资源编排 使用教程

产品版本：ZStack 2.5.1

文档版本：V2.5.1

ZStack

版权声明

商标说明

ZStack商标和其他云轴商标均为上海云轴信息科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受上海云轴公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，上海云轴公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 概述

资源编排服务是一款帮助云计算用户简化云资源管理和自动化部署运维的服务。通过资源栈模板，定义所需的云资源、资源间的依赖关系、资源配置等，可实现自动化批量部署和配置资源，轻松管理云资源生命周期，通过API和SDK集成自动化运维能力。

如图 *1:* 资源编排所示：

图 1: 资源编排

资源编排具有以下功能优势：

1. 用户只需创建资源栈模板或修改已有模板，定义所需的云资源、资源间的依赖关系、资源配置等，资源编排将通过编排引擎自动完成所有资源的创建和配置；
2. 可根据业务需要，动态调整资源栈模板，从而调整资源栈以灵活应对业务发展需要；
3. 如果不再需要某资源栈，可一键删除该栈及栈内所有资源；
4. 可重复使用已创建的资源栈模板快速复制整套资源，无需重复配置；
5. 可根据业务场景灵活组合云服务，以满足自动化运维的需求。

2 准备工作

admin请提前安装最新版本ZStack，并部署完成创建云主机必要的资源。

详情可参考用户手册安装部署章节。

本教程将详细介绍资源编排的使用方法。

3 典型使用流程

使用资源编排服务，用户可通过资源栈模板快速创建和配置一组资源，并便捷管理这组资源。

资源编排典型使用流程如下：

1. 准备资源栈模板。
   - 使用资源编排服务，首先需准备资源栈模板，资源编排将基于所准备的模板创建和配置相应资源栈；
   - 可先查看云平台提供的资源栈示例模板（即系统模板）是否满足业务需求，若满足，可直接使用示例模板创建资源栈；
   - 若示例模板不满足业务需求，可创建一个新模板或修改已有模板（即自定义模板）。如何创建自定义模板，请参考资源栈模板章节。
2. 通过模板创建资源栈。
   - 若示例模板满足业务需求，可直接使用示例模板创建资源栈。如何使用示例模板创建资源栈，请参考资源栈示例模板章节；
   - 用户也可使用自定义模板创建资源栈。如何使用自定义模板创建资源栈，请参考资源栈模板章节。
3. 管理资源栈。
   - 提供资源栈生命周期管理；
   - 可查看资源栈内所有资源的信息；
   - 删除资源栈时，默认会删除栈内编排创建的所有资源。
   - 管理资源栈的详细介绍，请参考资源栈章节。

4 资源栈

资源编排通过资源栈模板快速创建和配置一组资源，这组资源定义为一个资源栈，通过管理资源栈，维护这组资源。

资源栈支持以下操作：

- 创建资源栈
- 查看资源栈信息
- 删除资源栈

创建资源栈

在ZStack私有云主菜单，点击**平台运维** > **资源编排** > **资源栈**，进入**资源栈**界面，点击**创建资源栈**，弹出**创建资源栈**界面。

创建资源栈分为以下两步：

1. 可参考以下示例输入相应内容：

   - **区域**：自动显示当前区域
   - **名称**：设置资源栈名称
   - **简介**：可选项，可留空不填
   - **超时设置**：用于设置创建资源栈的超时时限，超时将失败，默认为60分钟
   - **失败回滚**：默认勾选，超时失败后将清理已创建的资源
   - **创建方式**：选择创建资源栈方式

     创建资源栈有以下三种方式：

     - **资源栈模板**：选择自定义模板或系统模板创建资源栈

       如图 2: 资源栈模板方式所示：

       图 2: 资源栈模板方式

注: 如何创建自定义模板，请参考资源栈模板章节。

- **上传文件**：直接上传已定义的UTF8编码格式的模板文件创建资源栈

  如图 *3:* 上传文件方式所示：

  图 3: 上传文件方式

注: 关于模板语法的详细介绍，请参考资源栈模板语法章节。

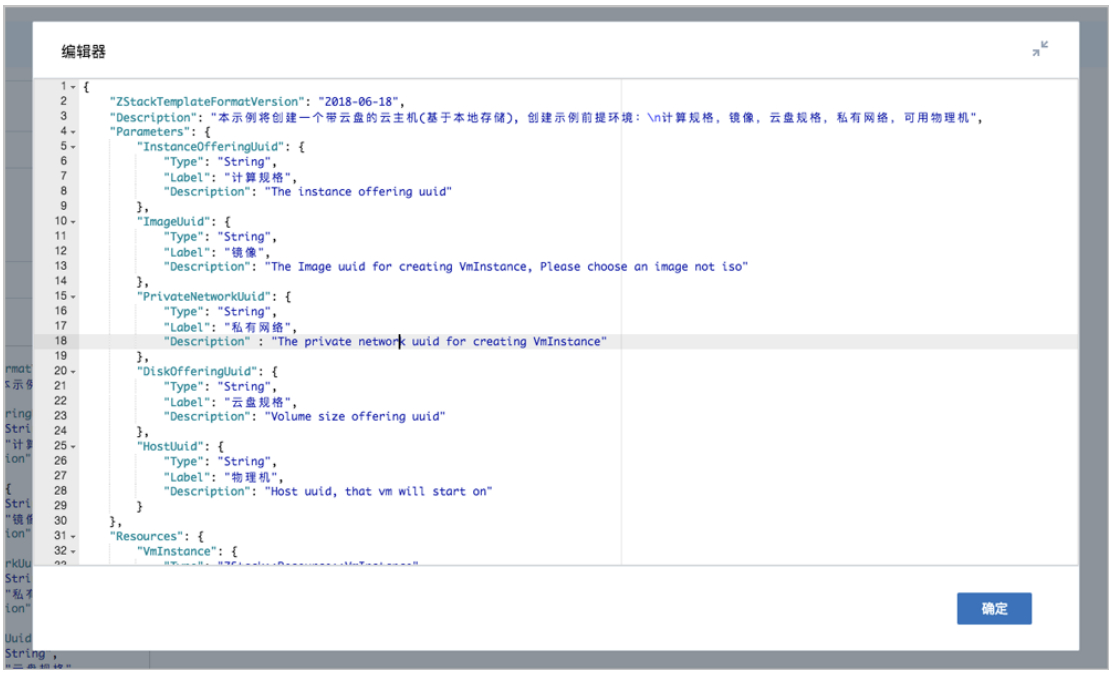
- **文本**：在文件编辑器中编辑模板创建资源栈

  如图 *4:* 文本方式：

  图 4: 文本方式

可展开编辑器，如图 *5: 展开编辑器*所示：

图 5: 展开编辑器



注: 关于模板语法的详细介绍，请参考资源栈模板语法章节。

如图 *6: 创建资源栈1*所示，点击**下一步**。

图 6: 创建资源栈1

2. 根据需要的资源信息输入各个参数，不同类型的资源栈需要输入的参数不同。以上述创建云主机带云盘的资源栈为例，可参考以下示例输入相应内容：

   - **计算规格**：选择创建云主机的计算规格
   - **镜像**：选择创建云主机的镜像
   - **私有网络**：选择创建云主机的网络，本示例需选择创建云主机的私有网络
   - **云盘规格**：选择云主机所带云盘的规格

- **物理机**：选择物理机以启动云主机

如图 7: 创建资源栈2所示，点击**确定**，开始创建资源栈。

图 7: 创建资源栈2

注:

- 开始创建资源栈前，可点击**预览**查看将要创建的资源列表。
- 创建资源栈需要一定时长，请等待创建完成。

查看资源栈信息

在**资源栈**界面，选择某一资源栈，展开其详情页，可查看当前创建的资源栈状态和信息，包括：基本属性、资源栈内容、资源、事件、审计。

- 基本信息：显示资源栈当前状态、名称、简介、栈UUID等信息，其中名称和简介支持修改
- 资源栈内容：包括模板数据和参数配置
  - 模板数据：显示当前资源栈所对应的模板信息
  - 参数配置：创建资源栈时指定的参数信息

