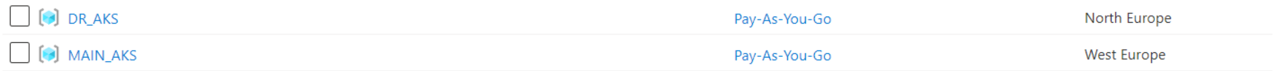
AURACHAIN

Azure DR Deployment

Table of Contents

For this scenario we will start by creating two resource groups (rg) “MAIN_AKS ” and “DR_AKS”



□	DR_AKS	Pay-As-You-Go	North Europe
□	MAIN_AKS	Pay-As-You-Go	West Europe

MAIN_AKS - will be deployed in West Europe

DR_AKS - will be deployed in North Europe

The two zones have direct connections in Azure Public Cloud and therefore is recommended that these zone to be used in this scenarios.

> For more zones that are directly connected please check Azure official documentation.

> Next all resources that will be deployed in one of the rg we will specify the location of the resource to match the location of the rg.

In each rg we will deploy one Virtual Network . In this example those are named mainAKS_vnet and drAKS-vnet.

Adress space for each vNet will be as follow:

- mainAKS_vnet : 10.0.0.0/16
- drAKS_vnet: 10.1.0.0/16

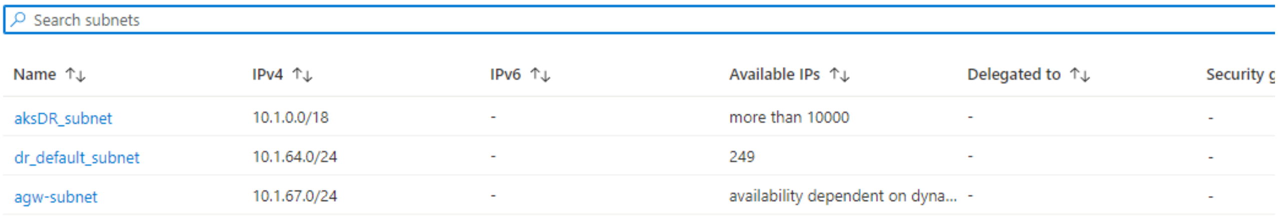
In each vNet we will create 3 subnet’s

Example:

aksMAIN_subnet	**10.0.0.0/18**
agw-subnet	10.0.67.0/24
main_default_subnet	10.0.64.0/24

> Change the name of the subnet and adress-space according to the vnet.

Example:

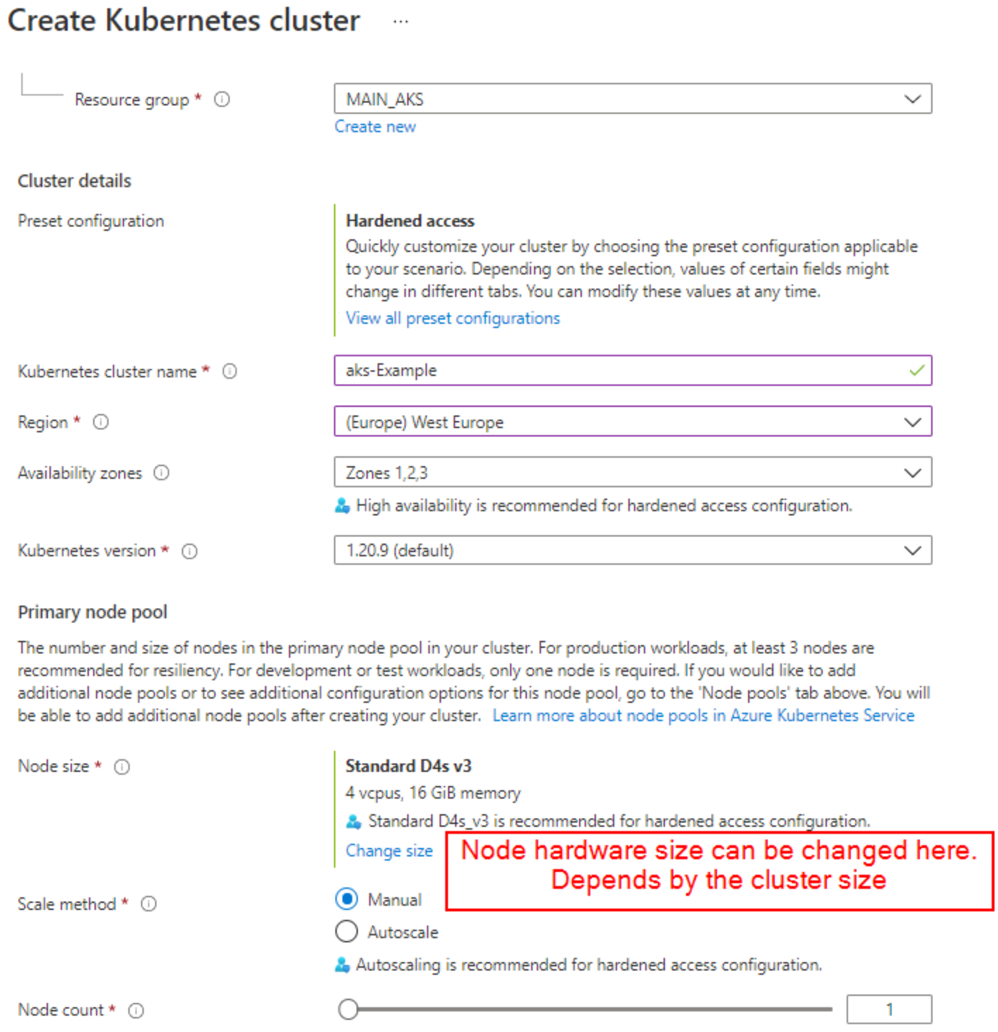


Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓	Security g
aksDR_subnet	10.1.0.0/18	-	more than 10000	-	-
dr_default_subnet	10.1.64.0/24	-	249	-	-
agw-subnet	10.1.67.0/24	-	availability dependent on dyna...	-	-

1 Azure AKS Deployment

In each rg we will deploy one Azure Kubernetes Service(AKS) and as example we can use as bellow images:

ⓘ The configuration must be similar in both zones.



Create Kubernetes cluster ···

Resource group * ⓘ: MAIN_AKS
Create new

Cluster details

Preset configuration: **Hardened access**
Quickly customize your cluster by choosing the preset configuration applicable to your scenario. Depending on the selection, values of certain fields might change in different tabs. You can modify these values at any time.
View all preset configurations

Kubernetes cluster name * ⓘ: aks-Example ✓

Region * ⓘ: (Europe) West Europe

Availability zones ⓘ: Zones 1,2,3
High availability is recommended for hardened access configuration.

Kubernetes version * ⓘ: 1.20.9 (default)

Primary node pool

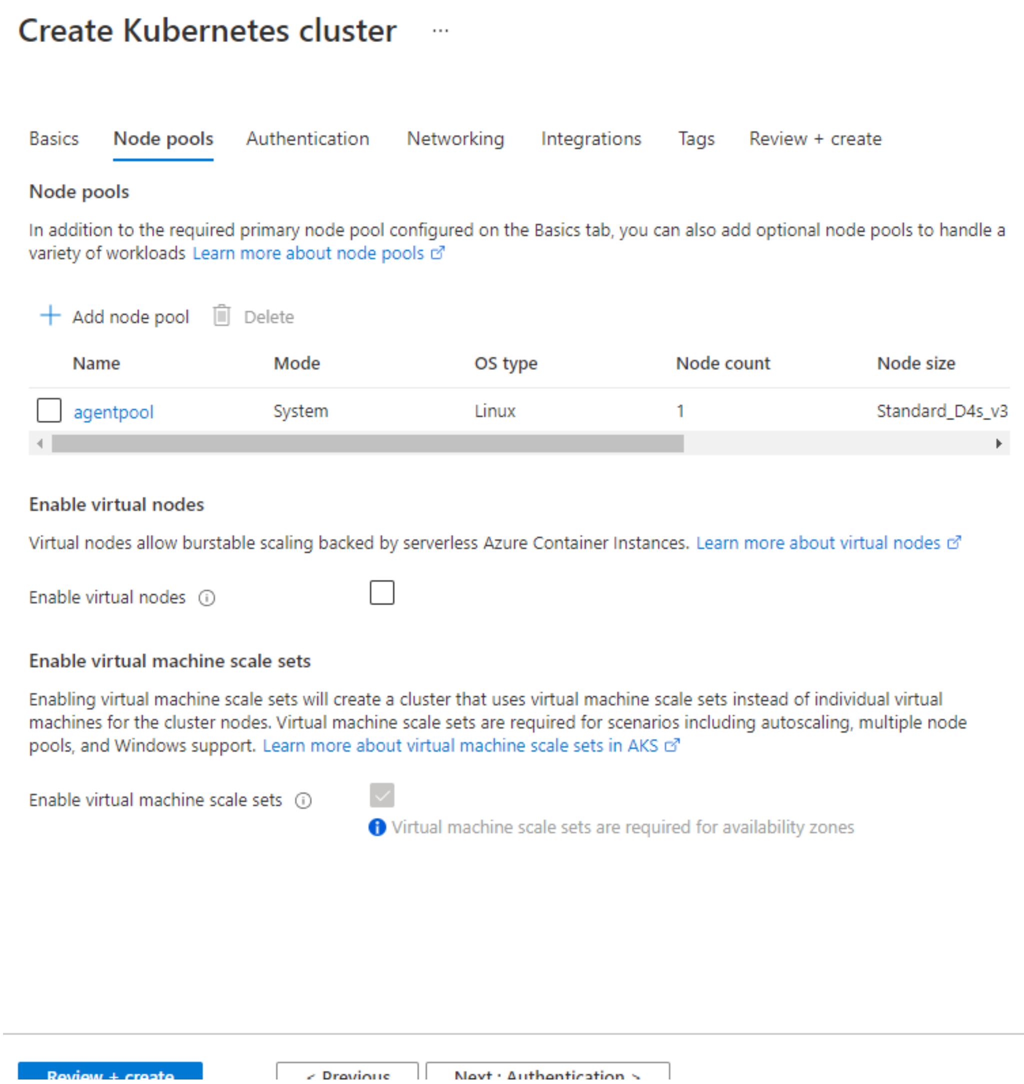
The number and size of nodes in the primary node pool in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. If you would like to add additional node pools or to see additional configuration options for this node pool, go to the 'Node pools' tab above. You will be able to add additional node pools after creating your cluster. Learn more about node pools in Azure Kubernetes Service

Node size * ⓘ: **Standard D4s v3**
4 vcpus, 16 GiB memory
Standard D4s_v3 is recommended for hardened access configuration.
Change size

Node hardware size can be changed here. Depends by the cluster size

Scale method * ⓘ:
● Manual
○ Autoscale
Autoscaling is recommended for hardened access configuration.

Node count * ⓘ: 1



Create Kubernetes cluster ...

Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads Learn more about node pools

	Name	Mode	OS type	Node count	Node size
□	agentpool	System	Linux	1	Standard_D4s_v3

Enable virtual nodes

Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. Learn more about virtual nodes

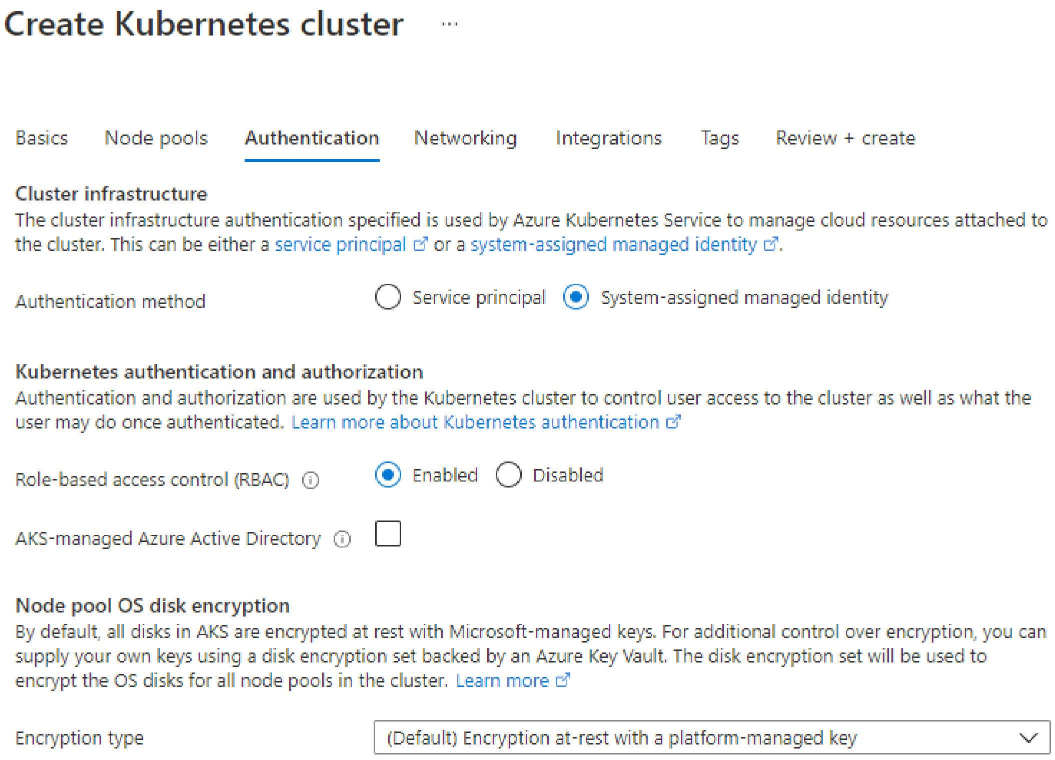
Enable virtual nodes ⓘ □

Enable virtual machine scale sets

Enabling virtual machine scale sets will create a cluster that uses virtual machine scale sets instead of individual virtual machines for the cluster nodes. Virtual machine scale sets are required for scenarios including autoscaling, multiple node pools, and Windows support. Learn more about virtual machine scale sets in AKS

Enable virtual machine scale sets ⓘ ☑

ⓘ Virtual machine scale sets are required for availability zones



Create Kubernetes cluster ...

Basics | Node pools | **Authentication** | Networking | Integrations | Tags | Review + create

Cluster infrastructure

The cluster infrastructure authentication specified is used by Azure Kubernetes Service to manage cloud resources attached to the cluster. This can be either a service principal or a system-assigned managed identity.

