

如何在AWS云中添加资源放置脚本？

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[如何在AWS云中添加资源放置脚本？](#)

简介

本文档介绍如何在Amazon Web Services(AWS)云中为CloudCenter 4.8.2添加用户资源放置脚本。

先决条件

要求

Cisco 建议您了解以下主题：

- Linux
- AWS EC2云
- 外壳脚本

使用的组件

本文档中的信息基于以下软件版本：

- CloudCenter版本4.8.2
- CCO(CiscoCloud Orchestrator)
- CCM(CiscoCloud Manager)
- Redhat7或Centos7
- 存储callout脚本的存储库

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

如何在AWS云中添加资源放置脚本？

CloudCenter支持根据用户设置放置资源，但有不同的云。本文档重点介绍在AWS云中设置资源放置标注脚本。

请按照以下步骤放置callout脚本以放置资源

步骤1.登录存储库/Linux VM，在该VM中可以创建用于资源放置的bash脚本。如果在任何Linux VM中创建脚本，请确保将这些文件复制到存储库中，以便可以访问该文件以部署应用。

步骤2. **vi callout.sh**并在脚本中输入内容。

```
#!/bin/bash

. /utils.sh

print_log "$ENV_imageName"

print_log "$Cloud_Setting_CloudFamily"

print_log "$ENV_parentJobName"

content="{\"vpcId\": \"vpc-31e88948\",
\"subnetId\": \"subnet-44f8bb0c\",
\"securityGroupList\": \"sg-0f05b97b\",
\"vmTagsList\": \"Name:RP_001,PayProfile:Dev,User:AdminUser\",
\"assignPublicIp\": \"true\",
\"nodeInfo\": \"VpcID:vpc-31ee2948, subnetId: subnet-44ftb40c,securityGroupList:sg-0f04b97b \"}"

print_ext_service_result "$content"
```

注意：将VPC ID、子网、安全组、vmTagList、node-info替换为您的信息。

所有信息都可从AWS EC2实例中获取

步骤3.将bash脚本保存在存储库中，并将权限更改为**755**。

步骤4. **#chmod 755 callout.sh**

步骤5.创建脚本后，您需要从CloudCenter Manager启用资源放置脚本。

a.登录到CloudCenter Manager GUI > Environment > New Environment。

General Settings

* NAME

Resource_pl

resource Placement

1

SERVICENOW EXTENSION ⓘ

None

APPROVAL REQUIRED TO DEPLOY TO THIS ENVIRONMENT ⓘ

NO

Cloud Selection

* CLOUD REGION / 1 SELECTED

* CLOUD ACCOUNT



AWS_RTP
US East (Virginia)

