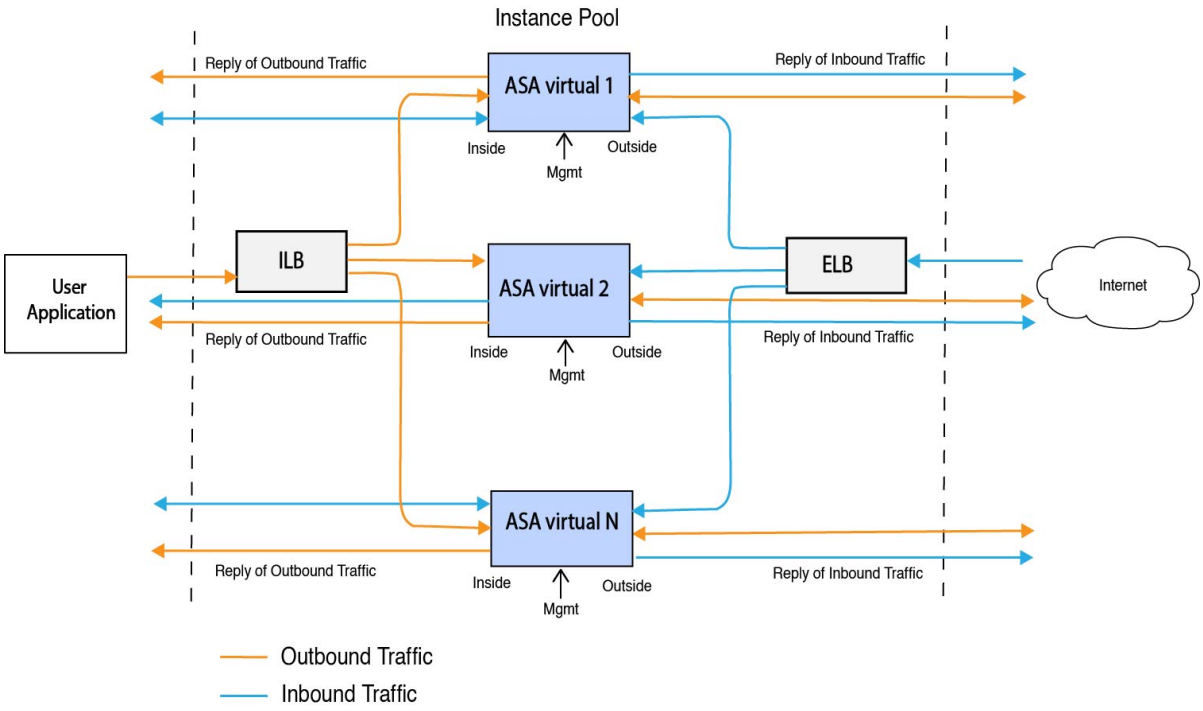
在 OCI 上部署 ASA Virtual Auto Scale 解决方案

使用案例

此 ASA Virtual 的使用案例 - OCI Autoscale 解决方案会显示在解决方案图中。面向互联网的负载均衡器具有使用侦听程序与目标组的组合启用的端口的公共 IP 地址。

图 *1:* 使用案例图



可以为网络流量实施基于端口的明细。这可通过 NAT 规则实现。以下各部分将介绍此配置示例。

前提条件

权限和策略

以下是实施解决方案所需的 OCI 权限和策略：

1. **用户和组**

注释 您必须是 OCI 用户或租户管理员才能创建用户和组。

创建 Oracle 云基础设施用户账户和用户账户所属的组。如果存在具有用户账户的相关组，则无需再进行创建。有关创建用户和组的说明，请参阅创建组和用户。

2. **组策略**

您需要创建策略，然后将其映射到组。要创建策略，请转至 **OCI** > **身份和安全 (Identity & Security)** > **策略 (Policies)** > **创建策略 (Create Policy)**。创建以下策略并将其添加到所需的组中：

- 允许组 <*Group_Name*> 使用隔离专区 <*Compartment_Name*> 中的指标
- 允许组 <*Group_Name*> 管理隔离专区 <*Compartment_Name*> 中的警报

- 允许组 <*Group_Name*> 管理隔离专区 <*Compartment_Name*> 中的主题
- 允许组 <*Group_Name*> 检查隔离专区 <*Compartment_Name*> 中的指标
- 允许组 <*Group_Name*> 读取隔离专区 <*Compartment_Name*> 中的指标
- 允许组 <*Group_Name*> 使用隔离专区 <*Compartment_Name*> 中的标记命名空间
- 允许组 <*Group_Name*> 读取隔离专区 <*Compartment_Name*> 中的日志组
- 允许组 <*Group_Name*> 使用隔离专区 <*Compartment_Name*> 中的实例池
- 允许组 <*Group_Name*> 使用租户中的 Cloud Shell
- 允许组 <*Group_Name*> 读取租户中的对象存储命名空间
- 允许组 <*Group_Name*> 管理租户中的存储库

注释 您也可以在租户级别创建策略。您可以自行决定如何提供所有的权限。

3. **Oracle 功能的权限**

   要让 Oracle 功能能够访问另一个 Oracle 云基础设施资源，请将该功能包含在动态组中，然后创建一个策略以授予该动态组对该资源的访问权限。

4. **创建动态组**

   要创建动态组，请转至 **OCI** > **身份和安全 (Identity & Security)** > **动态组 (Dynamic Group)** > **创建动态组 (Create Dynamic Group)**

   在创建动态组时指定以下规则：

   ```
   ALL {resource.type = 'fnfunc', resource.compartment.id = '<Your_Compartment_OCID>'}
   ```

   有关动态组的更多详细信息，请参阅：

   - https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm
   - https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/managingdynamicgroups.htm

