在 Microsoft Azure 上部署 ASA Virtual Auto Scale 解决方案

适用于 Azure 上的 ASA 虚拟的 Auto Scale 解决方案

概述

Auto Scale 解决方案支持资源分配，以满足性能要求并降低成本。如果资源需求增加，系统将确保根据需要分配资源。如果资源需求减少，则会取消分配资源以降低成本。

ASA virtual Auto Scale for Azure 是完整的无服务器实现，它利用 Azure 提供的无服务器基础架构（逻辑应用、Azure 函数、负载均衡器、安全组、虚拟机规模集等）。

ASA virtual Auto Scale for Azure 实现的一些主要功能包括：

- 基于 Azure Resource Manager (ARM) 模板的部署。
- 支持基于 CPU 的扩展指标。

注释 有关详细信息，请参阅Auto Scale 逻辑， 第 29 页。

- 支持 ASA virtual 部署和多可用性区域。
- 完全自动化配置会自动应用于横向扩展 ASA virtual实例。
- 对负载均衡器和多可用性区域的支持。
- 支持启用和禁用 Auto Scale 功能。
- 思科提供 Auto Scale for Azure 部署包以方便部署。

Azure 上的 ASA virtual Auto Scale 解决方案支持两种使用不同拓扑配置的使用案例：

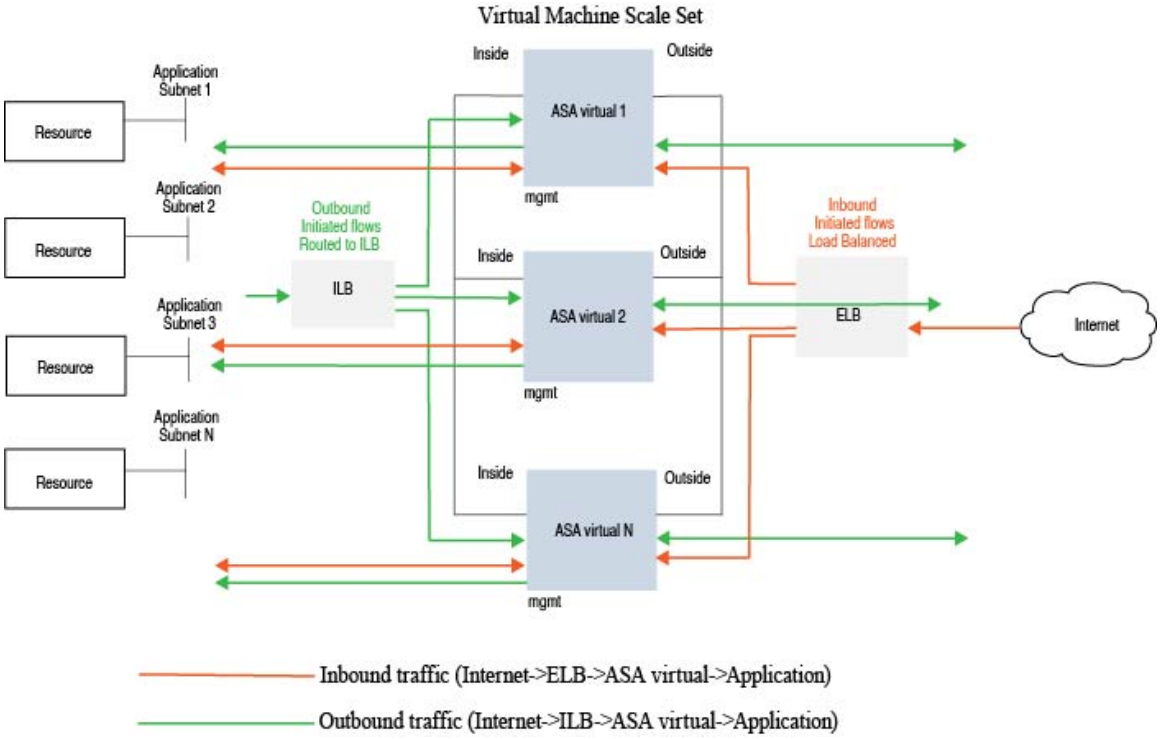
- 使用三明治拓扑的 Auto Scale – 它将 ASA virtual 规模集置于 Azure 内部负载均衡器 (ILB) 与 Azure 外部负载均衡器 (ELB) 之间。
- 使用 Azure 网关负载均衡器 (GWLB) 的 Auto Scale - Azure GWLB 与安全防火墙、公共负载均衡器和内部服务器集成，以简化防火墙的部署、管理和扩展。

使用三明治拓扑的 Auto Scale 使用案例

ASA virtual Auto Scale for Azure 是一种自动化水平扩展解决方案，它将 ASA virtual规模集置于 Azure 内部负载均衡器 (ILB) 与 Azure 外部负载均衡器 (ELB) 之间。

- ELB 将流量从互联网分发到规模集中的 ASA virtual实例；然后，防火墙将流量转发到应用程序。
- ILB 将出站互联网流量从应用程序分发到规模集中的 ASA virtual实例；然后，防火墙将流量转发到互联网。
- 网络数据包决不会在一个连接中同时穿过（内部和外部）负载均衡器。
- 规模集中的 ASA virtual实例数将根据负载条件自动进行扩展和配置。

图 *1:* 使用三明治拓扑的 ***ASA Virtual Auto Scale*** 使用案例



Auto Scale 与 Azure 网关负载均衡器使用案例

Azure 网关负载均衡器 (GWLB) 可确保安全防火墙检查进出 Azure VM（例如应用服务器）的互联网流量，而无需更改任何路由。Azure GWLB 与安全防火墙的集成简化了防火墙的部署、管理和扩展。这种集成还降低了操作复杂性，并为防火墙上的流量提供了单一的入口和出口点。应用和基础设施可以保持源 IP 地址的可视性，而这在某些环境中至关重要。

在 Azure GWLB Auto Scale 使用案例中，ASA virtual 只会使用两个接口：管理接口和一个数据接口。

注释

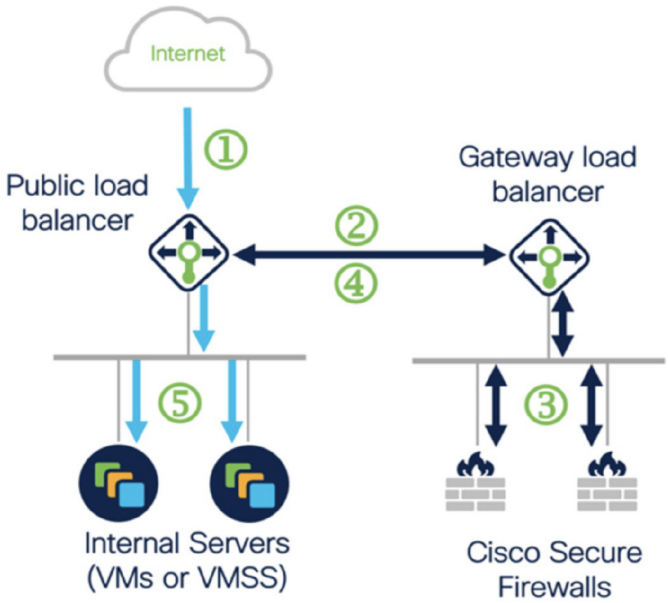
- 如果要部署 Azure GWLB，则不需要网络地址转换 (NAT)。
- 仅支持 IPv4。

许可

支持 BYOL。

入站流量使用案例和拓扑

下图显示了入站流量的流量。



① Inbound flow uses public IP of public load balancer

② Flow is forwarded transparently from the public load balancer to the gateway load balancer

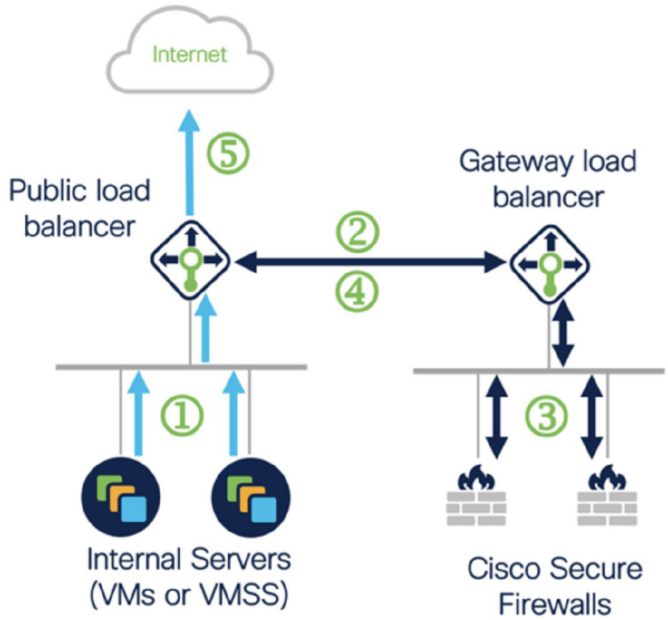
③ Flow is inspected by a firewall and returned to the gateway load balancer

④ Flow is returned to the public load balancer

⑤ Flow is forwarded to an internal server

出站流量使用案例和拓扑

下图显示了出站流量的流量。



① Outbound flow leaves the internal server

② Flow is forwarded transparently from the public load balancer to the gateway load balancer

