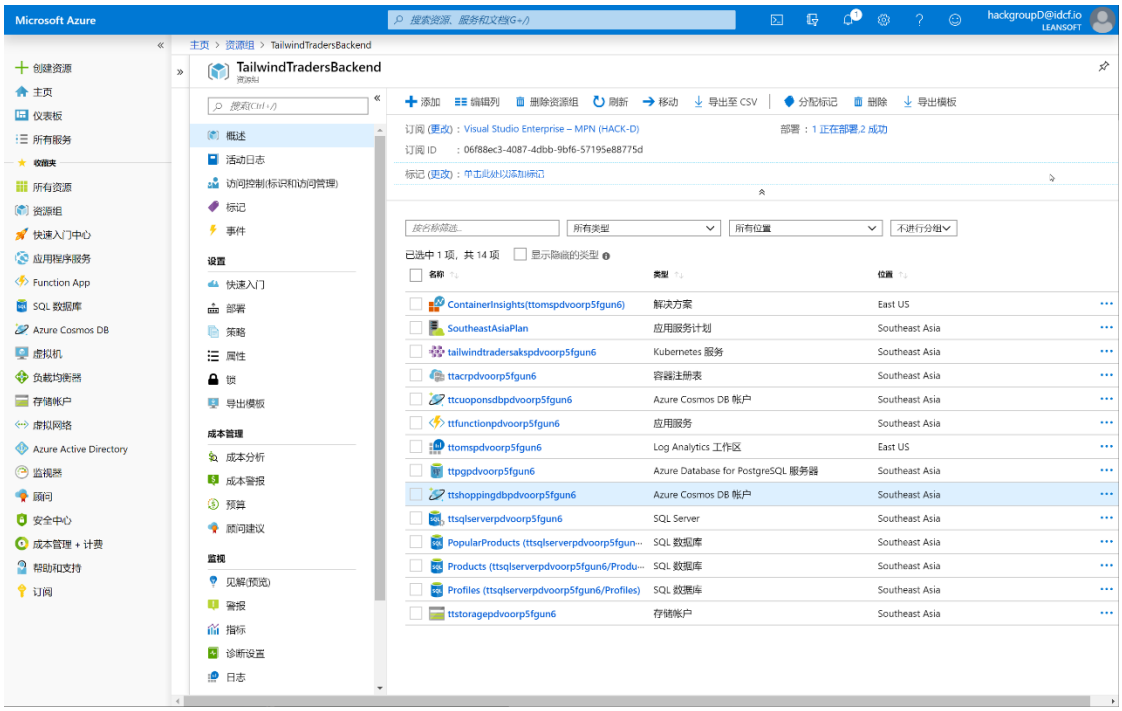
IDCF 黑客马拉松示例项目 TailwindTraders 后台应用

TailwindTraders 后台应用使用了以下应用组件提供一套微服务架构的应用模型，运行在微软云 Azure 平台之上。



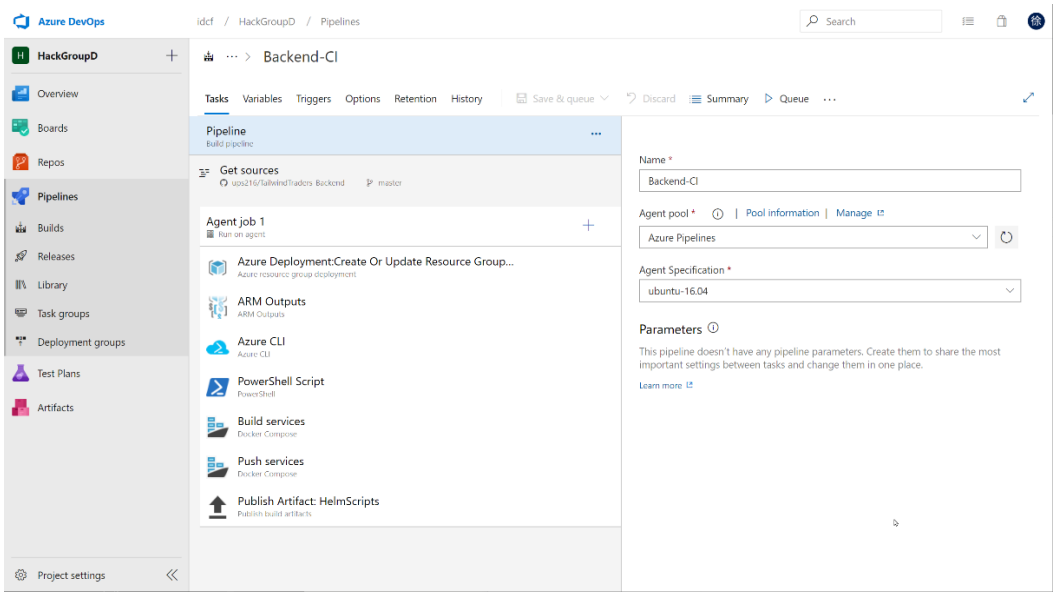
组件列表/项目结构

- 以下应用组件采用微服务应用架构，运行于 Azure Kubernetes Service
  - /Source/ApiGWs/
    - Tailwind.Traders.WebBff：web 端应用网关，提供网站对于后端服务环境的访问入口
    - Tailwind.Traders.Bff：手机端应用网关，提供 iOS/Android 应用对于后端应用的访问入口
  - /Source/Services/
    - TailwindTraders.Cart.Api: 购物车应用服务
    - TailwindTraders.Congnitive.Docker：AI 认知服务，提供针对图片的人工智能识别服务，用于根据用户上传图片自动搜索商品
    - TailwindTraders.Coupon.Api: 代金券服务
    - TailwindTraders.ImageClassifier.Api: 图片识别服务，通过调用 AI 认知服务完成图片识别
    - TailwindTraders.Login.Api: 用户登录认证服务
    - TailwindTraders.PopularProduct.Api: 流行商品服务
    - TailwindTraders.Product.Api: 商品品类服务
    - TailwindTraders.Profile.Api: 用户个人资料服务
    - TailwindTraders.Stock.Api: 商品库存服务
- 以下应用组件使用 Azure 云 PaaS 服务提供

    - 无服务计算(Serverless) Azure Funciton
