# 😊 Scale Set (Auto Scale)

Virtual Machine Scale Sets (VMSS) in Microsoft Azure allow you to manage and scale a set of virtual machines (VMs). The autoscale feature of VMSS automatically adjusts the number of VM instances running to match your application's needs. Here's how the autoscale feature works:

**Key Features of Autoscaling in VMSS**

1. **Automatic Scaling**: VMSS can automatically increase or decrease the number of VM instances based on predefined rules and real-time metrics. This ensures that your application can handle varying levels of demand without manual intervention.
2. **Customizable Rules**: You can define scaling rules based on metrics such as CPU usage, memory usage, disk I/O, network traffic, or custom metrics. These rules specify the conditions under which VM instances should be added or removed.
3. **Scheduled Scaling**: Besides metric-based scaling, you can also set up scheduled scaling. This allows you to increase or decrease the number of instances at specific times or on specific days, accommodating predictable workload patterns.
4. **Load Balancing**: VMSS integrates with Azure Load Balancer, ensuring that traffic is evenly distributed across all VM instances, which helps in maintaining application performance and reliability.
5. **Health Monitoring**: VMSS monitors the health of each VM instance. If an instance is found to be unhealthy, it can be automatically replaced to ensure the overall health of the scale set.
6. **Integration with Azure Monitor**: Autoscaling rules can be based on metrics collected by Azure Monitor, providing a comprehensive view of your application's performance and allowing more precise scaling decisions.

**Setting Up Autoscale**

1. **Define Autoscale Rules**:
    - **Metric-Based Scaling**: Set rules based on metrics like CPU usage, memory usage, etc. For example, if CPU usage exceeds 70% for 5 minutes, add more VM instances.
    - **Scheduled Scaling**: Define schedules for scaling operations. For example, scale out to 10 instances every Monday at 9 AM and scale in to 2 instances every Friday at 5 PM.
2. **Configure Notifications**: Set up alerts to notify you of scaling events or if scaling operations fail.
3. **Review and Apply**: Review the autoscale settings and apply them to your VMSS.

**Benefits**

- **Cost Efficiency**: By automatically scaling the number of VM instances based on demand, you can optimize costs by only running the necessary number of instances at any given time.
- **High Availability**: Autoscaling helps maintain application performance and availability, even under high load, by adding more instances when needed.
- **Operational Simplification**: Reduces the need for manual intervention in scaling operations, allowing you to focus on other aspects of your application.

**Use Cases**

- **Web Applications**: Handle varying traffic loads by automatically scaling web server instances.
- **Batch Processing**: Scale out instances to process large data sets and scale in once the processing is complete.
- **Microservices**: Manage and scale microservice instances independently based on their individual load and performance requirements.

**Example Scenario**

Imagine you have an e-commerce website hosted on Azure VMs. During peak shopping hours or special promotions, your website experiences high traffic, leading to increased CPU usage on your VMs. By setting up autoscale rules, you can automatically add more VM instances when CPU usage exceeds a certain threshold, ensuring that your website remains responsive and performs well under load. After the peak period, autoscale can reduce the number of instances to save costs.

In summary, the autoscale feature in VMSS provides a robust mechanism for dynamically adjusting the number of VM instances based on real-time metrics and scheduled rules, ensuring optimal performance and cost efficiency for your applications.

# 🤔 What are we doing in this lab?

In this process, you are setting up an Azure Virtual Machine Scale Set (VMSS) with autoscaling enabled. The steps include creating the VMSS, configuring its autoscale settings, and ensuring proper network and security configurations.

**Summary**

1. **Setup**: Create a VMSS by selecting a resource group, name, region, availability zone, and other basic settings.
2. **Orchestration**: Choose "Uniform" orchestration mode and keep the security type to default.
3. **Autoscaling Configuration**: Enable and configure autoscaling rules to automatically adjust the number of VMs based on CPU usage.
4. **Operating System and Size**: Select the operating system (Windows) and VM size.
5. **User Credentials**: Provide a username and password for the VMs.

