

AZURE PROJECT

WEB & APP DEPLOYMENT WITH LOAD BALANCERS

Submitted by:

Rachamalla Bindu

Objective :

- Create Azure resources (RG, VNet, subnets)
- Deploy two Linux VMs named web and app
- Configure NSGs to allow ports 22, 80 (both VMs) and 8080 (app VM)
- Install Nginx on web, Tomcat on app
- Deploy Public Load Balancer for Web (HTTP) and Private Load Balancer for App (8080)
- Verify end-to-end connectivity

Abstract :

- Deploy an Azure Resource Group containing a VNet with two subnets, two VMs (Web and App), Security Rules allowing SSH(22), HTTP(80), and App(8080).
- Install Nginx on Web VM and Tomcat on App VM.
- Create a public load balancer for the Web tier (port 80) and an internal/private load balancer for the App tier (port 8080).
- Verify connectivity by accessing public LB IP and confirming Nginx can reach the App through the private LB.

➤ 1. Create a Resource group :

Home > Resource Manager | Resource groups >

Create a resource group ...

Basics Tags Review + create

Automation Link

Basics

Subscription Azure subscription 1
Resource group name binduRG01
Region Central India

Tags

None

Previous

Next

Create

➤ 2. Create a Virtual Network with Two Subnets :

Home > Network foundation | Virtual networks >

Create virtual network ...

Basics Security IP addresses Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure subscription 1
Resource group * binduRG01
Create new

Instance details

Virtual network name * Vnet01
Region * (Asia Pacific) Central India
Deploy to an Azure Extended Zone

Previous

Next

Review + create

- Subnet 1 : WEBsubnet
- Subnet2 : Appsubnet

The screenshot shows the 'Create virtual network' wizard in the Microsoft Azure portal. The 'IP addresses' tab is selected. A summary table shows the main address space 10.1.0.0/16 with a range of 10.1.0.0 - 10.1.255.255 and 65,536 addresses. Below it, two subnets are listed: 'WEBsubnet' with a range of 10.1.0.0 - 10.1.0.255 and 'APPsubnet' with a range of 10.1.1.0 - 10.1.1.255, both with a size of /24 (256 addresses). Buttons for 'Previous', 'Next', and 'Review + create' are at the bottom.

➤ 3.Create WEB Virtual Machine :

- OS: Linux (Ubuntu)
- Subnet: Web Subnet
- Public IP: Yes (for SSH access)

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Instance details' step is active. It includes fields for 'Virtual machine name' (binduWEB01), 'Region' ((Asia Pacific) Central India), 'Availability options' (No infrastructure redundancy required), 'Security type' (Standard), and 'Image' (Ubuntu Server 22.04 LTS - x64 Gen2 (free services eligible)). Buttons for '< Previous', 'Next : Disks >', and 'Review + create' are at the bottom.

- Allow public inbound rules HTTP (80), SSH (22)

Virtual network* ⓘ Vnet01
Create new

Subnet* ⓘ WEBsubnet (10.1.0.0/24)
Manage subnet configuration

Public IP ⓘ (new) binduWEB01-ip
Create new
Public IP addresses have a nominal charge. [Estimate price](#)

NIC network security group ⓘ None (Basic is selected)

< Previous | Next : Management > | Review + create

- Here the WEBvm is created in virtual network in web subnet.

➤ 4. Install NGINX on Web VM :

