

# Application Gateway

- It is mainly works on url / path based routing
- Application gateway mainly we use for routing the traffic to the web pages that pages maintained in the different servers.
- Creating a virtual machine for Home page

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Project details' section is visible, showing 'Subscription' set to 'Azure subscription 1' and 'Resource group' set to 'sai-rg'. The 'Instance details' section includes fields for 'Virtual machine name' (set to 'HomePageServer'), 'Region' (set to '(Asia Pacific) Central India'), 'Availability options' (set to 'No infrastructure redundancy required'), 'Security type' (set to 'Standard'), and 'Image' (set to 'Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible)'). At the bottom, there are buttons for 'Next : Disks >' and 'Review + create'.

- Creating and selecting Vnet and subnet for the instance.
- Here we created two subnets one is for maintaining Vm's.
- We dedicate one subnet for the Application gateway.

The screenshot shows the 'Edit subnet' configuration page in the Microsoft Azure portal. On the left, the 'Subnets' table lists two subnets: 'subnet-alb' and 'subnet-instances'. The 'subnet-alb' row shows an IP address range of '172.16.0.0 - 172.16.255.255' and a size of '/16'. On the right, the 'Edit subnet' form is open, showing 'Subnet ID' as '/subscriptions/a3107e0d-376f-4650-b283...', 'Subnet purpose' as 'Default', and 'Name' as 'subnet-alb'. Under 'IPv4', 'Include an IPv4 address space' is checked, and the 'IPv4 address range' is set to '172.16.0.0 - 172.16.255.255'. Under 'IPv6', 'Include an IPv6 address space' is unchecked. A note at the bottom states 'Private subnet' with the subtext 'Private subnets enhance security by not enabling default outbound access. To enable...'. At the bottom right, there are 'Save' and 'Cancel' buttons.

- Home page server is created and installed nginx on the home page server.

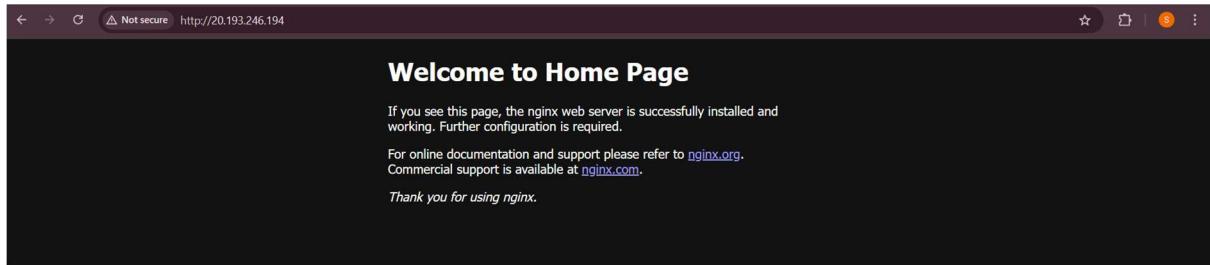
HomePageServer

Standard HDD OS disks will be retired on September 8, 2028.

**Properties**

Virtual machine	Networking
Computer name: HomePageServer	Public IP address: 4.213.17.131 (Network interface homepageserver155)
Operating system: Linux (ubuntu 24.04)	1 associated public IPs
VM generation: V2	Public IP address (IPv6): -
VM architecture: x64	Private IP address: 172.16.0.4
Agent status: Ready	Private IP address (IPv6): -
Agent version: 2.15.0.1	Virtual network/subnet: vnet-alb/subnet-alb
Hibernation: Disabled	DNS name: Configure
Host group: -	Size: Standard D2s v5
Host: -	vCPUs: 2
Proximity placement group: -	RAM: 4 GiB
Colocation status: N/A	
Capacity reservation group: -	
Disk controller type: SCSI	

→ Installed nginx and browse with public ip.



→ Creating a second Virtual machine for personal page.

→ And assign this vm to the same vnet and with the same subnet.

→ Extra subnet is for the application gateway.

