🔹 Azure Beginner Labs – Cheat Sheet

Lab 1: Create a Virtual Machine

1. Portal → Virtual Machines → + Create.
2. Basics: RG, Name, Region, Image (Linux/Windows).
3. Size: Select B1s (cheap).
4. Admin: Username + Password/SSH Key.
5. Networking: New VNet + Subnet.
6. Review + Create.
7. Connect:
   - Windows VM → RDP → mstsc.
   - Linux VM → SSH → ssh user@PublicIP.
8. Practice: Stop/Start, Resize → VM → Size.

👉 *Skills: Compute, Networking basics*

Lab 2: Virtual Network (VNet + Subnets)

1. Portal → Virtual networks → + Create.
2. Add 2 subnets: frontend, backend.
3. Create 2 VMs (each in different subnet).
4. Create NSGs:
   - Allow SSH (22) or RDP (3389).
   - Allow/block traffic between subnets.
5. Test by pinging/connecting between VMs.

👉 *Skills: Networking, Security*

Lab 3: Storage Account + Blob

1. Portal ➔ Storage accounts ➔ + Create.
2. Services ➔ Blob container ➔ + Create (public/private).
3. Upload file.
4. Right-click ➔ Generate SAS URL ➔ Test in browser.

👉 *Skills: Storage, Access Control*

Lab 4: File Share + VM Mount

1. In Storage Account ➔ File shares ➔ + Create.
2. On Windows VM ➔ Run net use Z: \\<storageaccount>.file.core.windows.net\<sharename> /u:<accountname> <key>.
3. Save/retrieve files from mounted drive (Z:).

👉 *Skills: VM Storage Integration*

Lab 5: Deploy Web App (App Service)

1. Portal ➔ App Services ➔ + Create.
2. Basics: RG, Name (unique), Runtime Stack.
3. Select App Service Plan ➔ Free or Basic.
4. Deploy sample: Deployment Center ➔ GitHub/Local.
5. Browse ➔ <appname>.azurewebsites.net.
6. Scaling ➔ App Service Plan ➔ Scale up.

👉 *Skills: PaaS, Deployment*

Lab 6: Azure SQL Database

1. Portal → SQL databases → + Create.
2. Server → New (admin user/pass).
3. Firewall: Allow Azure + Client IP.
4. Connect via Azure Data Studio / SSMS.
5. Run query: CREATE TABLE test (id INT, name NVARCHAR(50));.

👉 *Skills: Databases, Connectivity*

Lab 7: Load Balancer

1. Create 2 VMs (install IIS or Nginx).
   - IIS (Windows): Install-WindowsFeature -name Web-Server -IncludeManagementTools.
   - Nginx (Linux): sudo apt install nginx -y.
2. Create Public Load Balancer → Frontend IP.
3. Backend Pool → Add both VMs.
4. Health Probe → TCP:80.
5. Load Balancing Rule → Port 80.
6. Test in browser → LB Public IP.
7. Stop 1 VM → Check failover.

👉 *Skills: HA, Networking*

Lab 8: Azure Bastion

1. Portal → Bastions → + Create.
2. Virtual Network → must have subnet AzureBastionSubnet.
3. Public IP → Create new.
4. After deploy → Go to VM → Connect → Bastion.

5. Login (no public IP needed).

👉 *Skills: Secure Remote Access*

Lab 9: Azure Monitor + Alerts

1. Go to VM/App Service ➔ Monitoring ➔ Insights ➔ Enable.
2. Alerts ➔ + Create Alert Rule.
    - Scope: VM.
    - Condition: CPU % > 80.
    - Action: Email.
3. Simulate ➔ Stress test CPU (yes > /dev/null on Linux).

👉 *Skills: Monitoring, Ops*

Lab 10: Azure Backup & Recovery

1. VM ➔ Backup ➔ Enable.
2. Set Recovery Services Vault + Policy.
3. Run Backup Now.
4. Simulate recovery:
    - Restore Files ➔ Download agent + script.
    - Or Restore VM ➔ Create new VM from backup.

👉 *Skills: Backup, DR*

🔷 Azure Beginner Labs – Detailed Cheat Sheet

Lab 1: Create a Virtual Machine

Steps:

1. Portal → Search **Virtual Machines** → + Create.
2. **Basics Tab**:
   - RG: rg-vm-demo
   - Name: vm-demo
   - Region: (nearest)
   - Image: Windows Server 2019 / Ubuntu 22.04
   - Size: B1s (cheap).
3. **Admin**: Username + Password (or SSH Key for Linux).
4. **Disks**: Standard HDD (for cost).
5. **Networking**:
   - New VNet + Subnet
   - Public IP → Enabled
   - NSG → Allow RDP (3389) or SSH (22).
6. Review + Create.
7. **Connect**:
   - Windows → RDP: mstsc → IP → login.
   - Linux → SSH: ssh azureuser@<PublicIP>.
8. **Practice**:
   - Stop/Start VM → see IP change.
   - Resize → VM → Size → pick another SKU.