- 资源：显示资源栈所包括的全部资源信息
- 事件：显示资源栈生命周期中发生的每一个事件
- 审计：查看此资源栈的相关操作

删除资源栈

如果不再使用某一资源栈，可将该资源栈删除。

注:

- 删除资源栈默认会删除栈内编排创建的所有资源；
- 若资源栈所对应的模板事先已设置DeletionPolicy为Retain，栈内编排创建的所有资源将会被保留，详情可参考资源*(Resources)*章节。

5 资源栈模板

基于资源栈模板可快速创建资源栈。

资源栈模板分为系统模板和自定义模板，关于系统模板介绍，详情可参考章节，本章节主要介绍自定义模板。

资源栈模板支持以下操作：

- 创建资源栈模板
- 查看模板信息
- 启用模板
- 停用模板
- 创建资源栈
- 修改模板
- 删除模板

创建资源栈模板

在ZStack私有云主菜单，点击**平台运维** > **资源编排** > **资源栈模板**，进入**资源栈模板**界面，点击**创建资源栈模板**，弹出**创建资源栈模板**界面，可参考以下示例输入相应内容：

- **名称**：设置资源栈模板名称
- **简介**：可选项，可留空不填
- **创建方式**：选择创建资源栈模板方式

  创建资源栈模板有以下两种方式：

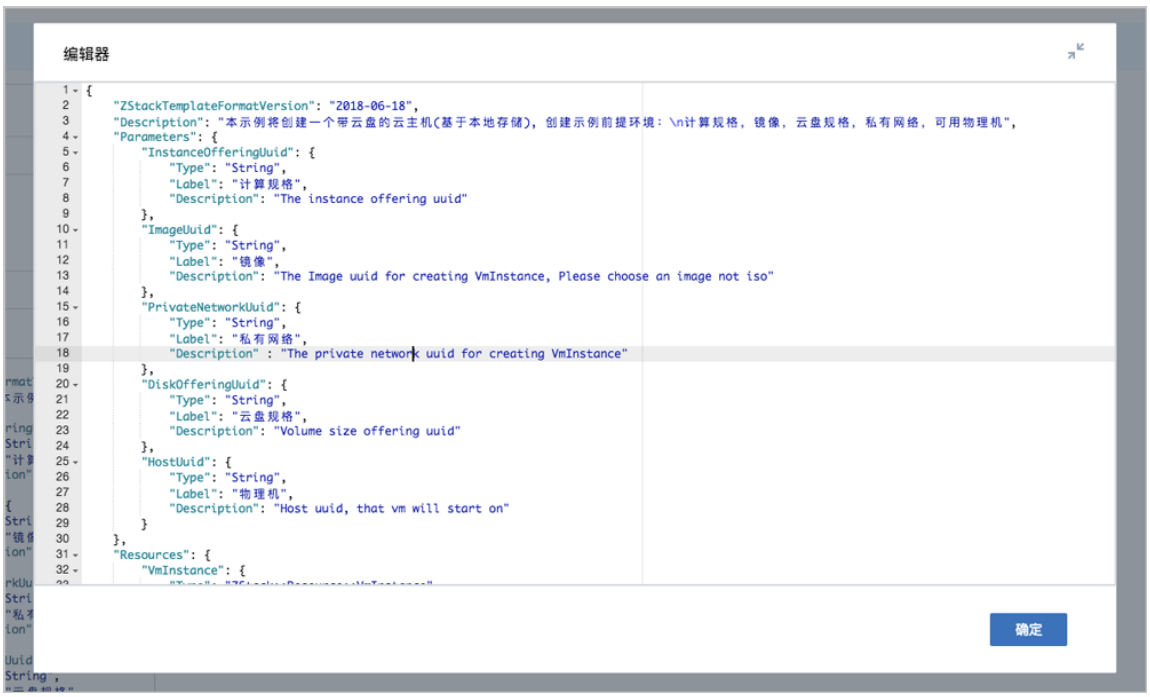
  - **文本**：在文件编辑器中编辑创建

    如图 *8:* 文本方式：

    图 8: 文本方式

可展开编辑器，如图 *9: 展开编辑器*所示：

图 9: 展开编辑器



注: 关于模板语法的详细介绍，请参考资源栈模板语法章节。

- **上传文件**：直接上传已定义的UTF8编码格式的文件创建

  如图 *10:* 上传文件方式所示：

  图 10: 上传文件方式

注: 关于模板语法的详细介绍，请参考资源栈模板语法章节。

如图 *11:* 创建资源栈模板所示，点击**确定**。

图 11: 创建资源栈模板

查看模板信息

在**资源栈模板**界面，选择某一模板，展开其详情页，可查看当前创建的模板状态和信息，包括：基本属性、资源栈模板内容、审计。

- 基本信息：显示模板当前状态、名称、简介、模板UUID、MD5码等信息，其中名称和简介支持修改
- 资源栈模板内容：显示模板具体内容，关于模板语法的详细介绍，请参考资源栈模板语法章节
- 审计：查看此模板的相关操作

启用/停用模板

- 启用模板：将已停用的模板启用
- 停用模板：将模板停用，停用的模板不能创建资源栈

创建资源栈

在**资源栈模板**界面，选择某一模板，点击**更多操作 > 创建资源栈**，弹出**创建资源栈**界面。

使用自定义模板创建资源栈分为以下两步：

1. 可参考以下示例输入相应内容：

   - **区域**：自动显示当前区域
   - **名称**：设置资源栈名称
   - **简介**：可选项，可留空不填
   - **超时设置**：用于设置创建资源栈的超时时限，超时将失败，默认为60分钟
   - **失败回滚**：默认勾选，超时失败后将清理已创建的资源
   - **资源栈模板**：自动显示已选择的模板

   如图 *11:* 创建资源栈模板所示，点击**下一步**。

   图 12: 创建资源栈1

2. 根据需要的资源信息输入各个参数，不同类型的资源栈需要输入的参数不同。以上述创建云主机带云盘的资源栈为例，可参考以下示例输入相应内容：

   - **计算规格**：选择创建云主机的计算规格
   - **镜像**：选择创建云主机的镜像

- **私有网络**：选择创建云主机的网络，本示例需选择创建云主机的私有网络
- **云盘规格**：选择云主机所带云盘的规格
- **物理机**：选择物理机以启动云主机

如图 *13:* 创建资源栈2所示，点击**确定**，开始创建资源栈。

图 13: 创建资源栈2

注:

- 开始创建资源栈前，可点击**预览**查看将要创建的资源列表。
- 创建资源栈需要一定时长，请等待创建完成。

修改模板

支持在文件编辑器中修改模板。

删除模板

如果不再使用某一模板，可将该模板删除。

约束条件

需注意：

- 每个模板文件大小不超过4MB
- 若通过API提交，则参数大小不可超过64K

6 资源栈示例模板

云平台提供了常用的示例模板，用户可基于已有示例模板创建资源栈。

资源栈模板支持以下操作：

- 查看示例模板信息
- 使用示例模板创建资源栈

查看示例模板信息

在ZStack私有云主菜单，点击**平台运维 > 资源编排 > 资源栈示例模板**，进入**资源栈示例模板**界面，选择某一示例模板，展开其详情页，可查看模板状态和信息，包括：基本属性、资源栈模板内容、审计。

- 基本信息：显示示例模板的状态、名称、简介、模板UUID、MD5码等信息

  注: 示例模板一直处于启用状态，且不允许任何修改。

- 资源栈模板内容：显示示例模板具体内容，关于模板语法的详细介绍，请参考资源栈模板语法章节
- 审计：查看此模板的相关操作

使用示例模板创建资源栈

在**资源栈示例模板**界面，选择某一示例模板，点击**更多操作 > 创建资源栈**，弹出**创建资源栈**界面。

使用示例模板创建资源栈分为以下两步：

1. 可参考以下示例输入相应内容：

   - **区域**：自动显示当前区域
   - **名称**：设置资源栈名称
   - **简介**：可选项，可留空不填
   - **超时设置**：用于设置创建资源栈的超时时限，超时将失败，默认为60分钟
   - **失败回滚**：默认勾选，超时失败后将清理已创建的资源
   - **资源栈模板**：自动显示已选择的模板