③ Flow is inspected by a firewall and returned to the gateway load balancer

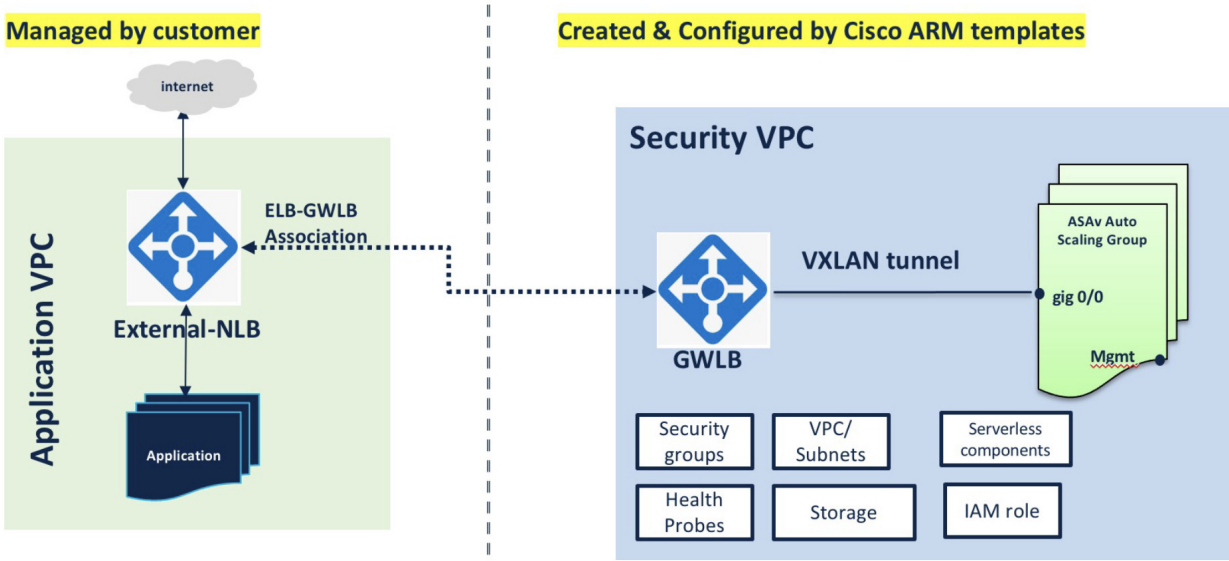
④ Flow is returned to the public load balancer

⑤ Flow is forwarded to the Internet by the public load balancer

应用 VPC 和安全 VPC 之间的流量

在下图中，流量从现有拓扑重定向至防火墙，以便由外部负载均衡器进行检查。然后，流量将被路由到新创建的 GWLB。路由到 ELB 的任何流量都会别转发到 GWLB。

然后，GWLB 将 VXLAN 封装的流量转发到 ASA virtual 实例。您必须创建两个 ASA virtual 关联，因为 GWLB 会对入口和出口流量使用两个单独的 VXLAN 隧道。ASA virtual 会解封装 VXLAN 封装的流量，对其进行检查，然后将流量路由到 GWLB。然后，GWLB 将流量转发到 ELB。



适用范围

本文档介绍部署 ASA virtual Auto Scale for Azure 解决方案的无服务器组件的详细步骤。

重要事项

- 请先阅读整个文档，然后再开始部署。
- 在开始部署之前，请确保满足前提条件。
- 请确保遵守此处所述的步骤和执行顺序。

下载部署软件包

面向 Azure 的 ASA virtual Auto Scale 解决方案是一个基于 Azure 资源管理器 (ARM) 模板的部署，它会利用 Azure 提供的无服务器基础设施（逻辑应用、Azure 函数、负载均衡器、虚拟机扩展设置等）。

下载启动面向 Azure 的 ASA virtual Auto Scale 解决方案所需的文件。您的版本的部署脚本和模板可从 GitHub 存储库获取。

注意 请注意，Cisco 提供的自动扩展部署脚本和模板作为开源示例提供，不在常规 Cisco TAC 支持范围内。定期检查 GitHub 以了解更新和自述文件说明。

有关如何构建 *ASM_Function.zip* 包的说明，请参阅通过源代码构建 Azure 函数，第 32 页。

Auto Scale 解决方案组件

以下组件构成了 ASA virtual Auto Scale for Azure 解决方案。

Azure Functions（函数应用）

函数应用是一组 Azure 函数。基本功能包括：

- 定期交流/探测 Azure 指标。
- 监控 ASA virtual 负载和触发内向扩展/外向扩展操作。

这些函数以压缩 Zip 包的形式提供（请参阅构建 Azure 函数应用包，第 10 页）。这些函数尽可能离散以执行特定任务，可以根据需要进行升级，以提供增强功能和新版本支持。

Orchestrator（逻辑应用）

Auto Scale 逻辑应用是一个工作流，即按照一定序列的步骤集合。Azure 函数是独立的实体，无法彼此通信。此协调器按顺序排列这些函数的执行，并在它们之间交换信息。

- 逻辑应用可用于编排 Auto Scale Azure 函数并在函数之间传递信息。
- 每个步骤代表一个 Auto Scale Azure 函数或内置标准逻辑。
- 逻辑应用作为 JSON 文件交付。
- 可以通过 GUI 或 JSON 文件自定义逻辑应用。

虚拟机规模集 (VMSS)

VMSS 是同构虚拟机（如 ASA virtual 设备）的集合。

- VMSS 可以向集合中添加新的相同虚拟机。
- 添加到 VMSS 的新虚拟机将自动与负载均衡器、安全组和网络接口连接。
- VMSS 具有内置 Auto Scale 功能，该功能对适用于 Azure 的 ASA virtual 禁用。
- 您不应在 VMSS 中手动添加或删除 ASA virtual 实例。

Azure Resource Manager (ARM) 模板

ARM 模板用于部署 ASA virtual Auto Scale for Azure 解决方案所需的资源。

ASA 虚拟 Auto Scale for Azure - ARM 模板**azure_asav_autoscale.json**为 Auto Scale Manager 组件提供输入，包括以下组件：

- Azure 函数应用
- Azure 逻辑应用
- 虚拟机规模集 (VMSS)
- 内部/外部负载均衡器。
- 部署所需的安全组和其他各种组件。

ASA 虚拟 Auto Scale with Azure GWLB - ARM 模板**azure_asav_autoscale_with_GWLB.json**为 Auto Scale Manager 组件提供输入，包括以下组件：

- Azure 函数应用
- Azure 逻辑应用
- 虚拟机 (VM) 或虚拟机规模集 (VMSS)
- 网络基础设施
- 网关负载均衡器
- 部署所需的安全组和其他各种组件

重要事项 ARM 模板在验证用户输入方面有限制，因此您需要在部署过程中负责验证输入。

前提条件

Azure 资源

资源组

部署此解决方案的所有组件需要一个现有的或新创建的资源组。

注释 记录资源组名称、创建它的区域，以及供以后使用的 Azure 订用 ID。

网络

确保虚拟网络可用或已创建。使用三明治拓扑的 Auto Scale 部署不会创建、更改或管理任何网络资源。 但请注意，使用 Azure GWLB 进行 Auto Scale 部署会创建网络基础设施。

ASA virtual 需要三个网络接口，因此您的虚拟网络需要三个子网以用于：

1. 管理流量
2. 内部流量
3. 外部流量

应在子网所连接的网络安全组中打开以下端口：

- SSH(TCP/22)

  负载均衡器与 ASA virtual 之间的运行状况探测所必需。

  无服务器函数与 ASA virtual 之间的通信所必需。

- 应用程序特定协议/端口

  任何用户应用程序所必需（例如，TCP/80 等）。

注释 记录虚拟网络名称、虚拟网络 CIDR、所有 3 个子网的名称，以及外部和内部子网的网关 IP 地址。

准备 ASA 配置文件

准备 ASA virtual配置文件并存储在 ASA virtual实例可访问的 http/https 服务器中。这是标准 ASA 配置文件格式。外向扩展的 ASA virtual将下载此文件并更新其配置。

ASA 配置文件应至少包含以下内容：

- 为所有接口设置 DHCP IP 分配。
- GigabitEthernet0/1 应为“内部”接口。
- GigabitEthernet0/0 应为“外部”接口。

注释 使用三明治拓扑的 Auto Sacle 部署需要两个数据接口。但是，使用 Azure GWLB 的 Auto Scale 部署只需要一个数据接口。

- 将网关设置为内部和外部接口。
- 在 Azure 实用程序 IP 的内部和外部接口上启用 SSH（用于运行状况探测）。
- 创建 NAT 配置以便将流量从外部转发到内部接口。
- 创建访问策略以允许所需流量。
- 许可配置。不支持 PAYG 计费。

注释 无需专门配置管理接口。

以下是 ASA virtual Auto Scale for Azure 解决方案的 ASA 配置文件示例。

```
ASA Version 9.13(1)
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address dhcp setroute
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address dhcp setroute
!
route outside 0.0.0.0 0.0.0.0 10.12.3.1 2
!
route inside 0.0.0.0 0.0.0.0 10.12.2.1 3
!
ssh 168.63.129.0 255.255.255.0 outside
!
ssh 168.63.129.0 255.255.255.0 inside
!
object network webserver
host 10.12.2.5
object service myport
service tcp source range 1 65535 destination range 1 65535
access-list outowebaccess extended permit object myport any any log disable
access-group outowebaccess in interface outside
object service app
service tcp source eq www
nat (inside,outside) source static webserver interface destination static interface any
service app app
object network obj-any
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) source dynamic obj-any interface destination static obj-any obj-any
configure terminal
dns domain-lookup management
policy-map global_policy
class inspection_default
inspect icmp
call-home
profile License
destination transport-method http
destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
license smart
feature tier standard
throughput level 2G
license smart register idtoken <TOKEN>
: end
```

以下是 ASA virtual Auto Scale with Azure GWLB 解决方案的 ASA 配置文件示例。

```
interface G0/0
nameif outside
ip address dhcp setroute
```

```
no shut
!s
sh 168.63.129.0 255.255.255.0 outside
route outside 0.0.0.0 0.0.0.0 192.168.2.1 2
nve 1
encapsulation vxlan
source-interface outside
peer ip 192.168.2.100
!i
nterface vni1
proxy paired
nameif GWLB-backend-pool
internal-port 2000
internal-segment-id 800
external-port 2001
external-segment-id 801
vtep-nve 1
!s
ame-security-traffic permit intra-interface
```