5. **为动态组创建策略**

   要添加策略，请转至 **OCI** > **身份和安全 (Identity & Security)** > **策略 (Policies)** > **创建策略 (Create Policy)**。将以下策略添加到组：

   ```
   允许动态组 <Dynamic_Group_Name> 管理隔离专区 <Compartment_OCID> 中的所有资源
   ```

从 GitHub 下载文件

- OCI Autoscale 解决方案已作为 存储库提供。您可以从存储库中提取或下载文件。

Python3 环境

可以在克隆存储库中找到 *make.py* 文件。此程序会将 Oracle 功能和模板文件压缩为 Zip 文件；将它们复制到目标文件夹。为了执行这些任务，应配置 Python 3 环境。

注释 此 Python 脚本只能用于 Linux 环境中。

基础设施配置

必须配置以下选项：

1. **VCN**

   根据应用的需要创建VCN。创建具有互联网网关的VCN，该网关至少有一个通过到互联网的路由连接的子网。

   有关创建 VCN 的信息，请参阅https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/creatingnetwork.htm。

2. **应用程序子网**

   有关创建子网的信息，请参阅 https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingVCNs_topic-Overview_of_VCNs_and_Subnets.htm#。

3. **外部子网**

   子网应该具有能够通过“0.0.0.0/0”连接互联网网关的路由。此子网包含思科 的外部接口和面向互联网的负载均衡器。确保为出站流量添加 NAT 网关。

   有关详情，请参阅以下文档：

   - https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingIGs.htm
   - https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/NATgateway.htm#To_create_a_NAT_gateway

4. **内部子网**

   这与具有或没有 NAT/互联网网关的应用程序子网类似。

注释 对于 运行状况探测，您可以通过端口 80 来访问元数据服务器 (169.254.169.254)。

5. **管理子网**

   管理子网应是公共子网，这样它才能支持对 的 SSH 可访问性。

6. **安全组 - 实例的网络安全组**

7. **对象存储命名空间**

此对象存储命名空间用于托管静态网站，包含 configuration.txt 文件。您必须为 configuration.txt 文件创建预身份验证请求。此预身份验证 URL 可在模板部署期间使用。

注释 确保 实例可通过 HTTP URL 访问已上传的以下配置。

当 启动时，它会执行以下命令 `$ copy /noconfirm <configuration.txt file's pre-authenticated request URL > disk0:Connfiguration.txt`

此命令支持要使用 configuration.txt 文件配置的 启动。

加密密码

注释 有关此程序的详细信息，请参阅创建保管库和密钥。

的密码用于配置自动扩展时使用的所有 实例，并且它还用于

因此，您需要不时地保存和处理密码。由于密码更改频繁且存在漏洞，因此不允许以纯文本格式编辑或保存密码。密码只能采用加密格式。

要以加密形式获取密码，请执行以下操作：

过程

步骤 1 创建保险库。

OCI 保险库提供安全创建和保存主加密密钥的服务，以及使用它们进行加密和解密的方法。因此，应在与 Autoscale 解决方案的其余部分相同的隔离专区中创建保险柜（如果尚未创建）。

转至 **OCI** > **身份和安全 (Identity & Security)** > **保管库 (Vault)** > **选择或创建新保管库 (Choose or Create New Vault)**

。

步骤 2 创建主加密密钥。

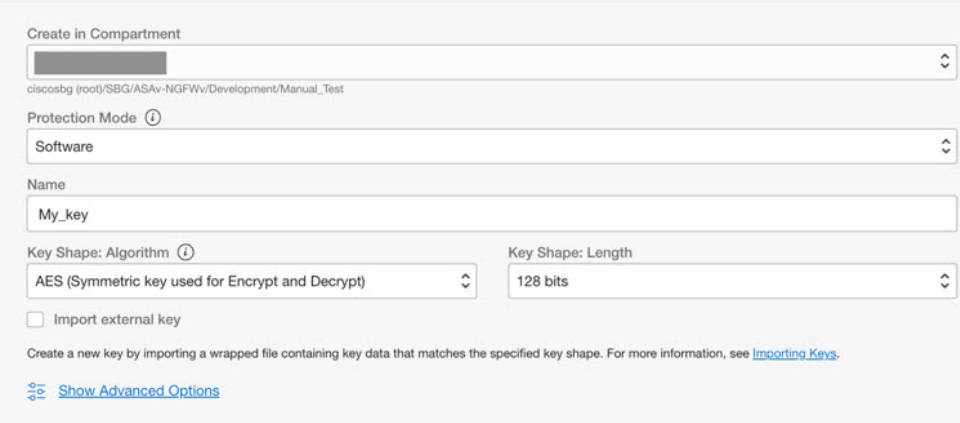
需要使用主加密密钥才能加密纯文本密码。

转至 **OCI** > **身份和安全 (Identity & Security)** > **保管库 (Vault)** > **选择或创建密钥 (Choose or Create Key)**

从任意给定算法中选择任意长度的密钥。

1. AES - 128、192、256
2. RSA - 2048、3072、4096
3. ECDSA - 256、384、521

图 *2:* 创建密钥



步骤 3 创建加密密码。。

1. 转至 **OCI** > **打开 CloudShell（OCI 云终端）(Open CloudShell [OCI Cloud Terminal])**
2. 通过替换 <*Password*> 作为密码来执行以下命令。

   ```
   echo -n '<Password>' | base64
   ```
3. 从选定的保险库中，复制加密终端和主加密密钥 OCID。替换以下值，然后执行 encrypt 命令：
   - 将 KEY_OCID 替换为您的密钥的 OCID
   - 将 Cryptographic_Endpoint_URL 替换为您的保险库的加密终端 URL
   - 将密码替换为您的密码