Create a virtual machine

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 \* (sai-rg)

Virtual machine name \* (PersonalPageServer)

Region \* ((Asia Pacific) Central India)

Availability options \* (No infrastructure redundancy required)

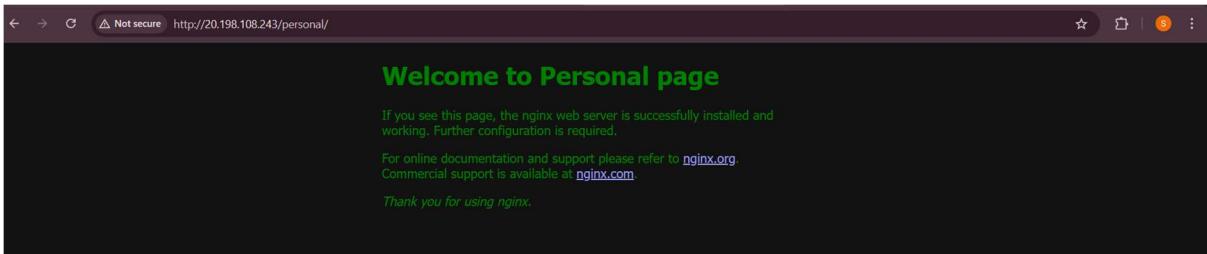
Security type (Standard)

Image \* (Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible))

This image is compatible with additional security features. Click here to swap to the Trusted launch security type.

< Previous Next : Disks > Review + create Give feedback

- Installed nginx for second Vm and change the default path.
- /var/www/html → /var/www/html/personal
- Changing the text to knowing it is personal page



- Search for the application gateway in the search bar.
- Select Application Gateway and click on create.

- Adding basic details for the application gateway.
- Selecting Resource group and add name for the application gateway.
- Select region where our Vm's created.
- Here I can't enable autoscaling.

→ Selecting Virtual network.

→ And Select the dedicated subnet to the Application Gateway.

Configure virtual network

Virtual network \* ⓘ vnet-alb Create new

Subnet \* ⓘ Subnet-AzAlb (172.16.2.0/24) Manage subnet configuration

Previous Next : Frontends >

→ Next in the Frontend Ip's We add the new public ip to access the azalb.

→ Free subscription can only assign 3 public ip's for a region. Make sure we can maintain it.

Home Create application gateway ...

⚠ Changes you make on this tab may affect any configuration you've done on other tabs. Review all options prior to creating the application gateway.

✓ Basics ⚡ Frontends ⓘ Backends ⓘ Configuration ⓘ Tags ⓘ Review + create

Traffic enters the application gateway via its frontend IP address(es). An application gateway can use a public IP address, private IP address, or one of each type. ⓘ

Frontend IP address type ⓘ

Frontend IPv4 address \* PersonalPageServer-ip (20.198.108.243)

Add new

Add a public IP

Name \* Azalbip

SKU Basic Standard

Assignment Dynamic Static

Availability ZoneRedundant zone

OK Cancel

Previous Next : Backends >

→ Adding backend pool for the application gateway.

→ Here the home pool contains one vm.

→ Personal pool contains another vm.

Microsoft Azure ⓘ Upgrade ⌂ Search resources, services, and docs (G+)

lsk2026@outlook.com ⓘ DEFAULT DIRECTORY (LSK2026...)

Home Create application gateway ...

✓ Basics ✓ Frontends ⚡ Backends ⓘ Configuration ⓘ Tags ⓘ Review + create

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN). ⓘ

Add a backend pool

Backend pool Targets

homePool > 1 target ...

Add a backend pool

Name \* PersonalPool

Add backend pool without targets Yes No

Backend targets

1 Item

Target type Target

Virtual machine personalpageserver997 (172.16...)

IP address or FQDN

Add Cancel

Previous Next : Configuration >

→ Here adding the routing rule.

→ In our case we use http and the port 80

The screenshot shows the Azure portal interface for creating an Application Gateway. On the left, there's a navigation pane with 'Home', 'Create application gateway', 'Basics', 'Frontends', 'Backends', 'Configuration' (which is selected), 'Tags', and 'Review + create'. Below this, it says 'Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second...'. There's a 'Frontends' section with a 'Public: (new) Azalbip' entry and a '+ Add a frontend IP' button. On the right, a modal window titled 'Add a routing rule' is open. It has a description: 'Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.' It asks for 'Rule name \*' (azalrule), 'Priority \*' (100), and 'Listener \* Backend targets'. Under 'Listener', it shows 'listener1' (Protocol: HTTP, Port: 80, Listener type: Basic). It also includes sections for 'Custom error pages' and 'Bad Gateway - 502' (Enter Html file URL). At the bottom of the modal are 'Add' and 'Cancel' buttons.

→ In the backend targets we add home pool and home settings for default path.