构建 Azure 函数应用包

ASA virtual Auto Scale 解决方案要求您构建一个存档文件：*ASM_Function.zip*，它以压缩 ZIP 包的形式提供一组离散的 Azure 函数。

有关如何构建 *ASM_Function.zip* 包的说明，请参阅通过源代码构建 Azure 函数 ， 第 32 页。

这些函数尽可能离散以执行特定任务，可以根据需要进行升级，以提供增强功能和新版本支持。

输入参数

下表定义了模板参数并提供了示例。确定这些值后，您可以在将 ARM 模板部署到 Azure 订用时使用这些参数创建 ASA virtual 设备。请参阅部署 Auto Scale ARM 模板 ， 第 14 页。在 Auto Scale with Azure GWLB 解决方案中，还会创建网络基础设施，因此必须在模板中配置其他输入参数。参数说明的含义不言自明。

表 *1:* 模板参数

参数名	允许的值/类型	说明	资源创建类型
resourceNamePrefix	字符串*（3-10 个字符）	所有资源都使用包含此前缀的名称创建。 注：只能使用小写字母。 示例：asav	New
virtualNetworkRg	字符串	虚拟网络资源组名称。 示例：cisco-virtualnet-rg	现有
virtualNetworkName	字符串	虚拟网络名称（已创建）。 示例：cisco-virtualnet	现有

参数名	允许的值/类型	说明	资源创建类型
mgmtSubnet	字符串	管理子网名称（已创建） 示例：cisco-mgmt-subnet	现有
insideSubnet	字符串	内部子网名称（已创建）。 示例：cisco-inside-subnet	现有
internalLbIp	字符串	内部子网的内部负载均衡器 IP 地址（已创建）。 例如：1.2.3.4	现有
outsideSubnet	字符串	外部子网名称（已创建）。 示例：cisco-outside-subnet	现有
softwareVersion	字符串	ASA virtual 版本（在部署期间从下拉列表中选择）。 默认值：914.1.0允许：914.1.0, 913.1.0	现有
vmSize	字符串	ASA virtual 实例的大小（在部署过程中从下拉列表中选择）。	不适用
asaAdminUserName	字符串*	ASA virtual 'admin' 用户的用户名。 密码的长度必须为 12 至 72 个字符，而且必须具有：小写、大写、数字及特殊字符；重复字符不得超过 2 个。 这**不能**是“admin”。请参阅 Azure 以了解 VM 管理员用户名准则。 **注释** 模板中不对此进行合规性检查。	New
asaAdminUserPassword	字符串*	ASA virtual管理员用户的密码。 密码的长度必须为 12 至 72 个字符，而且必须具有：小写、大写、数字及特殊字符；重复字符不得超过 2 个。 **注释** 模板中不对此进行合规性检查。	New

参数名	允许的值/类型	说明	资源创建类型
scalingPolicy	POLICY-1/POLICY-2	**POLICY-1**：当任何 ASA virtual 的平均负载在所配置的持续时间内超过外向扩展阈值时，将触发外向扩展。 **POLICY-2**：当自动扩展组中所有 ASA virtual 设备的平均负载在所配置的持续时间内超过外向扩展阈值时，将触发外向扩展。 在两种情况下，内向扩展逻辑都保持不变：当所有 ASA virtual 设备的平均负载在所配置的持续时间内低于内向扩展阈值时，将触发内向扩展。	不适用
scalingMetricsList	字符串	用于制定扩展决策的指标。 允许：CPU 默认值：CPU	不适用
scaleInThreshold	字符串	CPU 指标的内向扩展阈值。 默认值：10 当 ASA virtual 指标低于此值时，将触发扩展。 请参阅Auto Scale 逻辑，第 29 页。	不适用
scaleOutThreshold	字符串	CPU 指标的横向扩展阈值。 默认值：80 当 ASA virtual指标高于此值时，将触发横向扩展。 “scaleOutThreshold”应始终大于“scaleInThreshold”。 请参阅Auto Scale 逻辑，第 29 页。	不适用
minAsaCount	整数	在任何给定时间，规模集中可用的最小 ASA virtual 实例数。 示例：2	不适用

参数名	允许的值/类型	说明	资源创建类型
maxAsaCount	整数	规模集中允许的最大 ASA virtual 实例数。 示例：10 注释 Auto Scale 逻辑不会检查此变量的范围，因此请认真填写。	不适用
metricsAverageDuration	整数	从下拉列表中选择。 此数字表示计算指标平均值的时间（以分钟为单位）。 如果此变量的值为 5（即 5 分钟），则当计划 Auto Scale Manager 时，它将检查过去 5 分钟内的指标平均值，并且基于此平均值做出扩展决定。 注释 由于 Azure 限制，仅 1、5、15 和 30 是有效数字。	不适用
initDeploymentMode	BULK/STEP	主要适用于第一次部署，或者规模集不包含任何 ASA virtual 实例时。 BULK：Auto Scale 管理器将尝试一次并行部署“minAsaCount”数量的 ASA virtual 实例。 STEP：Auto Scale 管理器将按照计划间隔逐个部署“minAsaCount”数量的 ASA virtual 设备。	
configurationFile	字符串	ASA virtual 配置文件的完整路径。 示例：https://myserver/asavconfig/asaconfig.txt	不适用
*Azure 对新资源的命名约定有限制。查看限制，或者直接全部使用小写字母。**不要使用空格或任何其他特殊字符**。			

部署 Auto Scale 解决方案

部署 Auto Scale ARM 模板

使用三明治拓扑的ASA 虚拟 Auto Scale for Azure - 使用 ARM 模板**azure_asav_autoscale.json**来部署 ASA virtual Auto Scale for Azure 所需的资源。在给定资源组内，ARM 模板部署会创建以下各项：

- 虚拟机规模集 (VMSS)
- 外部负载均衡器
- 内部负载均衡器
- Azure 函数应用
- 逻辑应用
- 安全组（用于数据接口和管理接口）

ASA 虚拟 Auto Scale with Azure GWLB - 使用 ARM 模板**azure_asav_autoscale_with_GWLB.json**来部署 ASA virtual Auto Scale with Azure GWLB 解决方案所需的资源。在给定资源组内，ARM 模板部署会创建以下各项：

- 虚拟机 (VM) 或虚拟机规模集 (VMSS)
- 网关负载均衡器
- Azure 函数应用
- 逻辑应用
- 网络基础设施
- 部署所需的安全组和其他各种组件

开始之前

- 从 GitHub 存储库下载 ARM 模板 (https://github.com/CiscoDevNet/cisco-asav/tree/master/autoscale/azure)。

过程

步骤 1 如果您需要在多个 Azure 区域中部署 ASA virtual 实例，请基于部署区域中可用的区域编辑 ARM 模板。

示例：

```
"zones": [
    "1",
```

```
            "2",
            "3"
        ],
```

本示例显示了包含 3 个区域的“美国中部”区域。

步骤 2 编辑外部负载均衡器中所需的流量规则。您可以通过扩展此“json”数组来添加任意数量的规则。

示例：

```
       {
        "type": "Microsoft.Network/loadBalancers",
        "name": "[variables('elbName')]",
        "location": "[resourceGroup().location]",
        "apiVersion": "2018-06-01",
        "sku": {
          "name": "Standard"
        },
        "dependsOn": [
          "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"

        ],
        "properties": {
          "frontendIPConfigurations": [
            {
              "name": "LoadBalancerFrontEnd",
                "properties": {
                  "publicIPAddress": {
                    "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
                  }
                }
            }
          ],
          "backendAddressPools": [
            {
              "name": "backendPool"
            }
          ],
          "loadBalancingRules": [
            {
              "properties": {
                "frontendIPConfiguration": {
                  "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/frontendIpConfigurations/LoadBalancerFrontend')]"
                },
                "backendAddressPool": {
                  "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/backendAddressPools/BackendPool')]"
                },
                "probe": {
                  "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/probes/lbprobe')]"
                },
                "protocol": "TCP",
                "frontendPort": "80",
                "backendPort": "80",
                "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
              },
              "Name": "lbrule"
            }
```

```
],
```