AWS_RTP

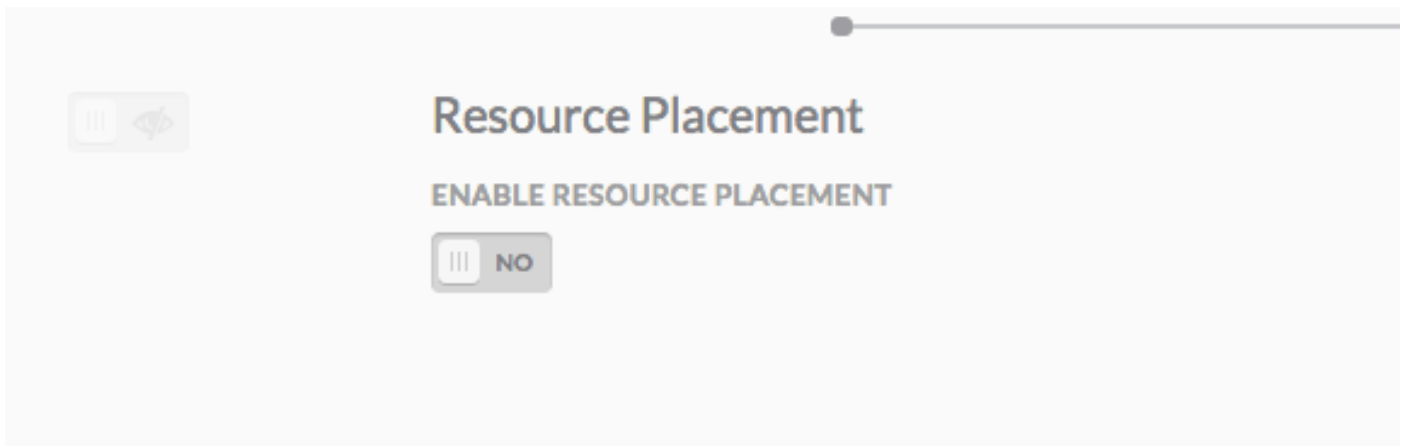
Simplified Networks

USE SIMPLIFIED NETWORKS ⓘ

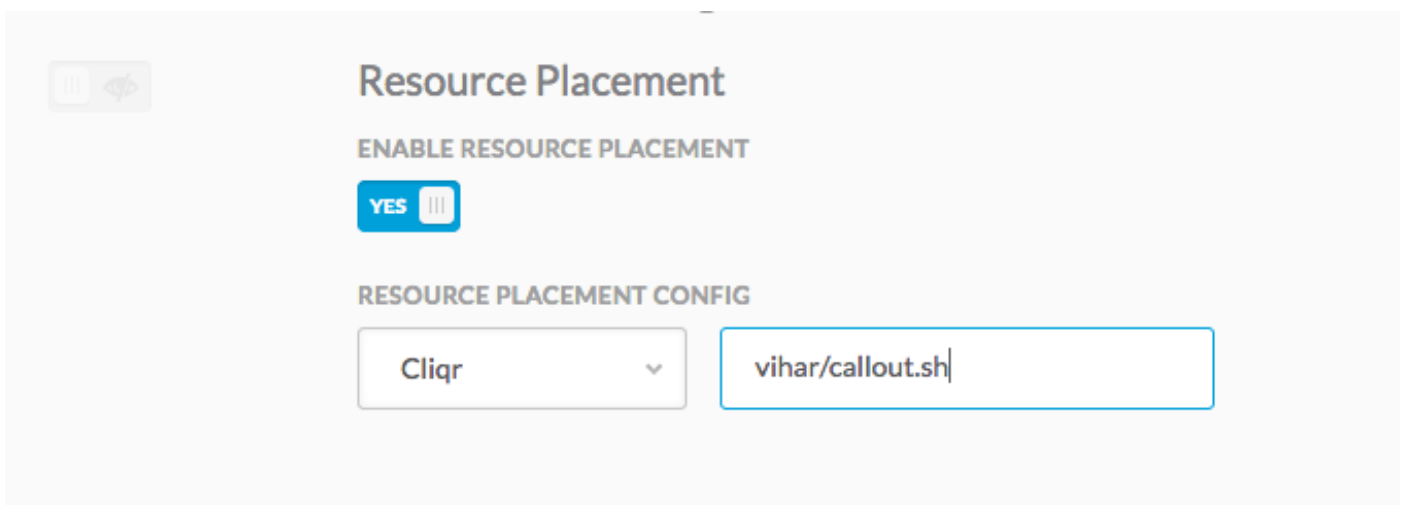
NO

b.单击“定义默认云设置”(DEFINE DEFAULTS CLOUD SETTINGS)选项卡。

c.选择您希望部署环境使用的实例类型并启用资源放置，如图所示。



d.单击“启用资源放置”后，即可获得设置“资源放置配置”文件的选项，如图所示。



注意：提供callout.sh的位置，该文件是您上传到存储库的文件，然后单击两次“完成”选项卡，从“部署环境”页退出。

步骤6.使用新创建的部署环境部署新实例。使用您创建的资源放置脚本成功部署VM。

提示：您可以从CCO检查gateway.log文件，以验证脚本是否成功执行。

这是日志文件中显示的输出。

```
2018-01-09 15:16:14,917 INFO service.LifecycleClusterStartAction [RxCachedThreadScheduler-6] - DeploymentJobID=37 requestNodeCount = 1 , minNodeCount=1 , createdCount = 1
```

```
2018-01-09 15:16:16,121 INFO strategy.DockerBaseCallout [threadPoolExecutor-21] - Output from Callout
```

```
CLIQR_EXTERNAL_SERVICE_LOG_MSG_START^M
```

```
Executing service resourcePlacement action with command:  
"/opt/remoteFiles/cliqr_local_file/callout.sh" from directory: ^M
```

```
CLIQR_EXTERNAL_SERVICE_LOG_MSG_END^M
```

```
CLOUD_CENTER_SCRIPT_OUTPUT^M
```

CLIQR_EXTERNAL_SERVICE_LOG_MSG_START^M

Executing script/command: /opt/remoteFiles/cliqr_local_file/callout.sh.^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_END^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_START^M

Ubuntu 14.04^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_END^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_START^M

^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_END^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_START^M

docker_cluster^M

CLIQR_EXTERNAL_SERVICE_LOG_MSG_END^M

CLIQR_EXTERNAL_SERVICE_RESULT_START^M

{"vpcId": " vpc-31e88948", ^M

"subnetId": " subnet-44f8bb0c ", ^M

"securityGroupList": " sg-0f05b97b ", ^M

"vmTagsList": "Name:RP_001, PayProfile:Dev, User:AdminUser", ^M

"assignPublicIp": "true", ^M

"nodeInfo": "VpcID: vpc-31e88948, subnetId: subnet-44f8bb0c, securityGroupList: sg-0f05b97b " } ^M

CLIQR_EXTERNAL_SERVICE_RESULT_END^M