        - /Source/Services/TailwindTraders.Visits.Function: 用户访问统计服务
    - 托管的 PostgreSQL 数据库服务
        - stockdb: TailwindTraders.Cart.Api 的后台数据库
    - 托管的 NoSQL 数据库服务 CosmosDB
        - MongoDB 兼容模式：TailwindTraders.Coupon.Api: 代金券服务数据库
        - CosmoDB 原生模式：TailwindTraders.Cart.Api: 购物车应用服务数据库
    - 托管的 SqlServer 数据库服务
        - TailwindTraders.PopularProduct.Api: 流行商品服务数据库
        - TailwindTraders.Product.Api: 商品品类服务数据库
        - TailwindTraders.Profile.Api: 用户个人资料服务书库
    - 对象存储服务 Azure Storage
        - 提供商品图片的高性能持久化存储以及 CDN 服务
- 应用使用性能分析(APM)，监控以及日志服务使用 Azure 云 PaaS 服务提供
    - 容器监控解决方案服务 Container Insight
    - 日志及分析服务 Log Analytics 工作区
- Docker 镜像仓库服务使用 Azure 云 PaaS 服务提供
    - 容器注册表服务

Azure DevOps 自动化构建流水线

Azure DevOps 中的构建流水线已经搭建完成，可以通过本组账号访问。



本项目的部署仍然需要手工完成，以下为部署脚本，请自行安装以下工具以便正常完成部署

- Azure CLI
- PowerShell

- Helm
- Visual Studio Code
- Git
- SSMS - 可用于更新后端应用的 Sql Server 数据库
- pgAdmin - 用于更新 PostgreSQL 中的数据
- Azure Storage Explorer - 用于更新图片存储