   加密命令

   ```
   oci kms crypto encrypt --key-id Key_OCID --endpoint
   Cryptographic_Endpoint_URL --plaintext <base64-value-of-password>
   ```
4. 从上述命令的输出中复制密文，然后根据需要使用它们。

准备 ASA 配置文件

确保应用程序已部署或为其制定了部署计划。

过程

步骤 1 在部署之前收集以下输入参数：

参数	数据类型	说明
tenancy_ocid	字符串	您的账户所属的租户的 OCID。要了解如何查找租户 OCID，请参阅此处。 租户 OCID 如下所示 - `ocid1.tenancy.oc1..<unique_ID>`
compartment_id	字符串	要在其中创建资源的隔离专区的 OCID。 示例：`ocid1.compartment.oc1..<unique_ID>`
compartment_name	字符串	隔离专区的名称
region	字符串	要在其中创建资源的区域的唯一标识符。 示例： `us-phoenix-1、us-ashburn-1`
lb_size	字符串	用于确定外部和内部负载均衡器的总预调配带宽（入口加出口）的模板。 支持的值：100Mbps、10Mbps、10Mbps-Micro、400Mbps、8000Mbps 示例：100Mbps
availability_domain	逗号分隔值	示例：Tpeb:PHX-AD-1 **注释** 在 Cloud Shell 中执行 **oci iam availability-domain list** 命令以获取可用性域名。
min_and_max_instance_count	逗号分隔值	您希望在实例池中保留的最小和最大实例数。 示例：1,5
autoscale_group_prefix	字符串	用于对通过使用模板创建的所有资源命名的前缀。例如，如果资源前缀为“autoscale”，则所有资源均按会如下方式命名 - autoscale_resource1、autoscale_resource2 等。

参数	数据类型	说明
asav_config_file_url	URL	上传到对象存储以用于配置 ASA Virtual 的配置文件的 URL。 注释 必须提供配置文件的预身份验证请求 URL 示例： https://objectstorage.<region-name>.oraclecloud.com/<object-storage-name>/oci-asav-configuration.txt
mgmt_subnet_ocid	字符串	要使用的管理子网的 OCID。
inside_subnet_ocid	字符串	要使用的内部子网的 OCID。
outside_subnet_ocid	字符串	要使用的外部子网的 OCID。
mgmt_nsg_ocid	字符串	要使用的管理子网网络安全组的 OCID。
inside_nsg_ocid	字符串	要使用的内部子网网络安全组的 OCID。
outside_nsg_ocid	字符串	要使用的外部子网网络安全组的 OCID。
elb_listener_port	逗号分隔值	外部负载均衡器侦听程序的通信端口列表。 示例： 80
ilb_listener_port	逗号分隔值	内部负载均衡器侦听程序的通信端口列表。 示例： 80
health_check_port	字符串	执行运行状况检查的负载均衡器的后端服务器端口。 示例：8080
instance_shape	字符串	要创建的实例的形状。形状可确定分配给实例的 CPU 数量、内存量和其他资源。 支持的形状：“VM.Standard2.4”和“VM.Standard2.8”

参数	数据类型	说明
lb_bs_policy	字符串	用于内部和外部负载均衡器后端的负载均衡器策略。要了解有关负载均衡器策略工作原理的更多信息，请参阅此处。 支持的值：“ROUND_ROBIN”、“LEAST_CONNECTIONS”、“IP_HASH”
image_name	字符串	用于创建实例配置的市场映像的名称。 默认值：“Cisco ASA virtual firewall (ASAv)” 注释 如果用户想要部署自定义映像，则用户必须配置 custom_image_ocid 参数。
image_version	字符串	要使用的 OCI Marketplace 中可用 ASA Virtual 映像的版本。目前，有 9.15.1.15 和 9.16.1 版本可用。 默认值：“Cisco ASA virtual firewall (ASAv)”
scaling_thresholds	逗号分隔值	用于内向扩展和外向扩展的 CPU 使用率阈值。以逗号分隔输入的形式指定内向扩展和外向扩展阈值。 示例：15,50 其中，15 是内向扩展阈值，50 是外向扩展阈值。
custom_image_ocid	字符串	如果未使用市场映像，用于创建实例配置的自定义映像的 OCID。 注释 custom_image_ocid 是可选参数
asav_password	字符串	ASA Virtual 采用加密形式的密码，用于通过 SSH 连接到 ASA Virtual 配置。有关如何加密密码的说明，请参阅配置指南，或参阅此处。

参数	数据类型	说明
cryptographic_endpoint	字符串	加密终端是用于解密密码的URL。它可以在保险库中找到。
master_encryption_key_id	字符串	用于加密密码的密钥的 OCID。它可以在保险库中找到。
配置文件名称 (Profile Name)		它是OCI中的用户配置文件名称。它可以在用户的配置文件部分下找到。 示例：oracleidentitycloudservice/<user>@<mail>.com
对象存储命名空间		它是在创建租户时创建的唯一标识符。您可以在 **OCI** > **管理 (Administration)** > **租户详细信息 (Tenancy Details)** 中找到此值
授权令牌		这用作 Docker 登录的密码，授权其将 Oracle-Functions 推送到 OCI 容器注册表中。要获取令牌，请转至 **OCI** > **身份 (Identity)** > **用户 (Users)** > **用户详细信息 (User Details)** > **身份验证令牌 (Auth Tokens)** > **生成令牌 (Generate Token)**。

