

PyCon China 2024

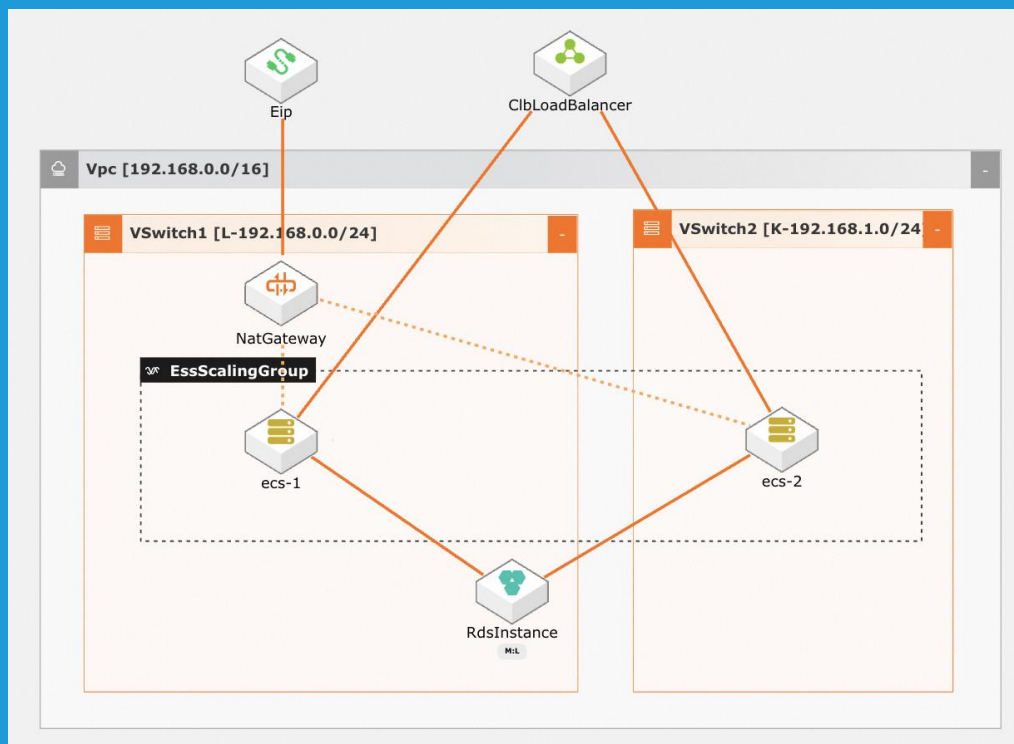
For Good . For fun.
2024/11/23 中国 上海

基于大模型自动生成IaC模板

阿里云弹性计算——阮正鑫



云上资源的部署



传统部署方案

控制台

OpenAPI

SDK



自动化部署
(IaC)

Infrastructure as code (IaC): 通过代码或者配置文件来定义和管理基础设施

IaC的优势

- 自动化
- 可重复
- 部署的标准和统一
- 版本控制和CICD集成

IaC 工具

云厂商一方IaC

```
VPC:
  Type: ALIYUN::ECS::VPC
  Properties:
    VpcName: myvpc
    CidrBlock: 192.168.0.0/16
VSwitch:
  Type: ALIYUN::ECS::VSwitch
  Properties:
    VpcId:
      Ref: VPC
    ZoneId:
      Ref: ZoneId
    CidrBlock: 192.168.0.0/24
SecurityGroup:
  Type: ALIYUN::ECS::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    SecurityGroupName: mysg
    SecurityGroupType: normal
```

Aliyun ROS

```
Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  MySubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: MyVPC
      CidrBlock: '10.0.1.0/24'
  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access
      VpcId:
        Ref: MyVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: '0.0.0.0/0'
```

