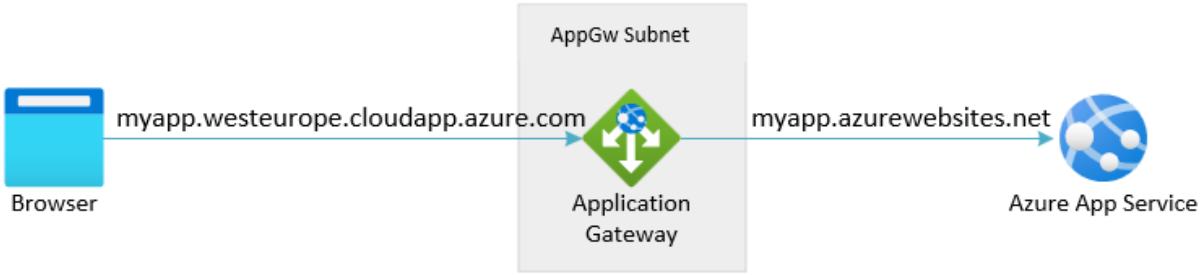
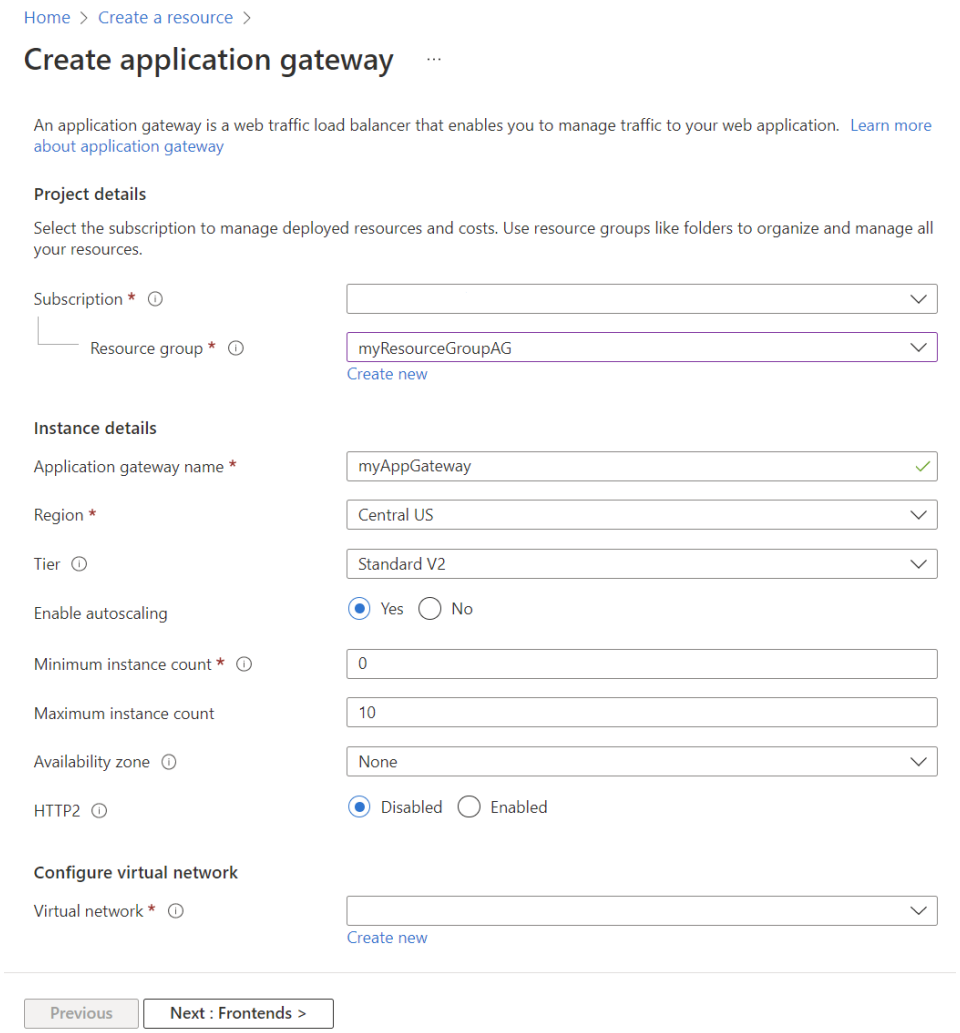
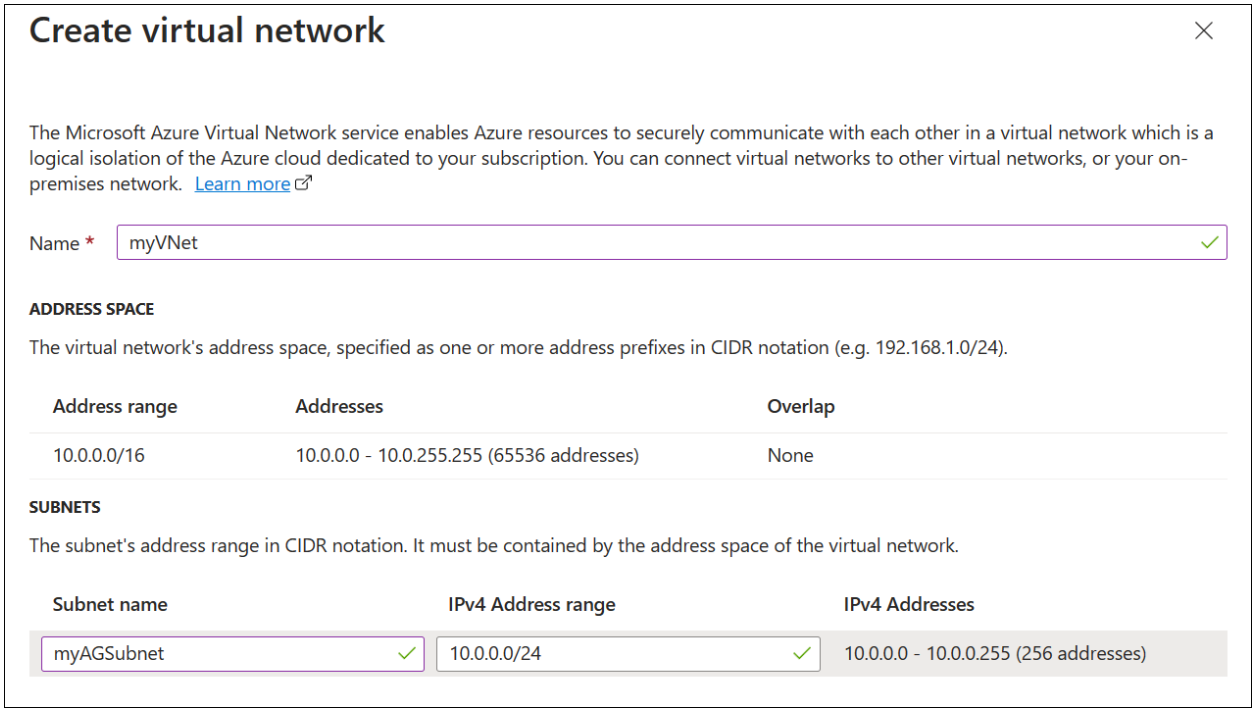