步骤 2 为负载均衡器运行状况探测和访问策略配置对象、许可、NAT 规则。

```
! Default route via outside
route outside 0.0.0.0 0.0.0.0 <Outside Subnet gateway> 2

! Health Check Configuration
object network metadata-server
host 169.254.169.254
object service health-check-port
service tcp destination eq <health-check-port>
object service http-port
service tcp destination eq <traffic port>
route inside 169.254.169.254 255.255.255.255 <Inside Subnet GW> 1

! Health check NAT
nat (outside,inside) source static any interface destination static interface metadata-server service
health-check-port http-port
nat (inside,outside) source static any interface destination static interface metadata-server service
health-check-port http-port

! Outbound NAT
object network inside-subnet
subnet <Inside Subnet> <Inside Subnet Gateway>
object network external-server
host <External Server IP>
nat (inside,outside) source static inside-subnet interface destination static interface external-server


! Inbound NAT
object network outside-subnet
```

```
subnet <Outside Subnet> <Outside Subnet GW>
object network http-server-80
host <Application VM IP>
nat (outside,inside) source static outside-subnet interface destination static interface http-server-80


!
dns domain-lookup outside
DNS server-group DefaultDNS

! License Configuration
call-home
profile license
destination transport-method http
destination address http <URL>
debug menu license 25 production
license smart
feature tier standard
throughput level <Entitlement>
licence smart register idtoken <License token> force
!
```

应在访问策略上允许这些运行状况探测连接和数据平面配置。

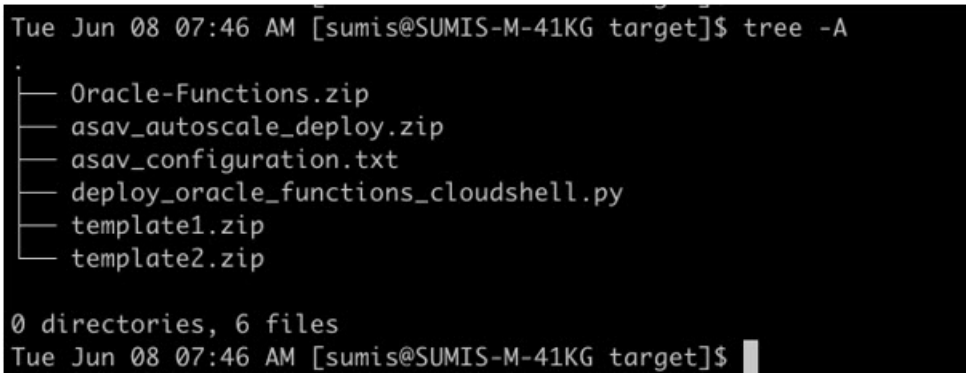
步骤 3 使用配置详细信息更新 *configuration.txt* 文件。

步骤 4 将 *configuration.txt* 文件上传到用户创建的对象存储空间，并为上传的文件创建预身份验证请求。

注释
确保在堆栈部署中使用 configuration.txt 的预身份验证请求 URL。

步骤 5 创建 Zip 文件。

可以在克隆存储库中找到 *make.py* 文件。执行 `python3 make.py build` 命令以创建 zip 文件。目标文件夹包含以下文件。



注释
如果您使用 Cloud Shell 部署自动扩展解决方案，请在执行 `python3 make.py build` 之前更新 *Easy_deploy/deployment_parameters.json* 文件。有关更新，请参阅输入参数的收集和部署 Oracle 功能部署。。

部署 Auto Scale 解决方案

在完成部署的必备步骤后，开始创建 OCI 堆栈。您可以使用 Cloud Shell 执行手动部署或。您的版本的部署脚本和模板可从 存储库获取。

手动部署

端到端 Autoscale 解决方案部署包括三个步骤：部署 Terraform 模板 1 堆栈、部署 Oracle 功能，然后部署 Terraform 模板 2。