Authentication method ○ Service principal ● System-assigned managed identity

Kubernetes authentication and authorization

Authentication and authorization are used by the Kubernetes cluster to control user access to the cluster as well as what the user may do once authenticated. Learn more about Kubernetes authentication

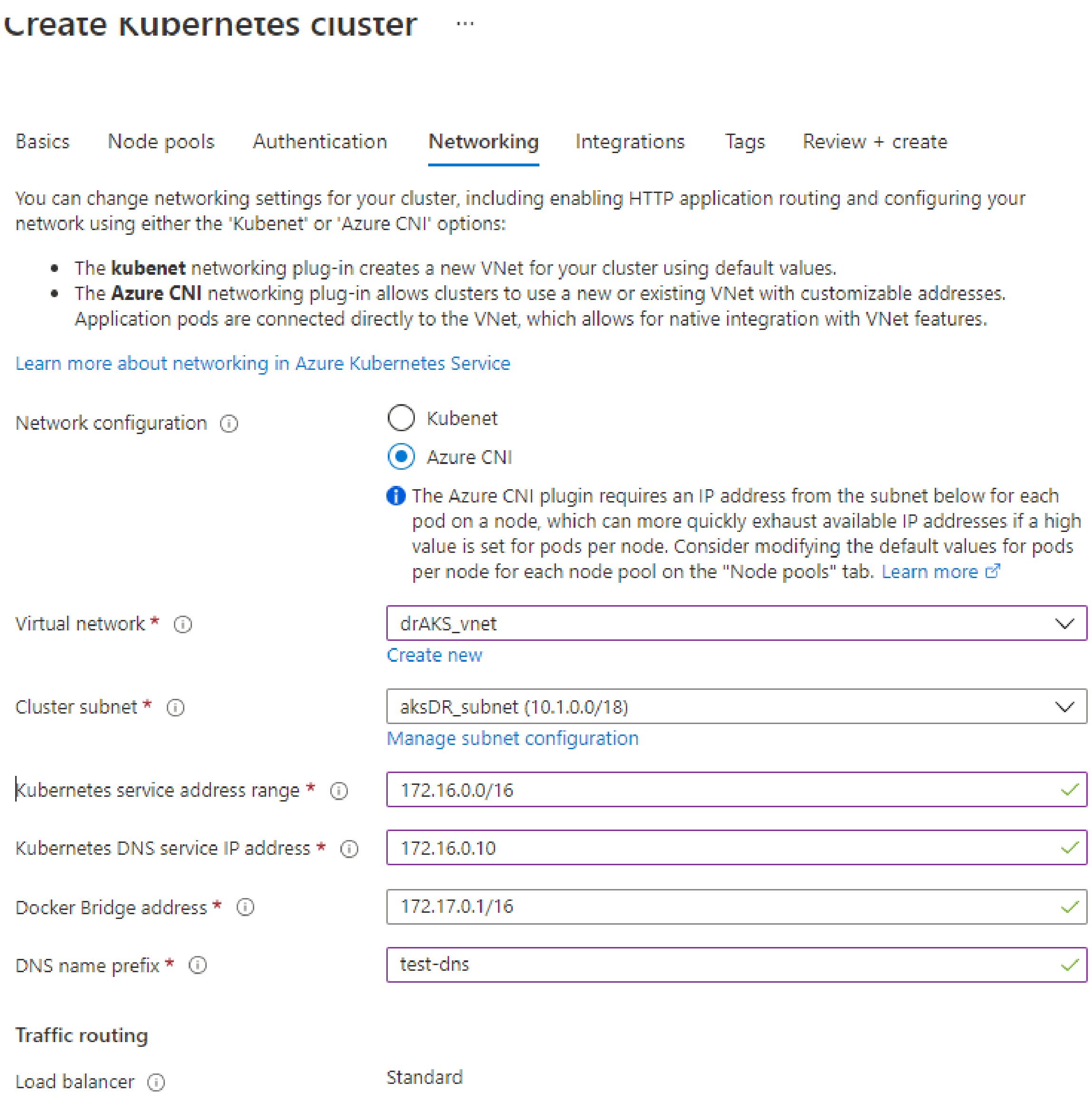
Role-based access control (RBAC) ● Enabled ○ Disabled

AKS-managed Azure Active Directory □

Node pool OS disk encryption

By default, all disks in AKS are encrypted at rest with Microsoft-managed keys. For additional control over encryption, you can supply your own keys using a disk encryption set backed by an Azure Key Vault. The disk encryption set will be used to encrypt the OS disks for all node pools in the cluster. Learn more

Encryption type: (Default) Encryption at-rest with a platform-managed key



Create Kubernetes cluster ···

You can change networking settings for your cluster, including enabling HTTP application routing and configuring your network using either the 'Kubenet' or 'Azure CNI' options:

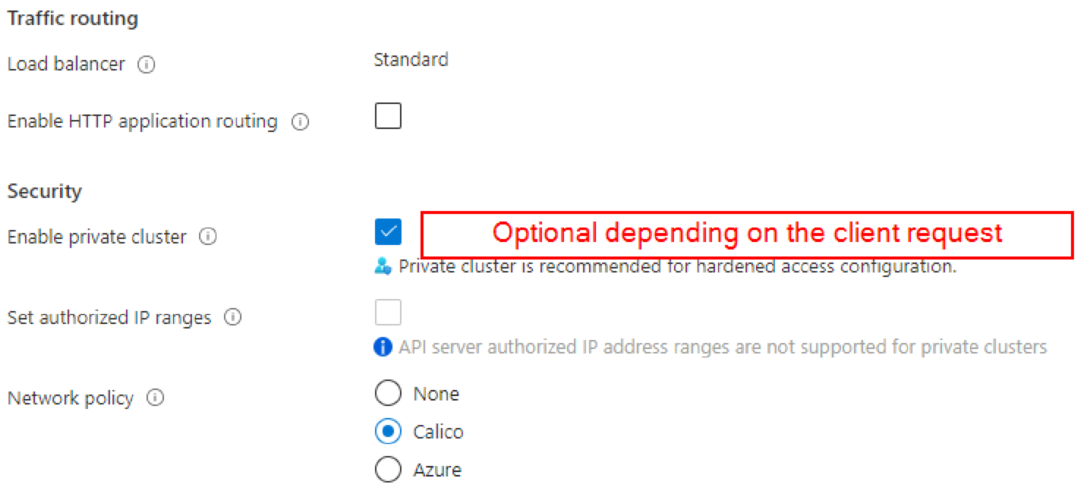
- The **kubenet** networking plug-in creates a new VNet for your cluster using default values.
- The **Azure CNI** networking plug-in allows clusters to use a new or existing VNet with customizable addresses. Application pods are connected directly to the VNet, which allows for native integration with VNet features.

Learn more about networking in Azure Kubernetes Service

Network configuration ⓘ	○ Kubenet ● Azure CNI ⓘ The Azure CNI plugin requires an IP address from the subnet below for each pod on a node, which can more quickly exhaust available IP addresses if a high value is set for pods per node. Consider modifying the default values for pods per node for each node pool on the "Node pools" tab. Learn more
Virtual network * ⓘ	drAKS_vnet Create new
Cluster subnet * ⓘ	aksDR_subnet (10.1.0.0/18) Manage subnet configuration
Kubernetes service address range * ⓘ	172.16.0.0/16 ✓
Kubernetes DNS service IP address * ⓘ	172.16.0.10 ✓
Docker Bridge address * ⓘ	172.17.0.1/16 ✓
DNS name prefix * ⓘ	test-dns ✓

Traffic routing

Load balancer ⓘ	Standard



Traffic routing

Load balancer: Standard

Enable HTTP application routing: □

Security

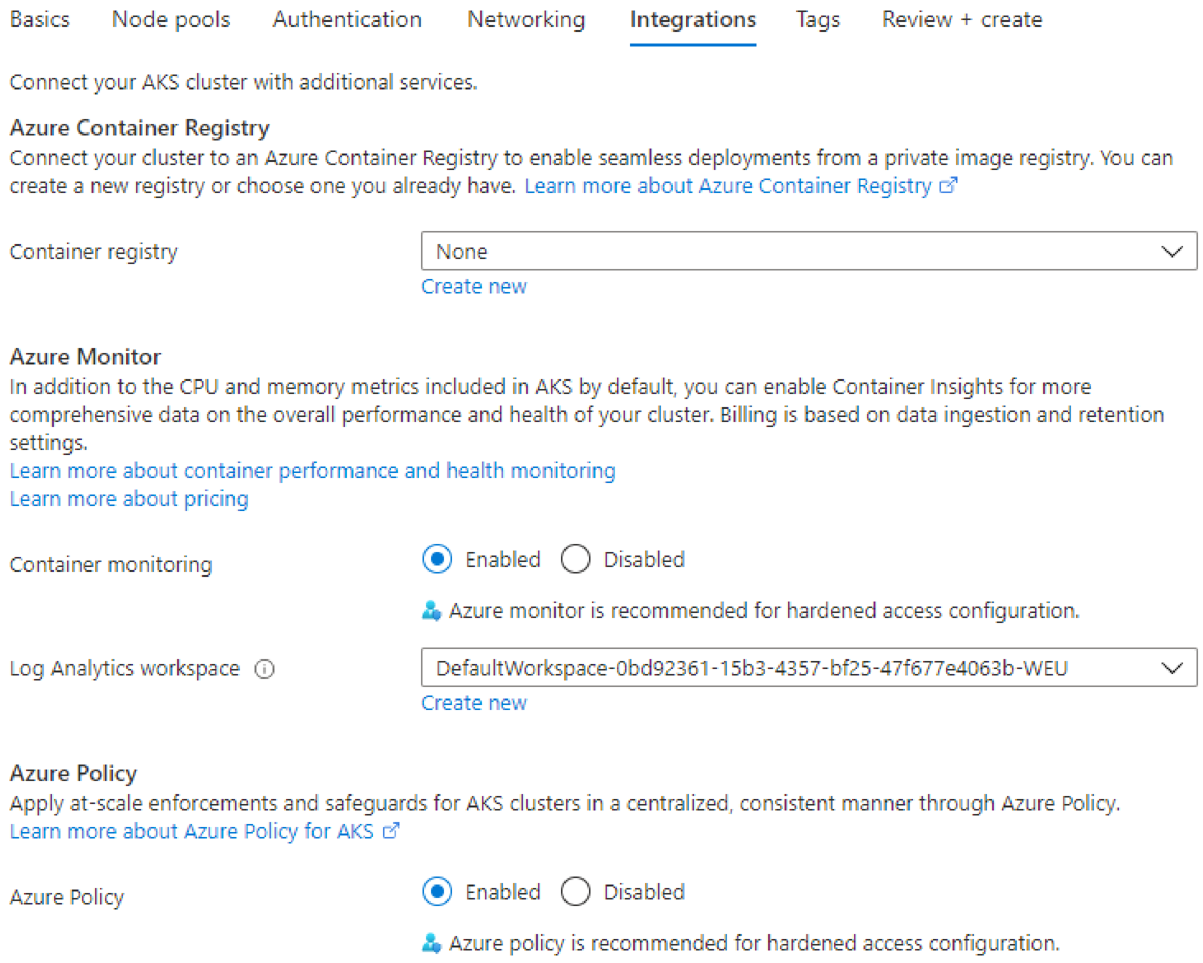
Enable private cluster: ☑ Optional depending on the client request

Private cluster is recommended for hardened access configuration.

Set authorized IP ranges: □

API server authorized IP address ranges are not supported for private clusters

Network policy:
○ None
● Calico
○ Azure



Connect your AKS cluster with additional services.

Azure Container Registry

Connect your cluster to an Azure Container Registry to enable seamless deployments from a private image registry. You can create a new registry or choose one you already have. Learn more about Azure Container Registry

Container registry: None

Create new

Azure Monitor

In addition to the CPU and memory metrics included in AKS by default, you can enable Container Insights for more comprehensive data on the overall performance and health of your cluster. Billing is based on data ingestion and retention settings.

Learn more about container performance and health monitoring

Learn more about pricing

Container monitoring: ● Enabled ○ Disabled

Azure monitor is recommended for hardened access configuration.

Log Analytics workspace: DefaultWorkspace-0bd92361-15b3-4357-bf25-47f677e4063b-WEU

Create new

Azure Policy

Apply at-scale enforcements and safeguards for AKS clusters in a centralized, consistent manner through Azure Policy.

Learn more about Azure Policy for AKS

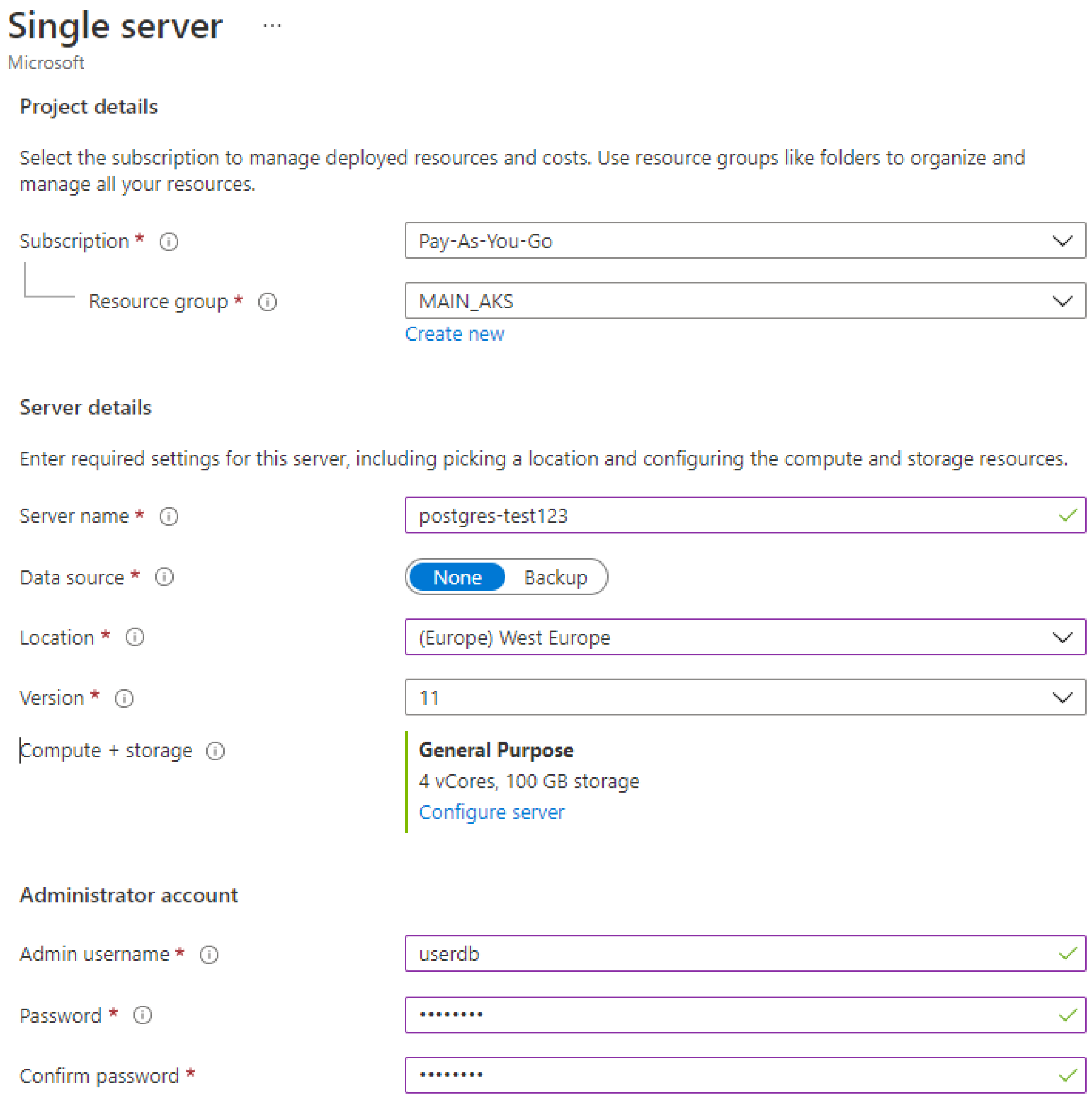
Azure Policy: ● Enabled ○ Disabled

Azure policy is recommended for hardened access configuration.

