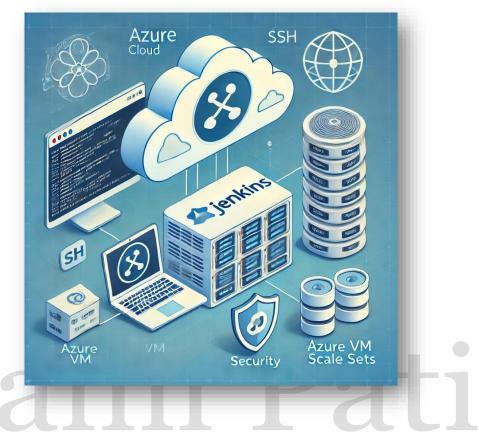
Week-4 | Deploy Jenkins on Azure VM | Azure VM Types Explained



Introduction

In this session, Abhishek provides an in-depth guide on deploying Jenkins on an Azure Virtual Machine (VM) while explaining the different Azure VM types and their practical applications. This session is crucial for DevOps engineers, cloud architects, and system administrators who want to understand Azure VM configurations, security best practices, networking setup, and Jenkins installation.

Azure Virtual Machines (VMs) provide scalability, flexibility, and cost-effectiveness for hosting applications, CI/CD tools, and cloud-based development environments. The session covers the process of deploying Jenkins on an Azure VM and configuring network security, SSH access, and Azure VM scale sets (VMSS) for auto-scaling.

Key Topics Covered

1. Understanding Virtualization in Azure

Before deploying a VM, it's important to understand **virtualization** and how Azure manages resources.

 Azure purchases physical servers and installs hypervisors to divide resources into multiple virtual machines.

- Virtualization Benefits:
 - Multi-tenancy: Multiple users can share the same infrastructure securely.

Resource Efficiency: Better utilization of physical hardware.

Cost Optimization: Pay for only what you use.

- o **High Availability**: Reduces hardware dependency risks.
- Azure's Virtualization Layers:
 - Physical Layer: Azure's data centers with physical servers.
 - Hypervisor Layer: Software that creates and manages virtual machines.
 Virtual Machine Layer: Users access VMs with assigned resources (CPU, RAM, storage).

2. Azure Virtual Machines: Pricing & Cost Management

Azure offers various VM pricing models to optimize cost and resource allocation.

- Azure Free Tier:
 - New users get 750 hours of free VM usage for 30 days.
 - After the free trial, a pay-as-you-go model applies.
- Azure Pricing Models:
 - Pay-as-you-go: Pay for the actual VM usage. Reserved Instances (RI):

 Pre-pay for 1 or 3 years at a discounted price.
 - Spot Instances: Discounted pricing for unused Azure capacity (best for non-critical workloads).
- Tip: Use Azure Pricing Calculator to estimate costs before deploying VMs.

3. Azure Virtual Machines (VMs) and Their Types

Azure provides different VM series optimized for specific workloads.

VM Series Purpose Use Cases

A-Series Low-cost, basic VMs Small applications, testing environments

B-Series Burstable VMs Development, proof-of-concept workloads

D-Series General-purpose Web applications, backend servers

E-Series Memory-optimized Databases, caching servers

F-Series Compute-optimized

High-performance computing (HPC)

VM Series Purpose

Use Cases

H-Series High-performance VMs AI, machine learning, simulations

L-Series Storage-optimized Big data analytics, databases

N-Series GPU-based VMs Graphics rendering, deep learning

Tip: Always choose a VM based on workload needs to avoid overpaying for unused resources.

4. Setting Up an Azure Virtual Machine

To deploy **Jenkins on Azure**, you first need to create a **virtual machine**.

Steps to Create an Azure VM:

- 1. Login to Azure Portal: https://portal.azure.com
- 2. Create a Resource Group:
 - o A resource group **organizes multiple Azure resources** together.
 - Best practice: Name the resource group according to project needs.
- 3. Create a New VM:
 - Choose Ubuntu 20.04 LTS for Jenkins.
 - Select an appropriate VM size (e.g., D2s_v3 for small workloads).
- 4. Configure Networking:
 - Allow SSH (port 22) for remote access.
 - Open port 8080 for Jenkins web access.
- 5. Generate & Download SSH Key:
 - Use Git Bash or PowerShell to generate an SSH key.
 - Secure your private key using chmod 400 mykey.pem.
- 6. Connect to VM using SSH:
- 7. ssh -i mykey.pem azureuser@<VM_Public_IP>

5. Installing Jenkins on Azure VM

Once the VM is set up, you can install **Jenkins**.

Step 1: Update System Packages sudo apt

update && sudo apt upgrade -y Step 2: Install

Java (Required for Jenkins) sudo apt install

openjdk-11-jdk -y

Step 3: Add Jenkins Repository & Install wget -q -O -

https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update sudo apt install

jenkins -y

Step 4: Start Jenkins & Enable Auto-start

sudo systemctl start jenkins sudo systemctl

enable jenkins Step 5: Allow Jenkins in

Firewall sudo ufw allow 8080 sudo ufw

enable

Step 6: Access Jenkins Web Interface

- Open a browser and visit:
- http://<VM_Public_IP>:8080
- Retrieve the **admin password**:
- sudo cat /var/lib/jenkins/secrets/initialAdminPassword · Complete the Jenkins setup wizard.

6. Securing the Azure VM

Security is critical when deploying services on the cloud.

Key Security Measures

- Use SSH keys instead of passwords for authentication.
- Disable root login in /etc/ssh/sshd_config.
- Enable firewall & limit open ports (sudo ufw status).
- Restrict public access to Jenkins (use a reverse proxy or VPN).

7. Introduction to Azure Virtual Machine Scale Sets (VMSS)

VM Scale Sets (VMSS) allow auto-scaling of VMs based on traffic load.

Why Use VMSS?

- Automatically adds or removes VMs based on CPU usage.
- Ensures high availability and load balancing.
- Cost-effective as resources scale dynamically.

Steps to Set Up VMSS

- 1. Create a VM Scale Set in Azure Portal.
- 2. **Configure autoscaling rules** (e.g., add instances if CPU > 70%).
- 3. **Attach a Load Balancer** to distribute traffic evenly.
- 4. **Enable auto-healing** (replaces unhealthy instances).



Final Thoughts & Key Takeaways

- Choosing the right Azure VM helps optimize performance and costs.
- ☑ Jenkins deployment requires proper networking and firewall settings.
- SSH security measures should be in place to prevent unauthorized access.
- VM Scale Sets (VMSS) provide auto-scaling for high-traffic applications.

Tip: VMSS is ideal for high-traffic web apps and microservices.