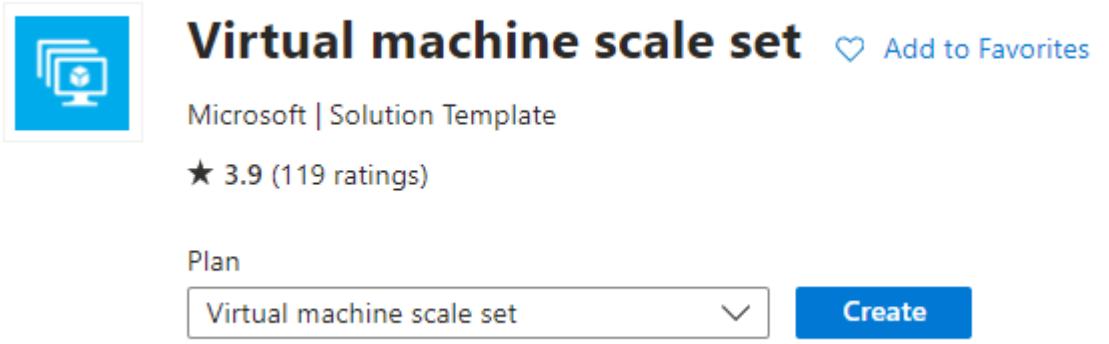
6. **Networking**: Configure networking settings, allowing RDP port access and enabling public IP addresses for individual VMs.
7. **Review and Create**: Review all settings and create the VMSS.
8. **Testing Autoscaling**: Verify autoscaling by adjusting the CPU threshold and observing the creation or deletion of VM instances.

**End Goal**

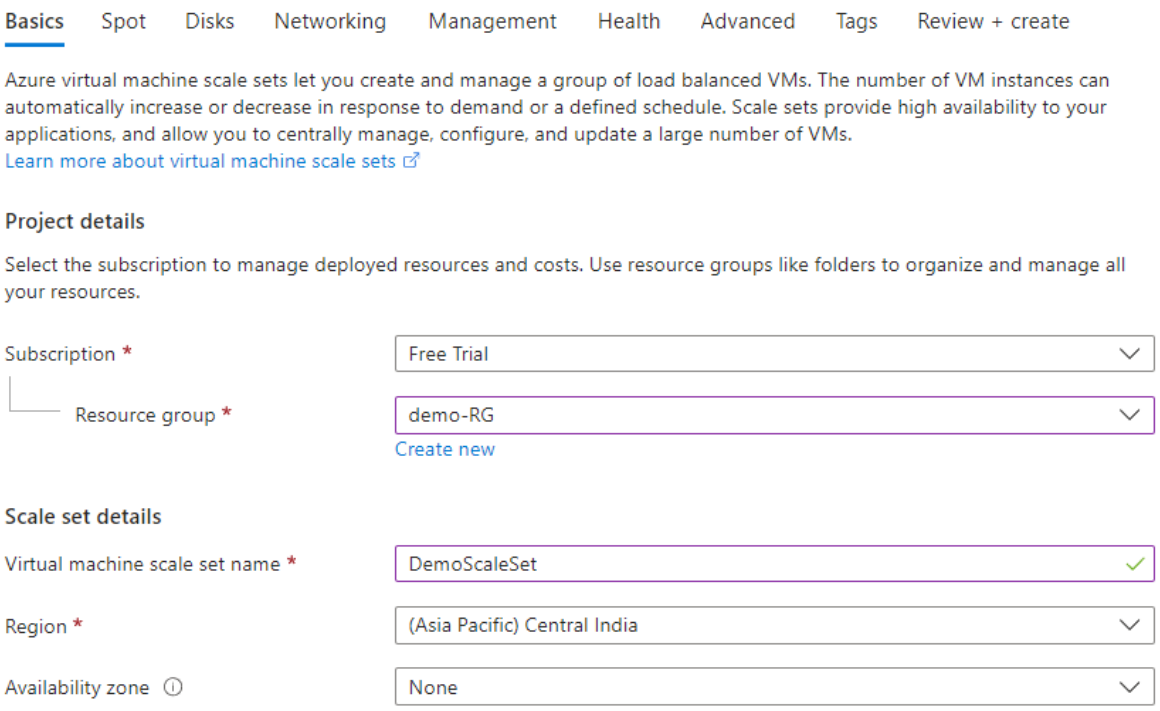
The end goal is to have a VMSS that can automatically scale the number of VM instances up or down based on the CPU usage, ensuring optimal performance and cost-efficiency. This setup provides high availability and reliability for your applications while minimizing manual intervention for scaling operations.