Create an application gateway

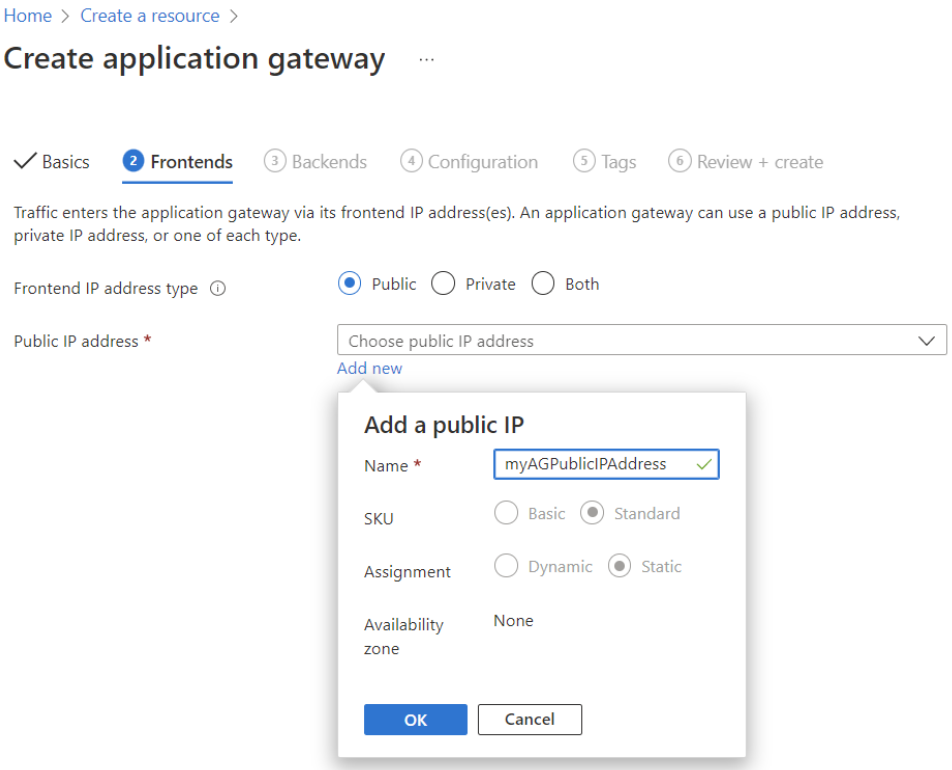
1. On the Azure portal menu or from the Home page, select Create a resource.
2. Under Categories, select Networking and then select Application Gateway in the Popular Azure services list.
3. On the Basics tab, enter these values for the following application gateway settings:
    a. Resource group: Select myResourceGroupAG for the resource group. If it doesn't exist, select Create new to create it.
    b. Application gateway name: Enter myAppGateway for the name of the application gateway.



