Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide the following key benefits:

* Easy to create and manage multiple VMs
* Provides high availability and application resiliency by distributing VMs across availability zones or fault domains
* Allows your application to automatically scale as resource demand changes
* Works at large-scale

**Why use virtual machine scale sets?**

To provide redundancy and improved performance, applications are typically distributed across multiple instances. Customers may access your application through a load balancer that distributes requests to one of the application instances. If you need to perform maintenance or update an application instance, your customers must be distributed to another available application instance. To keep up with extra customer demand, you may need to increase the number of application instances that run your application.

Azure virtual machine scale sets provide the management capabilities for applications that run across many VMs, automatic scaling of resources, and load balancing of traffic. Scale sets provide the following key benefits:

Easy to create and manage multiple VMs

When you have many VMs that run your application, it's important to maintain a consistent configuration across your environment. For reliable performance of your application, the VM size, disk configuration, and application installs should match across all VMs.

With scale sets, all VM instances are created from the same base OS image and configuration. This approach lets you easily manage hundreds of VMs without extra configuration tasks or network management.

Scale sets support the use of the Azure load balancer for basic layer-4 traffic distribution, and Azure Application Gateway for more advanced layer-7 traffic distribution and TLS termination.

Provides high availability and application resiliency

Scale sets are used to run multiple instances of your application. If one of these VM instances has a problem, customers continue to access your application through one of the other VM instances with minimal interruption.

For more availability, you can use Availability Zones to automatically distribute VM instances in a scale set within a single datacentre or across multiple datacentres.

Allows your application to automatically scale as resource demand changes

Customer demand for your application may change throughout the day or week. To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases.

Autoscale also minimizes the number of unnecessary VM instances that run your application when demand is low, while customers continue to receive an acceptable level of performance as demand grows and additional VM instances are automatically added. This ability helps reduce costs and efficiently create Azure resources as required.

Works at large-scale

Scale sets support up to 1,000 VM instances for standard marketplace images and custom images through the Azure Compute Gallery (formerly known as Shared Image Gallery). If you create a scale set using a managed image, the limit is 600 VM instances.

For the best performance with production workloads, use Azure Managed Disks.

Terraform Script:

terraform {  
 required\_version = ">=0.12"  
  
 required\_providers {  
 azurerm = {  
 source = "hashicorp/azurerm"  
 version = "~>2.0"  
 }  
 }  
}  
  
provider "azurerm" {  
 features {}  
}  
  
  
locals {  
 resource\_group="VMSS-RG"  
 location="Central India"  
}  
  
resource "azurerm\_resource\_group" "VMSS-RG"{  
 name=local.resource\_group  
 location=local.location  
}  
  
resource "azurerm\_virtual\_network" "VMSS-vnet" {  
 name = "VMSS-vnet"  
 location = "Central India"  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 address\_space = ["10.0.0.0/16"]  
  
 }  
resource "azurerm\_subnet" "vmss-subnet" {  
 name = "vmss-subnet"  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 virtual\_network\_name = azurerm\_virtual\_network.VMSS-vnet.name  
 address\_prefixes = ["10.0.0.0/24"]  
 }  
  
resource "azurerm\_network\_interface" "VMSS-nic" {  
 name = "VMSS-nic"  
 location = local.location  
 resource\_group\_name = local.resource\_group  
  
 ip\_configuration {  
 name = "internal"  
 subnet\_id = azurerm\_subnet.vmss-subnet.id  
 private\_ip\_address\_allocation = "Dynamic"  
 }  
  
 depends\_on = [  
 azurerm\_virtual\_network.VMSS-vnet  
 ]  
}  
  
resource "azurerm\_public\_ip" "app-public-ip" {  
 name = "app-public-ip"  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 location = "central India"  
 allocation\_method = "Static"  
}  
  
  
resource "azurerm\_lb" "VMSS-LB" {  
 name = "VMSS-LB"  
 location = azurerm\_resource\_group.VMSS-RG.location  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 sku="Standard"  
 sku\_tier = "Regional"  
 frontend\_ip\_configuration {  
 name = "frontend-ip"  
 public\_ip\_address\_id = azurerm\_public\_ip.app-public-ip.id  
 }  
  
 depends\_on=[  
 azurerm\_public\_ip.app-public-ip  
 ]  
}  
  
  
resource "azurerm\_lb\_backend\_address\_pool" "scalesetpool" {  
 loadbalancer\_id = azurerm\_lb.VMSS-LB.id  
 name = "scalesetpool"  
 depends\_on=[  
 azurerm\_lb.VMSS-LB  
 ]  
}  
  
resource "azurerm\_lb\_probe" "ProbeA" {  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 loadbalancer\_id = azurerm\_lb.VMSS-LB.id  
 name = "probeA"  
 port = 80  
 protocol = "Tcp"  
 depends\_on=[  
 azurerm\_lb.VMSS-LB  
 ]  
}  
  
resource "azurerm\_lb\_rule" "RuleA" {  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 loadbalancer\_id = azurerm\_lb.VMSS-LB.id  
 name = "RuleA"  
 protocol = "Tcp"  
 frontend\_port = 80  
 backend\_port = 80  
 frontend\_ip\_configuration\_name = "frontend-ip"  
 backend\_address\_pool\_ids = [ azurerm\_lb\_backend\_address\_pool.scalesetpool.id ]  
}  
  
resource "azurerm\_windows\_virtual\_machine\_scale\_set" "scale\_set" {  
 name = "scale-set"  
 resource\_group\_name = azurerm\_resource\_group.VMSS-RG.name  
 location = azurerm\_resource\_group.VMSS-RG.location  
 sku = "Standard\_D2s\_v3"  
 instances = 2  
 admin\_password = "Sharonraju@123"  
 admin\_username = "sharon"  
 upgrade\_mode = "Automatic"  
 source\_image\_reference {  
 publisher = "MicrosoftWindowsServer"  
 offer = "WindowsServer"  
 sku = "2019-Datacenter"  
 version = "latest"  
 }  
 os\_disk {  
 storage\_account\_type = "Standard\_LRS"  
 caching = "ReadWrite"  
}  
  
 network\_interface {  
 name = "VMSS\_nic"  
 primary = true  
  
 ip\_configuration {  
 name = "internal"  
 primary = true  
 subnet\_id = azurerm\_subnet.vmss-subnet.id  
 load\_balancer\_backend\_address\_pool\_ids =[azurerm\_lb\_backend\_address\_pool.scalesetpool.id]  
 }  
 }  
 depends\_on=[  
 azurerm\_virtual\_network.VMSS-vnet  
 ]  
}