2 Azure Postgres Deployment

For postgres we will deploy “Azure Database for PostgreSQL” in **Single server** mode. This service will be deployed in the MAIN rg.

Hardware will be chosen by project specs but in this example we will chose General Purpose tier with 4vCores , 100GB storage and backup Geo-Redundant.



Single server ···

Microsoft

Project details

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

Subscription * ⓘ	Pay-As-You-Go
Resource group * ⓘ	MAIN_AKS Create new

Server details

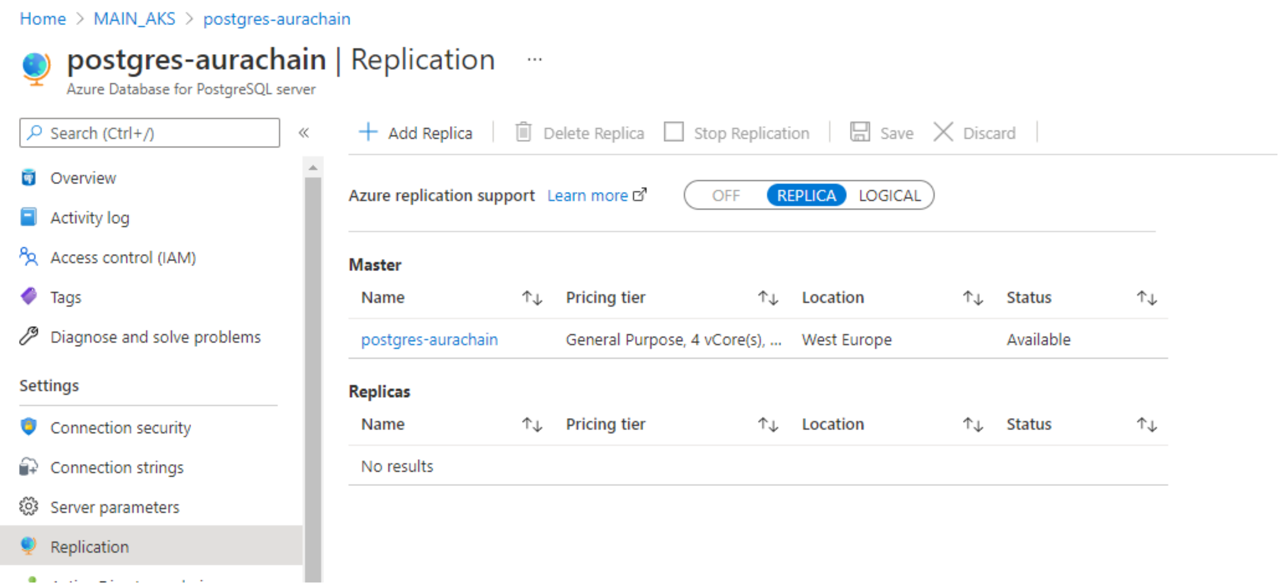
Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * ⓘ	postgres-test123 ✓
Data source * ⓘ	● None ○ Backup
Location * ⓘ	(Europe) West Europe
Version * ⓘ	11
Compute + storage ⓘ	**General Purpose** 4 vCores, 100 GB storage Configure server

Administrator account

Admin username * ⓘ	userdb ✓
Password * ⓘ	•••••••• ✓
Confirm password *	•••••••• ✓

After the first instance is deployed at Replication settings we will add a new replica set in North Europe and after build process is finished we will move the replica from MAIN rg to DR rg.



Home > MAIN_AKS > postgres-aurachain

postgres-aurachain | Replication ···

Azure Database for PostgreSQL server

Azure replication support Learn more

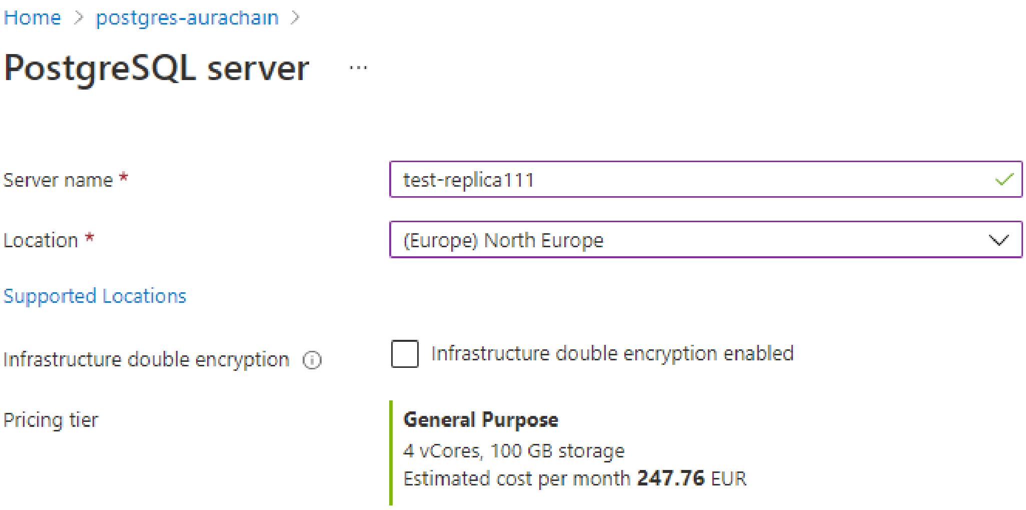
OFF [✓ REPLICA] LOGICAL

Master

Name	Pricing tier	Location	Status
postgres-aurachain	General Purpose, 4 vCore(s), ...	West Europe	Available

Replicas

Name	Pricing tier	Location	Status
No results			

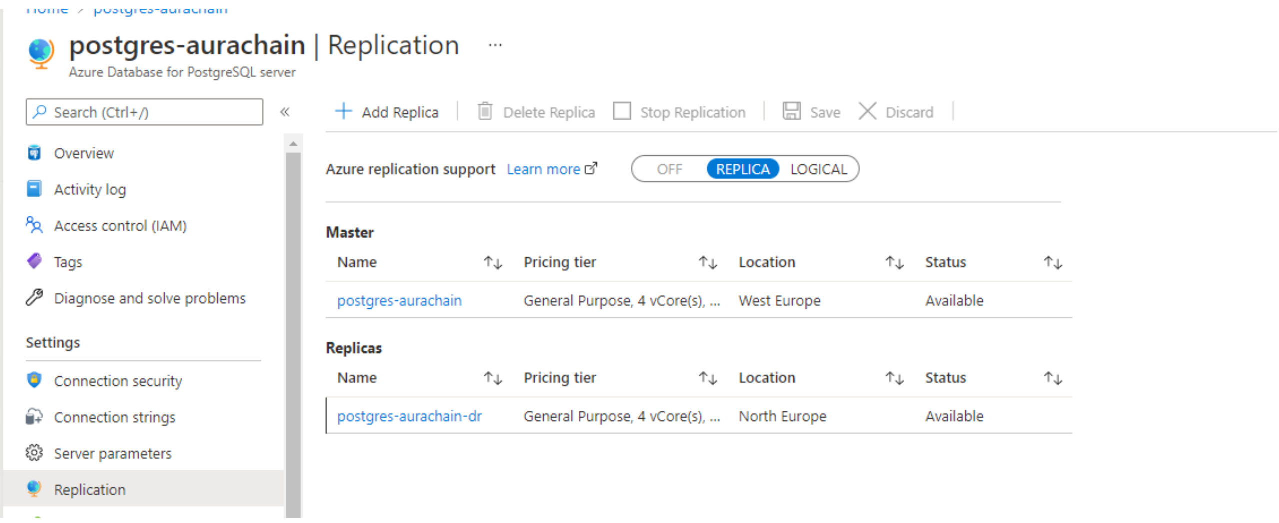


Home > postgres-aurachain >

PostgreSQL server ···

Server name *	test-replica111
Location *	(Europe) North Europe
Supported Locations	
Infrastructure double encryption ⓘ	□ Infrastructure double encryption enabled
Pricing tier	**General Purpose** 4 vCores, 100 GB storage Estimated cost per month **247.76** EUR

Example:



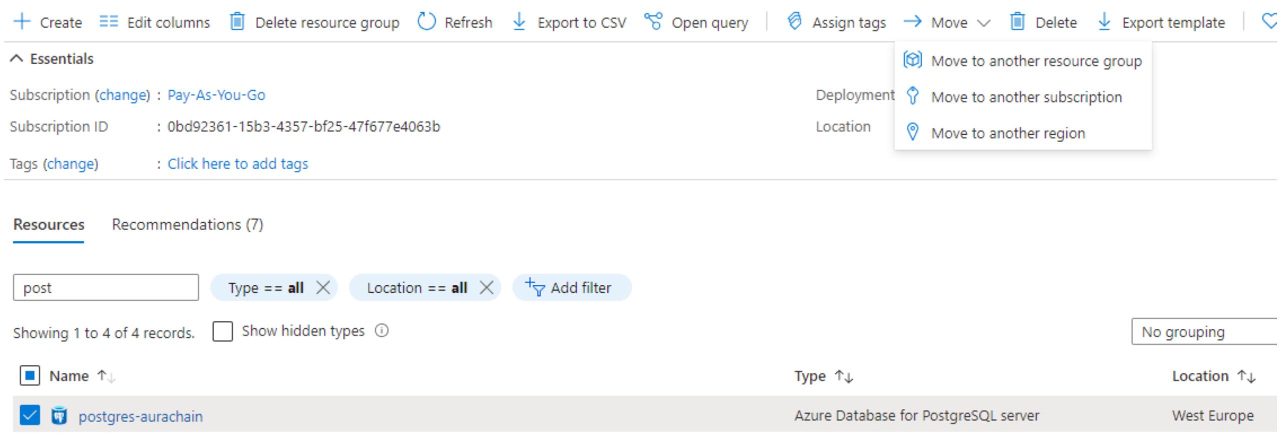
Azure replication support: REPLICA

Master

Name	Pricing tier	Location	Status
postgres-aurachain	General Purpose, 4 vCore(s), ...	West Europe	Available

Replicas

Name	Pricing tier	Location	Status
postgres-aurachain-dr	General Purpose, 4 vCore(s), ...	North Europe	Available



Subscription (change) : Pay-As-You-Go
Subscription ID : 0bd92361-15b3-4357-bf25-47f677e4063b
Tags (change) : Click here to add tags
Deployment
Location

post | Type == all | Location == all

Showing 1 to 4 of 4 records. □ Show hidden types

Name	Type	Location
☑ postgres-aurachain	Azure Database for PostgreSQL server	West Europe

Also after both resources are deployed and moved to specific rg “Private endpoint connections” will be build . For guidance check Azure official documentation.

In order to create databases we will connect to the main instance (the second instance is only read-replica by default) from the jump server by using the user defined at setup.

Example:

```
psql "host=postgres-test-aurachain.postgres.database.azure.com port=5432
dbname=postgres user=dbuser@postgres-aurachain password=mystrongpass sslmode=require"
```

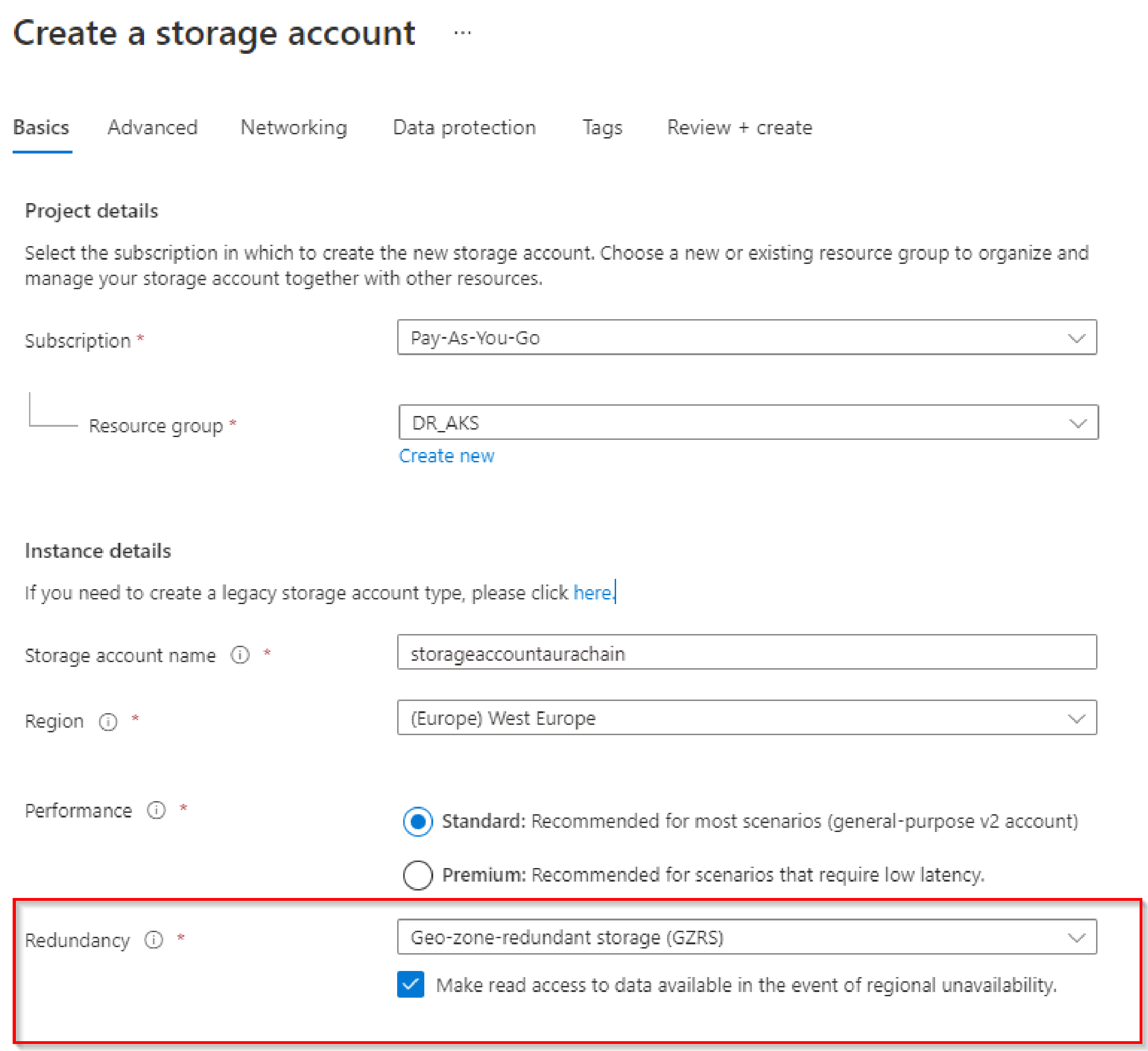
> ⚠ Because postgres service is build on windows servers we have to create databases manually and specify Windows specific ENCONDING , LC_TYPE etc

```
CREATE DATABASE <database_name> WITH ENCODING='utf8' OWNER=<owner_name> LC_COLLATE='English_United States.1252'  LC_CTYPE='English_United States.1252' CONNECTION LIMIT=-1;
```

> ⓘ In order to run “psql” commands please login to az cli and install necessary packages before following below example.