   如图 *14: 创建资源栈1*所示，点击**下一步**。

   图 14: 创建资源栈1

2. 根据需要的资源信息输入各个参数，不同类型的资源栈需要输入的参数不同。以上述模板**ZStack.System.v1.EIP**为例，通过该模板将创建一个弹性IP，并将弹性IP绑定到云主机上，可参考以下示例输入相应内容：

   - **计算规格**：选择创建云主机的计算规格
   - **镜像**：选择创建云主机的镜像
   - **私有网络**：选择创建云主机的网络，本示例需选择创建云主机的私有网络
   - **公有网络**：选择提供虚拟IP的公有网络，通过虚拟IP提供弹性IP服务

   如图 *15:* 创建资源栈*2*所示，点击**确定**，开始创建资源栈。

   图 15: 创建资源栈2

注:

- 开始创建资源栈前，可点击**预览**查看将要创建的资源列表。
- 创建资源栈需要一定时长，请等待创建完成。

7 快速实践

背景信息

本章节介绍如何使用一个资源栈示例模板**ZStack.System.v1.VPC**一键部署VPC网络。

操作步骤

1. 准备资源栈模板。

   本实践直接使用示例模板创建资源栈。

   在ZStack私有云主菜单，点击**平台运维** > **资源编排** > **资源栈示例模板**，进入**资源栈示例模板**界面，即可看到示例模板**ZStack.System.v1.VPC**，展开其详情页，查看资源栈模板内容，详情如下：

```
{
 "ZStackTemplateFormatVersion": "2018-06-18",
 "Description": "本示例会创建一个简单的VPC网络，需要用户提供下面正确的数据\n公有网络Uuid\n管理网络Uuid: 如果只有公有网络，则把公有网络当作管理网即可.\nVxlan网络的VTEP Cider",
 "Parameters": {
  "VrouterImageUrl": {
   "Type": "String",
                "Label":"云路由镜像链接地址",
   "Description":"云路由镜像链接地址",
   "DefaultValue": "http://cdn.zstack.io/product_downloads/vrouter/2.3/zstack-vrouter-2.3.2.qcow2"
  },
  "VmImageUrl": {
   "Type": "String",
                "Label": "云主机镜像链接地址",
   "Description":"测试云主机镜像，请确定ZStack 可以下载下面链接的镜像",
   "DefaultValue": "http://cdn.zstack.io/zstack_repo/latest/zstack-image-1.4.qcow2"
  },
  "BackupStorage":{
   "Type": "CommaDelimitedList",
                "Label": "镜像服务器",
   "Description":"镜像服务器Uuid"
  },
  "ManagementNetworkUuid":{
   "Type": "String",
                "Label": "管理网络",
   "Description":"管理网络Uuid,如果只有公有网络填入公有网络Uuid即可"
  },
  "PublicNetworkUuid":{
   "Type": "String",
                "Label": "公有网络",
   "Description":"公有网络Uuid"
  },
  "ZoneUuid":{
   "Type": "String",
                "Label": "区域",
   "Description":"区域Uuid"
```

```
  },
  "ClusterUuid":{
   "Type": "String",
                "Label": "集群",
   "Description":"集群Uuid"
  },
  "Cidr":{
   "Type": "String",
   "Label": "VTEP CIDR",
                "Description":"VTEP Cider",
   "DefaultValue":"{10.0.0.0/8}"
  },
  "Vni":{
   "Type": "Number",
   "DefaultValue":222
  },
  "StartVni":{
   "Type": "Number",
                "Label": "起始Vni",
   "DefaultValue":100
  },
  "EndVni":{
   "Type": "Number",
                "Label": "结束Vni",
   "DefaultValue":300
  },
  "StartIp":{
   "Type": "String",
                "Label": "起始IP",
   "DefaultValue":"192.168.20.2"
  },
  "EndIp":{
   "Type": "String",
                "Label": "结束IP",
   "DefaultValue":"192.168.20.200"
  },
  "Netmask":{
   "Type": "String",
                "Label": "子网掩码",
   "DefaultValue":"255.255.255.0"
  },
  "Gateway":{
   "Type": "String",
                "Label": "网关",
   "DefaultValue":"192.168.20.1"
  }
 },
 "Resources": {
  "VrouterImage": {
   "Type": "ZStack::Resource::Image",
   "Properties": {
    "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, {"Ref":"ZStack::StackUuid"}, {"Ref
":"ZStack::AccountUuid"}, {"Ref":"ZStack::AccountName"}, "Vrouter-Image"]]},
    "url": {"Ref":"VrouterImageUrl"},
    "system": true,
    "format": "qcow2",
    "backupStorageUuids":{"Ref":"BackupStorage"}
   }
  },
  "VMImage": {
   "Type": "ZStack::Resource::Image",
   "Properties": {
```

```
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "VmImage"]]},
  "url": {"Ref":"VmImageUrl"},
  "format": "qcow2",
  "backupStorageUuids":{"Ref":"BackupStorage"}
 }
},
"VirtualRouterOffering":{
 "Type":"ZStack::Resource::VirtualRouterOffering",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "Vrouter-Offering"]]},
  "zoneUuid":{"Ref":"ZoneUuid"},
  "managementNetworkUuid":{"Ref":"ManagementNetworkUuid"},
  "publicNetworkUuid":{"Ref":"PublicNetworkUuid"},
  "imageUuid":{"Fn::GetAtt":["VrouterImage", "uuid"]},
  "cpuNum":2,
  "memorySize":2147483648
 }
},
"VpcVRouter":{
 "Type":"ZStack::Resource::VpcVRouter",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "VPC-Router"]]},
  "virtualRouterOfferingUuid":{"Fn::GetAtt":["VirtualRouterOffering","uuid"]}
 }
},
"L2VxlanNetworkPool":{
 "Type":"ZStack::Resource::L2VxlanNetworkPool",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "L2VxlanNetworkPool"]]},
  "zoneUuid":{"Ref":"ZoneUuid"}
 }
},
"VniRange":{
 "Type":"ZStack::Resource::VniRange",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "VniRange"]]},
  "startVni":{"Ref":"StartVni"},
  "endVni":{"Ref":"EndVni"},
  "l2NetworkUuid":{"Fn::GetAtt":["L2VxlanNetworkPool","uuid"]}
 }
},
"L2VxlanNetwork":{
 "Type":"ZStack::Resource::L2VxlanNetwork",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "L2VxlanNetwork"]]},
  "poolUuid":{"Fn::GetAtt":["L2VxlanNetworkPool","uuid"]},
  "zoneUuid":{"Ref":"ZoneUuid"},
  "vni":{"Ref":"Vni"}
 }
},
"VpcL3Network":{
 "Type":"ZStack::Resource::L3Network",
 "Properties":{
  "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "VPC-Network"]]},
  "l2NetworkUuid":{"Fn::GetAtt":["L2VxlanNetwork","uuid"]},
  "category":"Private",
  "type":"L3VpcNetwork",
  "systemTags":["networkservices::VRouter"]
 }
},
"InstanceOffering":{
 "Type":"ZStack::Resource::InstanceOffering",
```

```
   "Properties":{
    "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "1cpu","4G"]]},
    "cpuNum": 1,
    "memorySize" : 4294967296
   }
  },

  "AttachL3ToVm":{
   "Type":"ZStack::Action::AttachL3NetworkToVm",
   "Properties":{
    "vmInstanceUuid": {"Fn::GetAtt":["VpcVRouter","uuid"]},
    "l3NetworkUuid":{"Fn::GetAtt":["VpcL3Network","uuid"]}
   },
   "DependsOn":[{"Ref":"AddIpRange"}]
  },
  "AddIpRange" :{
   "Type":"ZStack::Action::AddIpRange",
   "Properties":{
    "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "iprange"]]},
    "l3NetworkUuid":{"Fn::GetAtt":["VpcL3Network","uuid"]},
    "startIp":{"Ref":"StartIp"},
    "endIp":{"Ref":"EndIp"},
    "netmask":{"Ref":"Netmask"},
    "gateway":{"Ref":"Gateway"}
   }
  },
  "AttachL2NetworkToCluster":{
   "Type":"ZStack::Action::AttachL2NetworkToCluster",
   "Properties":{
    "l2NetworkUuid":{"Fn::GetAtt":["L2VxlanNetworkPool","uuid"]},
    "clusterUuid":{"Ref":"ClusterUuid"},
    "systemTags":[{"Fn::Join":["::",["l2NetworkUuid",{"Fn::GetAtt":["L2VxlanNetwork","uuid
"]},"clusterUuid",{"Ref":"ClusterUuid"},"cidr",{"Ref":"Cidr"}]]}]
   }
  },
  "TestVm":{
   "Type":"ZStack::Resource::VmInstance",
   "Properties":{
    "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"}, "TestVm"]]},
    "instanceOfferingUuid": {"Fn::GetAtt":["InstanceOffering","uuid"]},
    "l3NetworkUuids": [{"Fn::GetAtt":["VpcL3Network","uuid"]}],
    "imageUuid": {"Fn::GetAtt":["VMImage", "uuid"]}
   },
   "DependsOn":[{"Ref":"AttachL3ToVm"}]
  }
 },
 "Outputs": {
  "vpc": {
   "Value": {
    "Ref": "VpcL3Network"
   }
  }
 }
}
```