AWS CloudFormation

三方IaC

```
resource "alicloud_vpc" "main" {
  cidr_block = "192.168.0.0/16"
}
resource "alicloud_vswitch" "main" {
  vpc_id      = alicloud_vpc.main.id
  cidr_block  = "192.168.1.0/24"
  availability_zone = "cn-hangzhou-e"
}
resource "alicloud_security_group" "main" {
  vpc_id = alicloud_vpc.main.id
}
resource "alicloud_security_group_rule" "allow_http_inbound" {
  type            = "ingress"
  ip_protocol     = "tcp"
  security_group_id = alicloud_security_group.main.id
  port_range      = "80/80"
  cidr_ip         = "0.0.0.0/0"
}
resource "alicloud_security_group_rule" "allow_https_inbound" {
  type            = "ingress"
  ip_protocol     = "tcp"
  security_group_id = alicloud_security_group.main.id
  port_range      = "443/443"
  cidr_ip         = "0.0.0.0/0"
}
```

Terraform

```
const vpc = new alicloud.vpc.Network("my-vpc", {
  cidrBlock: "192.168.0.0/16",
  name: "pulumi-vpc",
});
const vswitch = new alicloud.vpc.Switch("my-vswitch", {
  cidrBlock: "192.168.1.0/24",
  vpcId: vpc.id,
  zoneId: "cn-hangzhou-e",
});
const securityGroup = new alicloud.ecs.SecurityGroup("my-security-group", {
  vpcId: vpc.id,
});
const allowHttp = new alicloud.ecs.SecurityGroupRule("allow-http", {
  type: "ingress",
  ipProtocol: "tcp",
  securityGroupId: securityGroup.id,
  portRange: "80/80",
  cidrIp: "0.0.0.0/0",
});
const allowHttps = new alicloud.ecs.SecurityGroupRule("allow-https", {
  type: "ingress",
  ipProtocol: "tcp",
  securityGroupId: securityGroup.id,
  portRange: "443/443",
  cidrIp: "0.0.0.0/0",
});
```

Pulumi

IaC模板编写难点

开发者会面临的问题是编写和管理基础设施模板（如 阿里云ROS、Terraform 等）可能涉及许多细节和配置选项，导致学习成本高

模板结构说明

参与数： 奇理、七格 | 更新时间： 2024-09-04 16:00:15

产品详情 相关技术圈

模板是一个JSON或YAML格式的文本文件，使用UTF-8编码。模板用于创建资源栈，是描述基础设施和架构的蓝图。模板编辑者在模板中定义云资源资源和配置细节，并说明资源间的依赖关系。

ROS模板结构

ROSTemplateFormatVersion: "2015-09-01"

Description: 模板描述信息，可用于说明模板的适用场景、架构说明等。

Metadata:

Parameters:

Mappings:

Conditions:

Resources:

Outputs:

Rules:

← 学习IaC模板语法

查阅资源定义 →

资源类型索引

参与数： Chandler、阿凡 等 2 人 | 更新时间： 2024-10-22 14:53:58

本文为您提供资源类型索引，便于您进行查询。

阿里云产品	资源资源
应用配置管理ACM	<ul style="list-style-type: none">ALIYUN::ACM::Configuration: 创建配置ALIYUN::ACM::Namespace: 创建命名空间
容器计算服务 ACS (Container Compute Service)	<ul style="list-style-type: none">ALIYUN::ACS::Cluster: 创建容器计算服务集群
操作审计 (ActionTrail)	<ul style="list-style-type: none">ALIYUN::ACTIONTRAIL::Trail: 创建跟踪，将事件投递到日志服务SLS或对象存储OSSALIYUN::ACTIONTRAIL::TrailLogging: 启用或关闭跟踪的日志记录
云原生数据仓库 AnalyticDB MySQL 版 (AnalyticDB for MySQL)	<ul style="list-style-type: none">ALIYUN::ADB::DBCluster: 创建一个AnalyticDB for MySQL集群ALIYUN::ADBSLAKER::DBCluster: 创建AnalyticDB MySQL湖仓版 (3.0) 集群ALIYUN::ADB::Account: 创建数据库集群账号

模板编辑

参与数： 奇理、七格 | 更新时间： 2024-09-04 16:00:15

产品详情 相关技术圈

模板是一个JSON或YAML格式的文本文件，使用UTF-8编码。模板用于创建资源栈，是描述基础设施和架构的蓝图。模板编辑者在模板中定义云资源资源和配置细节，并说明资源间的依赖关系。