部署 Terraform Template-1 堆栈

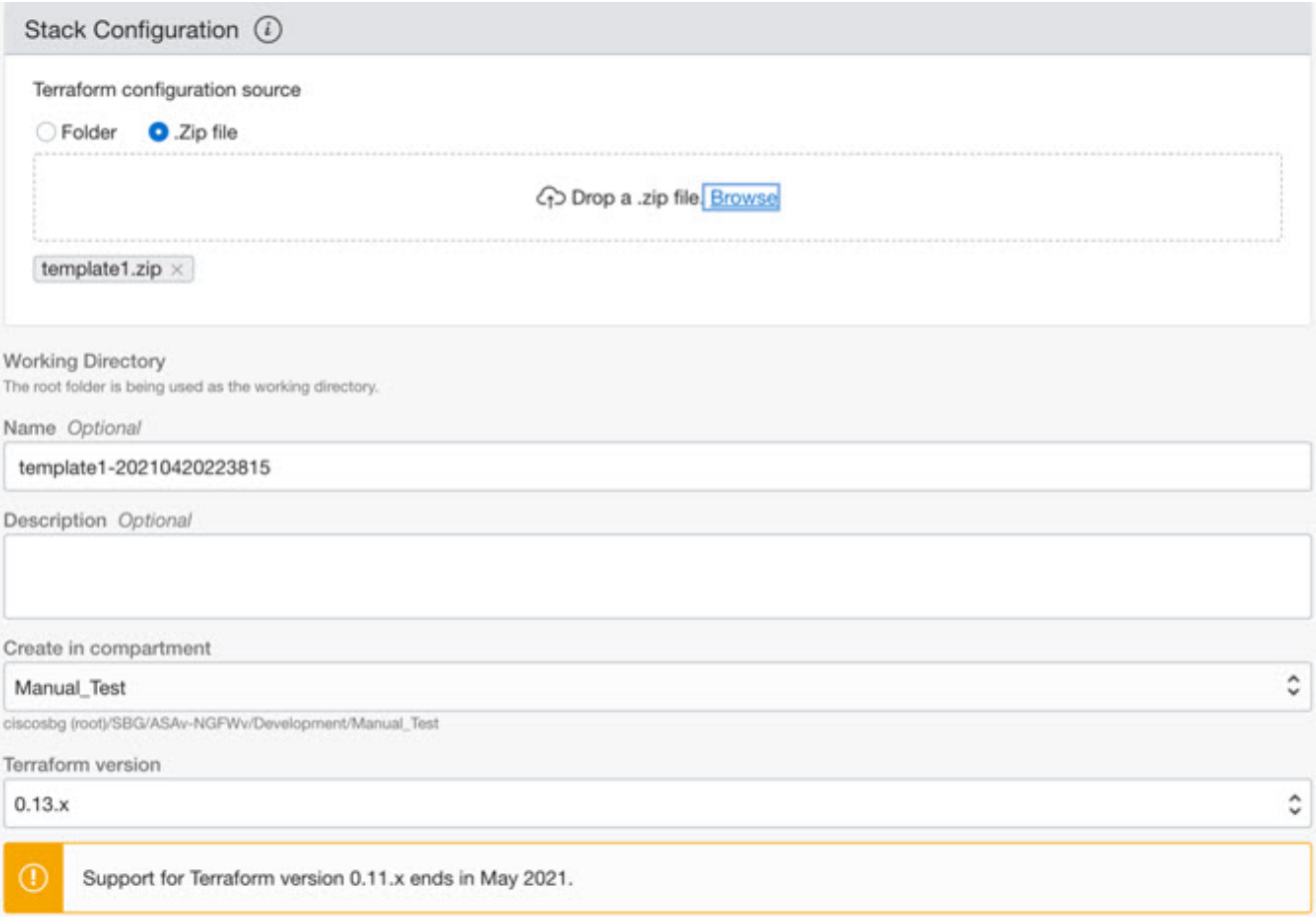
过程

步骤 1 登录 OCI 门户。

区域显示在屏幕的右上角。确保您在预期的区域内。

步骤 2 选择**开发人员服务 (Developer Service)** > **资源管理器 (Resource Manager)** > **堆栈 (Stack)** > **创建堆栈 (Create Stack)**

选择**我的配置 (My Configuration)**，然后选择目标文件夹中的 *Terraform template1.zip* 文件作为 Terraform 配置源，如下图所示。



步骤 3 在**转换版本 (Transform version)** 下拉列表中，选择 0.13.x 或 0.14.x。

步骤 4 在下一步中，输入 中收集的所有详细信息。

注释
输入有效的输入参数，否则堆栈部署可能会在后续步骤中失败。

步骤 5 在下一步中，选择 **Terraform 操作 (Terraform Actions)** > **应用 (Apply)**。

成功部署后，继续部署 Oracle 功能。

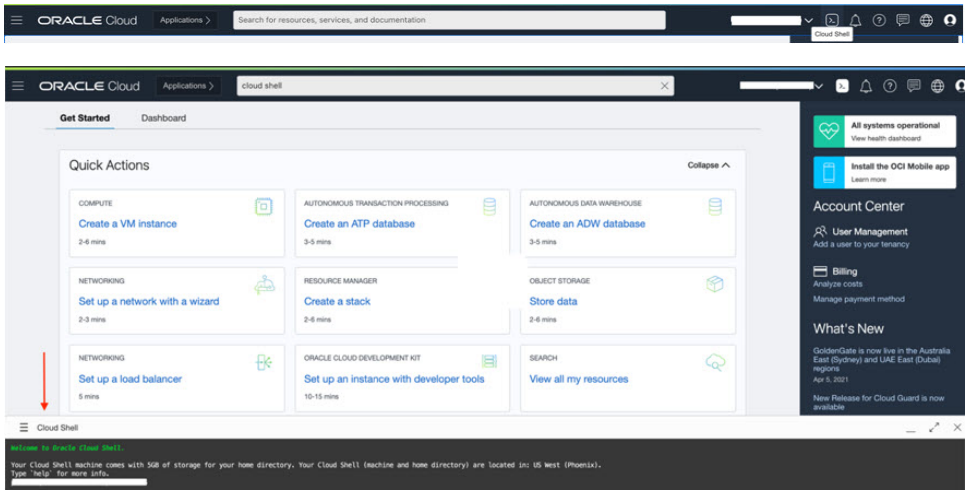
部署 Oracle 功能

注释 只有在 *Terraform Template-1* 部署成功后才能执行此步骤。

在 OCI 中，Oracle 功能会作为 Docker 映像上传，并会保存到 OCI 容器注册表中。在部署时，需要将 Oracle 功能推送到其中一个 OCI 应用（在 Terraform Template-1 中创建）中。