上述模板包含五个顶级字段：

- `"ZStackTemplateFormatVersion": "2018-06-18"`

  定义模板版本。

- "Description": "本示例会创建一个简单的VPC网络，需要用户提供下面正确的数据\n公有网络Uuid\n管理网络Uuid: 如果只有公有网络，则把公有网络当作管理网即可.\nVxlan网络的VTEP Cider"

  定义对模板的解释说明。

- "Parameters": { }

  定义模板的参数列表。

  本例中定义了以下参数：

  - 云路由镜像URL
  - 云主机镜像URL
  - 镜像服务器
  - 管理网络
  - 公有网络
  - 区域
  - 集群
  - VTEP CIDR
  - Vni、起始Vni、结束Vni
  - 起始IP、结束IP、子网掩码、网关

- "Resources": { }

  定义该模板将要创建的资源。

  本例中声明将要创建以下资源：

  - 添加一个云路由镜像
  - 添加一个云主机镜像
  - 创建一个云路由规格
  - 创建一个VPC路由器
  - 创建VXLAN网络池
  - 创建一个二层VXLAN网络
  - 创建一个VPC网络
  - 创建一个计算规格
  - 将VPC网络绑定到云主机

- 配置VPC网络的IP范围
- 将二层VXLAN网络加载到集群
- 创建一个云主机

这里声明的资源属性可以引用"Parameters": { }中定义的参数。

- "Outputs": { }

定义资源创建完成后，通过资源栈输出资源属性等有用信息。

关于模板语法的详细介绍，请参考资源栈模板语法章节。

2. 通过示例模板创建资源栈。

在**资源栈示例模板**界面，选择示例模板**ZStack.System.v1.VPC**，点击**更多操作** > **创建资源栈**，弹出**创建资源栈**界面。

1. 可参考以下示例输入相应内容：

- **区域**：自动显示当前区域
- **名称**：设置资源栈名称
- **简介**：可选项，可留空不填
- **超时设置**：用于设置创建资源栈的超时时限，超时将失败，默认为60分钟
- **失败回滚**：默认勾选，超时失败后将清理已创建的资源
- **资源栈模板**：自动显示已选择的模板

如图 *16: 创建资源栈1*所示，点击**下一步**。

图 16: 创建资源栈1

2. 根据需要的资源信息输入各个参数，不同类型的资源栈需要输入的参数不同。本例中可参考以下示例输入相应内容：

   - **云路由镜像URL**：添加云路由镜像以创建VPC路由器
   - **云主机镜像URL**：添加创建云主机的镜像
   - **镜像服务器**：选择合适的镜像服务器
   - **管理网络**：选择已提前创建的管理网络

     注: 出于安全和稳定性考虑，建议部署独立的管理网络，并与公有网络隔离。

   - **公有网络**：选择已提前创建的公有网络
   - **区域**：自动显示当前区域
   - **集群**：可选项，可选择VXLAN网络池加载的集群
   - **VTEP CIDR**：设置VTEP相应的CIDR

- **Vni**：可选项，可从VXLAN网络池中选择指定的Vni，若留空不填，则由系统动态随机分配
- **起始Vni**：设置VXLAN网络池的起始Vni
- **结束Vni**：设置VXLAN网络池的结束Vni
- **起始IP**：设置VPC网络的起始IP
- **结束IP**：设置VPC网络的结束IP
- **子网掩码**：设置VPC网络的子网掩码
- **网关**：设置VPC网络的网关

如图 *17:* 创建资源栈*2*所示，点击**确定**，开始创建资源栈。

图 17: 创建资源栈2


上一步
预览
确定
取消
创建资源栈
云路由镜像URL:
http://cdn.zstack.io/product_downloads/vrouter/2.3/zs
云主机镜像URL:
http://cdn.zstack.io/zstack_repo/latest/zstack-image-
镜像服务器: *
BS-1
管理网络: *
L3-管理网络
公有网络: *
L3-公有网络
区域: *
ZONE-1
集群: *
Cluster-1

注:

- 开始创建资源栈前，可点击**预览**查看将要创建的资源列表。
- 创建资源栈需要一定时长，请等待创建完成。

3. 管理资源栈。

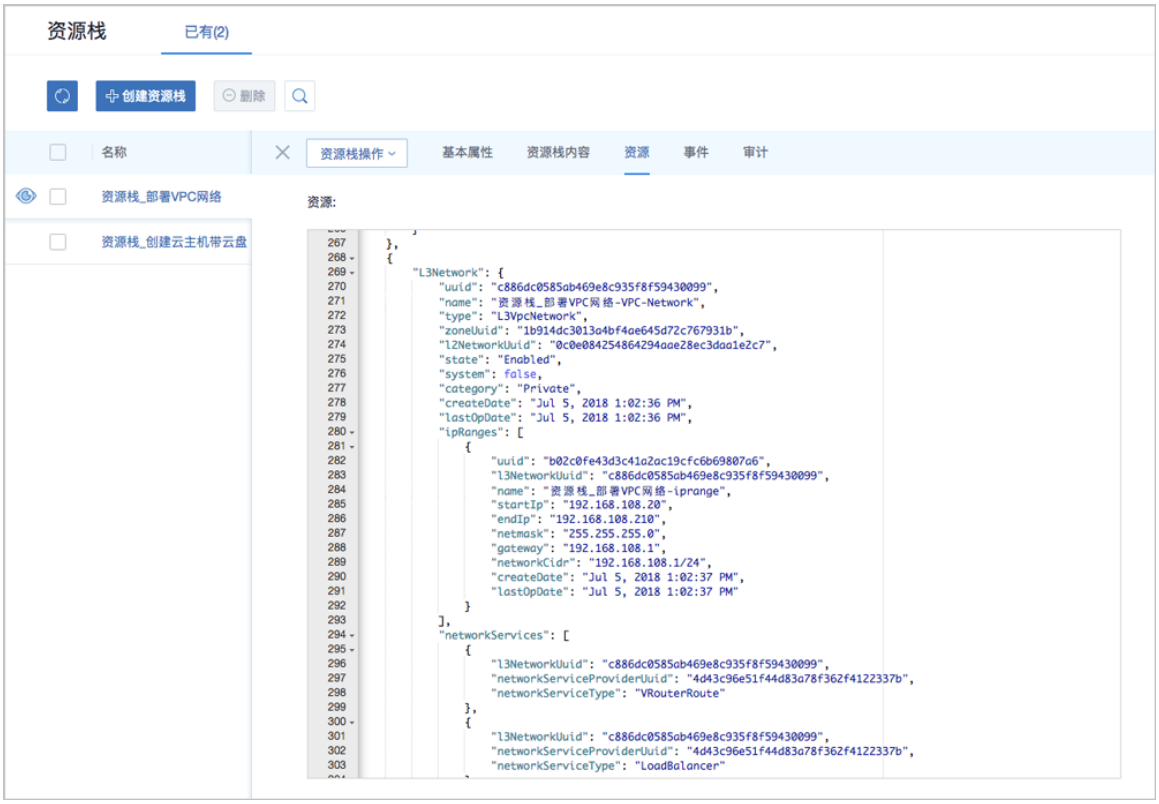
   资源栈创建成功后，可在**资源栈**界面，选中当前创建的资源栈，展开其详情页，查看栈状态和栈信息。