ROS模板结构

ROSTemplateFormatVersion: "2015-09-01"

Description: 模板描述信息，可用于说明模板的适用场景、架构说明等。

Metadata:

Parameters:

Mappings:

Conditions:

Resources:

Outputs:

Rules:

← 模板编写与调试

部署云资源 →

资源栈信息

参与数： 奇理、七格 | 更新时间： 2024-09-04 16:00:15

产品详情 相关技术圈

模板是一个JSON或YAML格式的文本文件，使用UTF-8编码。模板用于创建资源栈，是描述基础设施和架构的蓝图。模板编辑者在模板中定义云资源资源和配置细节，并说明资源间的依赖关系。

资源栈信息

参与数： 奇理、七格 | 更新时间： 2024-09-04 16:00:15

产品详情 相关技术圈

模板是一个JSON或YAML格式的文本文件，使用UTF-8编码。模板用于创建资源栈，是描述基础设施和架构的蓝图。模板编辑者在模板中定义云资源资源和配置细节，并说明资源间的依赖关系。

资源名称	资源类型	资源ID	资源状态	状态描述
MqProducerTransaction	ALYUN::ROCKETMQ::Group	64138	创建成功	state changed
MqConsumerTransaction	ALYUN::ROCKETMQ::Group	64138	创建成功	state changed
MqTopicTimer	ALYUN::ROCKETMQ::Topic	64138	创建成功	state changed
MqTopicNormal	ALYUN::ROCKETMQ::Topic	64138	创建成功	state changed
RdsDatabase	ALYUN::RDS::Database	64138	创建成功	state changed
MqProducerBasic	ALYUN::ROCKETMQ::Group	64138	创建成功	state changed
MqTopicNormal	ALYUN::ROCKETMQ::Topic	64138	创建成功	state changed
RdsAccountPrivilege	ALYUN::RDS::AccountPrivilege	64138	创建成功	state changed

直接找大模型帮忙

写一个ROS模板 部署一台ECS 安装dify

Qwen_Max

为了使用阿里云的ROS (Resource Orchestration Service) 部署一台ECS实例, 并在该实例上安装Dify, 你需要创建一个ROS模板。这

步骤 1: 创建ROS模板

首先, 你需要定义你的资源, 比如ECS实例、安全组等。以下是一个简单的ROS模板示例:

```
ROSTemplateFormatVersion: '2015-09-01'
Description: |
  A simple ROS template to create an ECS instance and install Dify on it.
Parameters:
  ImageId:
    Type: String
    Description: The image ID of the ECS instance.
  InstanceType:
    Type: String
    Description: The type of the ECS instance.
  SecurityGroupId:
    Type: String
    Description: The security group ID associated with the ECS instance.
  KeyPairName:
    Type: String
    Description: The key pair name for SSH access.
  VSwitchId:
    Type: String
    Description: The VSwitch ID for the ECS instance.
Resources:
  EcsInstance:
    Type: ALIYUN::ECS::Instance
    Properties:
      ImageId: !Ref ImageId
```