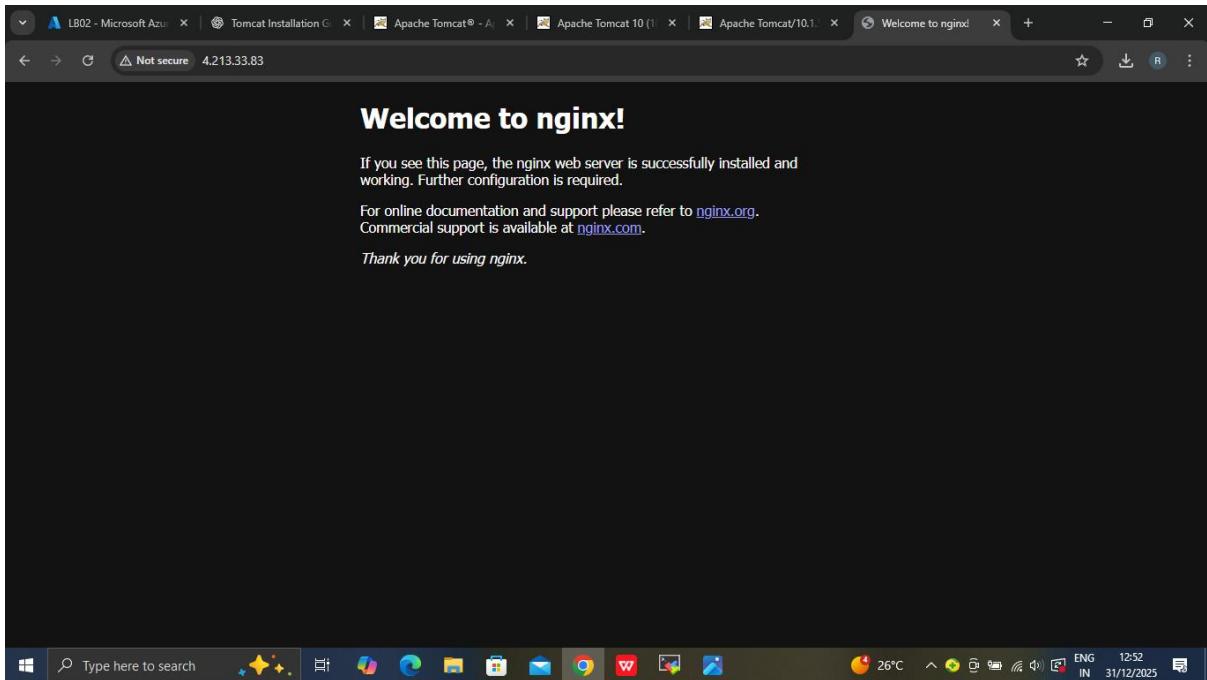
- By using below commands install nginx in mobaxterm
- Sudo su (to switch to root user)
- sudo apt update
- sudo apt install nginx -y
- sudo systemctl start nginx

```
bindu05@binduWEB01:~$ sudo su
root@binduWEB01:/home/bindu05# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@binduWEB01:/home/bindu05# apt install nginx -y
```

- The successful installation of NGINX was verified by checking the service status using the systemctl command, which showed the service as active and running in the above image.

➤ 5. Verify NGINX :

- Browse Web VM public IP
- NGINX default page should be displayed



➤ 6. Create Application Virtual Machine (App VM) :

- OS: Linux (Ubuntu)
- Subnet: App Subnet
- Public IP: Optional (recommended only for admin access)

- Allow inbound rules SSH (22)

➤ 7. Install Tomcat on App VM :

By using below commands install Tomcat in mobaxterm :