## 😄 To begin with the Lab:

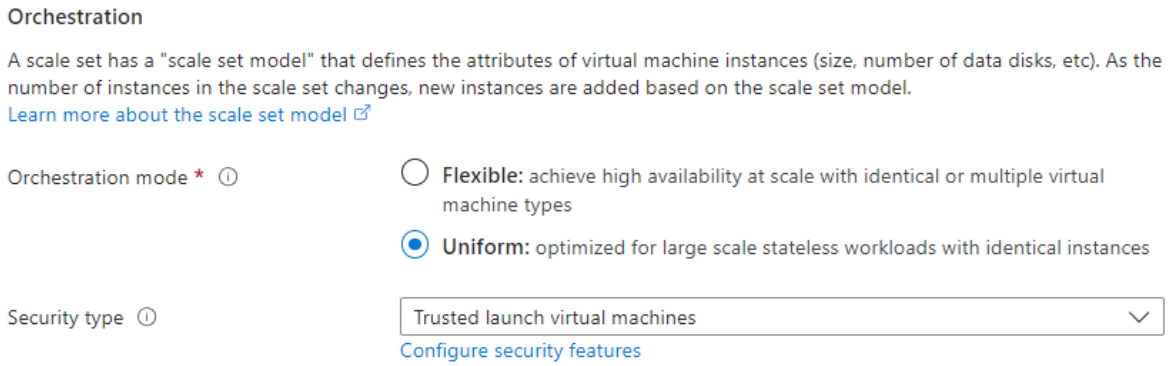
1. Log in to Azure Portal then from the marketplace search for Virtual Machine Scale Sets and choose the service accordingly.