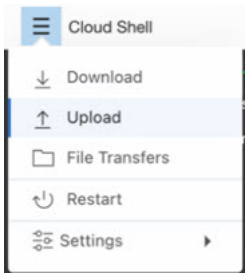
过程

步骤 1 打开 OCI Cloud Shell。

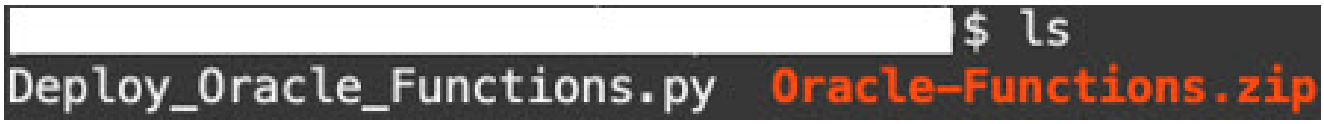


步骤 2 上传 *deploy_oracle_functions_cloudshell.py* 和 *Oracle-Functions.zip*。

从 Cloud Shell 的汉堡菜单中，选择**上传 (Upload)**。



步骤 3 使用 **ls** 命令来验证文件。



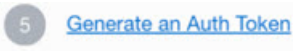
步骤 4 运行 `python3 Deploy_Oracle_Functions.py -h`。`deploy_oracle_functions_cloudshell.py` 脚本需要一些输入参数，可使用 help 参数找到其详细信息，如下图所示。

```
                                $ python3 Deploy_Oracle_Functions.py -h
usage: Deploy_Oracle_Functions.py [-h] -a  -r  -p  -c  -o  -t

*** Script to deploy Oracle Function for OCI ASAv Autoscale Solution ***

Instruction to find values of required arguments:
Application Name: Name of Application created by first Terraform Template
Region Identifier: OCI -> Administration -> Region Management
Profile Name: OCI -> Profile
Compartment OCID: OCI -> Identity -> Compartment -> Compartment Details
Object Storage Namespace: OCI -> Administration -> Tenancy Details
Authorization Token: OCI -> Identity -> Users -> User Details -> Auth Tokens -> Generate Token

optional arguments:
  -h, --help  show this help message and exit
  -a          Name of Application in OCI to which functions will be deployed
  -r          Region Identifier
  -p          Profile Name of User
  -c          Compartment OCID
  -o          Object Storage Namespace
  -t          Authorization Token for Docker Login (*Please Put in Quotes)
```

要运行脚本，请传递以下参数：

表 *1:* 参数和详细信息

参数	详细说明
应用名称	它是 Terraform Template-1 部署创建的 OCI 应用的名称。通过将 Template-1 中给出的“**autoscale_group_prefix**”和后缀“**_application**”组合在一起即可获得其值。
区域标识符	区域标识符是在不同区域的 OCI 中固定的区域代码字。 示例：表示凤凰城的“us-phoenix-1”或表示墨尔本的“ap-melbourne-1”。 要获取所有区域及其区域标识符的列表，请转至 **OCI** > **管理 (Administration)** > **区域管理 (Region Management)**。
配置文件名称	它是 OCI 中的简单用户配置文件名称。 示例：*oracleidentitycloudservice/<user>@<mail>.com* 该名称可以在用户的配置文件部分下找到。
隔离专区 **OCID**	它是隔离专区的 OCID（Oracle 云标识符）。用户拥有 OCI 应用的隔离专区 OCID。 转至 **OCI** > **身份 (Identity)** > **隔离专区 (Compartment)** > **隔离专区详细信息 (Compartment Details)**。
对象存储命名空间	它是在创建租户时创建的唯一标识符。 转至 **OCI** > **管理 (Administration)** > **租户详细信息 (Tenancy Details)**。

参数	详细说明
授权令牌	这用作 Docker 登录的密码，授权其将 Oracle-Functions 推送到 OCI 容器注册表中。在部署脚本中用引号指定令牌。 转至 **OCI** > **身份 (Identity)** > **用户 (Users)** > **用户详细信息 (User Details)** > **身份验证令牌 (Auth Tokens)** > **生成令牌 (Generate Token)**。 出于某种原因，如果您无法查看用户详细信息，请点击**开发人员服务 (Developer services)** > **功能 (Functions)**。转至 Terraform Template-1 创建的应用。点击**开始 (Getting Started)**，然后选择 Cloud Shell 设置，在这些步骤中，您将找到生成身份验证令牌的链接，如下所示。 

步骤 5 通过传递有效的输入参数运行 `python3 Deploy_Oracle_Functions.py` 命令。部署所有的功能需要一些时间。然后，您可以删除该文件并关闭 Cloud Shell。

部署 Terraform Template-2

模板 2 部署与警报创建相关的资源，包括警报、用于调用函数的 ONS 主题。模板 2 的部署与 Terraform Template-1 的部署类似。

过程

步骤 1 登录 OCI 门户。

区域显示在屏幕的右上角。确保您在预期的区域内。

步骤 2 选择**开发人员服务 (Developer Service)** > **资源管理器 (Resource Manager)** > **堆栈 (Stack)** > **创建堆栈 (Create Stack)**.

选择目标文件夹中的 *Terraform template template2.zip* 作为 Terraform 配置的源。

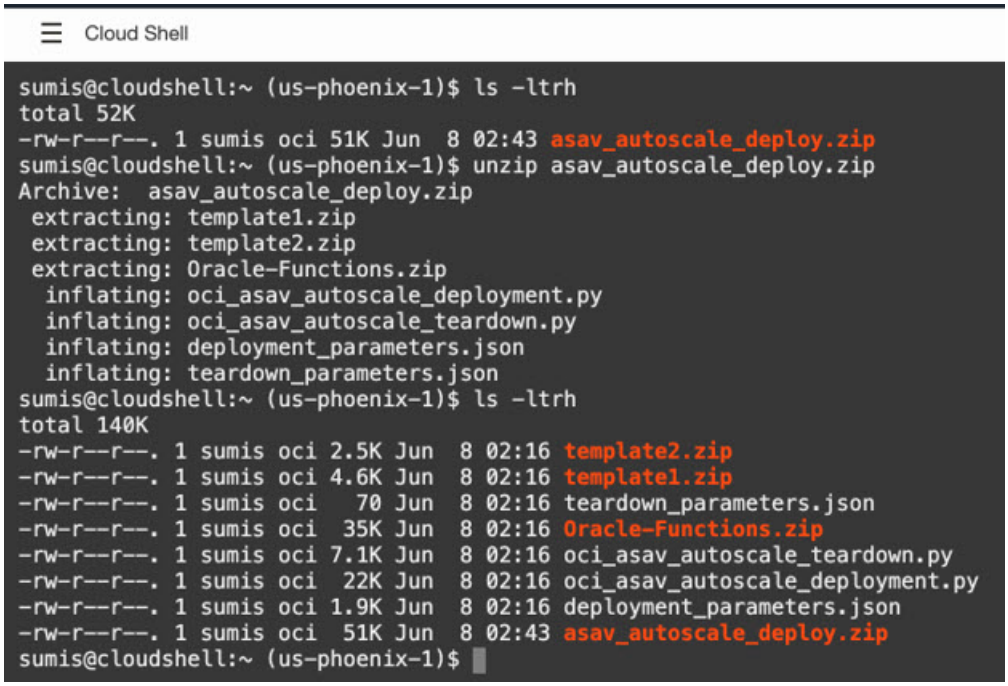
步骤 3 在下一步中，点击**Terraform 操作 (Terraform Actions)** > **应用 (Apply)**。