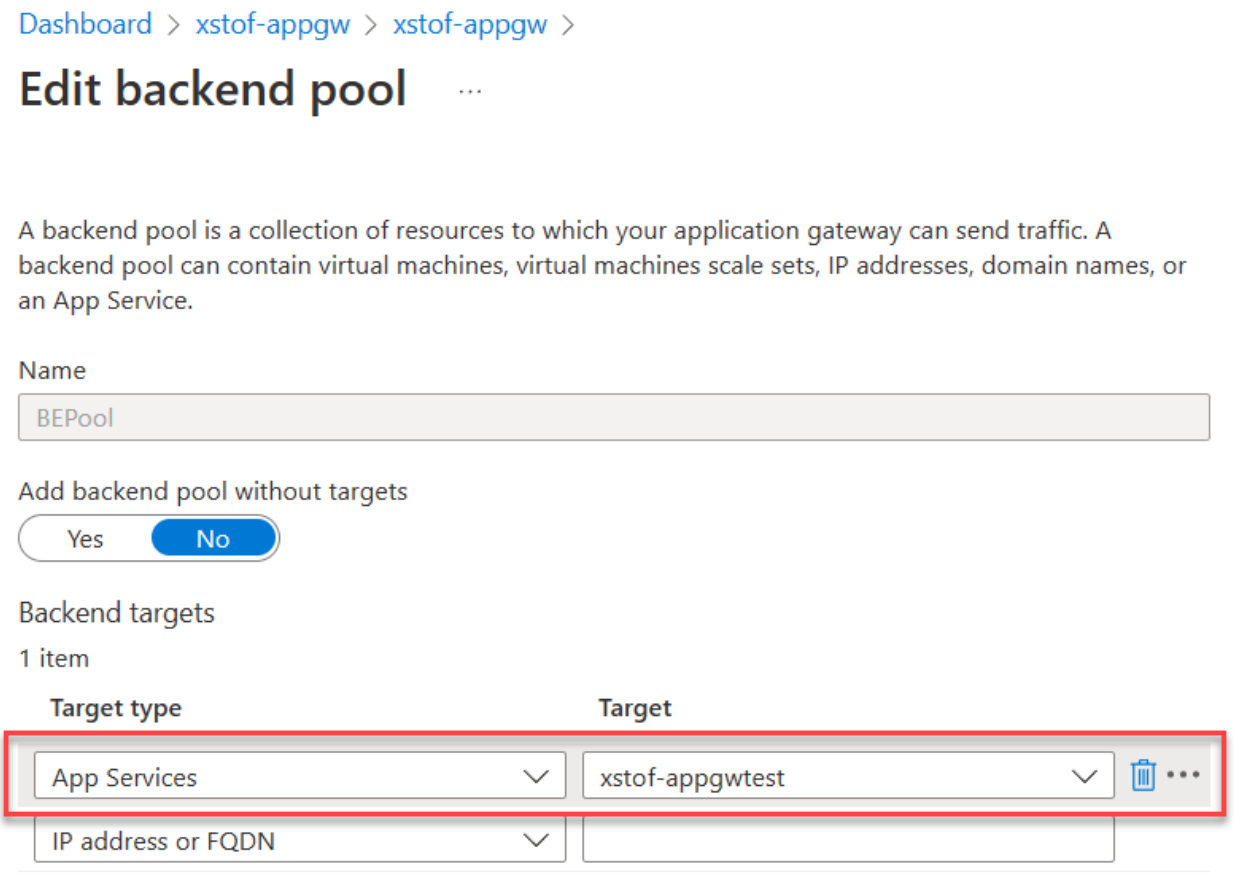
4. For Azure to communicate between the resources that you create, a virtual network is needed. You can either create a new virtual network or use an existing one. In this example, you'll create a new virtual network at the same time that you create the application gateway. Application Gateway instances are created in separate subnets. You create two subnets in this example: One for the application gateway, and another for the backend servers.
    a. Name: Enter myVNet for the name of the virtual network.
    b. Subnet name (Application Gateway subnet): The Subnets grid will show a subnet named default. Change the name of this subnet to myAGSubnet.
    c. The application gateway subnet can contain only application gateways. No other resources are allowed. The default IP address range provided is 10.0.0.0/24.