3 Azure Storage Account

Storage account resource will be deployed in to the MAIN rg and is mandatory to specify at the deployment GZRS redundancy to have a replica in North Europe.



Create a storage account ···

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * : Pay-As-You-Go

Resource group * : DR_AKS

Create new

Instance details

If you need to create a legacy storage account type, please click here.

Storage account name ⓘ * : storageaccountaurachain

Region ⓘ * : (Europe) West Europe

Performance ⓘ * :

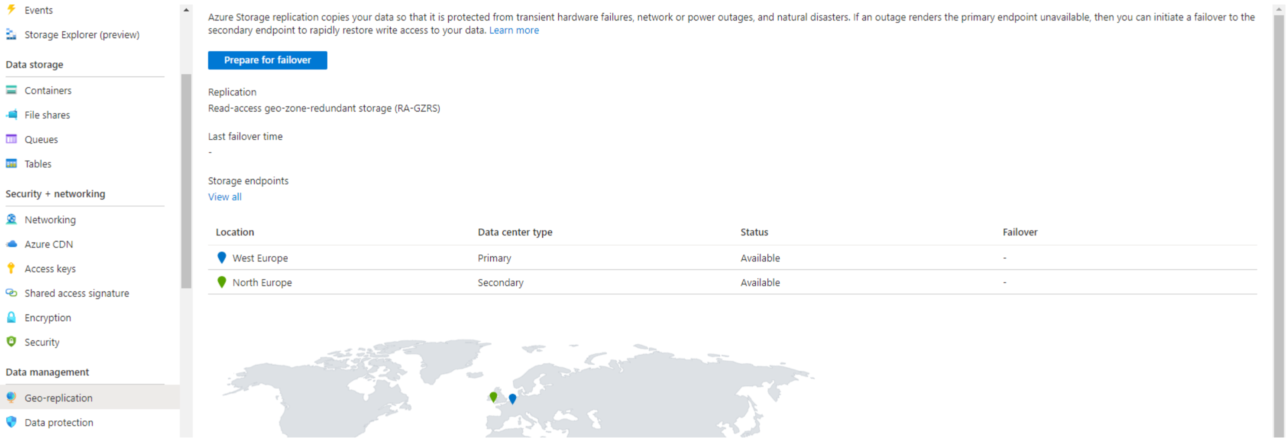
● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ * : Geo-zone-redundant storage (GZRS)

☑ Make read access to data available in the event of regional unavailability.

After deployment is finished we can check the availability by going to Geo-replication settings. Both zones must appear *Available*



Azure Storage replication copies your data so that it is protected from transient hardware failures, network or power outages, and natural disasters. If an outage renders the primary endpoint unavailable, then you can initiate a failover to the secondary endpoint to rapidly restore write access to your data. Learn more

Replication
Read-access geo-zone-redundant storage (RA-GZRS)

Last failover time
-

Storage endpoints
View all

Location	Data center type	Status	Failover
West Europe	Primary	Available	-
North Europe	Secondary	Available	-

In order to connect storage account to AKS and to create PV and PVC we will use bellow guide.

The same setup must be performed in both AKS clusters!

ⓘ In this example our storage account is named “aurachainstorage” in the “MAIN_AKS” rg and the file share that will be created is named: “aurachain-storage-common”

⚠ Please check closely below example and change name/labels/spec’s as per your setup!

ⓘ In order to run “az” and “kubectl” commands please login and install necessary packages before perform this setup

```
az storage account show-connection-string -n aurachainstorage -g MAIN_AKS -o tsv
export AZURE_STORAGE_CONNECTION_STRING=`az storage account show-connection-string -n
aurachainstorage -g MAIN_AKS -o tsv`
az storage share create -n aurachain-storage-common --connection-string
$AZURE_STORAGE_CONNECTION_STRING
STORAGE_KEY=$(az storage account keys list --resource-group MAIN_AKS --account-name
aurachainstorage --query "[0].value" -o tsv)
kubectl create secret generic azure-secret-storageaccount --from-
literal=azurestorageaccountname=aurachainstorage --from-
literal=azurestorageaccountkey=$STORAGE_KEY
```

Next we will use yaml files in order to create PV and PVC in desired namespace

```
cat pv-azure-create.yaml


apiVersion: v1
kind: PersistentVolume
metadata:
  name: aurachain-storage-prod-pv
  # The label is used for matching the exact claim
  labels:
    usage: aurachain-storage-pv
spec:
  capacity:
    storage: 100Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  azureFile:
    # Replace with your secret name
    secretName: azure-secret-storageaccount
    # Replace with correct storage share name
    shareName: aurachain-storage-common
    # In case the secret is stored in a different namespace
    #secretNamespace: default
    readOnly: false
```

```
cat pvc-azure-create.yaml


kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: common-storage
  namespace: aurachain-prod
  # Set this annotation to NOT let Kubernetes automatically create
  # a persistent volume for this volume claim.
  annotations:
    volume.beta.kubernetes.io/storage-class: ""
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  selector:
    # To make sure we match the claim with the exact volume, match the label
    matchLabels:
      usage: aurachain-storage-pv
```

```
kubectl apply -f pv-azure-create.yaml
kubectl apply -f pvc-azure-create.yaml
```

```
root@jump-main:~# kubectl get pvc,pv -n aurachain-prod
NAME                                   STATUS   VOLUME                      CAPACITY
ACCESS MODES   STORAGECLASS   AGE
persistentvolumeclaim/common-storage   Bound    aurachain-storage-prod-pv   100Gi
RWX                           3d4h

NAME                                            CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS   CLAIM                                    STORAGECLASS   REASON   AGE
persistentvolume/aurachain-storage-prod-pv        100Gi      RWX            Retain
Bound    aurachain-prod/common-storage                              3d4h
```

4 Azure VM deployment and ASR setup

Jump Machine:

In DR scenario we will install one vm per rg .

Jump vm will be used to run commands like: kubectl,az cli,psql,helm etc and also to manage and access all other resources that are in vnet and/or rg.

> ⓘ For each type of command we need to install specified packages and all the necessary tools.

At the deployment we will specify that internal network to be in defaul subnet , and also we need to create a public ip address that will be used for external access via ssh.

> ⚠ For security reason is important to secure access to jump (users, passcode, ssh keys, network security etc)

ELK stack:

In MAIN rg we will create one vm per each component all vm's will have one internal network in default subnet.

The installing of ELK services will be performed according to specified documentation.

For rg DR we don't need to create ELK vm's because ASR(Azure site recovery) and/or Backup services from Azure cloud will cover this aspect.

Monitoring:

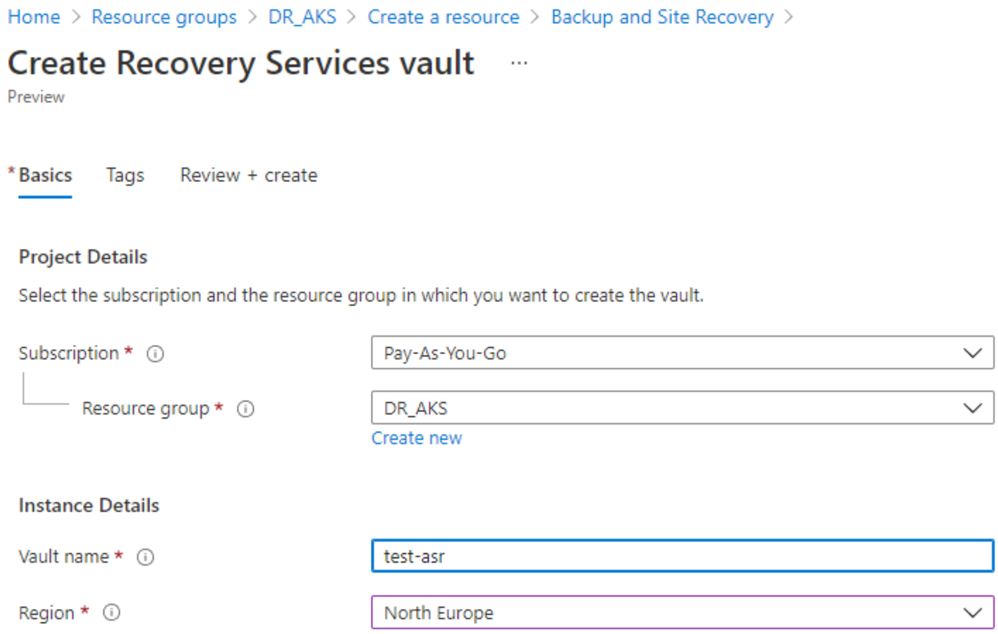
In MAIN rg we will create one vm, the vm will have one internal network in default subnet.

The installing of specific monitorig services will be performed according to specified documentation.

For rg DR we don't need to create monitoring VM because ASR(Azure site recovery) and/or Backup services from Azure cloud will cover this aspect.

ASR(Azure site recovery) SETUP:

In rg **DR** we will create the resource called “Backup and Site Recovery”



Home > Resource groups > DR_AKS > Create a resource > Backup and Site Recovery >

Create Recovery Services vault ...

Preview

*Basics Tags Review + create

Project Details

Select the subscription and the resource group in which you want to create the vault.

Subscription * : Pay-As-You-Go

Resource group * : DR_AKS

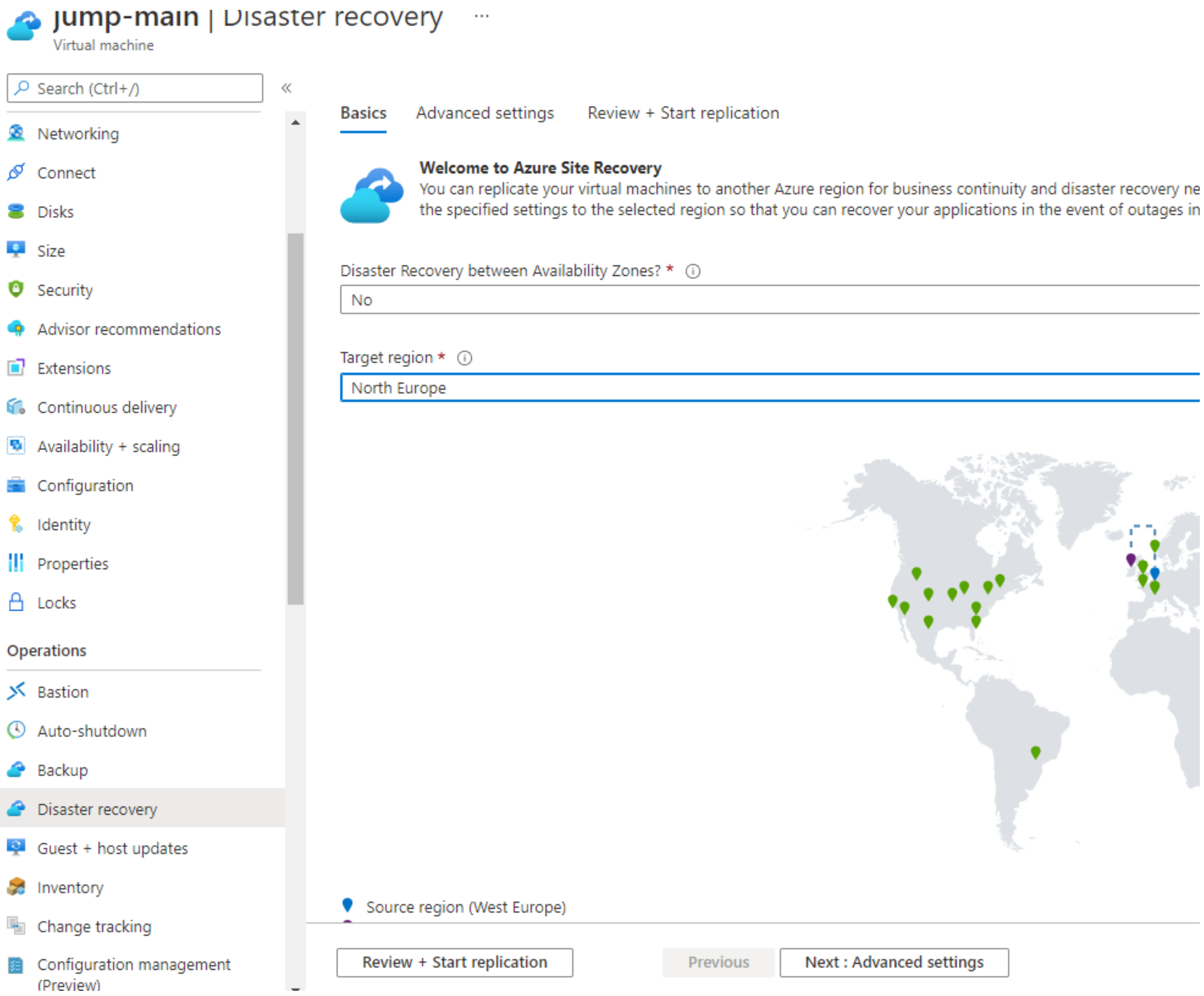
Instance Details

Vault name * : test-asr

Region * : North Europe

After the deployment on each vm that need to be protected we will create a disaster recovery job:

Example:



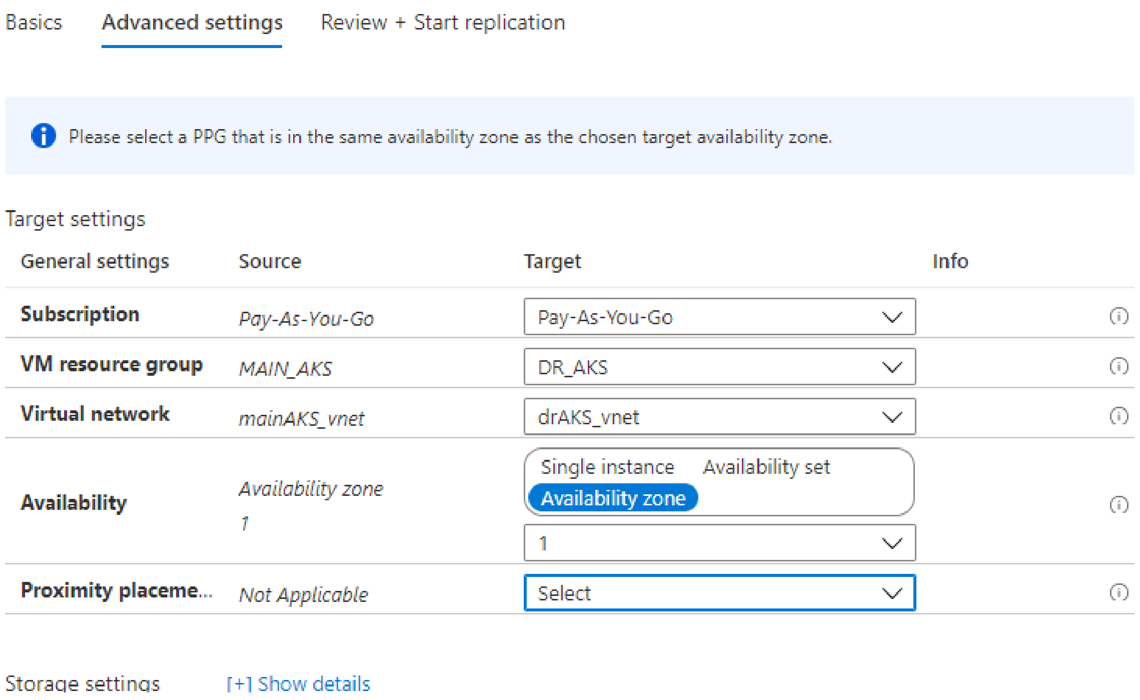
jump-main | Disaster recovery ...
Virtual machine

Welcome to Azure Site Recovery
You can replicate your virtual machines to another Azure region for business continuity and disaster recovery ne
the specified settings to the selected region so that you can recover your applications in the event of outages in

Disaster Recovery between Availability Zones? * ⓘ
No

Target region * ⓘ
North Europe

Source region (West Europe)

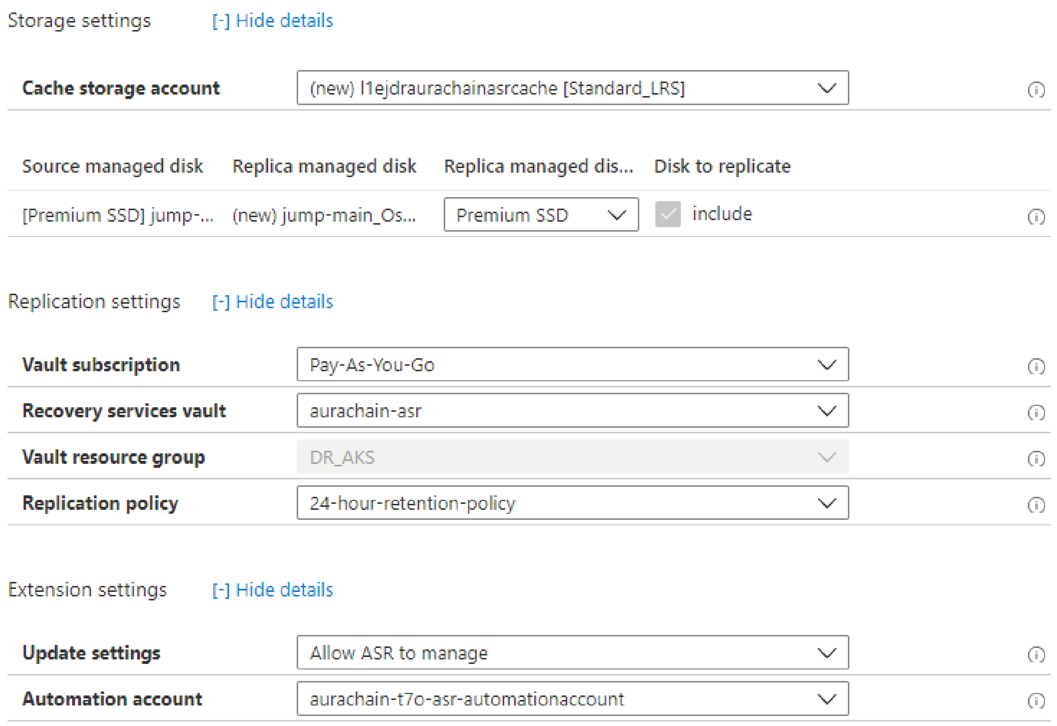


Please select a PPG that is in the same availability zone as the chosen target availability zone.

Target settings

General settings	Source	Target	Info
Subscription	*Pay-As-You-Go*	Pay-As-You-Go	ⓘ
VM resource group	*MAIN_AKS*	DR_AKS	ⓘ
Virtual network	*mainAKS_vnet*	drAKS_vnet	ⓘ
Availability	*Availability zone* *1*	Single instance Availability set [✓ Availability zone] 1	ⓘ
Proximity placeme...	*Not Applicable*	Select	ⓘ

Storage settings [+] Show details



Storage settings [-] Hide details

Cache storage account	(new) l1ejdraurachainasrcache [Standard_LRS]

Source managed disk	Replica managed disk	Replica managed dis...	Disk to replicate
[Premium SSD] jump-...	(new) jump-main_Os...	Premium SSD	☑ include

Replication settings [-] Hide details

Vault subscription	Pay-As-You-Go
Recovery services vault	aurachain-asr
Vault resource group	DR_AKS
Replication policy	24-hour-retention-policy

Extension settings [-] Hide details

Update settings	Allow ASR to manage
Automation account	aurachain-t7o-asr-automationaccount

> ⓘ In *Storage Settings* we must select both disks; os and also data disk.

For more informations please check Azure official documentation:

https://docs.microsoft.com/en-us/azure/site-recovery/azure-to-azure-quickstart

Backup SETUP:

In order to configure Backup functionality in we need to deploy in rg **MAIN** “Backup and Site Recovery” resource.

After deployment we need to configure Backup jobs, and backup policies in each vm that we need to be protected.

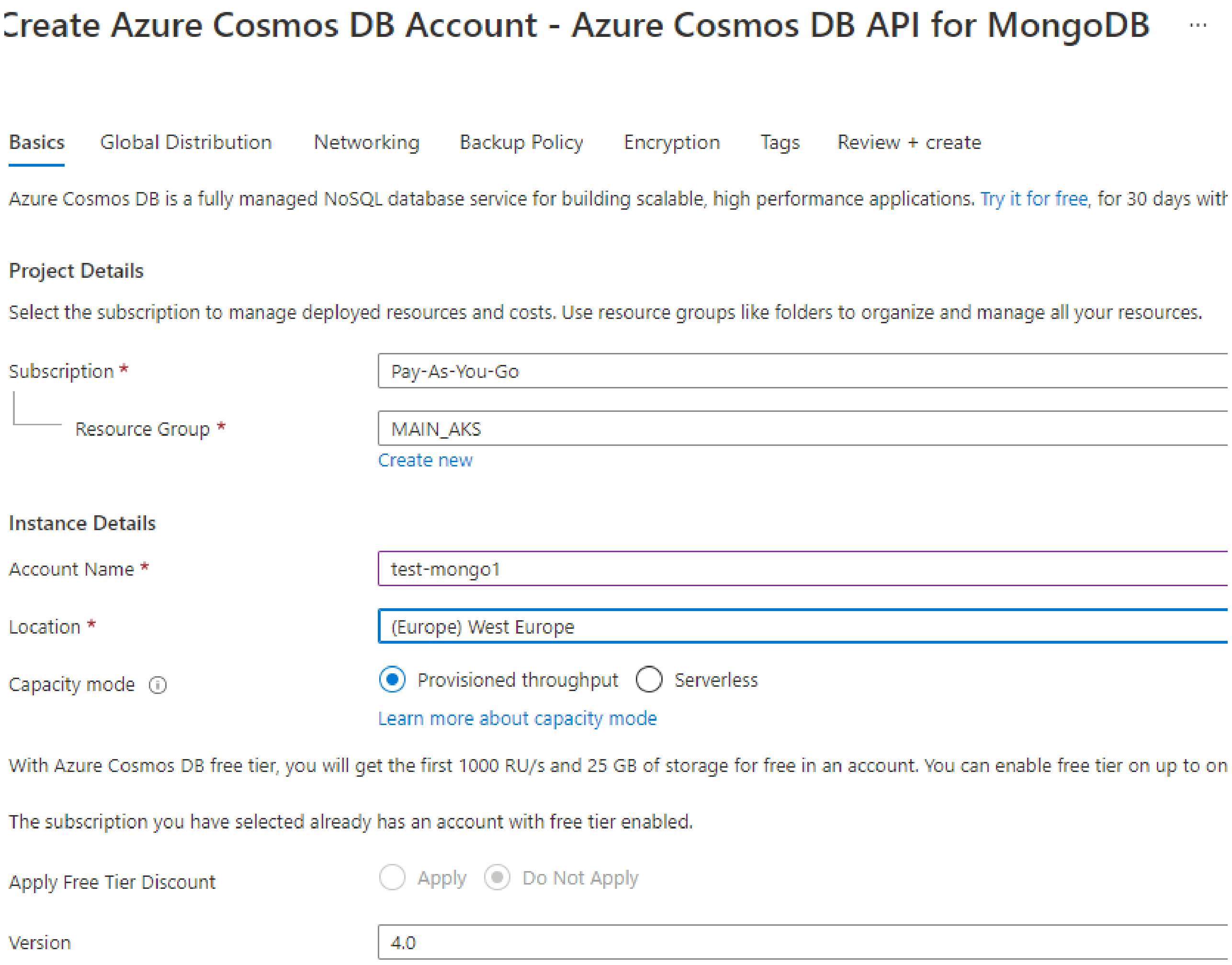
For more informations please check Azure official documentation:

https://docs.microsoft.com/en-us/azure/backup/backup-azure-vms-first-look-arm

https://docs.microsoft.com/en-us/azure/backup/quick-backup-vm-portal

5 Azure Cosmos MongoDB

In rg MAIN we will create the resource named: “Azure Cosmos DB API for MongoDB”



Create Azure Cosmos DB Account - Azure Cosmos DB API for MongoDB ...

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. Try it for free, for 30 days with

Project Details

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

Subscription *	Pay-As-You-Go
Resource Group *	MAIN_AKS Create new

Instance Details

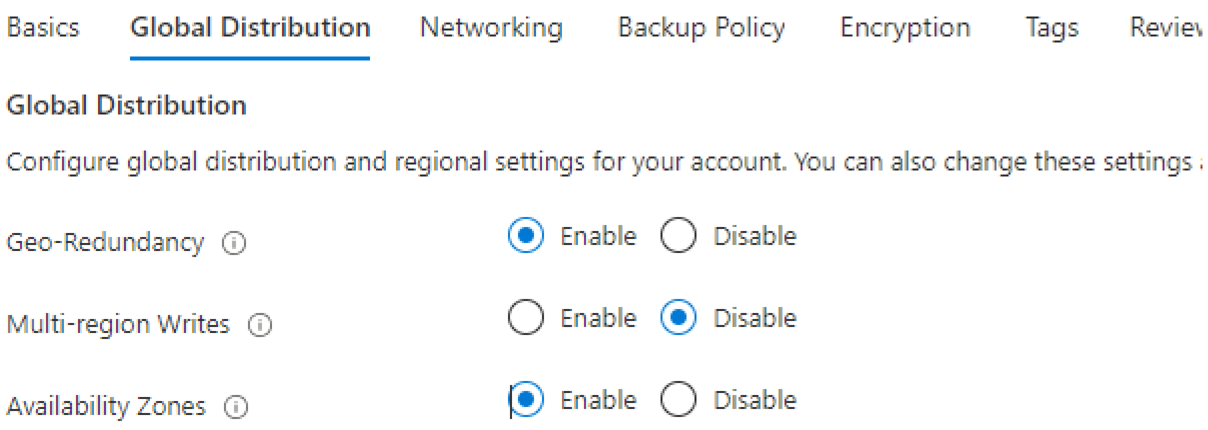
Account Name *	test-mongo1
Location *	(Europe) West Europe
Capacity mode ⓘ	● Provisioned throughput ○ Serverless Learn more about capacity mode

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to on

The subscription you have selected already has an account with free tier enabled.

Apply Free Tier Discount	○ Apply ● Do Not Apply
Version	4.0

ⓘ In order to cover DR scenario we will activate Geo-Redundancy functionality in this way we will our data available in DR location.



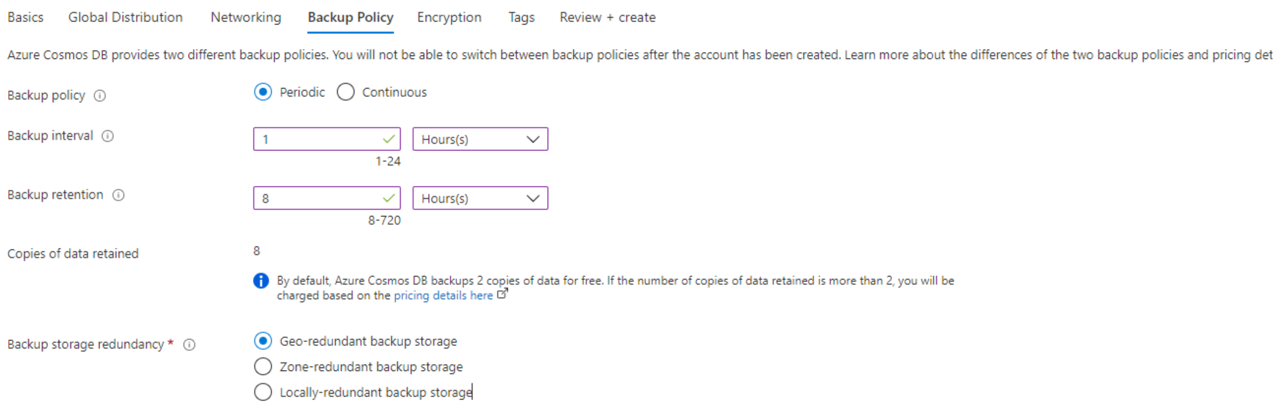
Global Distribution

Configure global distribution and regional settings for your account. You can also change these settings a

Geo-Redundancy ● Enable ○ Disable

Multi-region Writes ○ Enable ● Disable

Availability Zones ● Enable ○ Disable



Azure Cosmos DB provides two different backup policies. You will not be able to switch between backup policies after the account has been created. Learn more about the differences of the two backup policies and pricing det

Backup policy ● Periodic ○ Continuous

Backup interval 1 Hours(s) 1-24

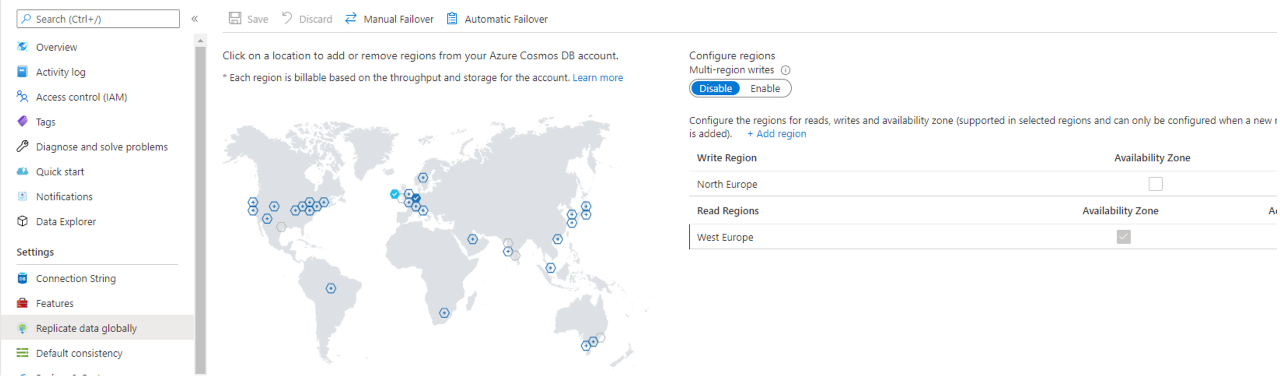
Backup retention 8 Hours(s) 8-720

Copies of data retained 8

By default, Azure Cosmos DB backups 2 copies of data for free. If the number of copies of data retained is more than 2, you will be charged based on the pricing details here

Backup storage redundancy *
● Geo-redundant backup storage
○ Zone-redundant backup storage
○ Locally-redundant backup storage

After the resource is successfully deployed you can check the status of the availability regions :



Click on a location to add or remove regions from your Azure Cosmos DB account.