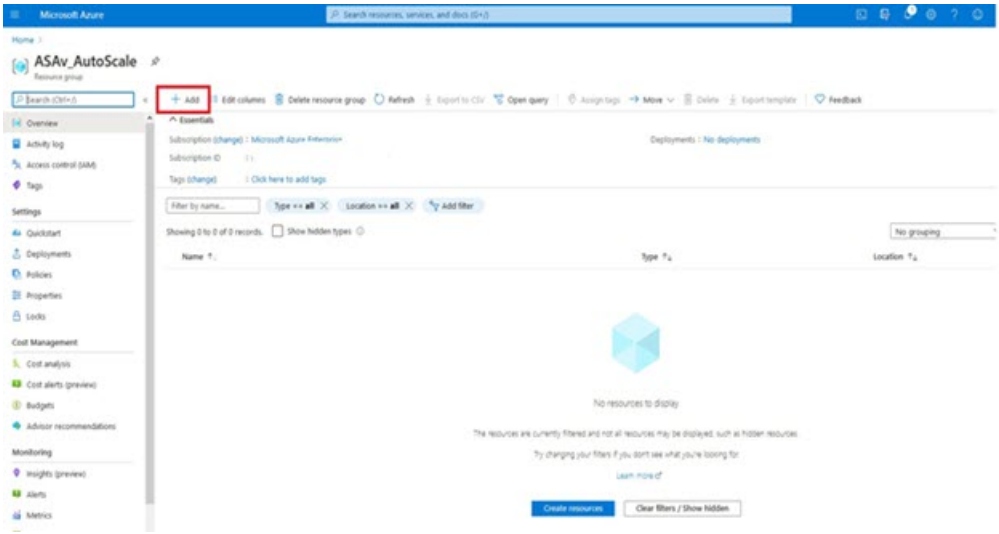
注释
如果您不想编辑此文件，也可以在部署后从 Azure 门户编辑此项。

步骤 3 使用您的 Microsoft 帐户用户名和密码登录 Microsoft Azure 门户。

步骤 4 点击服务菜单中的**资源组 (Resource groups)** 以访问资源组边栏选项卡。您将看到该边栏选项卡中列出您的订用中的所有资源组。

创建新资源组或选择现有的空资源组；例如，*ASA virtual_AutoScale*。

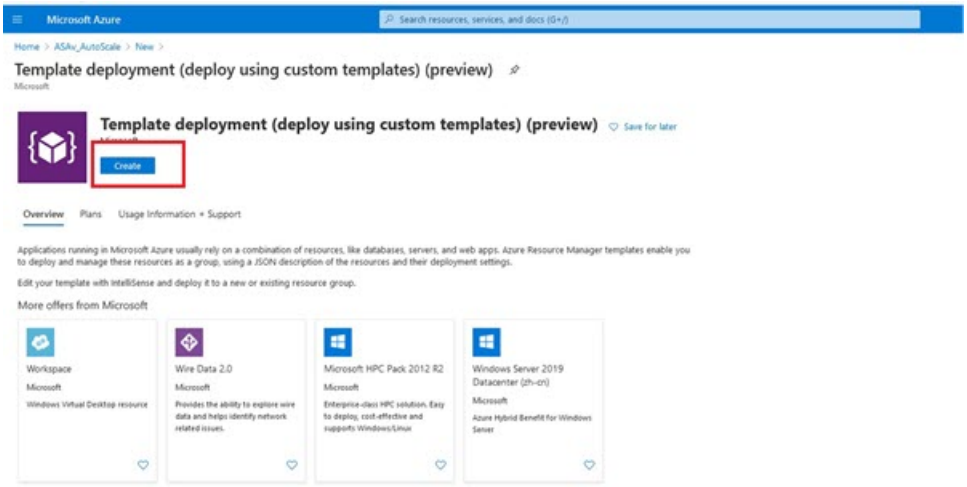
图 *2: Azure* 门户



步骤 5 点击**创建资源 (+) (Create a resource [+])**，为模板部署创建新资源。此时将显示“创建资源组”(Create Resource Group) 边栏选项卡。

步骤 6 在**搜索市场 (Search the Marketplace)** 中，键入**模板部署（使用自定义模板部署）**，然后按 **Enter**。

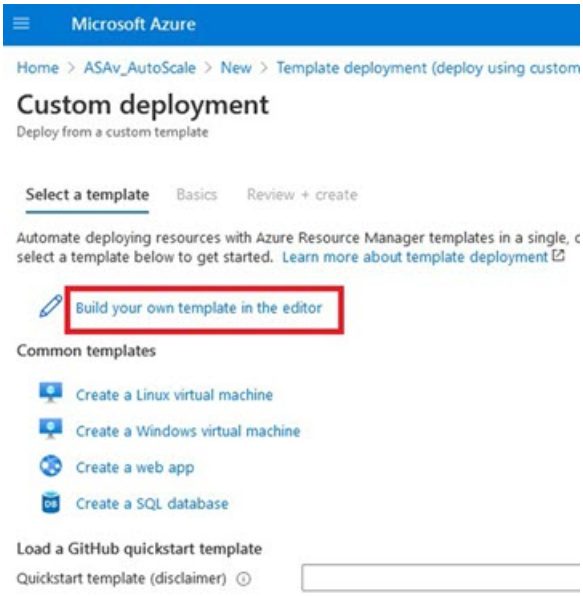
图 *3:* 自定义模板部署



步骤 **7** 点击创建 **(Create)**。

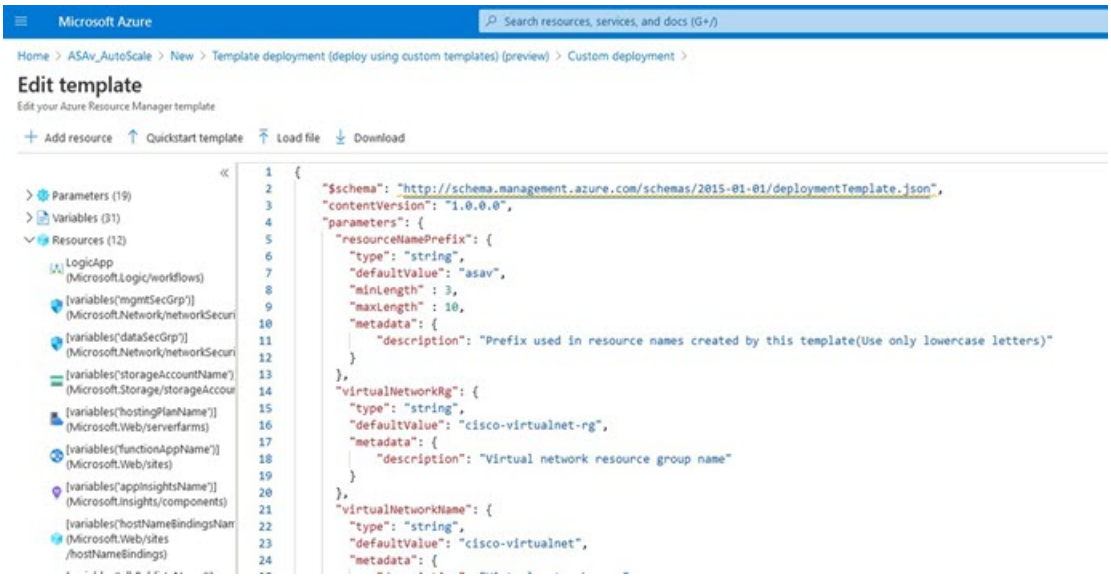
步骤 **8** 创建模板时有多个选项。选择在编辑器中选择构建您自己的模板 **(Build your own template in editor)**。

图 *4:* 构建您自己的模板



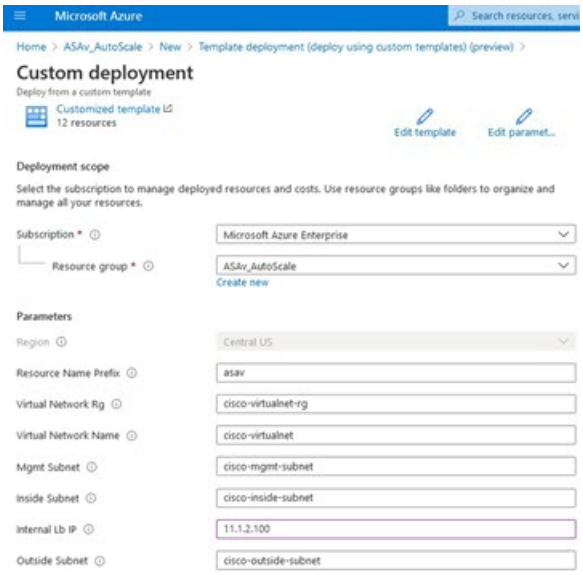
步骤 **9** 在编辑模板 **(Edit template)** 窗口中，删除所有默认内容并从更新的 *azure_asav_autoscale.json* 复制内容，然后点击保存 **(Save)**。

图 *5:* 编辑模板



步骤 10 在下一部分，填写所有参数。有关每个参数的详细信息，请参阅输入参数，第 10 页，然后点击**购买 (Purchase)**。

图 *6: ARM* 模板参数



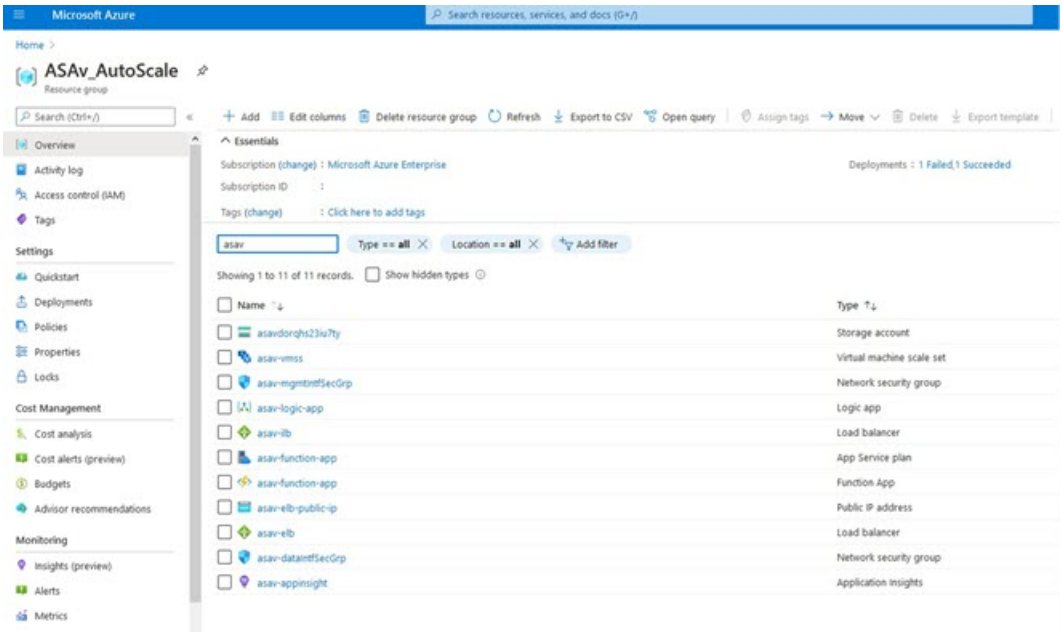
注释

您也可以点击**编辑参数 (Edit Parameters)**，然后编辑 JSON 文件或上传预填的内容。

ARM 模板的输入验证功能有限，因此您需要负责验证输入。

步骤 11 当成功部署模板后，它将为 ASA virtual Auto Scale for Azure 解决方案创建所有必要的资源。请参阅下图中的资源。“类型”(Type) 列描述了每个资源，包括逻辑应用、VMSS、负载均衡器、公共 IP 地址等。