   - 基本信息：显示资源栈当前状态、名称、简介、栈UUID等信息，其中名称和简介支持修改
   - 资源栈内容：包括模板数据和参数配置
     - 模板数据：显示当前资源栈所对应的模板信息

  - 参数配置：创建资源栈时指定的参数信息
- 资源：显示资源栈所包括的全部资源信息
- 事件：显示资源栈生命周期中发生的每一个事件
- 审计：查看此资源栈的相关操作

本例中，资源栈内的全部资源信息如图 *18:* 资源栈输出资源信息所示：

图 18: 资源栈输出资源信息



如果不再使用该资源栈，可将栈删除。

8 附录

8.1 资源栈模板语法

资源栈模板是一个UTF8编码格式的文件。

基于模板可快速创建资源栈，用户在模板中定义所需的云资源、资源间的依赖关系、资源配置等，资源编排将解析模板，自动完成所有资源的创建和配置。

资源栈模板结构

资源栈模板结构如下：

```
{
   "ZStackTemplateFormatVersion" : "YYYY-MM-DD",
   "Description" : "模板描述信息，可用于说明模板的适用场景、架构说明等。",
   "Parameters" : {
     // 定义创建资源栈时，用户可以定制化的参数。
   },
   "Mappings" : {
     // 定义映射信息表，映射信息是一种多层的Map结构。
   },
   "Resources" : {
     // 所需资源的详细定义，包括资源间的依赖关系、配置细节等。
   },
   "Outputs" : {
     // 用于输出一些资源属性等有用信息，可以通过API获取输出的内容。
  }
```

- **ZStackTemplateFormatVersion(必需)**

  模板版本号。

  - 格式为：YYYY-MM-DD
- **Description(可选)**

  模板描述信息，可用于说明模板的适用场景、架构说明等。

  - 对模板进行详细描述，有利于用户理解模板内容。
- **Parameters(可选)**

  定义创建资源栈时，用户可以定制化的参数。

  - 例如，用户将创建云主机的计算规格设计成一个参数。
  - 参数支持默认值。
  - 使用参数可以增强模板的灵活性，提高复用性。

  - 关于**参数(Parameters)**的详细介绍，请参考*参数(Parameters)*章节。
- **Mappings(可选)**

  定义映射信息表，映射信息是一种多层的Map结构。

  - 可通过Fn::FindInMap函数选择key对应的值。
  - 可根据不同的输入参数值作为key查找映射表。
  - 例如，可根据区域不同，自动查找区域-镜像映射表，从而找到适用的镜像。
  - 关于**映射(Mappings)**的详细介绍，请参考*映射(Mappings)*章节。
- **Resources(可选)**

  所需资源的详细定义，包括资源间的依赖关系、配置细节等。

  - 关于**资源(Resources)**的详细介绍，请参考*资源(Resources)*章节。
- **Outputs(可选)**

  用于输出一些资源属性等有用信息，可以通过API获取输出的内容。

  - 关于**输出(Outputs)**的详细介绍，请参考*输出(Outputs)*章节。

8.1.1 参数(Parameters)

参数(Parameters)：定义创建资源栈时，用户可以定制化的参数。

- 创建资源栈模板时，使用参数可以增强模板的灵活性，提高复用性。
- 创建资源栈时，可根据实际情况替换模板中的某些参数值。

语法

参数由参数名称和参数属性组成。

- 参数名称必须为字母数字，同一个模板中不能与其它参数名称重复。
- 可以用Label字段定义友好的参数名。

参数属性列表：

属性	描述	是否必需	举例
Type	参数类型，默认支持： • String • Number(整数或浮点)	是	"Type": "String"

属性	描述	是否必需	举例
	• CommaDelimited List(相当于Java里的List<String>) • Boolean		
Lable	参数别名，生成预览或正式表单时用	否	"Lable": "云主机密码"
Description	参数描述	否	"Description": "云主机登录密码"
NoEcho	该字段是否用*****替代，不填则不替代	否	"NoEcho": true **注:** 暂不支持
DefaultValue	参数默认值	否	"DefaultValue": "password"

资源编排还提供一些常量参数。

- 常量参数可直接引用，无需在Parameters中定义（也不可定义）。
- 其值在资源编排运行时确定。

常量参数列表

常量名	描述
ZStack::StackName	当前栈的名称
ZStack::StackUuid	当前栈的UUID
ZStack::AccountUuid	当前栈的AccountUuid
ZStack::AccountName	当前栈的AccountName

示例

代码段示例如下：

```
"Parameters" : {
  "username" : {
    "Label": "登录名",
    "Description" : "登录名",
    "DefaultValue": "root",
    "Type" : "String"
  },
  "password" : {
    "Label": "密码",
```

```
    "NoEcho" : "true",
    "Description" : "主机登录密码",
    "Type" : "String",
    "AllowedPattern" : "[a-zA-Z0-9]*"
  }
}
```

本例中Parameters声明两个参数：

- username
  - 参数属于**String**类型，默认值为**root**。
  - 可指定的最小长度为**2**，可指定的最大长度为**12**。

  注: username的默认值也必须符合长度限制和允许值限制。
- password
  - 参数属于**String**类型，无默认值。
  - 将NoEcho属性设置为**true**，可阻止查询栈接口返回参数值。

    注: NoEcho属性设置暂不支持。
  - 可指定的最小长度为**6**，可指定的最大长度为**41**。
  - 允许大、小写字母字符和数字。

8.1.2 资源(Resources)

资源(Resources)：所需资源的详细定义，包括资源间的依赖关系、配置细节等。

- Resources可引用前述Parameters、Mappings、以及Functions的内容。
- Resources可被其他Resources和Outputs所引用。

语法

资源由资源逻辑UUID和资源描述组成。

- 资源描述用大括号{ }括起。
- 如果声明多个资源，用逗号,分隔开。

资源关键字列表：

关键字	描述	是否必需	举例
Type	资源类型，包括以下两种类型： • Resource类型 • Action类型	是	• "Type": "ZStack::Resource::VmInstance" • "Type": "ZStack::Action::AddIpRange" • 详情请参考**资源类型*(Type)***
Properties	资源属性，为资源指定创建参数	是	详情请参考**资源属性*(Properties)***
DependsOn	资源依赖，定义资源所依赖的资源	否	• "DependsOn": [{"Ref": "WebServer1"}] • 详情请参考**资源依赖*(DependsOn)***
DeletionPolicy	删除策略 • 资源栈被删除时是否保留某个资源 • 若某个资源需要保留，则它所依赖的资源也要保留(系统自动为其保留) • 默认不保留	否	• "DeletionPolicy": "Retain" • 详情请参考**删除策略*(DeletionPolicy)***
Description	资源描述	否	• "Description" : "attach ip range to l3 network"

示例

代码段示例如下：

```
"Resources" : {
   "UUID-1" : {
      "Description" : "资源描述",
      "Type" : "资源类型",
      "Properties" : {
         资源属性描述
      }
   },
   "UUID-2" : {
```

```
      "Description" : "资源描述"
      "Type" : "资源类型",
      "Properties" : {
         资源属性描述
      } ,
      "DependsOn":"要依赖的资源，如UUID-1，注意上下文中必须包含此资源",
      "DeletionPolicy":"删除策略"
   }
}
```

本例中Parameters声明了两个资源，关键字说明如下：

- **输出UUID**
  - UUID-1、UUID-2均为资源逻辑UUID，且均为变量。
  - 在创建模板其它部分时，可以通过资源逻辑UUID引用该资源。
  - 资源逻辑UUID在模板中具有唯一性。
- **资源类型(Type)**
  - 表示正在声明的资源的类型，包括：Resource类型、Action类型。
  - 例如，"Type": "ZStack::Resource::VmInstance"表示云主机实例，"Type": "ZStack::Action ::AddIpRange"表示添加IP范围。
  - 关于资源编排支持的所有资源列表，详情请参考章节。
- **资源属性(Properties)**
  - 为资源指定创建参数。
  - 代码段示例如下：

```
"Resources" : {
   "InstanceOffering" : {
      "Type" : "ZStack::InstanceOffering",
      "Properties" : {
         "cpuNum" : "1",
         "cpuSpeed" : "1",
         "memorySize" : "1073741824",
         "name" : "instance-offering",
         "type" : "UserVm",
         "sortKey": 0,
         "allocatorStrategy": "LeastVmPreferredHostAllocatorStrategy"
      }
   }
}
```