* Each region is billable based on the throughput and storage for the account. Learn more

Configure regions

Multi-region writes

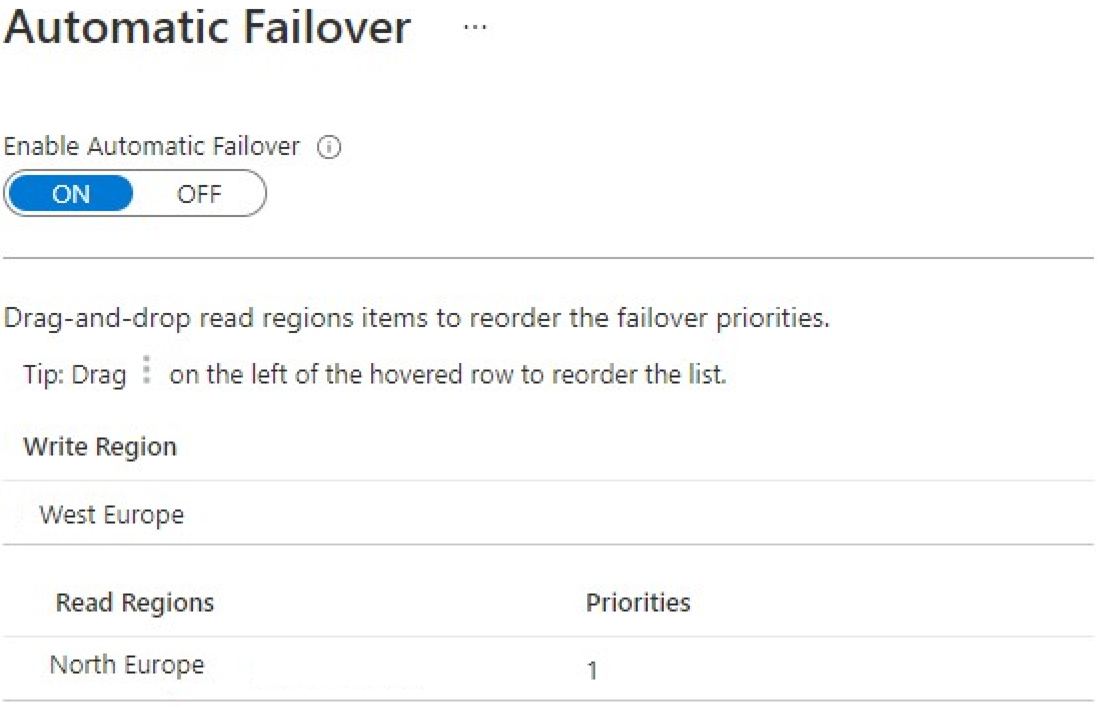
[✓ Disable] Enable

Configure the regions for reads, writes and availability zone (supported in selected regions and can only be configured when a new r is added). + Add region

Write Region	Availability Zone
North Europe	□

Read Regions	Availability Zone	A
West Europe	☑	

and also we need to enable Automatic Failover



Automatic Failover ...

Enable Automatic Failover

[✓ ON] OFF

Drag-and-drop read regions items to reorder the failover priorities.

Tip: Drag ⋮ on the left of the hovered row to reorder the list.

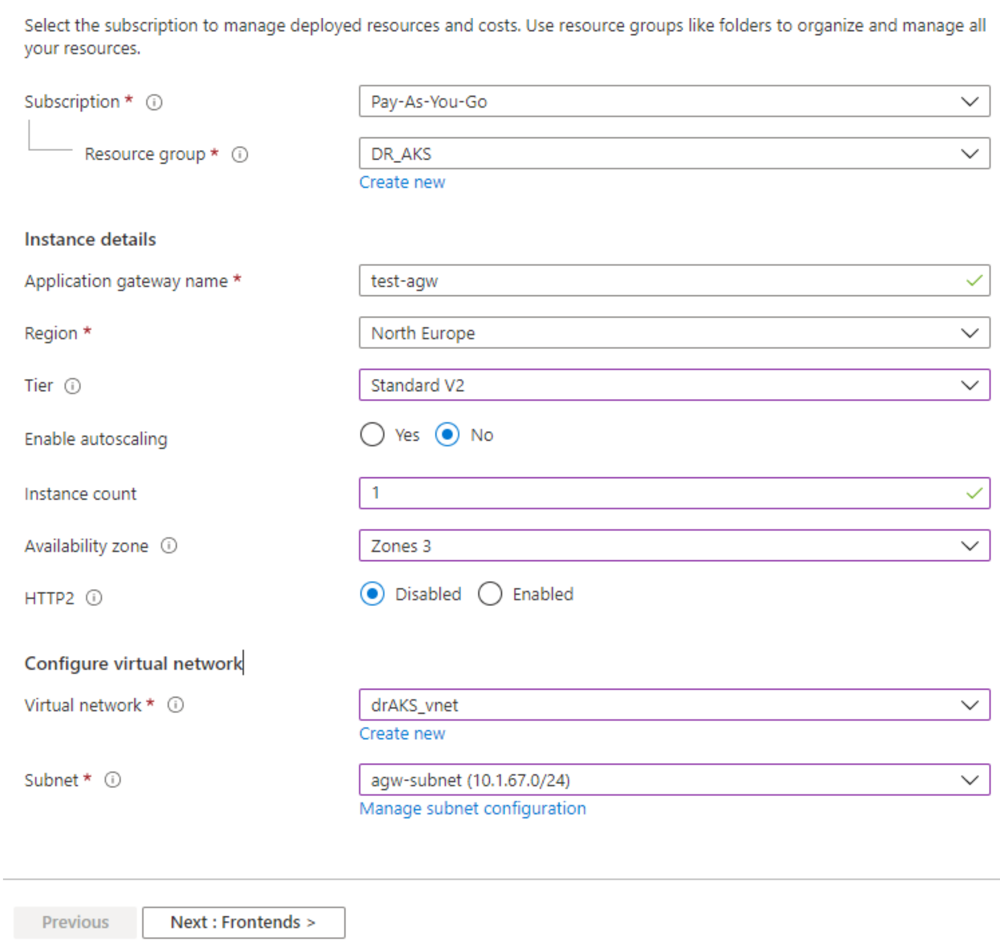
Write Region
West Europe

Read Regions	Priorities
North Europe	1

6 Azure Application gateway

In order for this service to work properly is better to deploy it last.

For deployment we will create in each rg/zone one instance of: “Application gateway“. Each instance will have one public ip and one internal network. For internal network “agw-subnet” will be used.



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

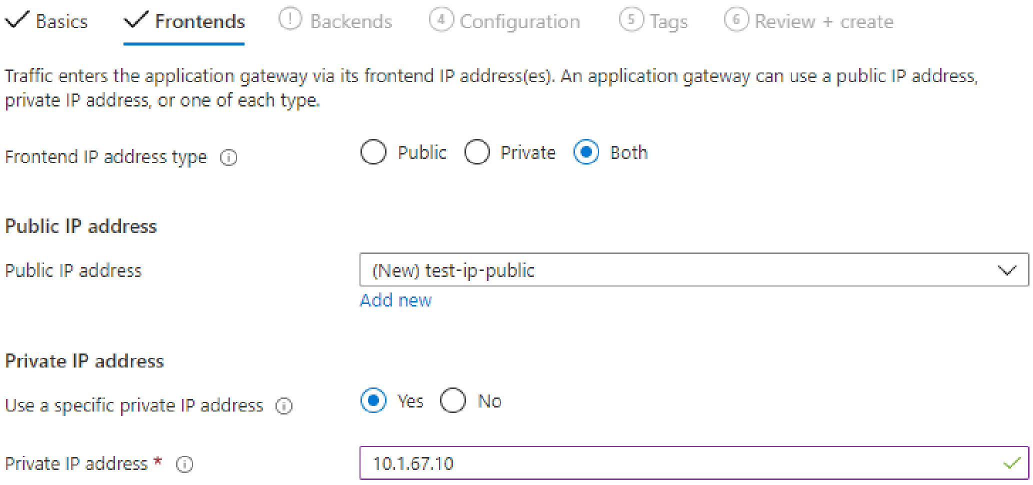
Subscription * ⓘ	Pay-As-You-Go
Resource group * ⓘ	DR_AKS Create new

Instance details

Application gateway name *	test-agw ✓
Region *	North Europe
Tier ⓘ	Standard V2
Enable autoscaling	○ Yes ● No
Instance count	1 ✓
Availability zone ⓘ	Zones 3
HTTP2 ⓘ	● Disabled ○ Enabled

Configure virtual network

Virtual network * ⓘ	drAKS_vnet Create new
Subnet * ⓘ	agw-subnet (10.1.67.0/24) Manage subnet configuration



✓ Basics ✓ Frontends ⓘ Backends ④ Configuration ⑤ Tags ⑥ 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 ⓘ ○ Public ○ Private ● Both

Public IP address

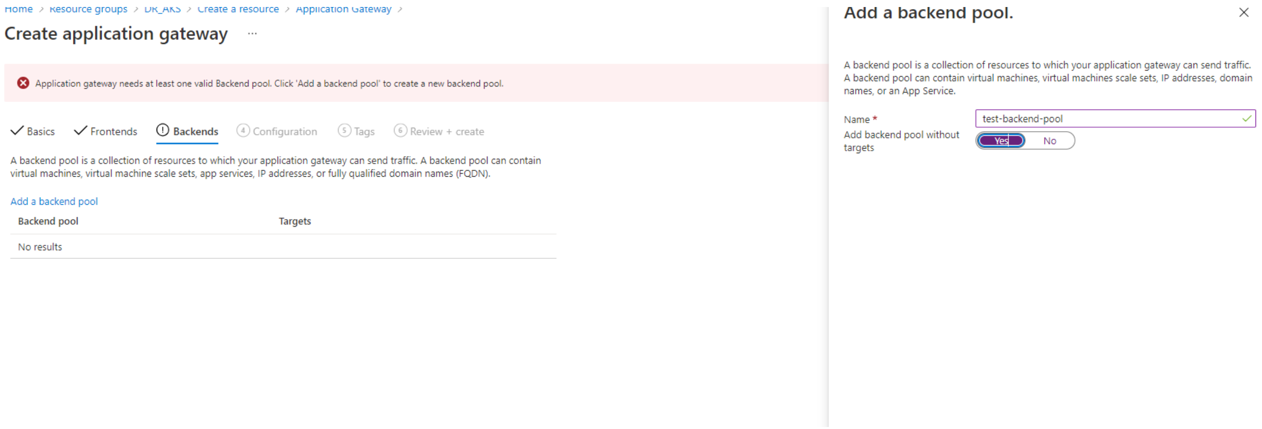
Public IP address: (New) test-ip-public

Add new

Private IP address

Use a specific private IP address ⓘ ● Yes ○ No

Private IP address * ⓘ 10.1.67.10 ✓



Home > Resource groups > DR_AKS > Create a resource > Application Gateway >

Create application gateway ...

Application gateway needs at least one valid Backend pool. Click 'Add a backend pool' to create a new backend pool.

✓ Basics ✓ Frontends ⓘ Backends ④ Configuration ⑤ Tags ⑥ 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
No results	

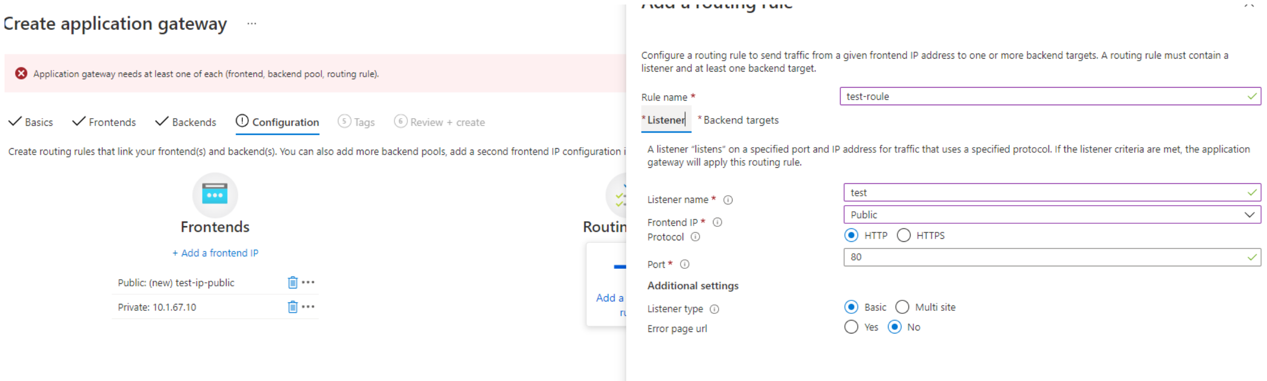
Add a backend pool.

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.

Name * test-backend-pool ✓

Add backend pool without targets [✓ Yes] No

ⓘ In order to finish deployment complete a dummy setup that will be erased later.



Create application gateway ···

Application gateway needs at least one of each (frontend, backend pool, routing rule).

✓ Basics ✓ Frontends ✓ Backends ⓘ Configuration ⑤ Tags ⑥ Review + create

Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration i

Public: (new) test-ip-public

Private: 10.1.67.10

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.

