# Dify on Azure Workshop

## Overview

这个实验展示了如何在Azure上部署Dify，使用了Azure 提供的多种的服务，包括Azure AI Search, Azure Database for PostgreSQL, Azure Cache for Redis, Azure Kubernetes Service等。

## Dify on Azure部署指南

### **Windows环境准备**

1. 已安装Terraform，若没有安装的话，请参考Windows上安装步骤Terraform进行安装。
2. 已安装Azure CLI，若没有安装的话，请参考Windows上安装Azure CLI指南进行安装。
3. 准备docker hub账号， 下载Dify镜像的时候需要用到。

**# Windows上安装步骤Terraform（如已安装可以略过该步骤）**

1. **下载Terraform**
2. 访问下载页面 https://www.terraform.io/downloads.html。
3. 在页面中找到适用于Windows的版本，点击下载链接。
4. **解压文件**
5. 下载后，找到下载的ZIP文件，如：`terraform\_1.0.0\_windows\_amd64.zip`。
6. 右键点击ZIP文件，选择“解压到当前文件夹”或使用您喜欢的解压工具解压文件。
7. **将Terraform添加到系统路径**
8. 将解压后的`terraform.exe`文件移动到您希望存放的目录，例如：`C:\terraform`。
9. 打开“控制面板”，选择“系统和安全”，然后选择“系统”，点击左侧的“高级系统设置”。在“系统属性”窗口中，点击“环境变量”按钮。
10. 在“环境变量”窗口中，找到“系统变量”部分，选择`Path`变量，然后点击“编辑”。
11. 在“编辑环境变量”窗口中，点击“新建”，然后输入`C:\terraform`（即`terraform.exe`所在的目录）。点击“确定”保存更改。
12. **验证安装**
13. 打开命令提示符（按`Win + R`，输入`cmd`，然后按回车）。
14. 在命令提示符中输入命令“terraform -v ”并按回车， 如果安装成功，您将看到Terraform的版本信息输出。

**## Windows上安装Azure CLI指南（如已安装可以略过该步骤）**

1. **下载Azure CLI安装程序**
   1. 打开浏览器，访问下载页面 (https://docs.microsoft.com/cli/azure/install-azure-cli-windows?tabs=azure-cli)。
   2. 在页面中找到适用于Windows的安装程序，点击下载链接。
2. **运行安装程序**
   1. 下载完成后，找到下载的安装程序文件（例如：`AzureCLI.msi`）。
   2. 双击安装。
3. **验证安装**
   1. 安装完成后，打开命令提示符或PowerShell。
   2. 输入命令“az --version”以验证Azure CLI是否安装成功，如果安装成功，您将看到Azure CLI的版本信息。
4. **登录到Azure**
5. 在命令提示符或PowerShell中，输入命令“az login”登录Azure账户。
6. 按照提示在浏览器中完成登录过程。

现在，您已经成功在Windows上安装了Azure CLI，并可以开始使用它来管理您的Azure资源。

### **Ubuntu环境准备**

1. 已安装Terraform，若没有安装的话，请参考Ubuntu上安装步骤Terraform进行安装。
2. 已安装Azure CLI，若没有安装的话，请参考Ubuntu上安装Azure CLI指南进行安装。
3. 准备docker hub账号， 下载Dify镜像的时候需要用到。

**# Ubuntu上安装步骤Terraform（如已安装可以略过该步骤）**

**参考链接：** [**Install Terraform | Terraform | HashiCorp Developer**](https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli)

**# Ubuntu上安装Azure CLI指南（如已安装可以略过该步骤）**

**参考链接：**[**Install the Azure CLI on Linux | Microsoft Learn**](https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-linux?pivots=apt)  
  
**部署前参数准备**

**在配置 Dify 之前，请检查并设置 dev-variables.tfvars 文件中的变量。**

在 dev-variables.tfvars 文件中，定义了6个变量：

1. sp-subscription-id Azure订阅ID
2. sp-client-id：这是服务主体的客户端 ID。它是一个唯一标识符，用于标识特定的服务主体。
3. sp-client-secret：这是服务主体的客户端密钥。它类似于密码，用于验证服务主体的身份。这个值应当保密，不应公开。
4. sp-tenant-id：这是租户 ID。它标识了 Azure Active Directory (AAD) 中的特定租户，通常用于多租户环境中区分不同的组织或用户组。
5. name：Dify平台的名字，用于创建或定义多个服务的名字（比如Azure资源组，Azure Private DNS zone等），需要全球唯一。
6. filename ：指定您的本地电脑中存储AKS凭据文件kubeconfig的路径
7. acr\_name: 根据具体情况修改名字， 需全球唯一
8. docker\_username: 在docker.io 上注册， 填入注册的user name
9. docker\_secret：在docker.io 上注册， 填入密码

**如何获得**sp-client-id **和**sp-client-secret

1. **执行下列命令，创建服务主体（service principal），根据您的azure环境修改命令中的参数**

az login

az ad sp create-for-rbac --name <your-service-principal-name> --role Owner --scopes /subscriptions/<your-subscription-id>

# 如果已经创建，查询之前创建的service-principal-name： 进入Azure portal： Microsoft Entra ID -> Manage -> User -> User Principal name

1. **执行上述命令后，你将获得一个 JSON 响应，其中包含 appId（即 sp-client-id）、password（即 sp-client-secret）和 tenant（即 sp-tenant-id）。**

示例输出：

{

  "appId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

  "displayName": "<your-service-principal-name>",

  "name": "http://<your-service-principal-name>",

  "password": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

}

1. **将这些值填入你的 dev-variables.tfvars 文件中：（appId即sp-client-id，password即sp-client-secret，tenant即sp-tenant-id）**

sp-client-id = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

sp-client-secret = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

sp-tenant-id = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

### 使用terraform将dify部署到Azure

1. 运行“terraform init”初始化
2. 通过运行以下terraform plan为开发环境生成计划。

terraform plan -out=dev-plan -var-file="./environments/dev-variables.tfvars"

1. 通过运行以下“terraform apply命令应用生成的计划。

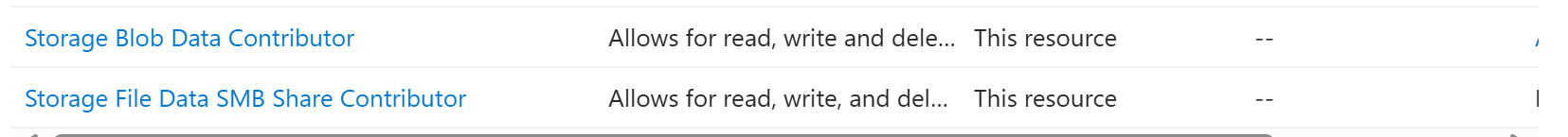
terraform apply "dev-plan"

现在，您已经成功在Azure上部署了Dify，接下来登陆Azure门户，从刚刚创建的资源组中找到Azure Kubernetes Service集群，并打开它。按图示找到您在azure上部署的Dify的公网地址。点击ingress的IP 地址，将会在浏览器中显示Dify。 Dify on Azure上的部署就完成了！A screenshot of a computer

AI-generated content may be incorrect.

Debugging

* 1. Dify-API, Dify-Worker 启动不成功，显示mount file permission deny。

解决方法： 给AKS对应的file 以下permission。

* 1. 无法访问，确认api gateway 的subnet里面的security group打开了80端口。