- Sudo su (to switch to root user)
- sudo apt update
- sudo apt install openjdk-11-jdk -y
- wget <https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.50/bin/apache-tomcat-10.1.50.tar.gz>
- tar -xvzf apache-tomcat-10.1.50.tar.gz
- mv apache-tomcat-10.1.50 tomcat
- ls
- cd /opt/tomcat/bin
- ./startup.sh

```

terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
Session 3. 4.247.162.178 (bindu05) 2. 4.213.179.21 (bindu05)
/home/bindu05/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
apache-tomcat-10.1.50/webapps/manager/images/asf-logo.svg
apache-tomcat-10.1.50/webapps/manager/images/favicon.ico
apache-tomcat-10.1.50/webapps/manager/images/tomcat.svg
apache-tomcat-10.1.50/webapps/manager/index.jsp
apache-tomcat-10.1.50/webapps/manager/status.xsl
apache-tomcat-10.1.50/webapps/manager/xform.xls
apache-tomcat-10.1.50/bin/bootstrap.sh
apache-tomcat-10.1.50/bin/ciphers.sh
apache-tomcat-10.1.50/bin/configtest.sh
apache-tomcat-10.1.50/bin/daemon.sh
apache-tomcat-10.1.50/bin/digest.sh
apache-tomcat-10.1.50/bin/makebase.sh
apache-tomcat-10.1.50/bin/migrate.sh
apache-tomcat-10.1.50/bin/setclasspath.sh
apache-tomcat-10.1.50/bin/startup.sh
apache-tomcat-10.1.50/bin/tool-wrapper.sh
apache-tomcat-10.1.50/bin/version.sh
root@binduAPP01:/opt# ls
apache-tomcat-10.1.50 apache-tomcat-10.1.50.tar.gz
root@binduAPP01:/opt# mv apache-tomcat-10.1.50 tomcat
root@binduAPP01:/opt# ls
apache-tomcat-10.1.50 tomcat
root@binduAPP01:/opt# cd tomcat
bash: /opt/tomcat: Is a directory
root@binduAPP01:/opt# cd tomcat/bin
root@binduAPP01:/opt/tomcat/bin# chmod +x *.sh
root@binduAPP01:/opt/tomcat/bin# ./startup.sh
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@binduAPP01:/opt/tomcat/bin# 

```

Remote monitoring Follow terminal folder

binduAPP01 0% 0.64 GB / 7.76 GB 0.01 Mb/s 0.00 Mb/s 38 min bindu05 (x2) 0% /boot/efi: 6%

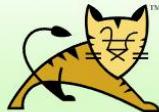
- Apache Tomcat was successfully installed and started, and the default Tomcat welcome page was verified by accessing it via the VM public IP on port 8080.

Not secure 4.247.162.178:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/10.1.50

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
\$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 10.1 access to the manager application is split between different users.
[Read more...](#)

Documentation
[Tomcat 10.1 Documentation](#)
[Tomcat 10.1 Configuration](#)
[Tomcat Wiki](#)
Find additional important configuration information in:
\$CATALINA_HOME/ RUNNING.txt
Developers may be interested in:

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:

tomcat-announce Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users User support and discussion
taglibs-user User support and discussion for Apache Taglibs

➤ 8. Create Azure Public Load Balancer :

A. Select standard load balancer and Create .

No load balancers to display

Azure Load Balancer enables your applications to be highly available and scalable. You can scale up and down based on your traffic patterns. Azure Load Balancer is best suited for network traffic requiring high performance and ultra-low latency.

[+ Create](#)

- Now in type select **Public** because it is public load balancer.

B. Frontend IP Configuration .

C. Backend Pool: Add Web VM .

Add IP configurations to backend pool

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

Resource Name	Resource group	Type	IP configurati...	IP Address	Availability set	Tags
binduAPP01	binduRG01	Virtual machine	ipconfig1	10.1.1.4	-	-
<input checked="" type="checkbox"/> binduWEB01	binduRG01	Virtual machine	ipconfig1	10.1.0.4	-	-

IP configurations

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

+ Add | × Remove

Save Cancel Give feedback

D. Health Probe :

- Protocol: TCP
- Port: 80

Add health probe

Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Name *	HP01
Protocol *	TCP
Port * (i)	80
Interval (seconds) * (i)	5
Used by * (i)	Not used

Save Cancel Give feedback

E. Load Balancing Rule:

- Frontend Port: 80
- Backend Port: 80
- Backend Pool: Web VM

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/-) | Copilot | ... | charanpoola13@gmail.com | DEFAULT DIRECTORY (CHARAN)

Home > CreateLoadBalancerBladeV2-20260101173334 | Overview > publicLB | Load balancing rules > Add load balancing rule

Name * LBrule01

IP version * IPv4

Frontend IP address * myfrontendIP (135.235.252.234)