  - 资源属性值定义规则：
    - 属性值可以是文本字符串、字符串列表、布尔值、引用参数、或者函数返回值。
    - 如果属性值为文本字符串或布尔值，该值会被双引号" "括起来。
    - 如果属性值为任一类型的字符串列表，该值会被中括号[ ]括起来。

- 如果值为内部函数或引用的参数，该值会被大括号{ }括起来。
- 将文字、列表、引用参数、和函数返回值合并起来取值时，上述规则适用。
- 以下示例说明如何声明不同的属性值类型：

```
"Properties" : {
    "String" : "string",
    "LiteralList" : [ "value1", "value2" ],
    "Boolean" : "true"
    "ReferenceForOneValue" :  { "Ref" : "ResourceID" } ,
    "FunctionResultWithFunctionParams" : {
        "Fn::Join" : [ "%", [ "Key=", { "Ref" : "SomeParameter" } ] ] }
}
```

- 如果资源不需要声明任何属性，可以忽略该资源的属性部分。

- **资源依赖(DependsOn)**
  - 定义资源所依赖的资源。
  - 为某个资源添加DependsOn属性后，该资源仅在DependsOn属性中指定的资源之后创建。
  - 代码段示例如下：

```
{
   "ZStackTemplateFormatVersion" : "2018-06-18",
   "Resources" : {
      "WebServer": {
         "Type": "ZStack::Resource::VmInstance",
         "DependsOn": "DatabseServer"
      },
      "DatabseServer": {
         "Type": "ZStack::Resource::VmInstance",
         "Properties": {
            "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"},"VM"]]},
            "instanceOfferingUuid": {"Ref":"InstanceOfferingUuid"},
            "imageUuid":{"Ref":"ImageUuid"},
            "l3NetworkUuids":[{"Ref":"PrivateNetworkUuid"}],
            "dataDiskOfferingUuids":[{"Ref":"DiskOfferingUuid"}],
        "hostUuid":{"Ref":"HostUuid"}
         }
      }
   }
}
```

  本例表示WebServer将在DatabaseServer创建成功后才开始创建。

- **删除策略(DeletionPolicy)**
  - 在模板中，设置DeletionPolicy属性，可以声明资源栈被删除时是否保留资源。
  - DeletionPolicy有Retain和Delete两个选项。
    - 默认为Delete，表示删除资源栈默认会删除栈内编排创建的所有资源。

- 若将DeletionPolicy设置为Retain，表示资源栈被删除时可保留资源。此时，该资源所依赖的资源也要保留（系统自动为其保留）。

  例如，模板对应的资源栈被删除时，保留栈内的云主机，代码段示例如下：

```
"Resources" : {
   "VMInstance" : {
      "Type" : "ZStack::Resource::VmInstance",
      "Properties": {
         "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"},"VM"]]},
         "instanceOfferingUuid": {"Ref":"InstanceOfferingUuid"},
         "imageUuid":{"Ref":"ImageUuid"},
         "l3NetworkUuids":[{"Ref":"PrivateNetworkUuid"}],
         "dataDiskOfferingUuids":[{"Ref":"DiskOfferingUuid"}],
    "hostUuid":{"Ref":"HostUuid"}
   },
   "DeletionPolicy" : "Retain"
 }
}
```

8.1.3 输出(Outputs)

输出(Outputs)：用于输出一些资源属性等有用信息，可以通过API获取输出的内容。

语法

输出由输出UUID和输出描述组成。

- 输出描述用大括号{ }括起。
- 如果声明多个输出项，用逗号,分隔开。

输出关键字列表：

关键字	描述	是否必需	举例
Description	输出描述	否	• "Description" : "print l3 network" • 详情请参考**输出描述*(Description)***
Value	输出内容	是	• "Value" : {"Ref": "WebServer1"} • 详情请参考**输出内容*(Value)***

示例

代码段示例如下：

```
"Outputs" : {
    "UUID-1" : {
        "Description" : "输出描述",
        "Value" : "输出内容"
    },
    "UUID-2" : {
        "Description" : "输出的描述",
        "Value" : "输出内容"
    }
}
```

本例中Output声明了两个输出项，关键字说明如下：

- **输出UUID**
  - 输出UUID在模板中具有唯一性。
- **输出描述(Description)**
  - 用于描述输出值的String类型。
- **输出内容(Value)**
  - 在调用查询堆栈接口时，返回的属性值。
  - 代码段示例如下：

```
{
 "ZStackTemplateFormatVersion": "2018-06-18",
 "Description": "本示例将创建一个带云盘的云主机(基于本地存储), 创建示例前提环境：\n
计算规格，镜像，云盘规格，私有网络，可用物理机",
 "Parameters": {
     "InstanceOfferingUuid": {
         "Type": "String",
             "Label": "计算规格",
   "Description": "The instance offering uuid"
     },
     "ImageUuid": {
   "Type": "String",
             "Label": "镜像",
   "Description": "The Image uuid for creating VmInstance, Please choose an image not
 iso"
     },
     "PrivateNetworkUuid": {
   "Type": "String",
             "Label": "私有网络",
   "Description" : "The private network uuid for creating VmInstance"
         },
     "DiskOfferingUuid": {
   "Type": "String",
             "Label": "云盘规格",
   "Description": "Volume size offering uuid"
     },
```

```
    "HostUuid": {
  "Type": "String",
          "Label": "物理机",
  "Description": "Host uuid, that vm will start on"
    }
},
"Resources": {
    "VmInstance": {
  "Type": "ZStack::Resource::VmInstance",
  "Properties": {
      "name": {"Fn::Join":["-",[{"Ref":"ZStack::StackName"},"VM"]]},
      "instanceOfferingUuid": {"Ref":"InstanceOfferingUuid"},
      "imageUuid":{"Ref":"ImageUuid"},
      "l3NetworkUuids":[{"Ref":"PrivateNetworkUuid"}],
      "dataDiskOfferingUuids":[{"Ref":"DiskOfferingUuid"}],
      "hostUuid":{"Ref":"HostUuid"}
  }
    }
},
"Outputs": {
    "VmInstance": {
      "Value": {
      "Ref": "VmInstance"
  }
    }
}
}
```

本例中，输出部分有1个输出项，将输出VmInstance的属性值。

8.1.4 函数(Functions)

资源编排提供多个内置函数，用于管理资源栈。可在定义资源(Resources)、输出(Outputs)和映射(Mappings)时，使用内置函数。

提供的内置函数列表：

- Fn::Base64
- Fn::FindInMap
- Fn::GetAtt
- Fn::Join
- Fn::Split
- Fn::Select
- Ref

Fn::Base64

返回输入字符串的Base64编码结果。

- **声明**

```
"Fn::Base64" : stringToEncode
```

- **参数**
  - stringToEncode：转换成Base64的字符串。
- **示例**

```
"Fn::Base64" : "password"
```

- **返回值**

  用Base64表示的原始字符串。

  本例中，返回"cGFzc3dvcmQ="，即"password"的Base64编码结果。

Fn::FindInMap

返回与Mappings声明的双层映射中的键对应的值。

- **声明**

```
"Fn::FindInMap" : ["MapName", "TopLevelKey", "SecondLevelKey"]
```

- **参数**
  - MapName：Mappings 中所声明映射的 ID，包含键和值。
  - TopLevelKey：第一级键，其值是一个键/值对列表。
  - SecondLevelKey：第二级键，其值是一个字符串或者数字。
- **示例**

```
"Fn::FindInMap" : ["RegionMap", "cn-shanghai", "32"]
```