以下为部署步骤

Step 1 - 使用 Azure CLI 连接到 Azure 订阅

```
## 使用 Azured 订阅管理员账号登录
az login

## 设置当前订阅为默认订阅
az account set -s {subscription id}}

## 创建 Service Principle 账号
az ad sp create-for-rbac

Retrying role assignment creation: 1/36
{
  "appId": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX",
  "displayName": "azure-cli-2019-09-05-13-17-11",
  "name": "http://azure-cli-2019-09-05-13-17-11",
  "password": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX",
  "tenant": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX"
}
```

记录以上输出信息为 clientid/servicePrincipleID = appId ServicePrincipleKey = password

以下为建议的数据统一用户名密码，也可以自行指定 sqlServerAdministratorLogin = sqladmin sqlServerAdministratorLoginPassword = P2ssw0rd@123

Step 2 - 设置本地 Kubectl 环境，获取访问密钥

```
## 获取 k8s 访问密钥
az aks get-credentials -g TailwindTradersBackend -n {aks-name}
## 测试是否可以访问 k8s
kubectl get nodes

## 启用 K8s 仪表盘
kubectl create clusterrolebinding kubernetes-dashboard --clusterrole=cluster-admin --serviceaccount=kube-system:kubernetes-
dashboard
## 测试 k8s 仪表盘工作正常
az aks browse -g TailwindTradersBackend -n {aks-name}
```

Step 3 - 安装 Tiller 和 Helm

```
## 创建系统服务账号 Tiller
kubectl create serviceaccount --namespace kube-system tiller
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
## 初始化 helm 使用 tiller 服务账号
helm init --service-account tiller

## 检查 Kube-system 命名空间中已经成功部署了 tiller 服务端
kubectl get pods -n kube-system
```

Step 4 - 获取容器镜像仓库密钥并设置到 k8s 密钥仓库

```
## 获取 ACR 服务密钥
az acr credential show -n {acr-name}} -g TailwindTradersBackend --output table

## 创建 k8s 密钥仓库对象
kubectl create secret docker-registry acr-auth --docker-server {acr-name}.azurecr.io --docker-username {acr-name} --docker-
password {acr-password} --docker-email not@used.com
```

Step 5 - 在 K8s 集群上启用 https/ssl 加密访问

```
## 安装 cert-manager 服务
helm install --name cert-manager --namespace kube-system  --version v0.4.1 stable/cert-manager

## 启动服务，并绑定证书
powershell .\Enable-Ssl.ps1 -sslSupport prod -aksName {aks-name} -resourceGroup TailwindTradersBackend
```

Step 6 - 部署应用服务容器镜像到 k8s

```
## 创建应用所使用的系统服务账号
kubectl create serviceaccount ttsa

## 使用从 Backend-CI 中下载的制品包完成部署，假设制品包被解压到 D:\HelmScripts\目录
powershell ./Deploy-Images-Aks.ps1 -name "my-tt" -resourceGroup TailwindTradersBackend -aksName {aks-name} -acrName {acr-name} -
tag prod -valuesFile "D:\HelmScripts\gvalue.yml" -tlsEnv prod

## 可选：此命名将删除所有部署，如果需要从新部署可以使用此命令清楚所有服务
FOR /f "tokens=*" %i IN ('helm list --short') DO helm del --purge %i

## 检查所有服务均已启动
kubectl get pods
kubectl get pod -o=custom-columns=NAME:.metadata.name,STATUS:.status.phase,NODE:.spec.nodeName --all-namespaces
```

Step7 - 上传商品图片到 Azure Storage

```
## 所有商品图片将会打包在制品包中，假设制品包被解压到 D:\HelmScripts\目录
powershell .\Deploy-Pictures-Azure.ps1 -resourceGroup TailwindTradersBackend -storageName {storage-account-name}} -imageRootFolder
"D:\HelmScripts"
```

更新版本方式

如果需要更新应用，可以使用以下方式进行。因为所有应用数据均已经通过云平台 PaaS 服务进行持久化，所以 k8s 中的任何服务均可以随时安全的删除和更新。

全量从新部署

执行步骤：

- 使用 Step 6 中的命令清除所有应用
- 从新执行 Step 5, 6

部分更新

执行步骤：

- 直接执行 Step 6 中的部署命令即可

Happy Coding ...