图 *7: ASA* 虚拟 *Auto Scale* 模板部署



部署 Azure 函数应用

部署 ARM 模板时，Azure 会创建一个主干函数应用，然后您需要为其更新和手动配置 Auto Scale Manager 逻辑所需的函数。

开始之前

- 构建 *ASM_Function.zip* 包。请参阅通过源代码构建 Azure 函数，第 32 页。

过程

步骤 1 转至您在部署 ARM 模板时创建的函数应用，然后确认不存在任何函数。在浏览器中，转至以下 URL：

https://<函数应用名称>.scm.azurewebsites.net/DebugConsole

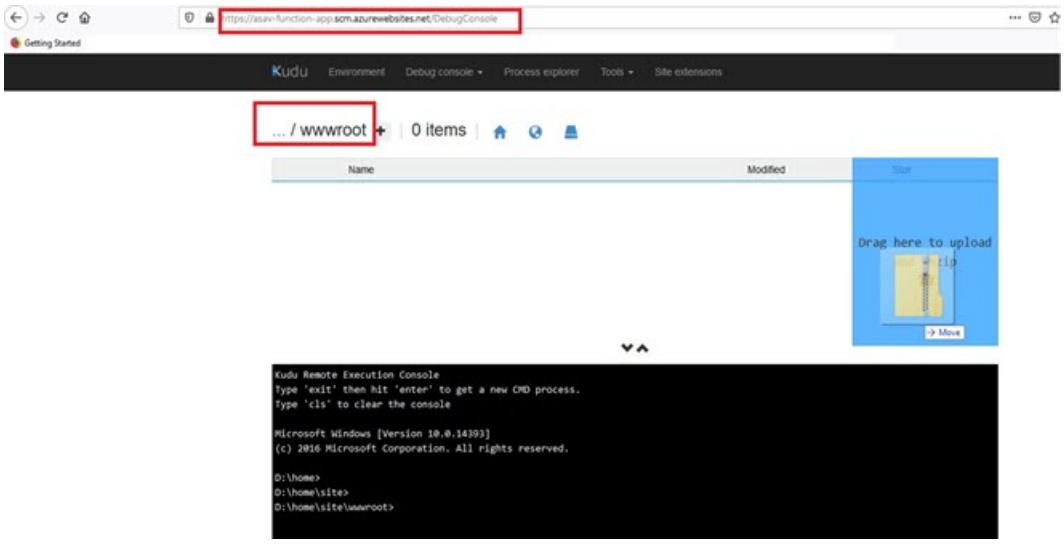
对于部署 Auto Scale ARM 模板，第 14 页中的示例：

https://asav-function-app.scm.azurewebsites.net/DebugConsole

步骤 2 在文件资源管理器中，导航到 **site/wwwroot**。

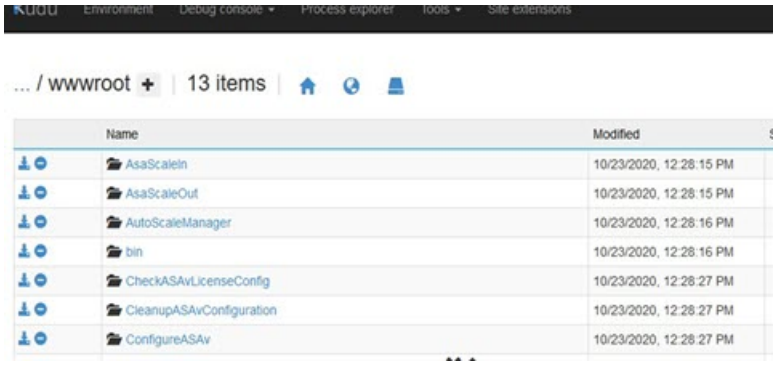
步骤 3 将 **ASM_Function.zip** 拖放到文件资源管理器的右侧。

图 *8:* 上传 ***ASA*** 虚拟 ***Auto Scale*** 功能



步骤 4 成功上传后，应该会显示所有无服务器函数。

图 *9:* ***ASA*** 虚拟 无服务器函数

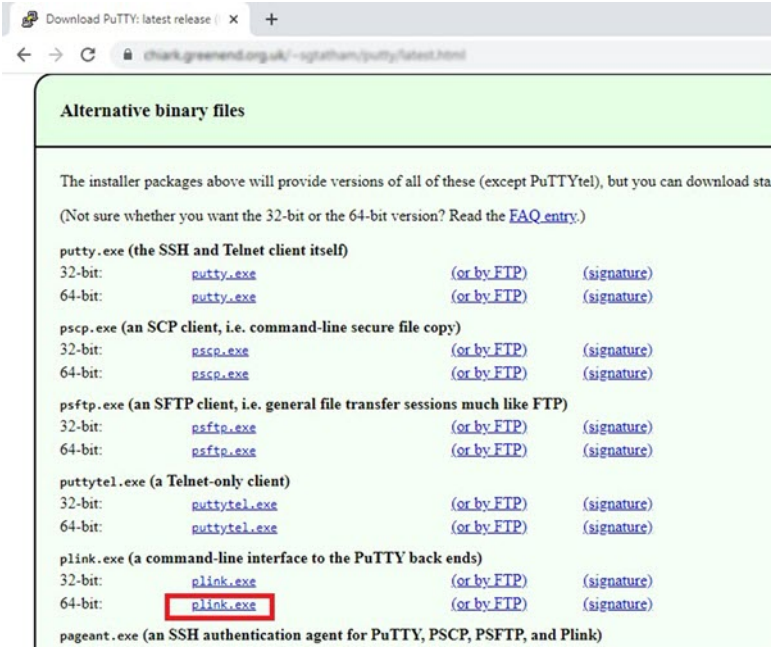


步骤 5 下载 PuTTY SSH 客户端。

Azure 函数需要通过 SSH 连接访问 ASA virtual。但是，无服务器代码中使用的开放源码库不支持 ASA virtual 所用的 SSH 密钥交换算法。因此，您需要下载预构建 SSH 客户端。

从 www.putty.org 将 PuTTY 命令行界面下载到 PuTTY 后端 (*plink.exe*)。

图 *10:* 下载 ***PuTTY***



步骤 6 将 SSH 客户端可执行文件 **plink.exe** 重命名为 **asassh.exe**。

步骤 7 将 **asassh.exe** 拖放到文件资源管理器的右侧，放到上一步中上传 **ASM_Function.zip** 的位置。

步骤 8 验证 SSH 客户端与函数应用程序一起存在。必要时刷新页面。

微调配置

有一些配置可用于微调 Auto Scale Manager 或在调试中使用。这些选项不会在 ARM 模板中显示，但可以在函数应用下编辑它们。

开始之前

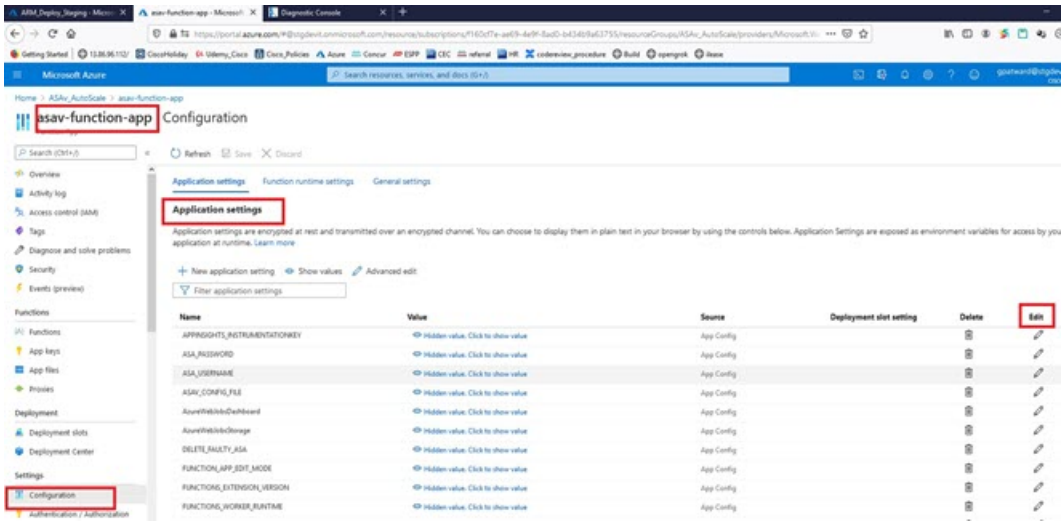
注释 可以随时编辑此项。按照以下顺序编辑配置。

- 禁用函数应用。
- 等待现有的计划任务完成。
- 编辑并保存配置。
- 启用函数应用。

过程

步骤 1 在 Azure 门户中，搜索并选择 ASA virtual 函数应用。

图 *11: ASA* 虚拟 函数应用



步骤 2 也可以在此处编辑通过 ARM 模板传递的配置。变量名称可能与 ARM 模板不同，但您可以轻松地从其名称中确定它们的用途。

大多数选项的名称不言自明。例如：

- 配置名称：“DELETE_FAULTY_ASA”（默认值：YES）

  在外向扩展期间，将会启动新的 ASA virtual 实例并通过配置文件对其进行配置。如果配置失败，则 Auto Scale Manager 将根据此选项决定保留该 ASA virtual 实例或将其删除。（YES：删除错误的 ASA virtual/NO：保留 ASA virtual 实例，即使配置失败）。

- 在函数应用设置中，有权访问 Azure 订用的用户都可以看到明文格式的所有变量（包括含安全字符串的变量，如“密码”）。

  如果用户对此有安全担忧（例如，如果在组织内的低权限用户之间共享 Azure 订用），可以使用 Azure 的 *Key Vault* 服务来保护密码。配置此项后，用户必须提供由存储密码的密钥保管库生成的安全标识符，而不是函数设置中的明文密码。