Rule name * test-roule

*Listener *Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener name * test

Frontend IP * Public

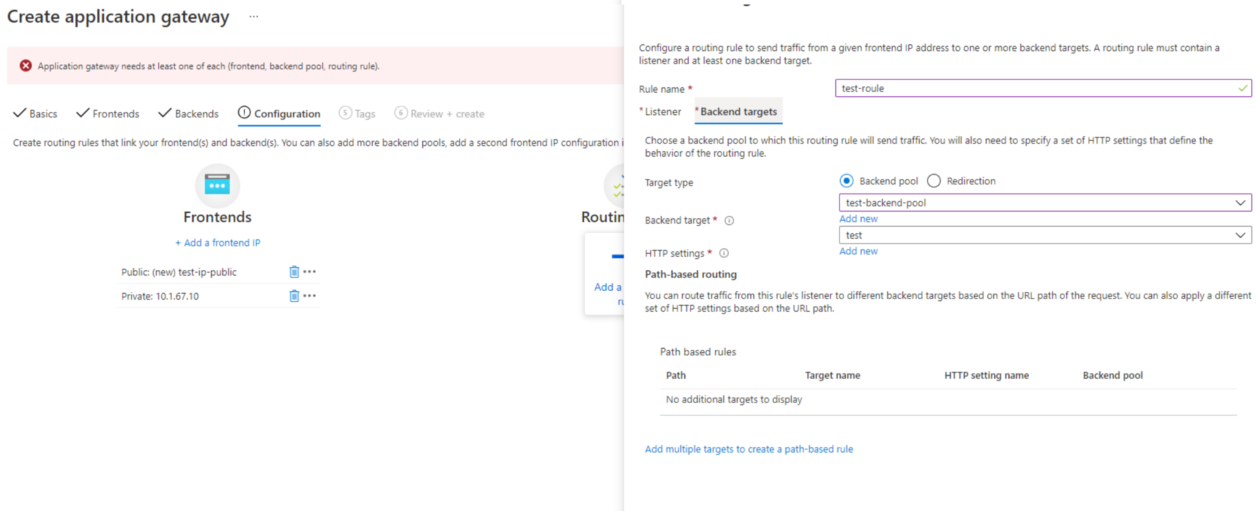
Protocol ● HTTP ○ HTTPS

Port * 80

Additional settings

Listener type ● Basic ○ Multi site

Error page url ○ Yes ● No



Create application gateway ···

Application gateway needs at least one of each (frontend, backend pool, routing rule).

✓ Basics ✓ Frontends ✓ Backends ⓘ Configuration ⑤ Tags ⑥ Review + create

Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration i

Public: (new) test-ip-public

Private: 10.1.67.10

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.

Rule name * test-roule

*Listener *Backend targets

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of HTTP settings that define the behavior of the routing rule.

Target type ● Backend pool ○ Redirection

Backend target * test-backend-pool

Add new

HTTP settings * test

Add new

Path-based routing

You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of HTTP settings based on the URL path.

Path based rules

Path	Target name	HTTP setting name	Backend pool
No additional targets to display			

Add multiple targets to create a path-based rule

After the resource is successfully deployed we can do the configuration:

The order for setup is:

1. Backend Pools - We define here the ip/hosts/fqdn of the services that we need to expose to internet

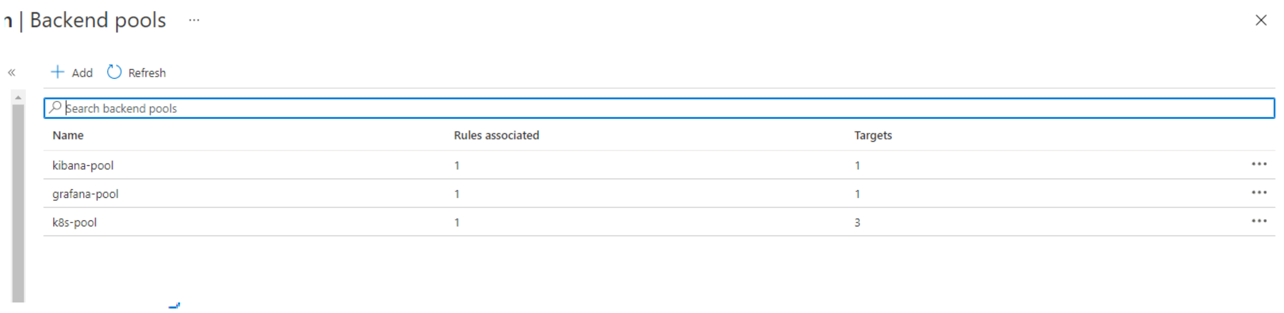
2. HTTP settings - Defines the ports that the services from backend are listenting.

3. Listeners - We define the ports that will be opened to the internet and also here we can set up certificates , timeout settings, cookies etc… .

4 Rules - At this final test we are binding together all settings defined previous.

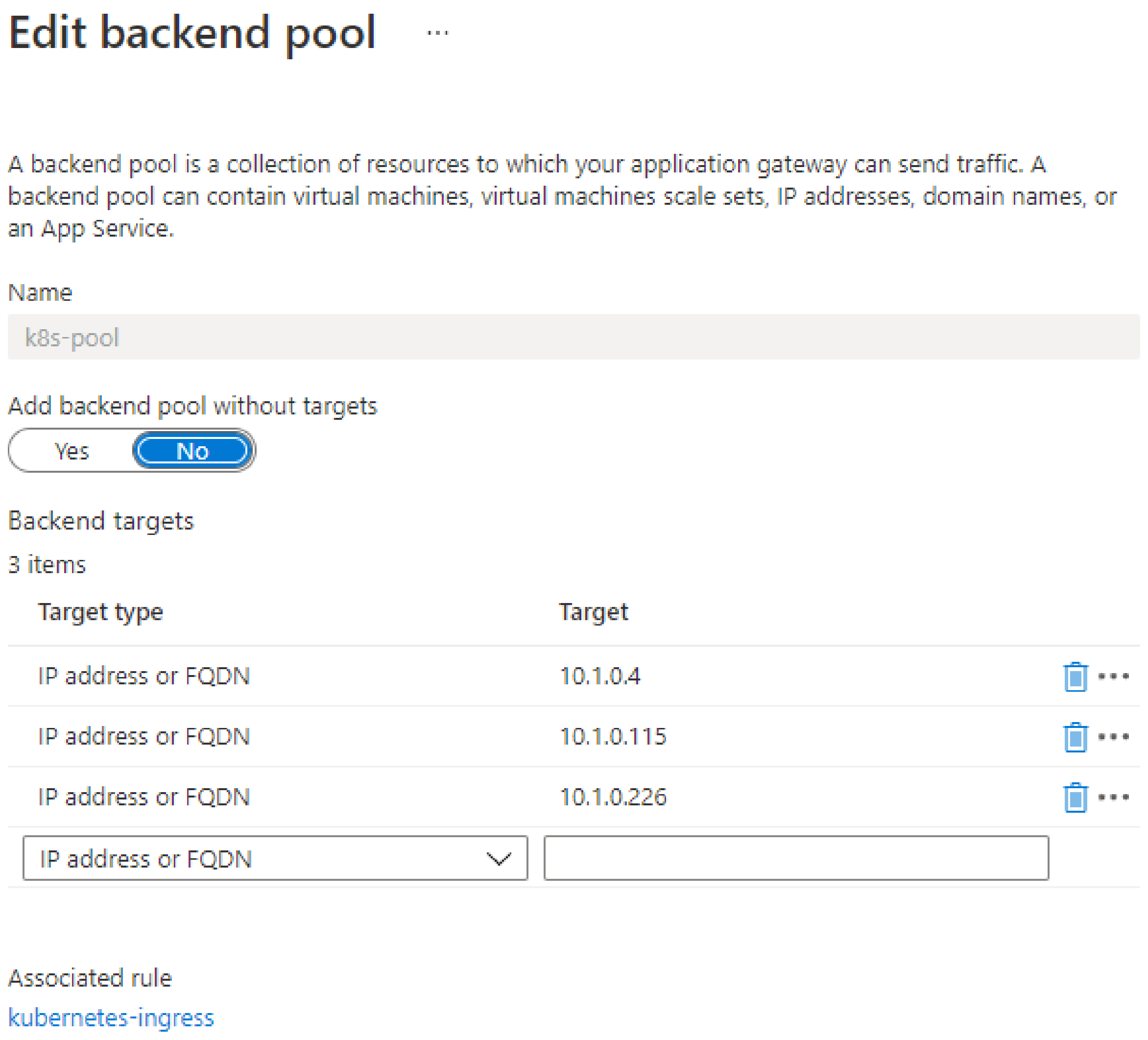
Examples:

Backend pools:



ı | Backend pools ···

Name	Rules associated	Targets
kibana-pool	1	1
grafana-pool	1	1
k8s-pool	1	3



Edit backend pool ···

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.

Name
k8s-pool

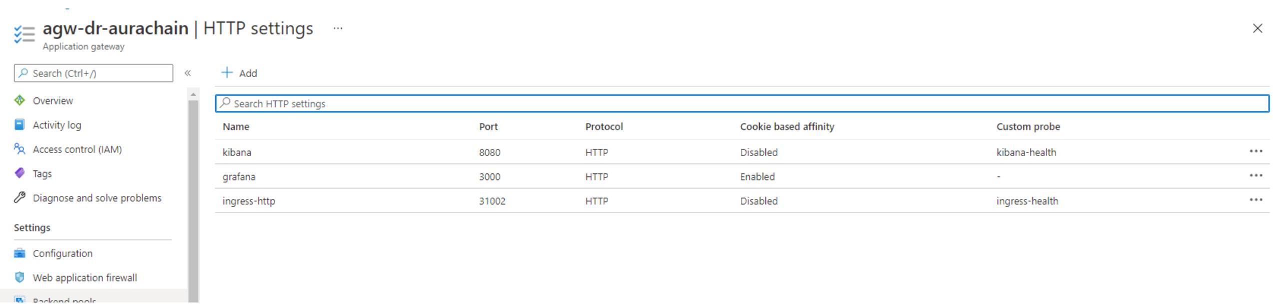
Add backend pool without targets
Yes [✓ No]

Backend targets
3 items

Target type	Target
IP address or FQDN	10.1.0.4
IP address or FQDN	10.1.0.115
IP address or FQDN	10.1.0.226
IP address or FQDN	

Associated rule
kubernetes-ingress

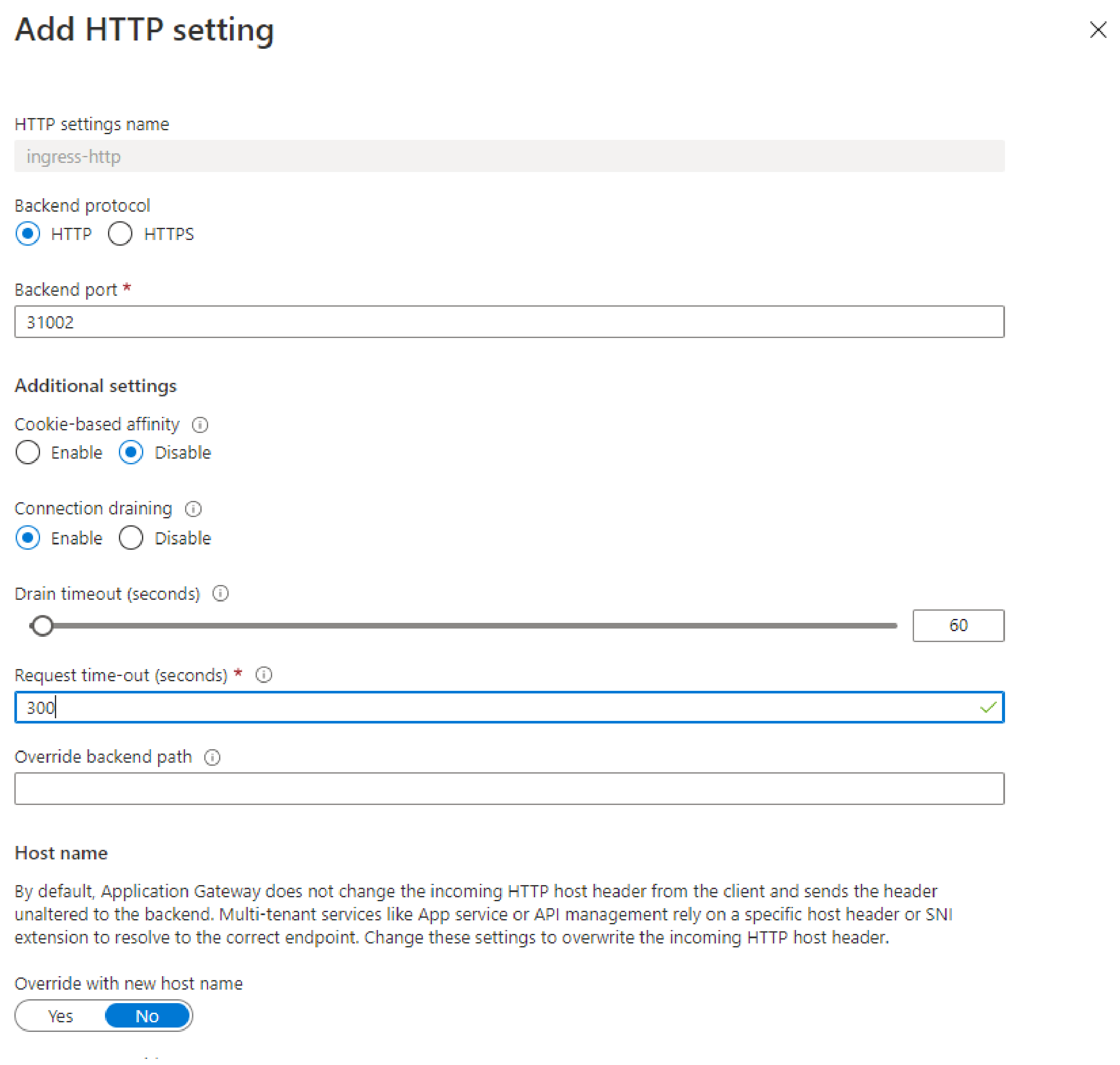
HTTP Settings:



agw-dr-aurachain | HTTP settings ···

Application gateway

Name	Port	Protocol	Cookie based affinity	Custom probe
kibana	8080	HTTP	Disabled	kibana-health
grafana	3000	HTTP	Enabled	-
ingress-http	31002	HTTP	Disabled	ingress-health



Add HTTP setting

HTTP settings name
ingress-http

Backend protocol
● HTTP ○ HTTPS

Backend port *
31002

Additional settings

Cookie-based affinity ⓘ
○ Enable ● Disable

Connection draining ⓘ
● Enable ○ Disable

Drain timeout (seconds) ⓘ
60

Request time-out (seconds) * ⓘ
300 ✓

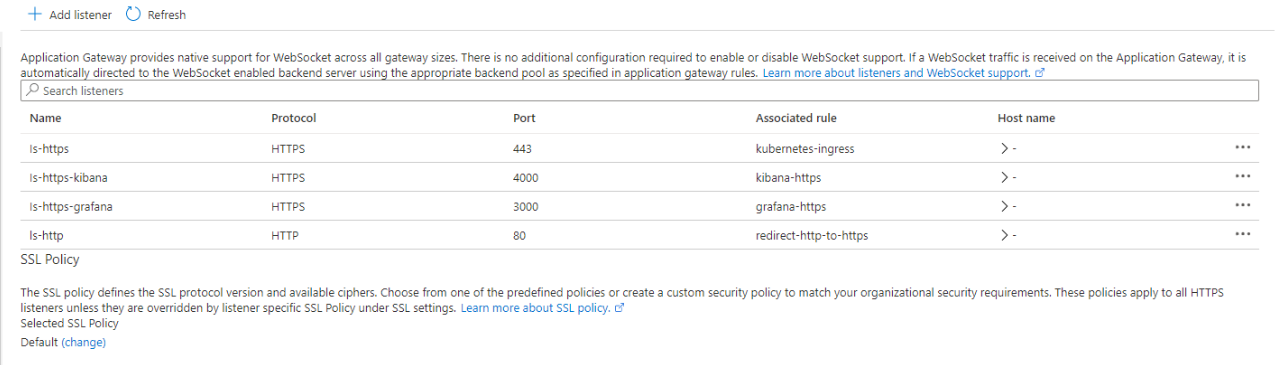
Override backend path ⓘ

Host name

By default, Application Gateway does not change the incoming HTTP host header from the client and sends the header unaltered to the backend. Multi-tenant services like App service or API management rely on a specific host header or SNI extension to resolve to the correct endpoint. Change these settings to overwrite the incoming HTTP host header.