- **返回值**

  分配给SecondLevelKey的值。

  本例中，返回"RegionMap"中"cn-shanghai"对应的键/值对列表里，键为"32"对应的值。

- **支持的函数**

  可在Fn::FindInMap函数中嵌套使用以下函数：

  - Fn::FindInMap
  - Ref

Fn::GetAtt

返回模板中的资源的属性值。

- **声明**

```
"Fn::GetAtt": ["resourceUuid", "attributeName"]
```

- **参数**
  - resourceUuid：目标资源的逻辑UUID。
  - attributeName：目标资源的属性名称。
- **示例**

```
"Fn::GetAtt" : ["MyVMInstance", "ImageUuid"]
```

- **返回值**

  属性值。

  本例中，返回resourceUuid为"MyVMInstance"的"ImageUuid"属性。

Fn::Join

将一组值连接起来，用特定分隔符隔开。

- **声明**

```
"Fn::Join" : ["delimiter", ["string1", "string2", ...]]
```

- **参数**
  - delimiter：分隔符。分隔符可为空，可将所有的值直接连接起来。
  - ["string1", "string2", ...]：被连接起来的值列表示例。
- **示例**

```
"Fn::Join" : ["-", ["a", "b", "c"]]
```

- **返回值**

  被连接起来的字符串。

  本例中，返回"a-b-c"

- **支持的函数**

  可在Fn::Join函数中嵌套使用以下函数：

  - Fn::Base64

- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Split

通过指定分隔符对字符串进行切片，并返回所有切片组成的列表。

- **声明**

```
"Fn::Split" : ["delimiter", "original_string"]
```

- **参数**
  - delimiter：分隔符，例如：,，;，\n，\t 等。
  - original_string：将要被切片的字符串。
- **示例**

```
"Fn::Split": [";", "foo; bar; achoo"]
```

- **返回值**

  切片后所有字符串组成的列表。

  本例中，返回["foo", " bar", "achoo"]

- **支持的函数**

  可在Fn::Split函数中嵌套使用以下函数：

  - Fn::Base64
  - Fn::FindInMap
  - Fn::GetAtt
  - Fn::Join
  - Fn::Select
  - Ref

Fn::Select

通过索引返回数据元列表中的单个数据元。

- **声明**

- 数据元列表可为一个数组：

```
"Fn::Select" : ["index", ["value1", "value2", ...]]
```

- 数据元列表可为一个映射表：

```
"Fn::Select" : ["index", {"key1": "value1", ...}]
```

- **参数**
  - index：待检索数据元的索引。
    - 如果数据元列表是一个数组，则索引是0到N-1之间的某个值，其中N代表阵列中元素的数量。
    - 如果数据元列表是一个映射表，则索引是映射表中的某个键。
    - 如果找不到索引对应的值，则返回空字符串。
- **示例**
  - 示例一：数据元列表是一个数组

```
"Fn::Select" : ["2", ["foo", " bar", "achoo"]]
```

  - 示例二：数据元列表是一个映射表

```
"Fn::Select" : ["shape", {"shape": "circle", "height": "80"}]
```

  - 示例三：数据元列表是一个CommaDelimitedList

```
"Parameters" : {
 "userParam": {
        "Type": "CommaDelimitedList",
        "Default": "10.0.100.0/24, 10.0.101.0/24, 10.0.102.0/24"
    }
},
"Resources": {
 "resourceUuid": {
        "Properties": {
           "CidrBlock": {"Fn::Select" : ["0", {"Ref":"userParam"}]}
        }
    }
}
```

- **返回值**

  选定的数据元。

  - 示例一：返回"achoo"
  - 示例二：返回"circle"
  - 示例三：返回"10.0.100.0/24"

- **支持的函数**
  - 对于Fn::Select索引值，可在Fn::Select函数中嵌套使用Ref函数。
  - 对于对象的Fn::Select列表，可在Fn::Select函数中嵌套使用以下函数：
    - Fn::Base64
    - Fn::FindInMap
    - Fn::GetAtt
    - Fn::Join
    - Fn::Select
    - Ref

Ref

返回指定参数或资源的值。

- 如果指定参数是resourceUuid，则返回资源的值。
- 否则系统将认为指定参数是参数，将尝试返回参数的值。
- **声明**

```
"Ref":"logicalName"
```

- **参数**
  - logicalName：要引用的资源或参数的逻辑名称。
- **示例**

  若diskOfferingParam被定义为：

```
"diskOfferingParam": {
 "allocatorStrategy": "DefaultPrimaryStorageAllocationStrategy",
      "diskSize": "21474836480",
      "type": "DefaultDiskOfferingType",
      "sorkKey": "0"
}
```

```
"Ref":"diskOfferingParam"
```

- **返回值**

  资源的值或者参数的值。

  本例中，返回diskOfferingParam的值：

```
{
 "allocatorStrategy": "DefaultPrimaryStorageAllocationStrategy",
```

```
        "diskSize": "21474836480",
        "type": "DefaultDiskOfferingType",
        "sorkKey": "0"
}
```

- **支持的函数**

  不能在Ref函数中嵌套使用任何函数。必须指定为资源逻辑UUID的字符串。

8.1.5 映射(Mappings)

定义映射信息表，映射信息是一种多层的Map结构。

- 映射是一个Key-Value映射表。
- 在模板的Resources和Outputs中，可使用内置函数Fn::FindInMap，通过指定Key而获取映射表的Value。

语法

映射由Key-Value键值对组成。

- 其中Key和Value可以为字符串类型或者数字类型。
- 如果声明多个映射，用逗号,分隔开。
- 每个映射的名称不能重复。

示例

代码段示例如下：

```
"Mappings" : {
 "Mapping01" : {
        "Key01" : {
           "Name" : "Value01"
        },
        "Key02" : {
           "Name" : "Value02"
        },
        "Key03" : {
           "Name" : "Value03"
        }
    }
}
```

使用内置函数Fn::FindInMap返回对应的值示例：

```
{
 "ZStackTemplateFormatVersion": "2018-06-18",
      "Parameters": {
         "regionParam": {
            "Description": "选择创建云主机的区域",
            "Type": "String",
            "AllowedValues": ["cn-hangzhou", "cn-shanghai"]
```

```
        }
    },
    "Mappings" : {
        "ImageInRegions" : {
            "cn-hangzhou" : { "32" : "imageUuid-1", "64" : "imageUuid-2" },
            "cn-shanghai" : { "32" : "imageUuid-3", "64" : "imageUuid-4" }
        }
    },
    "Resources": {
        "WebServer": {
            "Type": "ZStack::Resource::VmInstance",
            "Properties": {
                "name" : "test-vm",
                "imageUuid" : {"Fn::FindInMap": ["ImageInRegions", {"Ref":"regionParam"}, "64"]},
                "instanceOfferingUuid": {"Ref":"instanceOfferingUuid"},
                "l3NetworkUuids": [{"Ref":"l3NetworkUuid"}]
            },
            "DeletionPolicy": "Retain"
        }
    }
}
```

8.2 资源索引

创建资源栈模板时，可根据**资源类型(Type)**和**资源属性(Properties)**信息，申明对所需资源的具体要求。

资源编排支持的**资源类型(Type)**包括以下两种：

- Resource类型
- Action类型

8.2.1 Resource类型

表 1: Resource类型资源索引表

Resouce类型	说明
ZStack::Resource::VmInstance	创建云主机*(CreateVmInstance)*
ZStack::Resource::DataVolume	创建云盘*(CreateDataVolume)*
ZStack::Resource::Image	添加镜像*(AddImage)*
ZStack::Resource::RootVolumeTemplate	***从根云盘创建根云盘镜像(CreateRootVolumeTemplateFromRootVolume)***
ZStack::Resource::DataVolumeTemplate	从云盘创建数据云盘镜像*(CreateDataVolumeTemplateFromVolume)*
ZStack::Resource::AffinityGroup	创建亲和组*(CreateAffinityGroup)*