  注释
  搜索 Azure 文档，查找保护应用程序数据的最佳实践。

在虚拟机规模集中配置 IAM 角色

Azure 身份及访问管理 (IAM) 作为 Azure 安全和访问控制的一部分，用于管理和控制用户的身份。Azure 资源的托管身份为 Azure 服务提供 Azure Active Directory 中自动托管的身份。

这将允许函数应用控制虚拟机规模集 (VMSS)，无需显式身份验证凭证。

过程

步骤 1 在 Azure 门户中，转至 VMSS。

步骤 2 点击**访问控制 (IAM) (Access control [IAM])**。

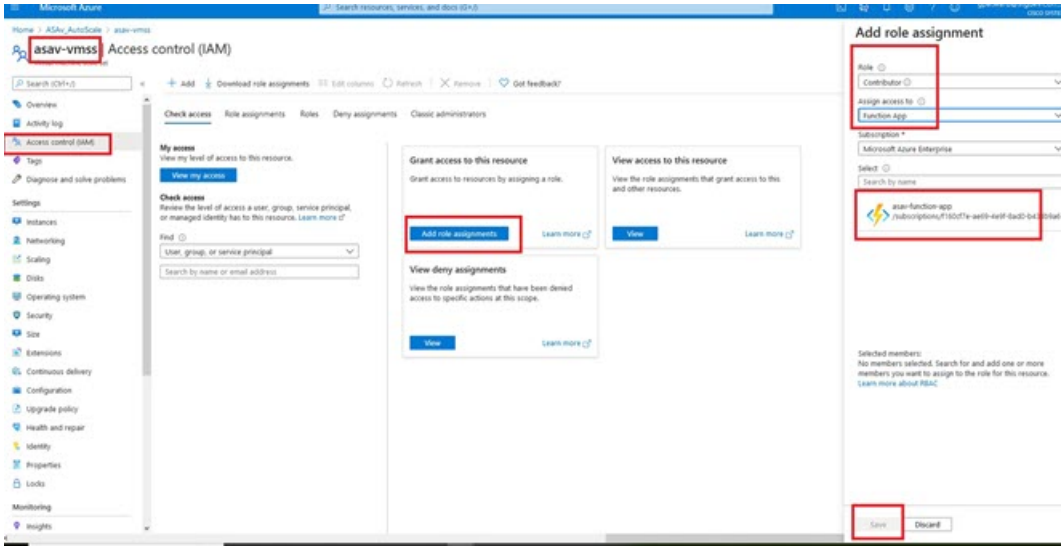
步骤 3 点击**添加 (Add)** 以添加角色分配

步骤 4 从**添加角色分配 (Add role assignment)**下拉列表中选择**参与者 (Contributor)**。

步骤 5 从**分配访问 (Assign access to)** 下拉列表中选择**函数应用 (Function App)**。

步骤 6 选择 ASA virtual 函数应用。

图 *12: AIM* 角色分配



步骤 7 点击**保存 (Save)**。

注释
此外，还应确认尚未启动任何 ASA virtual 实例。

更新安全组

ARM 模板创建两个安全组，一个用于管理接口，一个用于数据接口。管理安全组将只允许 ASA virtual 管理活动所需的流量。不过，数据接口安全组将允许所有流量。

过程

根据您的部署的拓扑和应用程序需求，微调安全组规则。

注释
数据接口安全组至少应允许来自负载均衡器的 SSH 流量。

更新 Azure 逻辑应用

逻辑应用充当Autoscale功能的协调器。ARM模板会创建一个主干逻辑应用，然后您需要手动更新，提供使之作为 Auto Scale 协调器发挥作用所需的信息。

过程

步骤 1 从存储库中将文件 *LogicApp.txt* 恢复到本地系统，然后如下所示进行编辑。

重要事项
在继续之前，阅读并理解所有这些步骤。

这些手动步骤不会在 ARM 模板中自动执行，以便稍后只能独立升级逻辑应用。

a) 必需： 查找所有“SUBSCRIPTION_ID”并替换为您的订用 ID 信息。
b) 必需： 查找所有“RG_NAME”并替换为您的资源组名称。
c) 必需： 查找所有“FUNCTIONAPPNAME”并替换为您的函数应用名称。

以下示例显示了 *LogicApp.txt* 文件中的几行：

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_ASA": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
```

```
                                    "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                                }
.
.
                        "DeviceDeRegister": {
                            "inputs": {
                                "body": "@body('AutoScaleManager')",
                                "function": {
                                    "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                                }
                            },
                            "runAfter": {
                                "Delay_For_connection_Draining": [
```

d) （可选）编辑触发间隔，或保留默认值(5)。这是定期触发 Autoscale 的时间间隔。以下示例显示了 *LogicApp.txt* 文件中的几行：

```
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
                },
```

e) （可选）编辑要进行排空的时间，或保留默认值 (5)。这是内向扩展操作期间，在删除设备之前从 ASA virtual 中排空现有连接的时间间隔。以下示例显示了 *LogicApp.txt* 文件中的几行：

```
         "actions": {
              "Branch_based_on_Scale-In_or_Scale-Out_condition": {
                  "actions": {
                      "Delay_For_connection_Draining": {
                          "inputs": {
                              "interval": {
                                  "count": 5,
                                  "unit": "Minute"
                              }
```

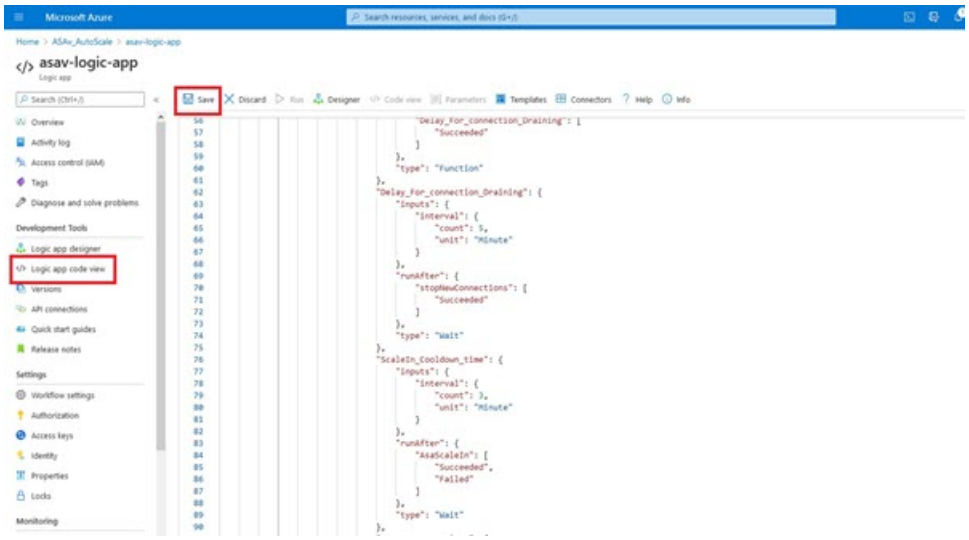
f) （可选）编辑冷却时间，或保留默认值(10)。这是在外向扩展完成后不执行任何操作的时间。以下示例显示了 *LogicApp.txt* 文件中的几行：