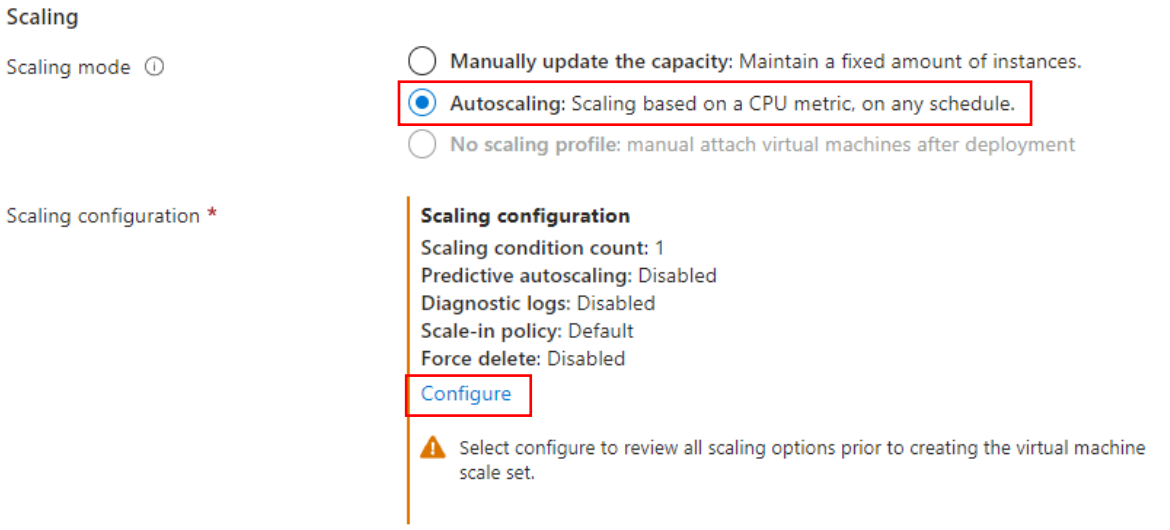

2. Now you need to choose your resource group, give your scale set a name, and choose your desired region. You can also choose the availability zone.