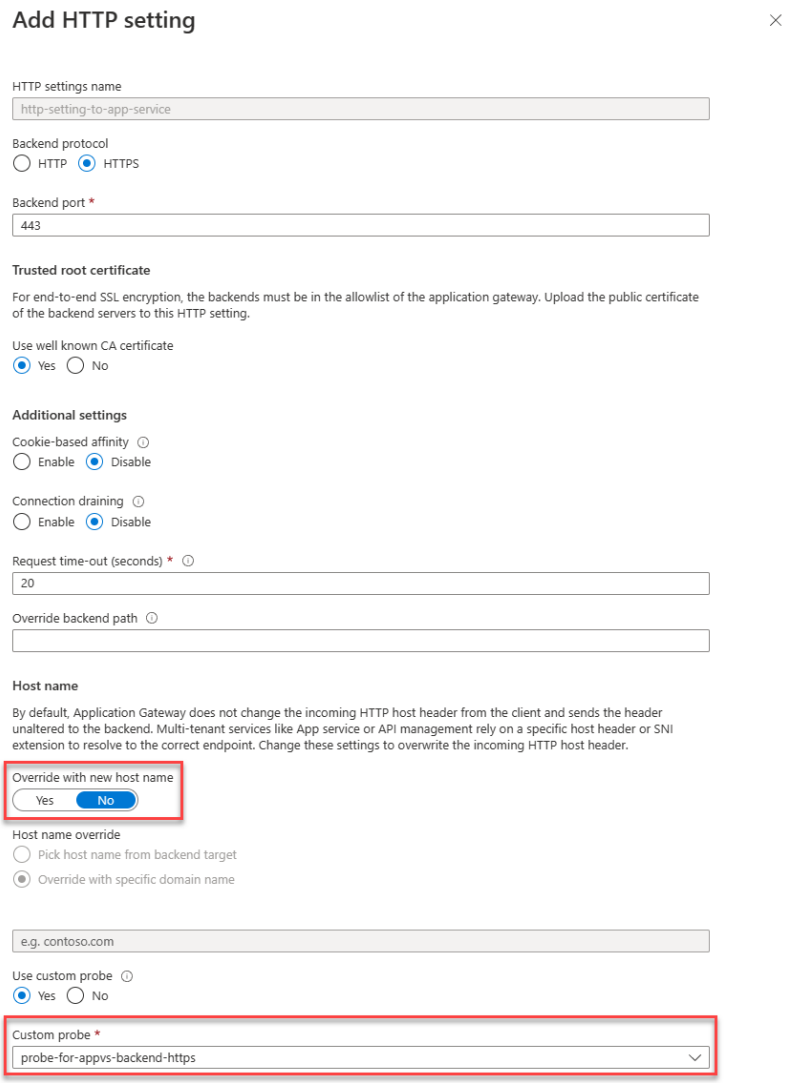
5. On the Frontends tab, verify Frontend IP address type is set to Public.
   You can configure the Frontend IP to be Public or Private as per your use case. In this example, you'll choose a Public Frontend IP.
6. Select Add new for the Public IP address and enter myAGPublicIPAddress for the public IP address name, and then select OK.