```
                "actions": {
                    "Branch_based_on_Scale-Out_or_Invalid_condition": {
                        "actions": {
                            "Cooldown_time": {
                                "inputs": {
                                    "interval": {
                                        "count": 10,
                                        "unit": "Second"
                                    }
```

注释
这些步骤也可以从 Azure 门户完成。有关详细信息，请参阅 Azure 文档。

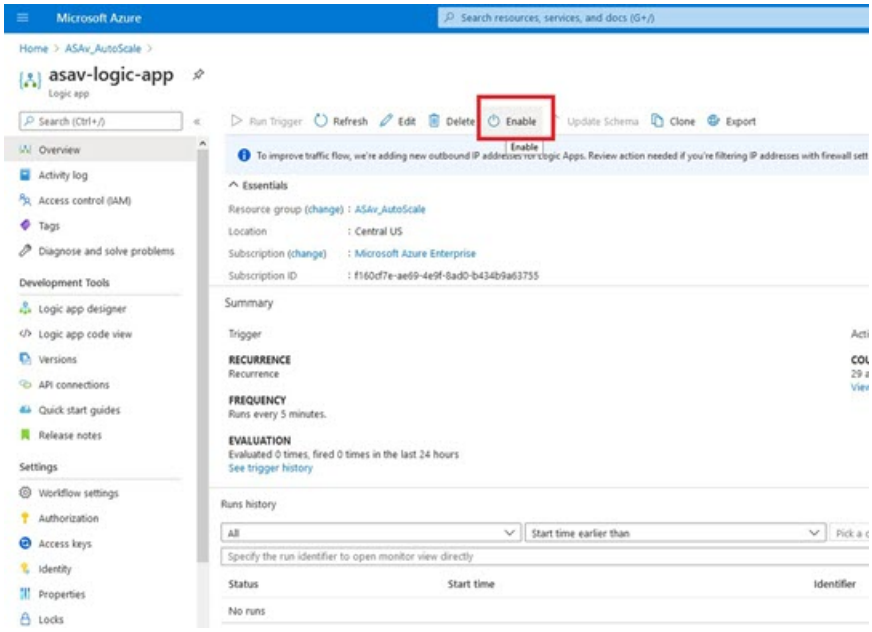
步骤 2 转至**逻辑应用代码视图 (Logic App code view)**，删除默认内容并粘贴编辑后的 *LogicApp.txt* 文件内容，然后点击**保存 (Save)**。

图 *13:* 逻辑应用代码视图



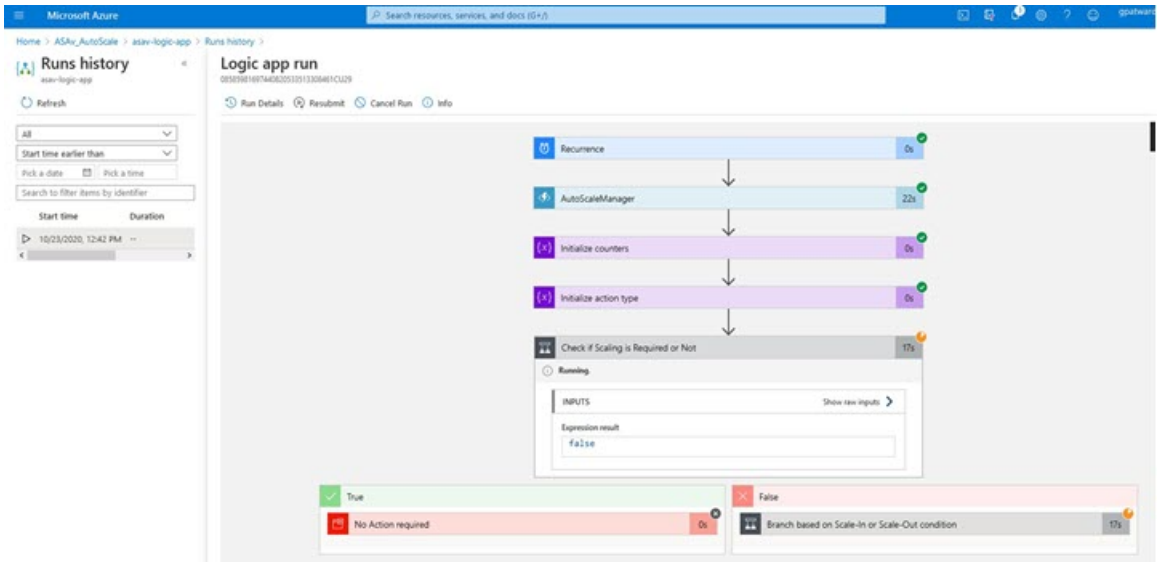
步骤 3 保存逻辑应用时，它处于“禁用”状态。当要启动 Auto Scale Manager 时，请点击**启用 (Enable)**。

图 *14:* 启用逻辑应用



步骤 4 启用后，任务就会开始运行。点击“正在运行”(Running) 状态可查看活动。

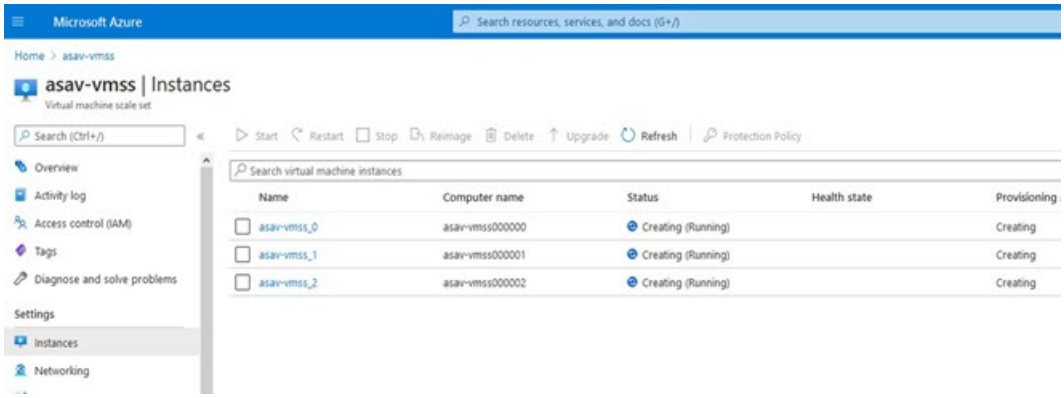
图 *15:* 逻辑应用运行状态



步骤 5 逻辑应用启动后，所有与部署相关的步骤都将完成。

步骤 6 在 VMSS 中验证是否正在创建 ASA virtual 实例。

图 *16: ASA* 虚拟 实例运行



在此示例中，由于在 ARM 模板部署中将'minAsaCount' 设置为“3”并将“initDeploymentMode”设置为“批量”，因此启动了三个 ASA virtual 实例。

升级 ASA virtual

ASA virtual升级仅支持采用虚拟机规模集 (VMSS) 映像升级的形式。因此，您需要通过 Azure REST API 接口升级 ASA virtual。

注释 您可以使用任何 REST 客户端来升级 ASA virtual。

开始之前

- 获取市场中提供的新 ASA virtual 映像版本（例如：914.001）。
- 获取用于部署原始规模集的 SKU（例如：asav-azure-byol）。
- 获取资源组和虚拟机规模集名称。

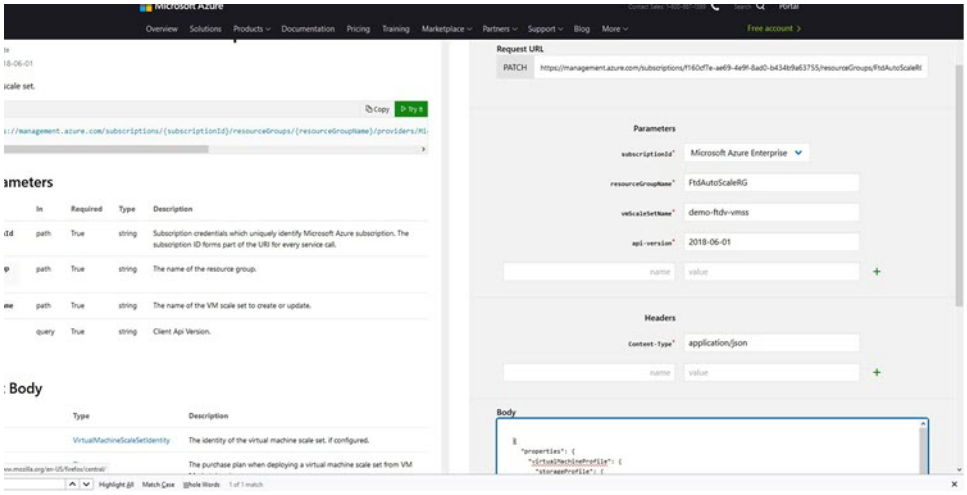
过程

步骤 1 在浏览器中，转至以下 URL：

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

步骤 2 在参数部分输入详细信息。

图 *17:* 升级 *ASA virtual*



步骤 3 在主体 **(Body)** 部分输入包含新 ASA virtual 映像版本、SKU 和触发器运行的 JSON 输入。

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                     "imageReference": {
                           "publisher": "cisco",
                           "offer": "cisco-asav",
                           "sku": "asav-azure-byol",
                            "version": "650.32.0"
                     }
               },
```

```
                }
            }
}
```

步骤 4 Azure 成功响应意味着 VMSS 已接受更改。

新映像将在新的 ASA virtual 实例中使用，而这些新实例将在外向扩展操作过程中启动。

- 虽然位于同一规模集中，但现有的 ASA virtual 实例将继续使用旧软件映像。
- 您可以覆盖上述行为，手动升级现有的 ASA virtual 实例。要执行此操作，请点击 VMSS 中的**升级 (Upgrade)** 按钮。它将重新启动并升级选定的 ASA virtual 实例。您必须手动重新注册并重新配置这些升级后的 ASA virtual 实例。**请注意，不建议使用此方法**。

Auto Scale 逻辑

外向扩展逻辑

- **POLICY-1**：当**任何** ASA virtual 的平均负载在所配置的持续时间内超过外向扩展阈值时，将触发外向扩展。
- **POLICY-2**：当**所有** ASA virtual 设备的平均负载在所配置的持续时间内超过外向扩展阈值时，将触发外向扩展。

内向扩展逻辑

- 如果**所有** ASA virtual 设备的 CPU 利用率在所配置的持续时间内低于配置的内向扩展阈值。

说明

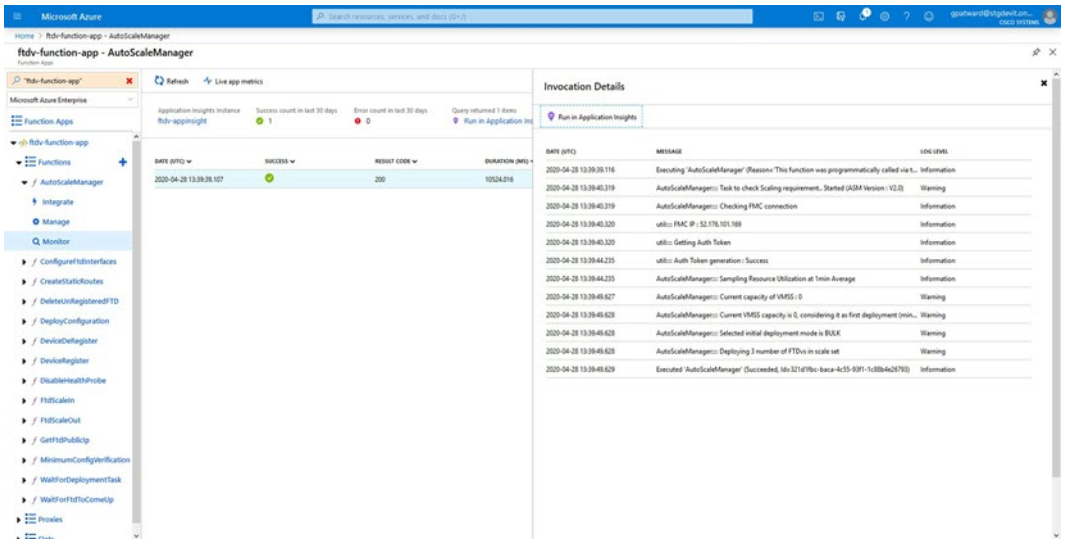
- 内向扩展/外向扩展以 1 为单位发生（即一次仅内向扩展/外向扩展 1 个 ASA virtual）。
- 上述逻辑基于以下假设：负载均衡器将尝试在所有 ASA virtual设备之间平均分配连接，一般来说，所有 ASA virtual设备应平均加载。