问题

- 通用大模型可能没有IaC相关领域知识
- 模型幻觉
 - 编造出不存在的资源
 - 指定错误的资源属性
 - 输出错误的资源返回值

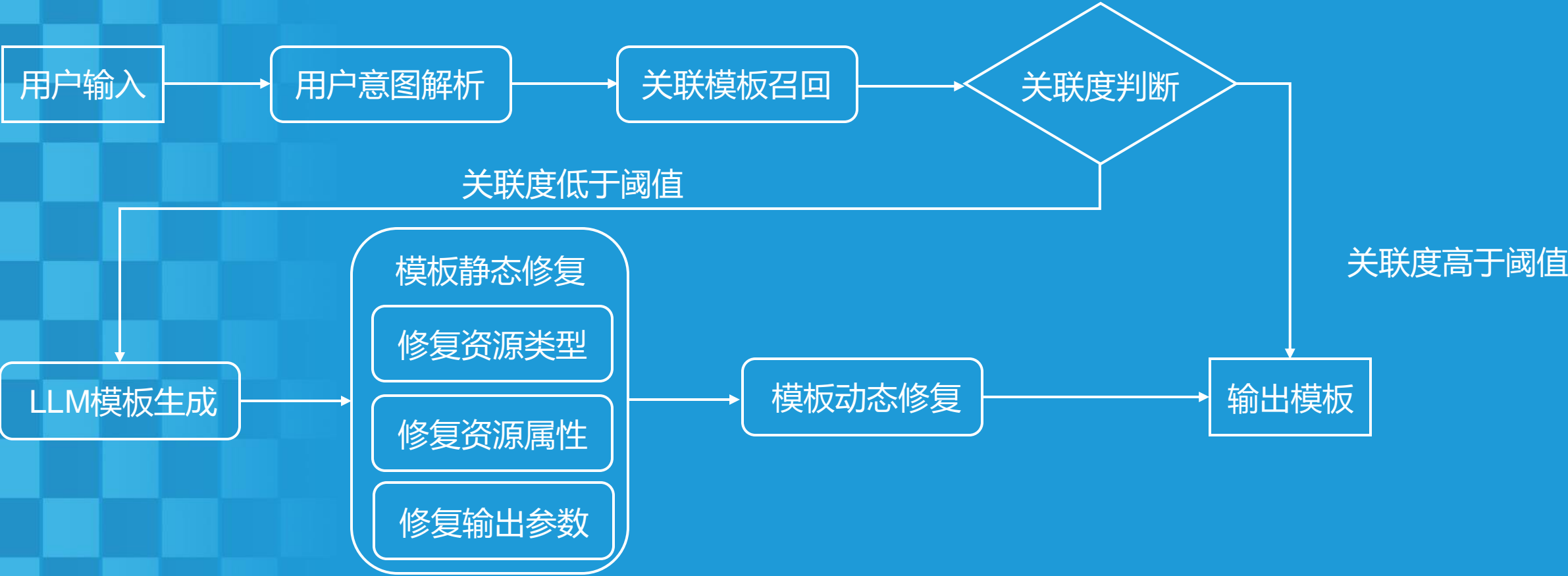
解决方法

- 通用大模型可能没有IaC相关领域知识
 - 预置IaC相关的知识库，通过RAG召回匹配度高的模板，提升大模型回复准确性
- 模型幻觉
 - 对大模型返回的模板进一步修复
 - 通过Function Call查询IaC定义的资源列表、资源属性定义，由大模型选择匹配的资源类型以及资源属性配置，实现模板的静态修复
 - 对模板进行校验、资源栈创建预检，使用大模型根据报错信息对模板进行动态的修改。

```
session = create_template_session(
    task_desc='我有几组资源名、错误的资源类型和可能的正确的待选资源类型的列表， '
    '你的工作就是必须从待选列表中找到正确的资源类型，不允许选择提供的错误的资源类型。'
    '结果为json格式，内容为资源名和正确资源类型的映射关系。',
    task_start_tip='接下来我会给出资源名、错误的资源类型和可能的正确的资源类型的列表。',
)
r = await session.ask('\n'.join(tokens))
resource_type_mappings = r.extract_json()
if not resource_type_mappings:
    ...
for res_name, res_type in bad_items:
    if res_name not in resource_type_mappings:
        ...
    if res_type in self.CommonResourceTypeMapping:
        new_res_type = self.CommonResourceTypeMapping[res_type]
    else:
        new_res_type = resource_type_mappings[res_name]
    if new_res_type not in resource_types:
        ...
    msg = f'Find match for resource type: {res_type} --> {new_res_type}.'
    ...
    tpl_resources[res_name]['Type'] = new_res_type
```