使用 Cloud Shell 部署 Autoscale

为避免部署开销，您可以通过调用简单的端到端部署脚本来部署 AutoScale 解决方案（Terraform template1、template2 和 oracle 功能）。

过程

步骤 1 将目标文件夹中的 *asav_autoscale_deploy.zip* 文件上传到 Cloud Shell 并提取文件。



步骤 2 在执行 *python3 make.py* 构建命令之前，请确保您已更新 `deployment_parameters.json` 中的输入参数。

步骤 3 要启动 Autoscale 解决方案部署，请在 Cloud Shell 上运行 `python3 oci_asav_autoscale_deployment.py` 命令。

完成解决方案部署大约需要 10-15 分钟。

如果在解决方案部署过程中出现任何错误，则错误日志会被保存。

验证部署

验证是否已部署所有资源，并且 Oracle 功能是否已与警报和事件连接。默认情况下，实例池的最小和最大实例数为零。您可以使用所需的最小和最大数量在 OCI UI 中编辑实例池。这将触发新的 实例。

我们建议您仅启动一个实例并检查其工作流程，并验证其行为以确保符合预期。完成验证后，您可以部署 的实际要求。

注释 将 实例的最小数量指定为**受扩展保护 (Scale-In protected)**，以避免被 OCI 扩展策略删除。

升级

升级 Autoscale 堆栈

此版本不支持升级。应重新部署堆栈。

升级 VM

此版本不支持升级 VM。应使用所需的 映像来重新部署堆栈。

实例池

1. 要更改实例池中的最小和最大实例数，请执行以下操作：

   点击**开发人员服务** > **功能** > **应用名称（通过 Terraform Template-1 创建）** > **配置**。

   分别更改 min_instance_count 和 max_instance_count。

2. 删除/终止实例不等于内向扩展。如果实例池中的任何实例因外部操作而并非内向扩展操作而被删除/终止，则实例池会自动启动新实例进行恢复。

3. Max_instance _count 定义外向扩展操作的阈值限制，但可以通过 UI 更改实例池的实例计数来超过此限制。确保 UI 中的实例计数小于在 OCI 应用中设置的 max_instance_count。否则，请相应地增大阈值。

4. 直接从应用减少实例池中的实例计数不会执行以编程方式设置的清理操作。由于这些后端不会从两个负载均衡器中耗尽和删除，如果 有许可证，它将丢失。

5. 由于某些原因，如果 实例在一段时间内运行状况不佳、无响应且无法通过 SSH 访问，则实例会被强制从实例池中删除，任何许可证都可能丢失。

Oracle 功能

- Oracle 功能实际上就是 Docker 映像。这些映像会别保存到 OCI 容器注册表的根目录中。这些映像不应被删除，否则也会删除在 Autoscale 解决方案中使用的功能。
- 通过 Terraform Template-1 创建的 OCI 应用包含 Oracle 功能正常工作所需的关键环境变量。除非强制要求，否则不应更改这些环境变量的值和格式。所做的任何更改只会反映在新实例中。

负载均衡器后端集

在 OCI 中，仅支持使用配置为 中的管理接口的主接口来连接到实例池的负载均衡器。因此，内部接口会连接到内部负载均衡器的后端集；外部接口会连接到外部负载均衡器的后端集。这些 IP 不会自动添加到后端集或从后端集中删除。我们的 Auto Scale 解决方案会以编程方式处理这两个任务。但在进行任何外部操作、维护或故障排除时，可能会有需要手动完成此操作的情况。

根据要求，可以使用侦听程序和后端集在负载均衡器上打开更多端口。即将启用的实例 IP 会被自动添加到后端集中，但应手动添加现有实例 IP。

在负载均衡器中添加侦听程序

要在负载均衡器中添加某个端口作为侦听程序，请转至**OCI** > **网络 (Networking)** > **负载均衡器 (Load Balancer)** > **侦听程序 (Listener)** > **创建侦听程序 (Create Listener)**。

将后端注册到后端集

要将 实例注册到负载均衡器，应将 实例外部接口 IP 配置为外部负载均衡器后端集中的后端。内部接口 IP 应配置为内部负载均衡器后端集中的后端。确保您正在使用的端口已被添加到侦听程序中。

从 OCI 中删除 Autoscale 配置

可以使用 OCI 中的资源管理器以相同的方式删除使用 Terraform 部署的堆栈。删除堆栈会删除其创建的所有资源，并且与这些资源关联的所有信息都会被永久删除。

注释 在堆栈删除的情况下，建议将实例池中的最小实例数设置为 0，然后等待实例终止。这样将有助于删除所有实例，并且不会留下任何残留。

您可以执行手动删除或使用 Cloud Shell。

手动删除

删除端到端 Auto Scale 解决方案包括三个步骤：删除 Terraform 模板 2 堆栈、删除 Oracle 功能，然后是删除 Terraform 模板 1 堆栈。