Resouce类型	说明
ZStack::Resource::InstanceOffering	创建云主机规格*(CreateInstanceOffering)*
ZStack::Resource::DiskOffering	创建云盘规格*(CreateDiskOffering)*
ZStack::Resource::L2VxlanNetworkPool	创建*VXLAN*网络池*(CreateL2VxlanNetworkPool)*
ZStack::Resource::L2NoVlanNetwork	创建普通二层网络*(CreateL2NoVlanNetwork)*
ZStack::Resource::L2VlanNetwork	创建二层*VLAN*网络*(CreateL2VlanNetwork)*
ZStack::Resource::L2VxlanNetwork	创建*VXLAN*网络*(CreateL2VxlanNetwork)*
ZStack::Resource::L3Network	创建三层网络*(CreateL3Network)*
ZStack::Resource::VRouterRouteTable	创建云路由路由表*(CreateVRouterRouteTable)*
ZStack::Resource::VpcVRouter	创建*VPC*云路由*(CreateVpcVRouter)*
ZStack::Resource::SecurityGroup	创建安全组*(CreateSecurityGroup)*
ZStack::Resource::SecurityGroupRule	添加规则到安全组*(AddSecurityGroupRule)*
ZStack::Resource::Vip	创建虚拟*IP(CreateVip)*
ZStack::Resource::Eip	创建弹性*IP(CreateEip)*
ZStack::Resource::PortForwardingRule	创建端口转发规则*(CreatePortForwardingRule)*
ZStack::Resource::LoadBalancer	创建负载均衡器*(CreateLoadBalancer)*
ZStack::Resource::LoadBalancerListener	创建负载均衡监听器*(CreateLoadBalancerListener)*
ZStack::Resource::IPsecConnection	创建*IPsec*连接*(CreateIPsecConnection)*
ZStack::Resource::VirtualRouterOffering	创建云路由规格*(CreateVirtualRouterOffering)*
ZStack::Resource::VniRange	创建*Vni Range(CreateVniRange)*

8.2.2 Action类型

表 2: Action类型资源索引表

Action类型	说明
ZStack::Action::AddIpRange	添加*IP*地址范围*(AddIpRange)*
ZStack::Action::AddDnsToL3Network	向三层网络添加*DNS(AddDnsToL3Network)*
ZStack::Action::AddVmToAffinityGroup	添加云主机到亲和组*(AddVmToAffinityGroup)*

Action类型	说明
ZStack::Action::AddVRouterRouteEntry	添加云路由路由条目*(AddVRouterRouteEntry)*
ZStack::Action::AddCertificateToLoadBalancerListener	添加证书到负载均衡*(AddCertificateToLoadBalancerListener)*
ZStack::Action::AddIpRangeByNetworkCidr	通过网络*CIDR*添加*IP*地址范围*(AddIpRangeByNetworkCidr)*
ZStack::Action::AddVmNicToLoadBalancer	添加云主机网卡到负载均衡器*(AddVmNicToLoadBalancer)*
ZStack::Action::AddVmNicToSecurityGroup	添加虚拟机网卡到安全组*(AddVmNicToSecurityGroup)*
ZStack::Action::AddRemoteCidrsToIPsecConnection	添加远端*CIDR*到*IPsec*连接*(AddRemoteCidrsToIPsecConnection)*
ZStack::Action::AttachEip	绑定弹性*IP(AttachEip)*
ZStack::Action::AttachDataVolumeToVm	挂载云盘到云主机上*(AttachDataVolumeToVm)*
ZStack::Action::AttachPortForwardingRule	挂载规则到虚拟机网卡上*(AttachPortForwardingRule)*
ZStack::Action::AttachIsoToVmInstance	加载*ISO*到云主机*(AttachIsoToVmInstance)*
ZStack::Action::AttachPciDeviceToVm	绑定*PCI*设备到云主机*(AttachPciDeviceToVm)*
ZStack::Action::AttachUsbDeviceToVm	云主机挂载所在物理机*USB*设备*(AttachUsbDeviceToVm)*
ZStack::Action::AttachL2NetworkToCluster	挂载二层网络到集群*(AttachL2NetworkToCluster)*
ZStack::Action::AttachL3NetworkToVm	加载*L3*网络到云主机*(AttachL3NetworkToVm)*
ZStack::Action::AttachNetworkServiceToL3Network	挂载网络服务到三层网络*(AttachNetworkServiceToL3Network)*
ZStack::Action::AttachSecurityGroupToL3Network	挂载安全组到*L3*网络*(AttachSecurityGroupToL3Network)*
ZStack::Action::AttachL3NetworksToIPsecConnection	添加三层网络到*IPsec*连接*(AttachL3NetworksToIPsecConnection)*
ZStack::Action::AttachVRouterRouteTableToVRouter	绑定云路由路由表到云路由器*(AttachVRouterRouteTableToVRouter)*

Action类型	说明
ZStack::Action::AddCertificateToLoadBalancerListener	添加证书到负载均衡*(AddCertificateToLoadBalancerListener)*
ZStack::Action::AddHostRouteToL3Network	向三层网络添加主机路由*(AddHostRouteToL3Network)*

术语表

访问密钥（AccessKey）

用于调用阿里云API或大河云联API的唯一凭证，AccessKey包括AccessKeyID（用于标识用户）和AccessKeySecret（用于验证用户密钥）。

数据中心（Data Center）

包含阿里云的地域和可用区等地域资源，用于匹配阿里云资源的地域属性。

地域（Region）

物理的数据中心，划分地区的基本单位，ZStack混合云的地域对应了阿里云端的地域。

可用区（Identity Zone）

在同一地域内，电力和网络互相独立的物理区域，ZStack混合云的可用区对应了阿里云端的可用区（Zone）。

存储空间（Bucket）

用于存储对象（Object）的容器，ZStack使用对象存储（OSS）里的Bucket来上传镜像文件。

ECS云主机（Elastic **Compute** Service）

阿里云端创建的ECS实例，可在ZStack混合云界面进行ECS云主机生命周期的管理。

专有网络VPC（Virtual **Private** Cloud）

用户基于阿里云构建的一个隔离的网络环境，不同的专有网络之间逻辑上彻底隔离。

虚拟交换机（VSwitch）

组成专有网络VPC的基础网络设备，可以连接不同的云产品实例。ZStack混合云的虚拟交换机对应了阿里云VPC下的虚拟交换机。

虚拟路由器（VRouter）

专有网络VPC的枢纽，可以连接专有网络的各个虚拟交换机，同时也是连接专有网络与其它网络的网关设备。ZStack支持查看VPC下的虚拟路由器。

路由表（Route Table）

虚拟路由器上管理路由条目的列表。

路由条目（Route Entry）

路由表中的每一项是一条路由条目。路由条目定义了通向指定目标网段的网络流量的下一跳地址。路由条目包括系统路由和自定义路由两种类型。ZStack支持自定义类型的路由条目。

安全组（Security Group）

针对云主机进行第三层网络的防火墙控制。ZStack混合云的安全组对应了阿里云端ECS云主机三层隔离的防火墙约束。

镜像（Image）

云主机使用的镜像模板文件，一般包括操作系统和预装的软件。ZStack支持上传本地镜像到阿里云，以及使用阿里云端镜像。

弹性公网IP（EIP）

阿里云端公有网络池中的IP地址，绑定弹性公网IP的ECS实例可以直接使用该IP进行公网通信。

VPN连接（VPN Connection）

通过建立点对点的IPsec VPN通道，实现企业本地数据中心的私有网络与阿里云端VPN网络进行通信。

VPN网关（VPN Gateway）

一款基于Internet，通过加密通道将本地数据中心和阿里云专有网络VPC安全可靠连接起来的服务。用户在阿里云VPC创建的IPsec VPN网关，与本地数据中心的用户网关配合使用。

VPN用户网关（Customer Gateway）

本地数据中心的VPN服务网关。可通过ZStack混合云创建VPN用户网关，并将VPN用户网关与VPN网关连接起来。

高速通道（Express Connect）

通过物理专线（即租用运营商的专线：电缆或光纤），连通本地数据中心到阿里云专线接入点，与阿里云VPC环境打通，实现云上云下不同网络间高速，稳定，安全的私网通信。

边界路由器（VBR）

用户申请的物理专线接入交换机的产品映射。用户在物理专线上可以创建边界路由器，边界路由器负责专线上的数据在阿里云上进行转发。通过边界路由器，用户数据可以直达阿里云VPC网络。

路由器接口（Router Interface）

一种虚拟的网络设备，可以挂载在路由器并与其他路由器接口进行高速通道互联，实现不同网络间的内网互通。