Backend pool * B-pool

Protocol TCP

Port * 80

Backend port * 80

Health probe * HP01 (TCP:80)

Save **Cancel** **Give feedback**

➤ 9. Create Azure Private Load Balancer :

A. Create Private Load Balancer

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/-) | Copilot | ... | charanpoola13@gmail.com | DEFAULT DIRECTORY (CHARAN)

Home > Load balancing and content delivery | Load balancers > Create load balancer

Project details

Subscription * Azure subscription 1

Resource group * binduRG01

Create new

Instance details

Name * privateLB

Region * Central India

SKU * Standard (Distribute traffic to backend resources)

Type * Internal

Tier * Regional

Review + create **< Previous** **Next : Frontend IP configuration >** **Download a template for automation** **Give feedback**

Here in type select internal because it is private load balancer .

B. Frontend IP Configuration:

- Private IP from AppSubnet

Add frontend IP configuration

Name * myfrontendIP02

IP version IPv4

Virtual network * Vnet01

Subnet * APPsubnet (10.1.0.0/24)

Assignment Dynamic

Availability zone * Zone-redundant

C. Backend Pool : Add App VM

Add IP configurations to backend pool

Resource Name	Resource group	Type	IP configurati...	IP Address	Availability set	Tags
binduAPP01	binduRG01	Virtual machine	ipconfig1	10.1.1.4	-	-
binduWEB01	binduRG01	Virtual machine	ipconfig1	10.1.0.4	-	-

D. Health Probe:

- Protocol: TCP
- Port: 8080

Home > Load balancing and content delivery | Load balancers > privateLB | Health probes >

Add health probe

Name * HP-02

Protocol * TCP

Port * 8080

Interval (seconds) * 5

Used by * Not used

Save **Cancel** **Give fee**

E. Load Balancing Rule:

- Frontend Port: 8080
- Backend Port: 8080
- Backend Pool: App VM

Home > Load balancing and content delivery | Load balancers > privateLB | Load balancing rules >

Add load balancing rule

Name * LBrule02

IP version * IPv4

Frontend IP address * myfrontendIP02 (10.1.1.5)

Backend pool * B-pool2

High availability ports

Protocol TCP

Port * 8080

Backend port * 8080

Save **Cancel** **Give fee**

➤ 10. Connectivity Testing (Main Validation Step) :

- Test connection from Web VM to Private Load Balancer
- Connect to Web VM using SSH
- With WEB virtual machine public Ip and username.
- Run telnet command.

```

root@binduWEB01:/home/bindu05# cd /var/www/html/
root@binduWEB01:/var/www/html# ls
index.nginx-debian.html
root@binduWEB01:/var/www/html# telnet 10.1.1.5:8080
telnet 10.1.1.5:8080: command not found
root@binduWEB01:/var/www/html# telnet 10.1.1.5 8080
Trying 10.1.1.5...
Connected to 10.1.1.5.
Escape character is '['.
Connection closed by foreign host.
root@binduWEB01:/var/www/html#

```

This confirms:

- Traffic flows from the public load balancer to the web VM and that Nginx can reach Tomcat via the private load balancer.

➤ Security Validation :

- Tomcat is not exposed publicly.
- Only Private Load Balancer IP is accessible from Web VM.
- Application layer is protected inside the VNet.
- End users cannot directly access Tomcat from the internet.
- This prevents unauthorized external access to the application layer.
- External traffic cannot reach the Private Load Balancer.
- Only the Web VM (NGINX) is exposed through the Public Load Balancer.
- Backend Application services remain isolated from the internet .

➤ Conclusion:

- In this project, we successfully implemented Azure Public and Private Load Balancers to design a secure two-tier architecture. The Public Load Balancer exposes the Web layer (NGINX) to the internet, while the Private Load Balancer ensures secure internal communication with the Application layer (Tomcat).
- Connectivity was validated using telnet, confirming proper backend pool configuration and network security. This architecture follows real-time industry best practices for scalability, security, and high availability.