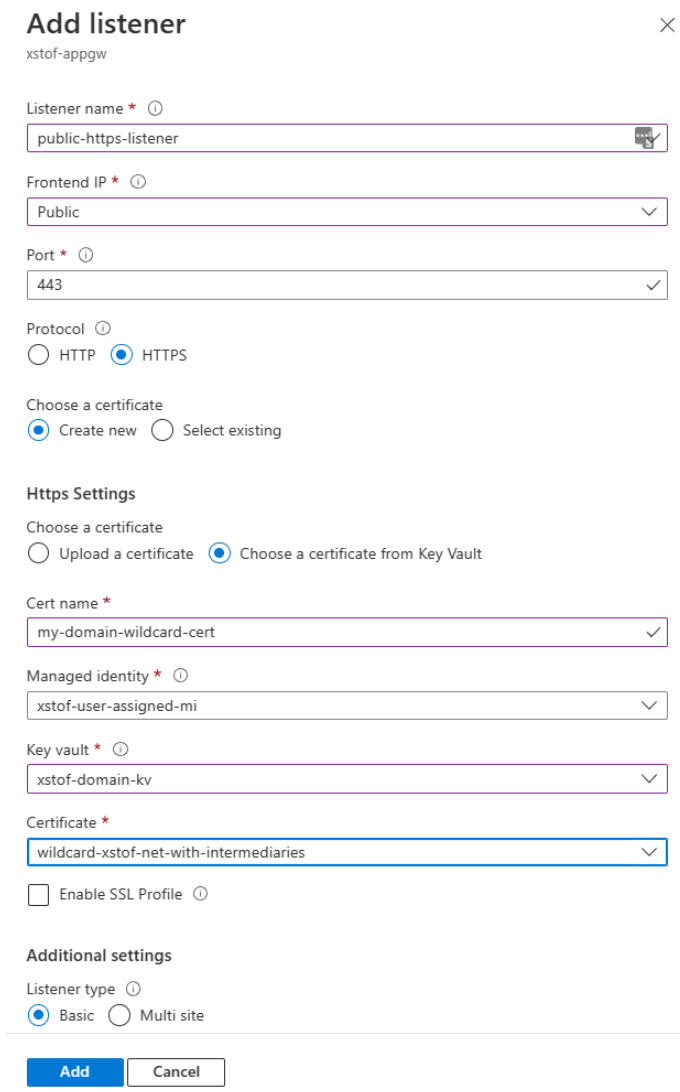
7. Under Backend pools, select the backend pool.
8. Under Target type, select App Services.
9. Under Target select your App Service.



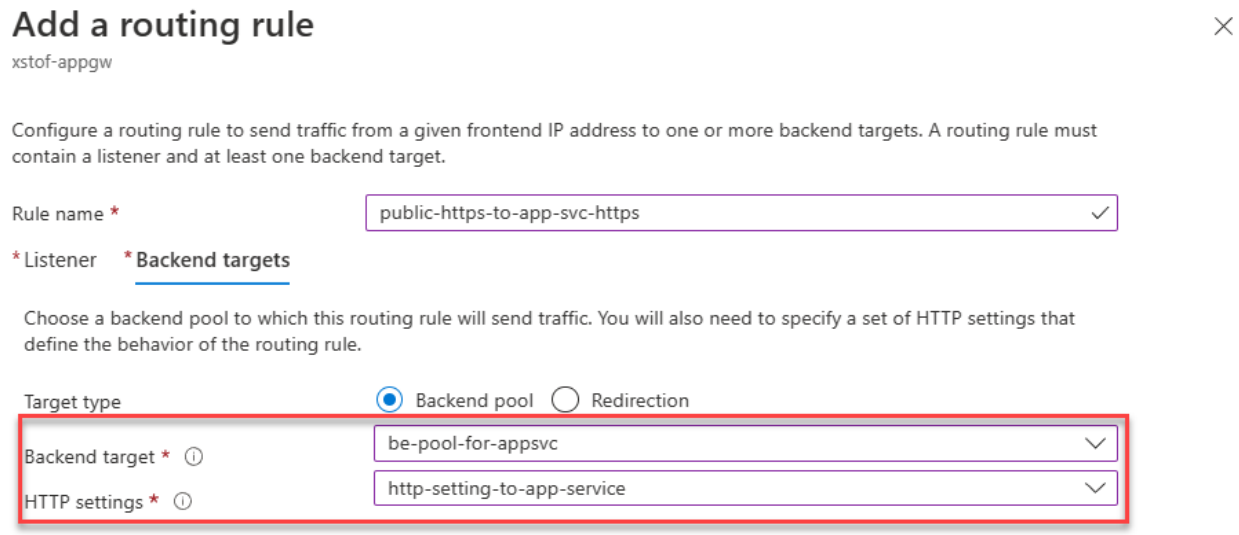
10. Under HTTP Settings, select an existing HTTP setting or add a new one.
11. When creating a new HTTP Setting, give it a name
12. Select HTTP as the desired backend protocol using port 443
13. If the certificate is signed by a well known authority, select "Yes" for "User well known CA certificate". Alternatively Add authentication/trusted root certificates of backend servers
14. Make sure to set "Override with new host name" to "No"
15. Select the custom HTTPS health probe in the dropdown for "Custom probe".