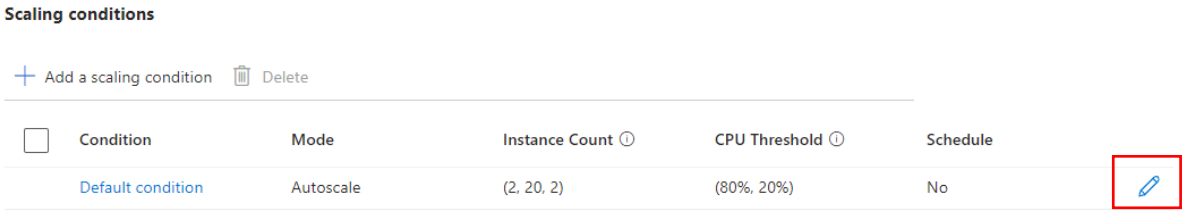

3. Then in the orchestration choose Uniform and keep the security type to default.



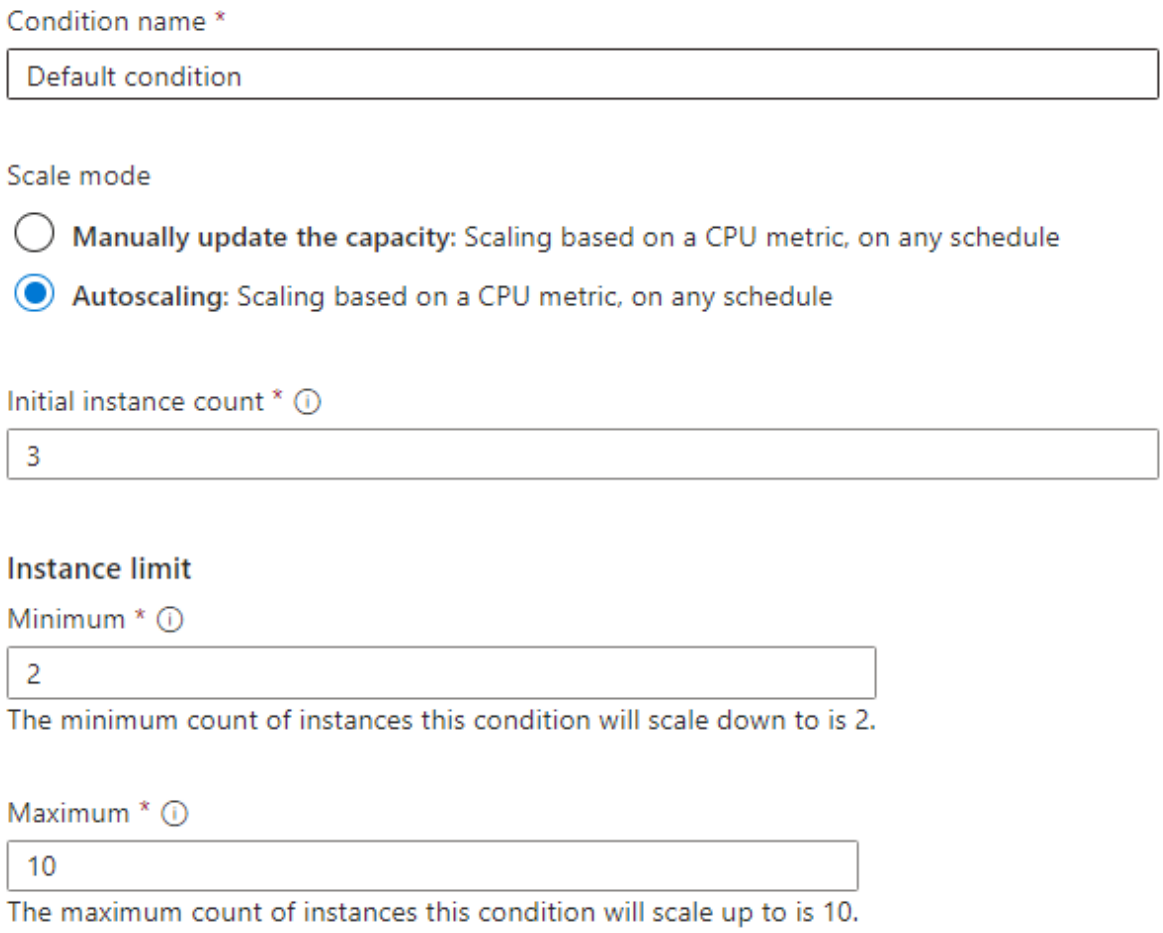

4. Now for scaling you need to choose Autoscaling and you need to click on Configure.



5. Then in the scaling condition you need to change the default condition and click on the highlighted pencil icon.




6. Now you need to follow the snapshots mentioned below and choose the same properties. Once you have chosen the properties then just click on save and you do not need to make any other changes so just click on save again.



### Scale out

CPU threshold greater than * ⓘ

80

Every time the average CPU usage is greater than 80%.

Increase instance count by * ⓘ

1

The condition will increase the instance count by 1 instances

### Scale in

CPU threshold less than * ⓘ

30

Every time the average CPU usage is less than 30%.

Decrease instance count by * ⓘ

1

The condition will decrease the instance count by 1 instances

### Query duration

Minutes * ⓘ

5

7. Now choose your operating system accordingly. Here we have chosen Windows OS. Then choose its size and move forward.

**Instance details**

Image * ⓘ — Windows Server 2019 Datacenter - x64 Gen2 (free services eligible)

See all images | Configure VM generation

VM architecture ⓘ — Arm64 / x64

Arm64 is not supported with the selected image.

Run with Azure Spot discount ⓘ

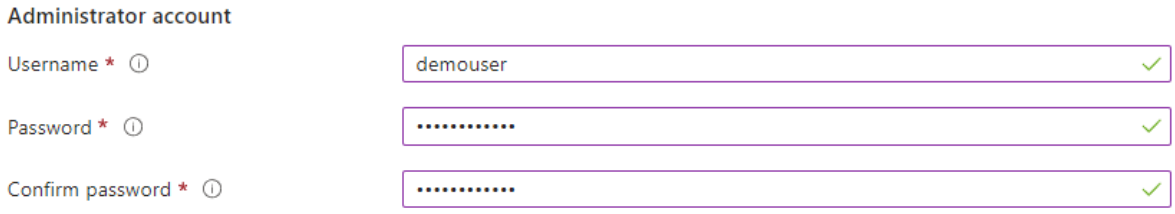
Size * ⓘ — Standard_B1ms - 1 vcpu, 2 GiB memory (₹1,603.33/month)