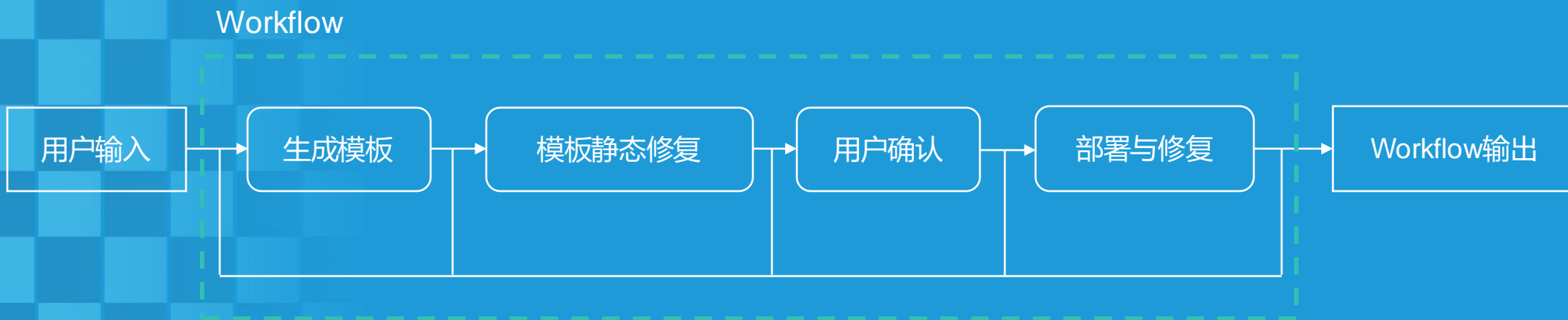
```
session = create_template_session(
    task_desc=f'我有一个有问题的模板片段，其中资源属性取值存在一些错误，具体报错信息为：\n{msg}\n'
    '你的任务就是根据错误信息，帮我修正这个模板中部分资源属性的取值，并输出新模板。'
    '# 要求 #\n'
    '模板代码请使用放在markdown代码块中，为json格式不需要注释。'
    '新模板需要保持完整，允许省略之前模板内容。'
    '报错信息中不涉及的部分，不允许进行修改。',
    task_start_tip='接下来我会给出有问题的模板，使用json格式。',
    model='qwen-max'
)
new_tpl = await self._repair_template(session, tpl)
if not new_tpl:
    return False
if res_name:
    res_names = [res_name]
else:
    res_names = list(new_tpl['Resources'].keys())
if not self._merge_props(ctx, res_names, new_tpl) or not ctx.result:
    return False
```

workflow节点的功能实现



总结简述

模板生成与修复的异步流程以工作流的形式实现，每个step节点实现一个基础能力，方便组合复用。通过获取工作流状态的形式，展示日志。



每一个Step都会有一个StepOutput作为一个阶段性的结果（可能是中间态的）
一步Step内可能会有多个Event 实现与大模型的多轮对话 需要输出日志信息

一个简单的Case

- 部署热门的大模型编排应用Dify

资源编排 ROS / 我的模板 / 创建模板

← 创建模板

模板配置 ^

模板类型

☒ ROS模板 ☐ Terraform模板

创建方式

手动创建 **AI助手创建** 可视化编辑器创建

* 请根据您的业务需求输入要生成的环境、架构或资源

创建VPC、安全组、ECS实例，配置Dify应用开发平台（CentOS 7），含Docker与Dify安装，提供公网访问。

若希望模板变得更通用，可设置模板参数

例如：将实例规格、数量作为参数

若希望看到资源的输出属性，可设置模板输出

例如：将实例ID和私网地址作为输出

生成模板 查看最近生成记录



生成模板已完成 预发2.0

✓ 模板生成输入预处理

Generate template input preprocessor started.

Generate template input preprocessor success.

✓ ROS模板生成

ROS template generator started.

Current generate ROS template number of retries: 1.

Template generate successfully.

```
1  ROSTemplateFormatVersion: '2015-09-01'
2  Parameters:
3    ZoneId:
4      Type: String
5      Label:
6        en: Availability Zone
7        zh-cn: 可用区ID
8      AssociationProperty: ALIYUN::ECS::Instance::ZoneId
9      Description:
10       en: <b><font color='blue'>This template will pull the
11         be network problems when using it in domestic
12         regions.</font></b>
13       zh-cn: <b><font color='blue'>本模板会拉取 Docker 镜像，在
14         国内地域使用可能会有网络问题，可切换至海外地域.</font></b>
14  InstanceType:
15    Type: String
```