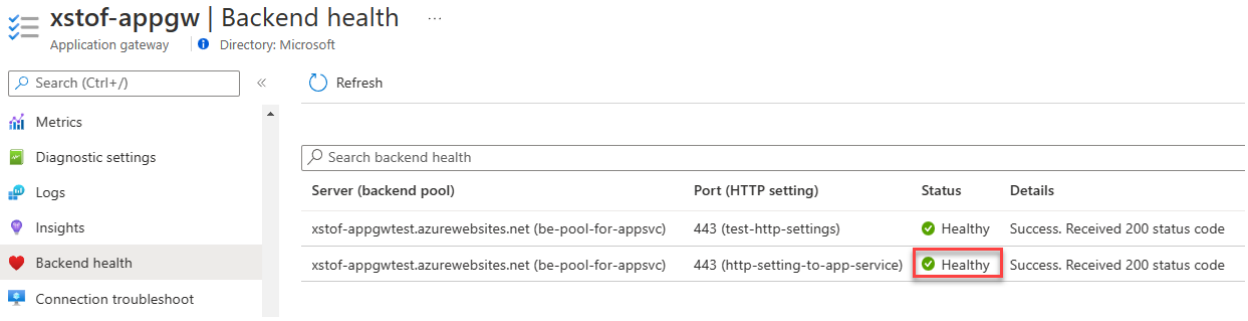
16. Open the "Listeners" section and choose "Add listener" or click an existing one to edit
17. For a new listener: give it a name
18. Under "Frontend IP", select the IP address to listen on
19. Under "Port", select 80
20. Under "Protocol", select "HTTP"
21. Under "Listener Type", select "Basic"
22. Click "Add" to add the listener



23. Under "Rules", click to add a new "Request routing rule"
24. Provide the rule with a name
25. Select an HTTP or HTTPS listener that is not bound yet to an existing routing rule
26. Under "Backend targets", choose the Backend Pool in which App Service has been configured
27. Configure the HTTP settings with which Application Gateway should connect to the App Service backend
28. Select "Add" to save this configuration



Open the "Backend health" section and ensure the "Status" column indicates the combination for HTTP Setting and Backend Pool shows as "Healthy".