See all sizes

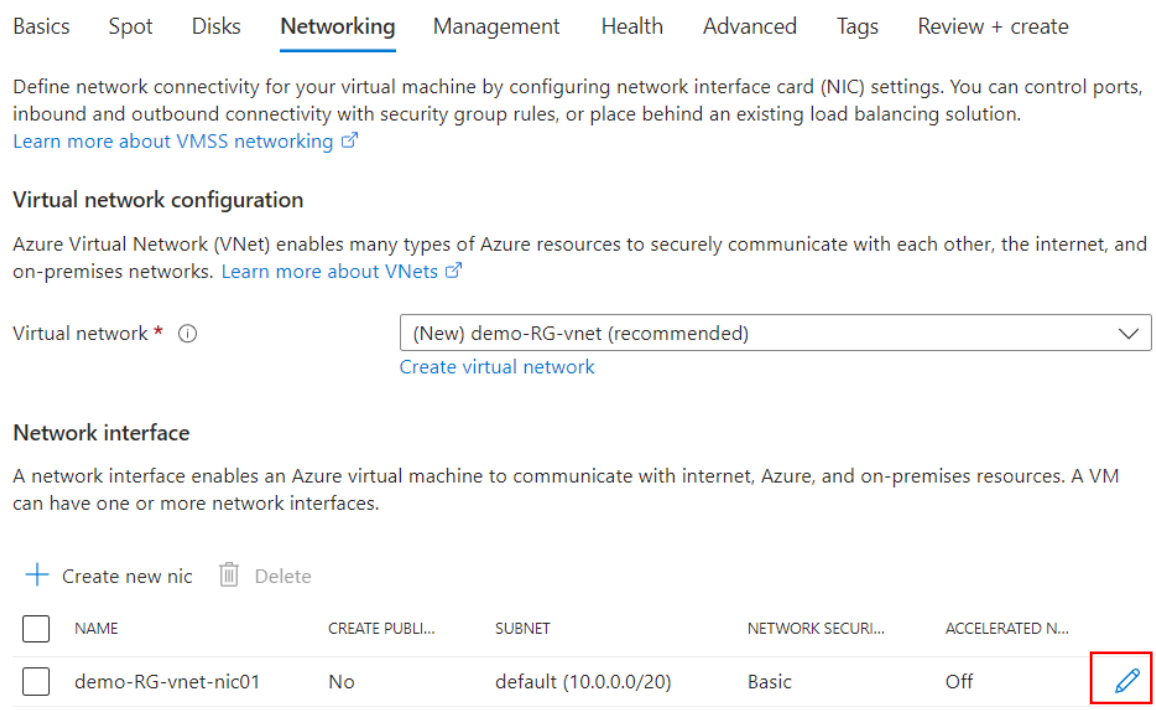
Enable Hibernation ⓘ

Hibernate does not currently support Uniform Orchestration mode. Learn more

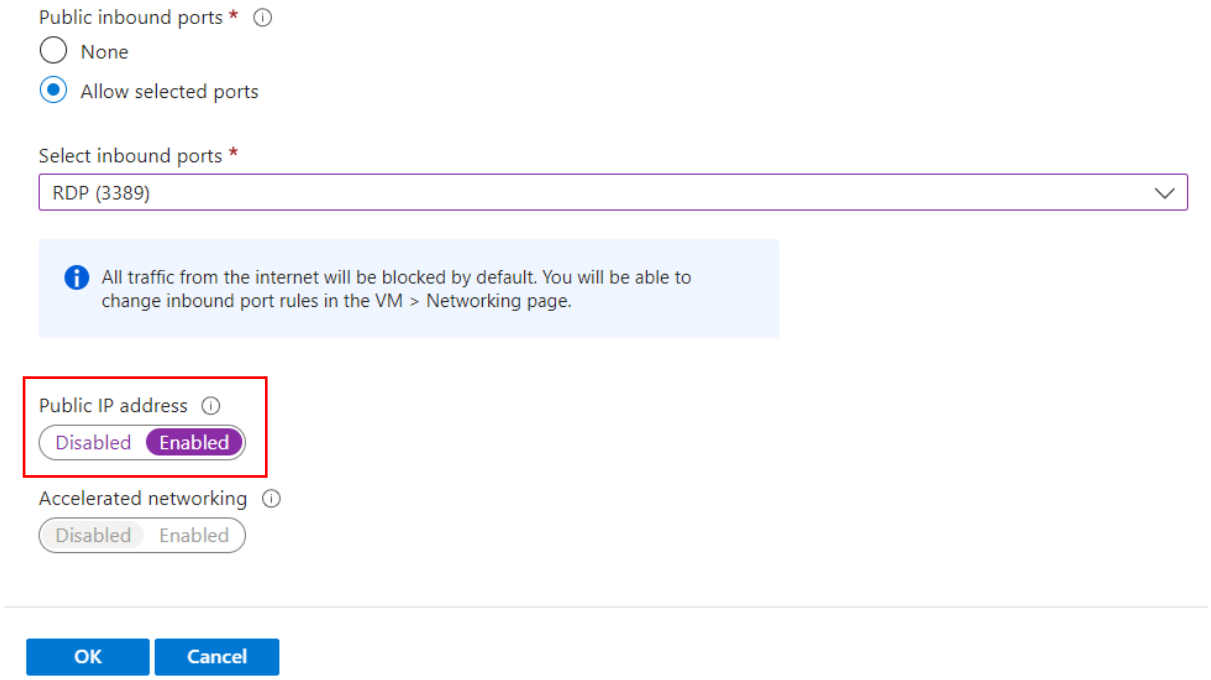
8. Then you need to give it a username and password.



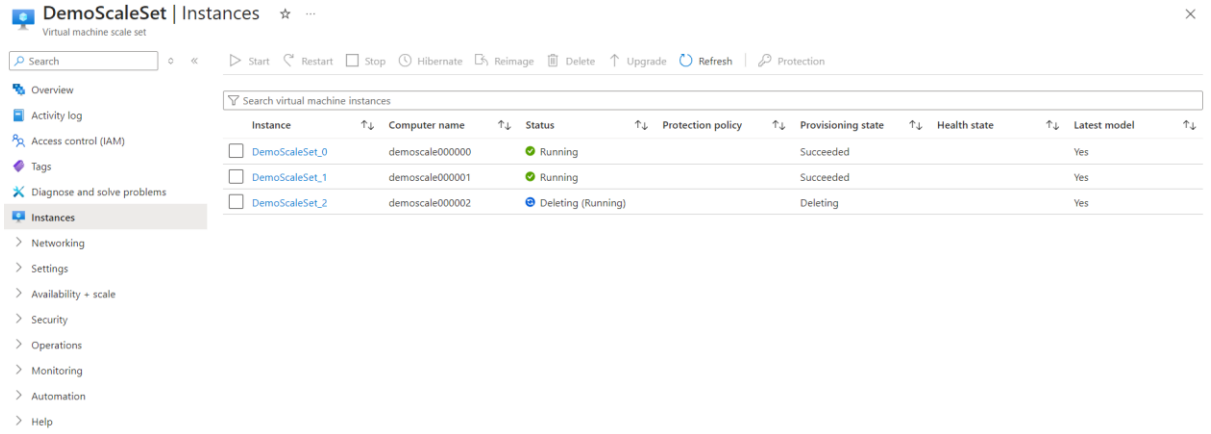

9. After that move to networking and click on the Pencil icon highlighted in the snapshot.




10. Now you need to allow the selected ports option and choose RDP port then you need to enable public IP address. So, this option will allow our Virtual machines to have separate public IP addresses which will help us to log in to them separately.
11. After that click on OK and move to the review page to create your VM Scale Set.




12. This will take some time to create. Once it is created go to it and then go to Instances. Here you will see your three instances but because the normal CPU threshold is below 30% our one CPU is being deleted.




13. Now if you login to your VM and increase the CPU threshold beyond 80% then the auto scale will start to created new VMs.