**Background**

Azure Cloud Services and Azure App Service both offer a staged deployment capability that allows you to deploy new application code to a staging slot and then swap it with the production slot once the deployment has been verified. While Azure Virtual Machine Scale Sets do not provide this feature we can emulate this capability using the approach outlined below.

**Blue / Green Deployment Approach**

1. Define two virtual machine scale sets.
2. The scale set hosting the current production application code is scaled out to the number of instances needed to support production traffic. We will call this scale set Blue.
3. The scale set hosting a previous version of the application code is scaled down to a minimum number of instances (1). We will call this scale set Green.
4. The load balancer is configured with two backend pools. A backend pool named Blue associated with the Blue scale set. A backend pool named Green associated with the Green scale set.
5. The Backend pool setting on the load balancing rules of the load balancer is always configured to point at whichever backend pool is currently hosting the current production application code.

**Deploying new application code**

For the sake of the example below let’s say that the Blue scale set is hosting the current production application code and the load balancer is configured to send traffic to the Blue backend pool.

1. To deploy a new set of production code we execute the deployment process against the Green scale set. This could be a complete image swap or an incremental deployment via the PowerShell DSC script extension or the custom script extension. [nimccoll/VMSSADODeployPipelines: Contains sample ARM templates, scripts, and ADO pipelines for deploying an ASP.Net application to a VM scale set (github.com)](https://github.com/nimccoll/VMSSADODeployPipelines)
2. Once the deployment has completed successfully on the Green scale set we scale it out to meet the needs of the production traffic load. [Manage Virtual Machine Scale Sets with Azure PowerShell - Azure Virtual Machine Scale Sets | Microsoft Learn](https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-manage-powershell#change-the-capacity-of-a-scale-set)
3. Once the Green scale set has been scaled out to support the production load, we update the load balancing rules of the load balancer to change the Backend pool setting to the Green backend pool. [Set-AzLoadBalancerRuleConfig (Az.Network) | Microsoft Learn](https://learn.microsoft.com/en-us/powershell/module/az.network/set-azloadbalancerruleconfig?view=azps-9.3.0#example-2-modify-a-load-balancing-rule-configuration-to-have-two-backend-address-pools)
4. With the Backend pool setting updated on the load balancing rules to send traffic to the Green scale set, we can scale the Blue scale set back down to a minimum number of instances. [Manage Virtual Machine Scale Sets with Azure PowerShell - Azure Virtual Machine Scale Sets | Microsoft Learn](https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-manage-powershell#change-the-capacity-of-a-scale-set)
5. The Green scale set is now hosting the current production code.

During the next deployment cycle new production application code will be deployed to the Blue scale set and we will repeat the process above with the roles of the two scale sets being reversed.

To determine which scale set is serving the production code at any time we can examine the load balancing rule of the load balancer and look at the Backend pool setting.