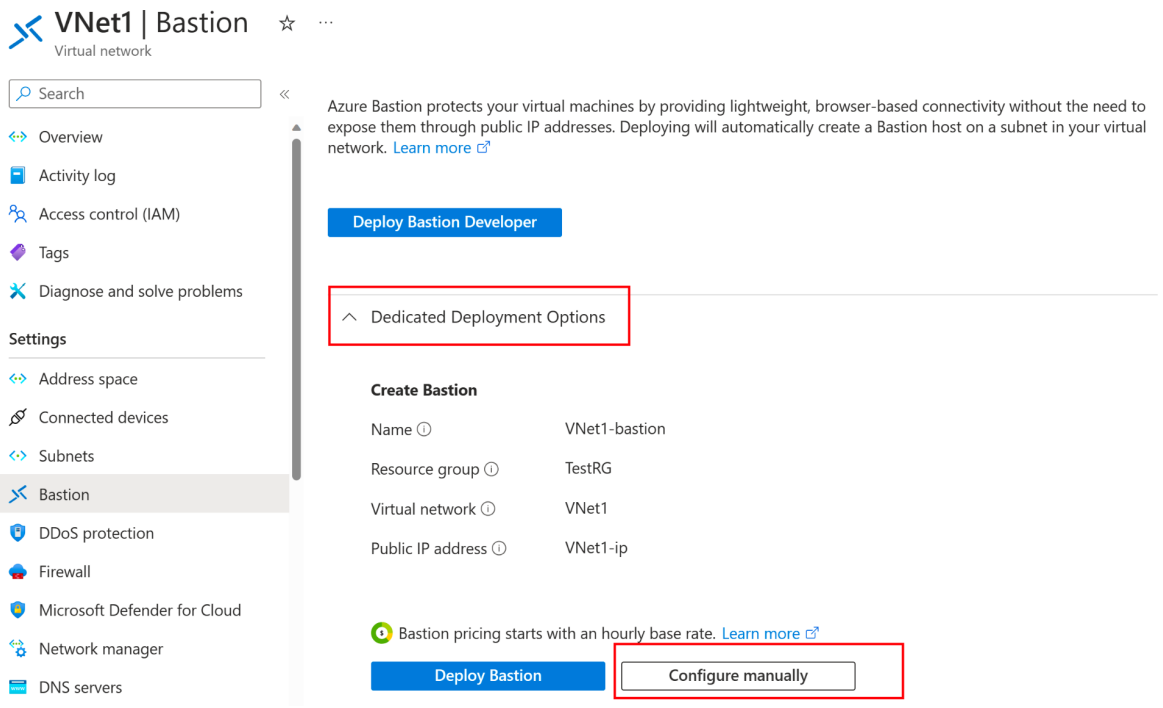
Setup Azure Firewall

1. Navigate to the Azure Portal
    a. Go to the Azure Portal.
    b. Sign in with your Azure credentials.
2. Find your WAF Policy
    a. In the left-hand menu, click on "All services" and then search for "Web Application Firewall policies."
    b. Select your WAF policy from the list. If you haven't already created a policy, you would need to create one.
3. Set the Policy in Prevention Mode
    a. In the WAF policy dashboard, under "Settings," click on "Policy settings."
    b. For the option "Mode," select "Prevention."

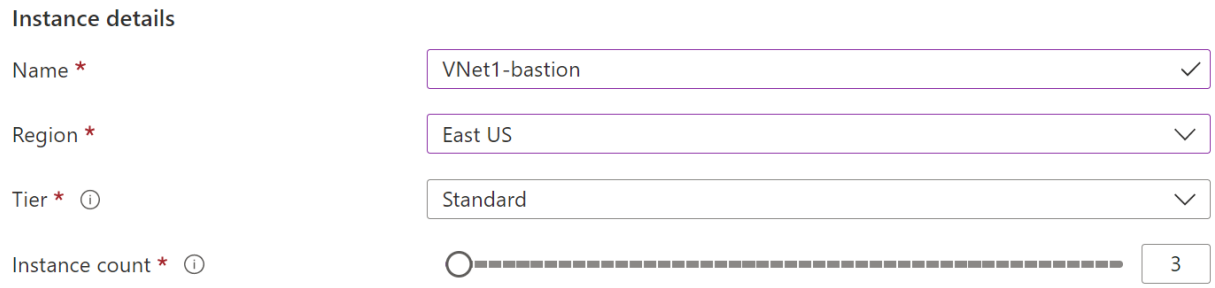
    c. Click the "Save" button to apply the changes.
4. Enable or Disable Specific Firewall Rules
    a. Still in the WAF policy dashboard, under "Settings," click on "Custom rules" if you're looking to enable/disable custom rules. For managed rules, click on "Managed rules."
    b. Here you'll see a list of the rules.
        i. For Custom Rules:
            1. Click on the rule you want to enable or disable.
            2. Toggle the "State" option to "Enabled" or "Disabled."
            3. Click "Save."
        ii. For Managed Rules:
            1. Find the rule set you are interested in.
            2. Click on the specific rule set to expand and view individual rules.
            3. For each rule, you can toggle it "On" or "Off."
            4. After making changes, go back to the "Managed rules" page and click "Save" at the top.
5. Verify the Changes
    a. Check the operational logs or perform tests to ensure that the WAF is operating in prevention mode.
    b. For the rules you enabled or disabled, test to ensure they are acting as expected.

Deploy a bastion Host

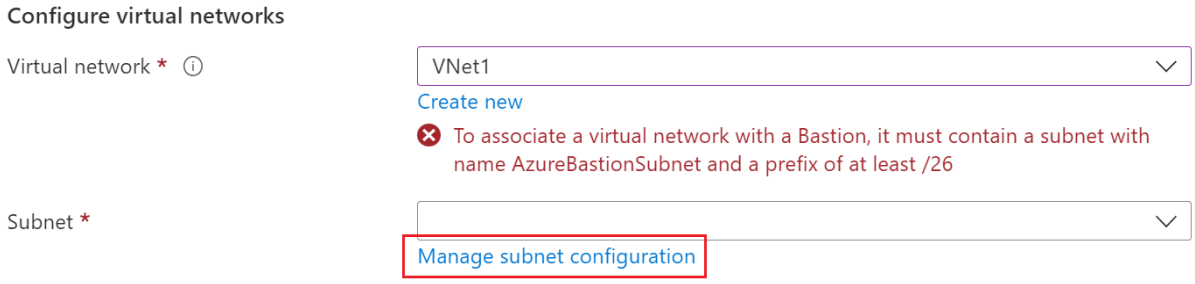
1. Sign in to the Azure portal.
2. Go to your virtual network.
3. On the page for your virtual network, in the left pane, select Bastion to open the Bastion page.
4. On the Bastion page, expand Dedicated Deployment Options.
5. Select Configure manually. This lets you configure specific additional settings (such as the SKU) when deploying Bastion to your virtual network.