👍

🗨

对比

```
Resources:
  VPC:
    Type: ALIYUN::ECS::VPC
    Properties:
      CidrBlock: !Ref VpcCidrBlock
      VpcName: DifyVPC
  VSwitch:
    Type: ALIYUN::ECS::VSwitch
    Properties:
      VpcId: !Ref VPC
      CidrBlock: !Ref VSwitchCidrBlock
      ZoneId: {{Select(0, !GetAZs)}}
      VSwitchName: DifyVSwitch
  SecurityGroup:
    Type: ALIYUN::ECS::SecurityGroup
    Properties:
      VpcId: !Ref VPC
      SecurityGroupName: !Ref SecurityGroupName
      SecurityGroupIngress:
        - PortRange: !Join ['-', [!Ref SSHPort, !Ref SSHPort]]
          Priority: 1
          NicType: intranet
          SourceCidrIp: 0.0.0.0/0
          IpProtocol: tcp
          Policy: Accept
  ECSInstance:
    Type: ALIYUN::ECS::Instance
    Properties:
      ImageId: !Ref ImageId
      InstanceType: !Ref InstanceType
      VpcId: !Ref VPC
      VSwitchId: !Ref VSwitch
      SecurityGroupId: !Ref SecurityGroup
      KeyPairName: !Ref KeyPairName
      SystemDiskCategory: !Ref SystemDiskCategory
      SystemDiskSize: !Ref SystemDiskSize
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          curl -fsSL https://get.docker.com -o get-docker.sh
          sh get-docker.sh
          systemctl start docker
          systemctl enable docker
          git clone https://github.com/dify-ai/dify.git /opt/dify
          cd /opt/dify
          docker-compose up -d
```

错误的函数引用

错误的端口配置

```
Resources:
  Vpc:
    Type: ALIYUN::ECS::VPC
    Properties:
      CidrBlock: 192.168.0.0/16
  VSwitch:
    Type: ALIYUN::ECS::VSwitch
    Properties:
      ZoneId:
        Ref: ZoneId
      VpcId:
        Ref: Vpc
      CidrBlock: 192.168.0.0/24
  SecurityGroup:
    Type: ALIYUN::ECS::SecurityGroup
    Properties:
      VpcId:
        Ref: Vpc
      SecurityGroupIngress_80:
        Type: ALIYUN::ECS::SecurityGroupIngress
        Properties:
          PortRange: 80/80
          SecurityGroupId:
            Ref: SecurityGroup
          SourceCidrIp: 0.0.0.0/0
          ...
  InstanceGroup:
    Type: ALIYUN::ECS::InstanceGroup
    Properties:
      VpcId:
        Ref: Vpc
      VSwitchId:
        Ref: VSwitch
      SecurityGroupId:
        Ref: SecurityGroup
      ...
  InstallDify:
    Type: ALIYUN::ECS::RunCommand
    Properties:
      CommandContent: |
        #!/bin/bash
        echo "#####"
        echo "# Install Docker"
        echo "#####"
        wget -O /etc/yum.repos.d/docker-ce.repo https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
        yum -y install docker-ce
        systemctl start docker
        systemctl enable docker

        echo "#####"
        echo "# Install Dify"
        echo "#####"
        yum -y install git
        git clone --depth 1 https://github.com/langgenius/dify.git /opt/dify
        cd /opt/dify/docker
        docker compose up -d
        docker compose ps
```

正确的参数引用

正确的端口范围设置

直接提问大模型

laC模板生成

Agent能力评估



评测方式：

- 1、输入IaC模板助手生成的模板与对比的标准模板，使用大模型进行对比、评分
- 2、输入IaC模板助手生成的模板与模板需求Query，使用大模型推理生成的模板的功能，与输入的Query进行对比评分

评测标准：

- 1、模板对比的标准：模板的资源类型、属性、输出值的正确性，模板资源间依赖正确性，模板功能的匹配度
- 2、用户输入需求对比的标准：对云资源的需求相近度，对软件部署的需求相近度

PyCon China 2024

For Good . For fun.
2024/11/23 中国 上海