👉 Skills: Compute, Networking basics

Lab 2: Virtual Network (VNet + Subnets)

Steps:

1. Portal → Search **Virtual networks** → + Create.
   - Name: vnet-demo
   - Address space: 10.0.0.0/16
   - Add Subnet 1: frontend → 10.0.1.0/24
   - Add Subnet 2: backend → 10.0.2.0/24.
2. Create **2 VMs**:
   - VM-frontend → in subnet frontend.
   - VM-backend → in subnet backend.
3. Create **NSG rules**:
   - Allow RDP (3389) or SSH (22).
   - Block all traffic between subnets (default deny).
   - Create allow rule for ICMP or RDP between them.
4. Test:
   - From frontend → try to ping backend.
   - Apply rules to allow/deny traffic.

👉 Skills: Networking, Security

Lab 3: Storage Account + Blob

Steps:

1. Portal → Search **Storage accounts** → + Create.
   - Name: stgdemolab.
   - Region: same as RG.
   - Performance: Standard.

   - Redundancy: LRS.
2. After creation → Go to **Blob containers** → + Container.
   - Name: testcontainer.
   - Public access: Private (default).
3. Upload file (test.txt).
4. Generate **SAS URL**:
   - Right-click → **Generate SAS** → Permissions: Read.
   - Copy URL → test in browser.

👉 Skills: Storage, Access Control

Lab 4: File Share + VM Mount

Steps:

1. In Storage Account → **File shares** → + Create.
   - Name: myshare.
   - Quota: 5 GB.
2. Go to your **Windows VM** → Connect via RDP.
3. Run PowerShell:
4. net use Z: \\stgdemolab.file.core.windows.net\myshare /u:stgdemolab <storage-key>

(Find **Access Key** under Storage → Access Keys).

5. Save a file into Z: drive.
6. Verify file is visible in Azure Portal.

👉 Skills: VM + Storage integration

Lab 5: Deploy Web App (App Service)

Steps:

1. Portal → **App Services** → + Create.
2. Basics:
    - Name: mywebappdemo123 (must be unique).
    - Runtime: Node.js / .NET / Python.
    - Publish: Code.
    - App Service Plan: Free (F1).
3. Deployment Center → GitHub or Local Git.
    - Or choose sample → "Starter app".
4. After deploy → browse URL:
    - https://mywebappdemo123.azurewebsites.net.
5. Scaling:
    - App Service → Scale up (change plan).

👉 Skills: PaaS, Deployment

Lab 6: Azure SQL Database

Steps:

1. Portal → **SQL Databases** → + Create.
    - DB Name: sqldemodb.
    - Create new SQL server (name, admin login, password).
2. Networking:
    - Allow Azure services.
    - Add client IP.
3. Create.
4. Install **Azure Data Studio / SSMS** → connect with server + login.
5. Run queries:
6. CREATE TABLE users (id INT PRIMARY KEY, name NVARCHAR(50));

7. INSERT INTO users VALUES (1, 'John Doe');
8. SELECT * FROM users;

👉 Skills: Databases, Connectivity

Lab 7: Load Balancer

Steps:

1. Create **2 VMs** in same VNet.
   - Install web server:
     - Windows: Install-WindowsFeature -name Web-Server -IncludeManagementTools.
     - Linux: sudo apt install nginx -y.
   - Update index.html on each VM with unique text (VM1 / VM2).
2. Create **Public Load Balancer**:
   - Frontend IP → New.
   - Backend Pool → Add both VMs (with NICs).
   - Health Probe → Port 80.
   - Rule → Port 80 → Backend Pool.
3. Test → Browse Load Balancer Public IP.
   - Refresh → see traffic switching.
4. Stop 1 VM → test failover.

👉 Skills: HA, Networking

Lab 8: Azure Bastion

Steps:

1. Portal → **Bastions** → + Create.
   - Virtual network → must have subnet named AzureBastionSubnet.

    - Assign Public IP.

2. Deploy.
3. Go to VM → Connect → Bastion.
4. Login directly (no public IP on VM).

👉 Skills: Secure Remote Access

Lab 9: Azure Monitor + Alerts

Steps:

1. Go to VM → Monitoring → Insights → Enable.
2. Alerts → + Create Rule.
    - Scope: This VM.
    - Condition: Metric → CPU % → > 80.
    - Action: Email (create Action Group).
3. Trigger test:
    - Linux: yes > /dev/null (CPU stress).
    - Windows: Run PowerShell loop.
4. Check email notification.

👉 Skills: Monitoring, Operations

Lab 10: Azure Backup & Recovery

Steps:

1. Go to VM → Operations → **Backup** → Enable.
    - Create Recovery Services Vault.
    - Assign backup policy (daily snapshot).
2. Run Backup Now.
3. Simulate Recovery:

    - **File Recovery** → Download script + mount as disk.
    - **VM Recovery** → Restore VM → New VM from backup.

👉 Skills: Backup, Disaster Recovery