This is a detailed view of the 'Add a routing rule' dialog. It starts with the configuration for a 'routing rule' (Rule name: azalrule, Priority: 100). It then moves to the 'Listener' section, where 'Backend targets' are selected. The 'Target type' is set to 'Backend pool' (selected over 'Redirection'). Under 'Backend target', 'homePool' is chosen, with 'Add new' and 'Homepoolsettings' options available. The 'Path-based routing' section follows, stating that traffic can be routed based on URL paths. A table for 'Path based rules' is shown, with a single entry: Path '/personal/\*', Target name 'Personal', Backend setting name 'personalsetting', and Backend pool 'PersonalPool'. At the bottom are 'Add' and 'Cancel' buttons.

→ Click on the Add multiple targets to create a path based rule

→ Add the path and target name and backend settings and backend target.

This screenshot shows the 'Azalbroutingrule' configuration dialog. It has a back arrow with the text '← Discard changes and go back to routing rules'. The 'Target type' is set to 'Backend pool' (selected over 'Redirection'). The 'Path' is '/personal/\*'. The 'Target name' is 'personal'. The 'Backend settings' is 'personalsetting'. The 'Backend target' is 'Personalpool'. All fields have green checkmarks indicating they are valid. At the bottom are 'Add' and 'Cancel' buttons.

→ Finally review and click on create the Application gateway looks like.

The screenshot shows the Azure portal's Resource Manager interface for an Application gateway named 'Sai-Azalb'. In the 'Essentials' section, it lists the resource group as 'sai-rg', location as 'Central India (Zone 1, 2, 3)', and subscription as 'Azure subscription 1'. It also shows the virtual network/subnet as 'vnet-alb/Subnet-AzAlb', frontend public IP address as '20.204.184.165 (albip)', and tier as 'Standard V2'. The availability zone is set to '1, 2, 3'. A prominent green checkmark icon is displayed at the bottom right, indicating that the resource is following light best practices.

→ Add health probes for both.

→ home probe for the firstVm

This screenshot shows the 'Add health probe' dialog box. The 'Name' field is set to 'homeprobe'. The 'Protocol' field is set to 'HTTP'. The 'Host' field contains '20.193.246.194'. The 'Path' field is set to '/'. The 'Interval (seconds)' is set to 30. The 'Timeout (seconds)' is set to 30. The 'Unhealthy threshold' is set to 3. The 'Use probe matching conditions' section has 'No' selected. The 'Backend settings' dropdown is set to 'HomeSettings'. A checkbox at the bottom left is checked, stating 'I want to test the backend health before adding the health probe'. There are 'Test' and 'Cancel' buttons at the bottom.

→ Personal health probe here we can set the path to the personal page.

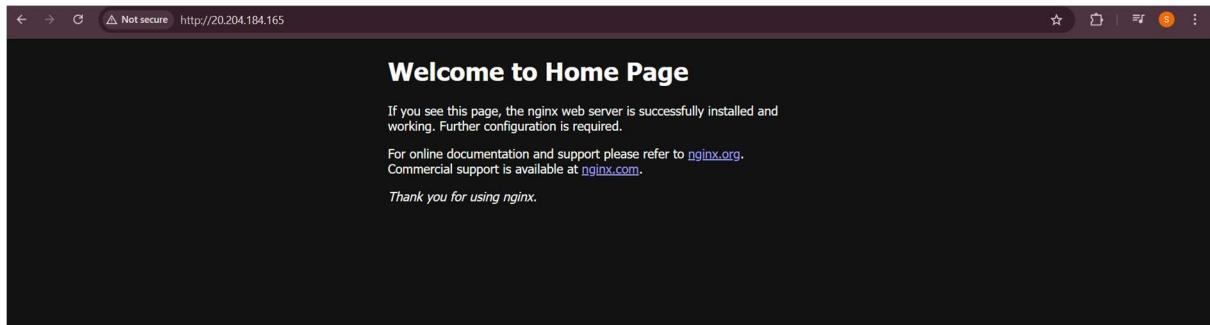
This screenshot shows the configuration for a 'personal' health probe. The 'Name' field is set to 'personal'. The 'Protocol' field is set to 'HTTP'. The 'Host' field contains '20.198.108.243'. The 'Path' field is set to '/personal/'. The 'Interval (seconds)' is set to 30. The 'Timeout (seconds)' is set to 30. The 'Unhealthy threshold' is set to 3. The 'Use probe matching conditions' section has 'Yes' selected. The 'HTTP response status code match' field is set to '200-399'. The 'HTTP response body match' field is empty. A checkbox at the bottom left is checked, stating 'I want to test the backend health before adding the health probe'. There are 'Test' and 'Discard' buttons at the bottom.

→ Take the public ip address of the application gateway and search in the browser.

→ In the default path home page we see.

→ In the /personal path we see personal page.

→ This is the home page for the application gateway.



→ This is the personal page in the application gateway.