Auto Scale 日志记录和调试

无服务器代码的每个组件都有自己的日志记录机制。此外，还会将日志发布到应用程序洞察。

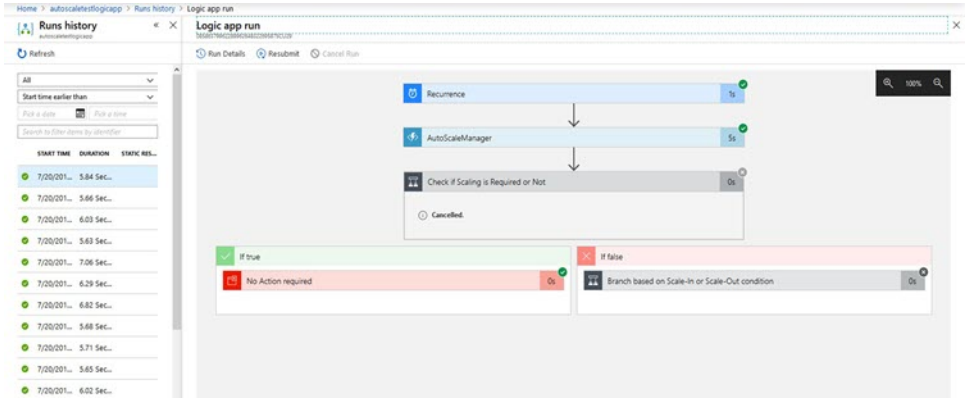
- 可以查看个别 Azure 函数的日志。

图 *18: Azure* 函数日志



- 可以查看每个逻辑应用及其各个组件每次运行的类似日志。

图 *19:* 逻辑应用运行日志



- 如果需要，可以随时停止/终止逻辑应用中任何正在运行的任务。但是，被启动/终止的当前运行 ASA virtual 设备将处于不一致状态。
- 在逻辑应用中可以看到每个运行/个别任务所花费的时间。
- 通过上传新的 zip，可以随时升级函数应用。在升级函数应用之前，先停止逻辑应用并等待所有任务完成。

Auto Scale 准则和限制

部署 ASA virtual Auto Scale for Azure 时，请注意以下准则和限制：

- 扩展决定基于 CPU 使用率。

- ASA virtual 管理接口配置为具有公共 IP 地址。
- 仅支持 IPv4。
- ARM 模板的输入验证功能有限，因此您需要负责提供正确的输入验证。
- Azure管理员可以在函数应用环境中看到明文形式的敏感数据（如管理登录凭证和密码）。您可以使用 *Azure Key Vault* 服务保护敏感数据。
- 配置中的任何更改都不会自动反映在运行中的实例上。更改将仅反映在未来的设备上。应手动将此类更改推送到现有设备。
- 如果您在现有实例上手动更新配置时遇到问题，我们建议从扩展组中删除这些实例并将其替换为新实例。

故障排除

以下是 ASA virtual Auto Scale for Azure 的常见错误情况和调试提示：

- 无法通过 SSH 连接到 ASA virtual：检查是否通过模板将复杂密码传递到 ASA virtual；检查安全组是否允许 SSH 连接。
- 负载均衡器运行状况检查失败：检查 ASA virtual 是否在数据接口上响应 SSH；检查安全组设置。
- 流量问题：检查负载均衡器规则、ASA virtual 中配置的 NAT 规则/静态路由；检查模板和安全组规则中提供的 Azure 虚拟网络/子网/网关详细信息。
- 逻辑应用无法访问 VMSS：检查 VMSS 中的 IAM 角色配置是否正确。
- 逻辑应用运行很长时间：在外向扩展 ASA virtual 设备上检查 SSH 访问；检查 Azure VMSS 中 ASA virtual 设备的状态。
- 与订用 ID 相关的 Azure 函数抛出错误：验证您的帐户中是否选择了默认预订。
- 内向扩展操作失败：有时 Azure 会花费很长时间删除实例，在这种情况下，内向扩展操作可能会超时并报告错误，但最终实例将被删除。
- 在做出任何配置更改之前，请确保禁用逻辑应用程序，并等待所有正在运行的任务完成。

如果在 ASA virtual Auto Scale 与 Azure GWLB 部署期间遇到任何问题，请查看以下故障排除提示：

- 检查 ELB-GWLB 关联。
- 检查 GWLB 中的运行状况探测状态。
- 通过验证 ASA virtual 物理和逻辑接口上的流量来检查 VXLAN 配置。
- 检查安全组规则。

通过源代码构建 Azure 函数

系统要求

- Microsoft Windows 桌面/笔记本电脑。
- Visual Studio（使用 Visual Studio 2019 版本 16.1.3 进行测试）

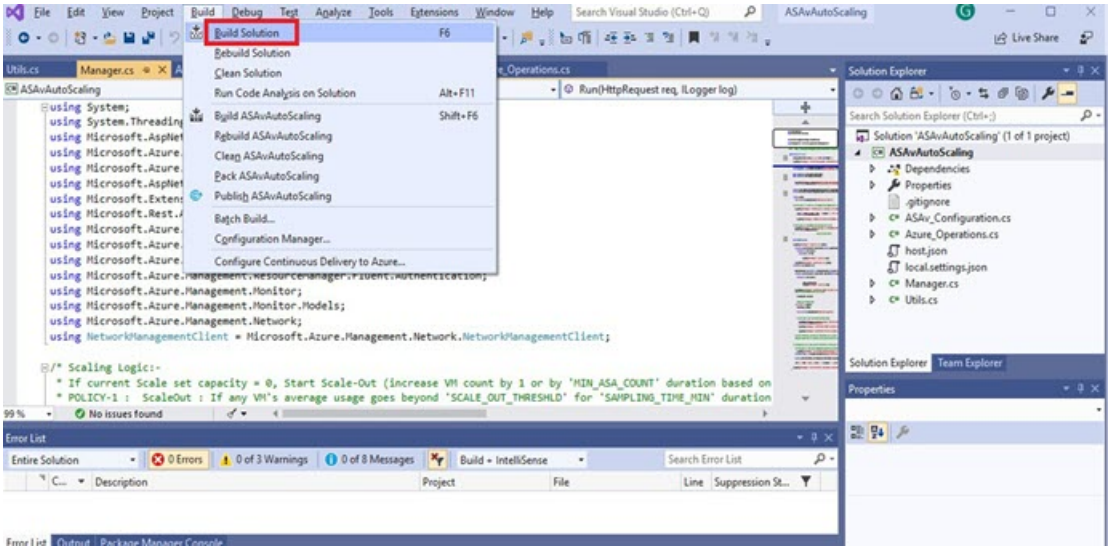
注释 Azure 函数是使用 C# 编写的。

- “Azure Development”工作负载需要安装在 Visual Studio 中。

使用 Visual Studio 构建

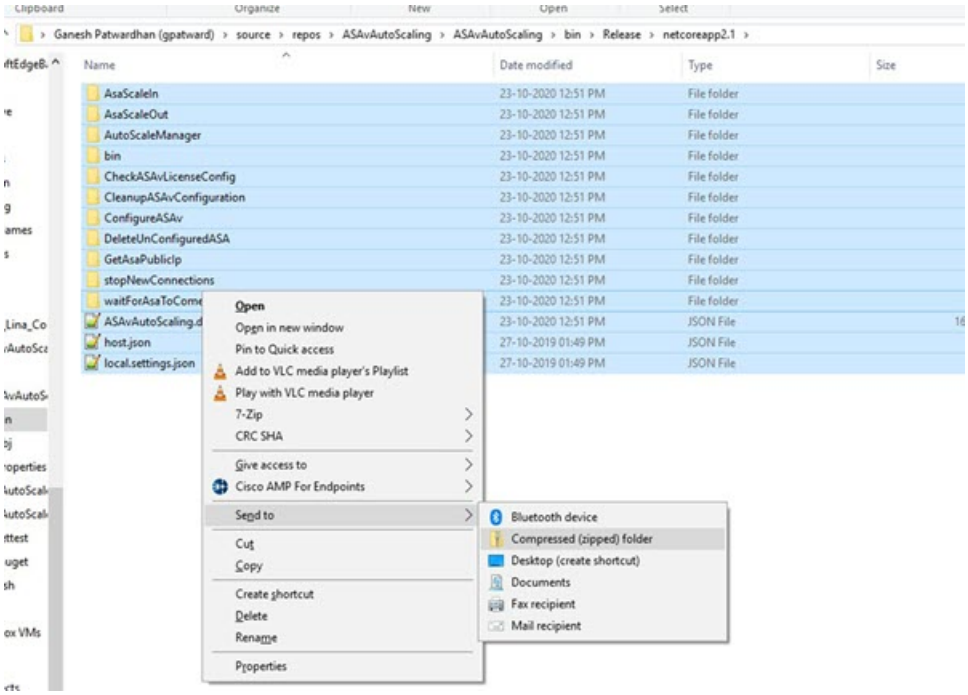
1. 将“code”文件夹下载到本地计算机。
2. 导航到文件夹“ASAAutoScaling”。
3. 在 Visual Studio 中打开项目文件“ASAAutoScaling.csproj”。
4. 使用 Visual Studio 标准程序进行清理和构建。

图 *20: Visual Studio* 内部版本



5. 成功编译内部版本后，导航到 **\bin\Release\netcoreapp2.1** 文件夹。
6. 选择所有内容，点击 **发送到 (Send to)** > **压缩 (zipped) 文件夹 (Compressed [zipped] folder)**，然后将 ZIP 文件保存为 *ASM_Function.zip*。

图 ***21:*** 生成 ***ASM_Function.zip***



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。