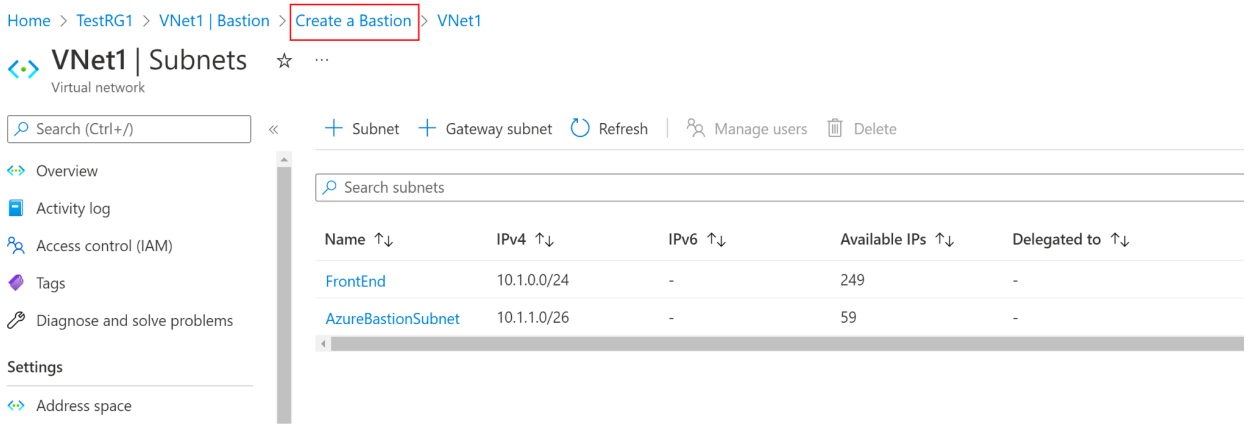
6. On the Create a Bastion page, configure the settings for your bastion host. Project details are populated from your virtual network values. Configure the Instance details values.
    a. Name: Type the name that you want to use for your bastion resource.
    b. Region: The Azure public region in which the resource will be created. Choose the region in which your virtual network resides.
    c. Tier: The tier is also known as the SKU. For this tutorial, select Standard. For information about the features available for each SKU, see Configuration settings - SKU
    d. Instance count: This is the setting for host scaling and is available for the Standard SKU. Host scaling is configured in scale unit increments. Use the slider or type a number to configure the instance count that you want. For this tutorial, you can select the instance count you'd prefer. For more information, see Host scaling and Pricing.



7. Configure the virtual networks settings. Select your virtual network from the dropdown. If you don't see your virtual network in the dropdown list, make sure you selected the correct Region in the previous settings on this page.
8. To configure the AzureBastionSubnet, select Manage subnet configuration.



9. On the Subnets page, select +Subnet to open the Add subnet page.
10. On the Add subnet page, create the 'AzureBastionSubnet' subnet using the following values. Leave the other values as default.
    a. The subnet name must be AzureBastionSubnet.
    b. The subnet must be at least /26 or larger (/26, /25, /24 etc.) to accommodate features available with the Standard SKU.
    c. Select Save at the bottom of the page to save your values.
11. At the top of the Subnets page, select Create a Bastion to return to the Bastion configuration page



12. The Public IP address section is where you configure the public IP address of the Bastion host resource on which RDP/SSH will be accessed (over port 443). The public IP address must be in the same region as the Bastion resource you're creating. Create a new IP address. You can leave the default naming suggestion.
13. When you finish specifying the settings, select Review + Create. This validates the values.
14. Once validation passes, you can deploy Bastion. Select Create. You'll see a message letting you know that your deployment is in process. Status displays on this page as the resources are created. It takes about 10 minutes for the Bastion resource to be created and deployed.

Deploy infrastructure with bicep

1. Navigate to the Bicep Folder
    a. Access the /bicep/ directory.
2. Edit the Parameter File
    a. Open the file /configDev/newEnv.parameters.json.
    b. Modify the value of the “projectName” parameter.
3. Deploy Using Bicep Template

First, ensure you're logged in and have set the correct subscription:

```
az login
az account set --subscription [Your Subscription ID]
```

Then, deploy the app service using the following command:

```
az deployment group create --resource-group [Your Resource Group Name]
--template-file generic.main.bicep --parameters
./configDev/newEnv.parameters.json
```

4. Discuss the created resources