删除 Terraform Template-2 堆栈

要删除 Autoscale 配置，您必须先删除 Terraform Template-2 堆栈。

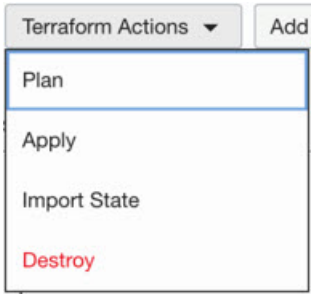
过程

步骤 1 登录 OCI 门户。

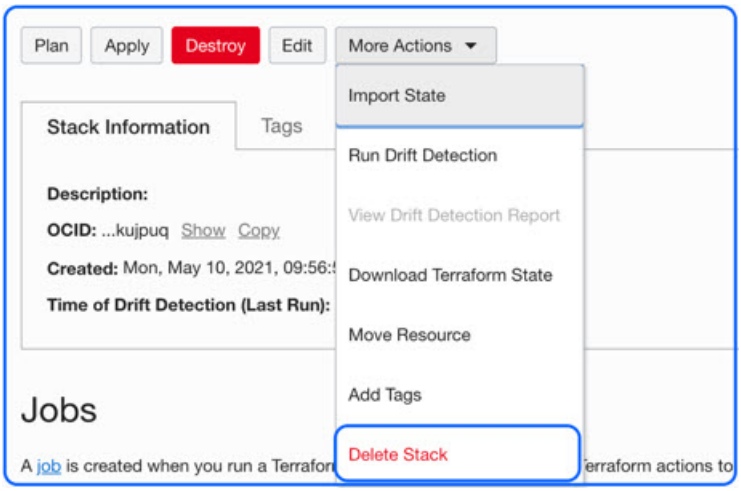
区域显示在屏幕的右上角。确保您在预期的区域内。

步骤 2 选择**开发人员服务 (Developer Services)** > **资源管理器 (Resource Manager)** > **堆栈 (Stack)**。

步骤 3 选择 Terraform Template-2 创建的堆栈，然后选择**Terraform 操作 (Terraform Actions)** 下拉菜单中的**销毁 (Destroy)**，如图所示。



将创建销毁作业，逐个删除资源需要一些时间。您可以在销毁作业完成后删除堆栈。如下图所示：



步骤 4 继续删除 Oracle 功能。

删除 Oracle 功能

Oracle 功能部署不是 Terraform 模板堆栈部署的一部分，它要使用 Cloud Shell 单独上传。因此，Terraform 堆栈删除也不支持其删除。您必须删除通过 Terraform Template-1 创建的 OCI 应用内的所有 Oracle 函数。

过程

步骤 1 登录 OCI 门户。

区域显示在屏幕的右上角。确保您在预期的区域内。

步骤 2 选择**开发人员服务 (Developer Services)** > **功能 (Functions)**。选择在模板 1 堆栈中创建的应用名称。

步骤 3 在此应用中，访问每个函数并将其删除。

删除 Terraform Template-1 堆栈

注释 只有在删除所有 Oracle 功能之后，才能成功删除模板 1 堆栈。

与 Terraform Template-2 删除相同。

过程

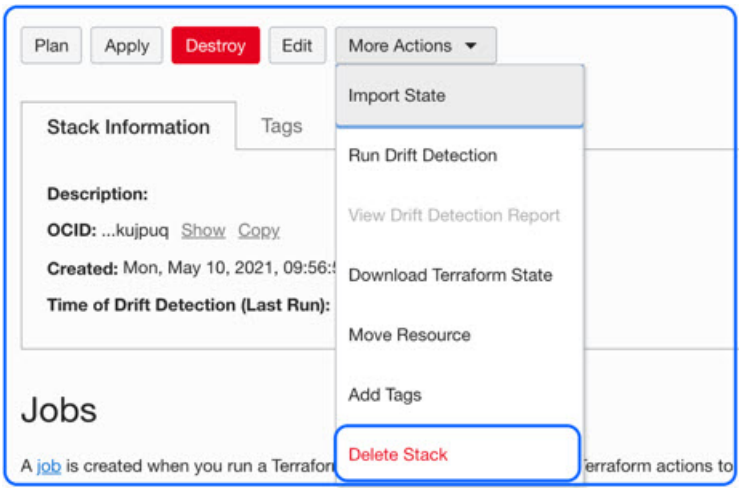
步骤 1 登录 OCI 门户。

区域显示在屏幕的右上角。确保您在预期的区域内。

步骤 2 选择**开发人员服务 (Developer Services)** > **资源管理器 (Resource Manager)** > **堆栈 (Stack)**。

步骤 3 选择 Terraform Template-2 创建的堆栈，然后点击 **Terraform 操作 (Terraform Actions)** 下拉菜单中的**销毁 (Destroy)**。系统将创建销毁作业，逐个删除资源需要一些时间。

步骤 4 销毁作业完成后，您可以从**更多操作 (More Actions)** 下拉菜单中删除堆栈，如下图所示。



成功删除 Terraform Template-1 堆栈后，您必须验证是否所有资源均已删除，并且没有任何类型的残留。

使用 Cloud Shell 来删除 Autoscale

用户可以在 Cloud Shell 中执行 命令，一般使用脚本删除堆栈和 Oracle 功能。如果堆栈是手动部署的，请更新 stack1 和 stack2 的堆栈 ID，然后更新 *teardown_parameters.json* 文件中的应用 ID。

当地语言翻译版本说明

思科可能会在某些地方提供本内容的当地语言翻译版本。请注意，翻译版本仅供参考，如有任何不一致之处，以本内容的英文版本为准。