Override with new host name
Yes [✓ No]

Listners:



Application Gateway provides native support for WebSocket across all gateway sizes. There is no additional configuration required to enable or disable WebSocket support. If a WebSocket traffic is received on the Application Gateway, it is automatically directed to the WebSocket enabled backend server using the appropriate backend pool as specified in application gateway rules. Learn more about listeners and WebSocket support.

Search listeners

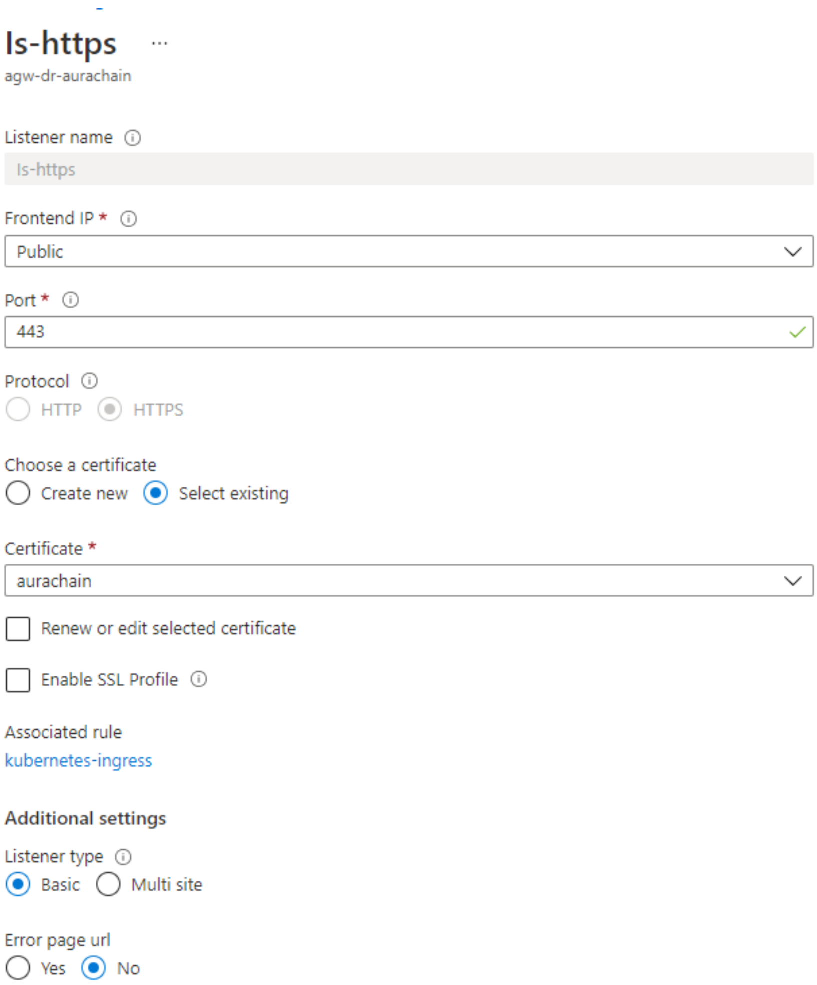
Name	Protocol	Port	Associated rule	Host name	
ls-https	HTTPS	443	kubernetes-ingress	> -	···
ls-https-kibana	HTTPS	4000	kibana-https	> -	···
ls-https-grafana	HTTPS	3000	grafana-https	> -	···
ls-http	HTTP	80	redirect-http-to-https	> -	···

SSL Policy

The SSL policy defines the SSL protocol version and available ciphers. Choose from one of the predefined policies or create a custom security policy to match your organizational security requirements. These policies apply to all HTTPS listeners unless they are overridden by listener specific SSL Policy under SSL settings. Learn more about SSL policy.

Selected SSL Policy

Default (change)



ls-https ···

agw-dr-aurachain

Listener name: ls-https

Frontend IP *: Public

Port *: 443

Protocol: ○ HTTP ● HTTPS

Choose a certificate: ○ Create new ● Select existing

Certificate *: aurachain

□ Renew or edit selected certificate

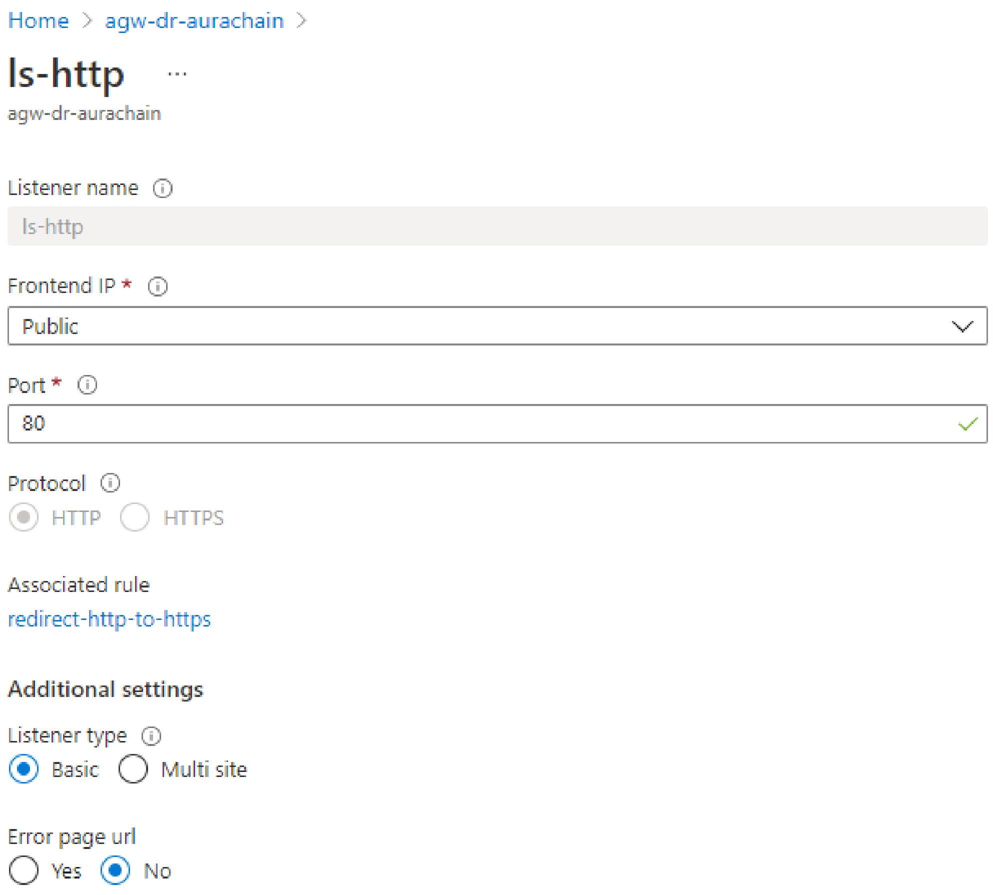
□ Enable SSL Profile

Associated rule: kubernetes-ingress

Additional settings

Listener type: ● Basic ○ Multi site

Error page url: ○ Yes ● No



ls-http ···

agw-dr-aurachain

Listener name ⓘ

ls-http

Frontend IP * ⓘ

Public

Port * ⓘ

80 ✓

Protocol ⓘ

● HTTP ○ HTTPS

Associated rule

redirect-http-to-https

Additional settings

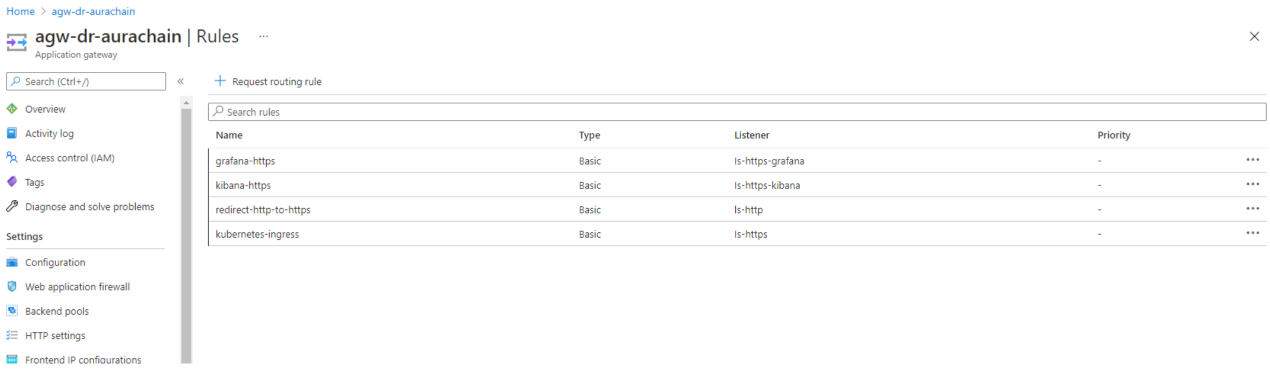
Listener type ⓘ

● Basic ○ Multi site

Error page url

○ Yes ● No

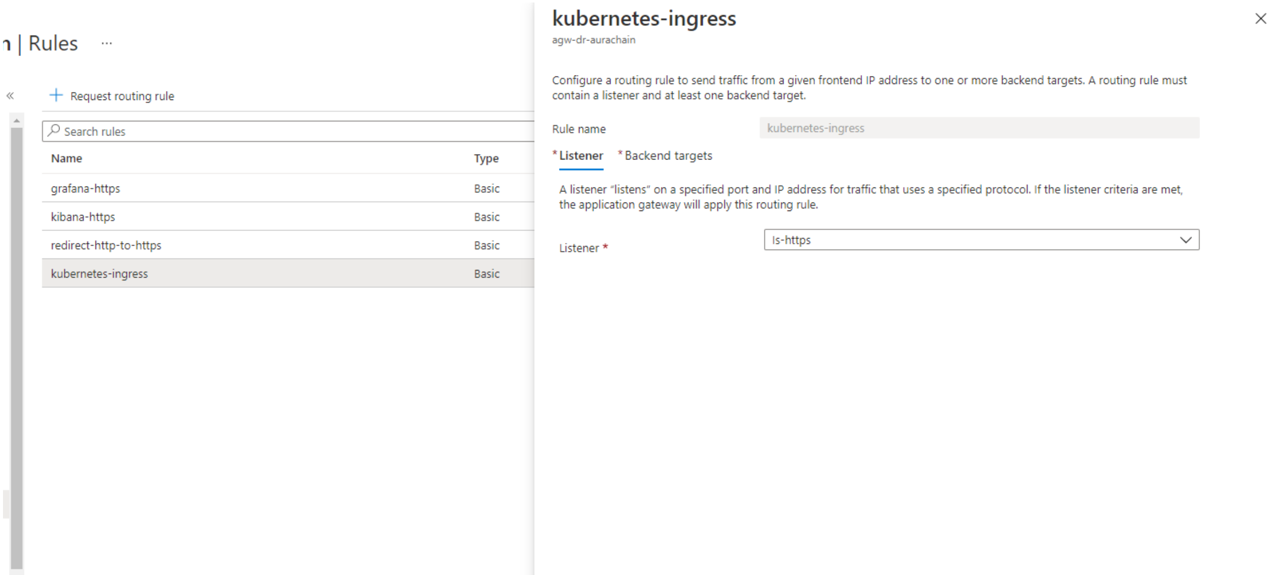
Rules:



agw-dr-aurachain | Rules ···

Application gateway

Name	Type	Listener	Priority
grafana-https	Basic	ls-https-grafana	-
kibana-https	Basic	ls-https-kibana	-
redirect-http-to-https	Basic	ls-http	-
kubernetes-ingress	Basic	ls-https	-



ı | Rules ···

« + Request routing rule

Search rules

Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

kubernetes-ingress

agw-dr-aurachain

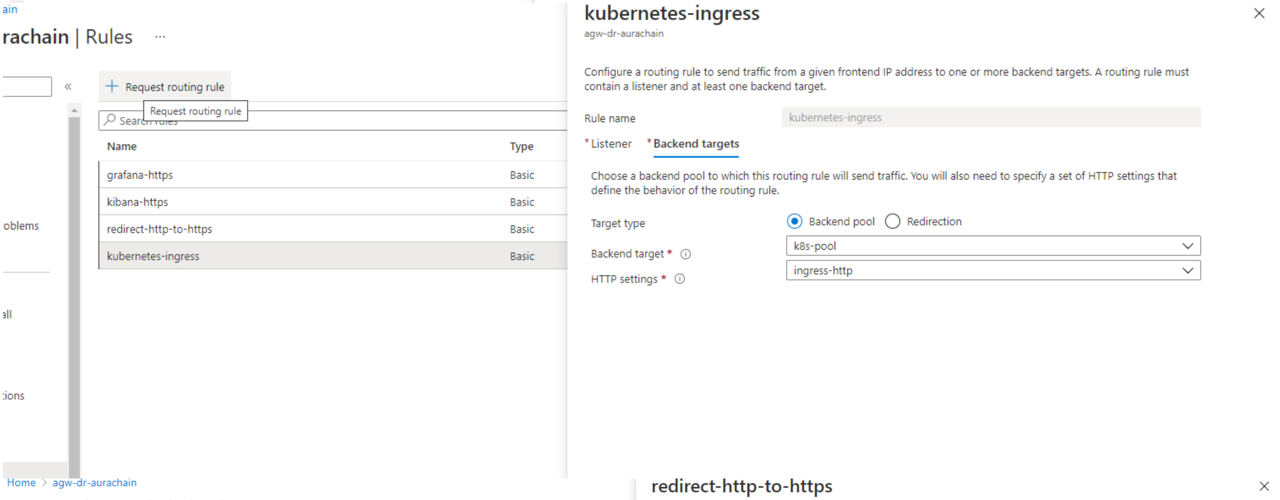
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name: kubernetes-ingress

* Listener * Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener *: ls-https



ain

rachain | Rules ···

« + Request routing rule

Request routing rule

Search rules

oblems

all

ions

Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

kubernetes-ingress

agw-dr-aurachain

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.

Rule name: kubernetes-ingress

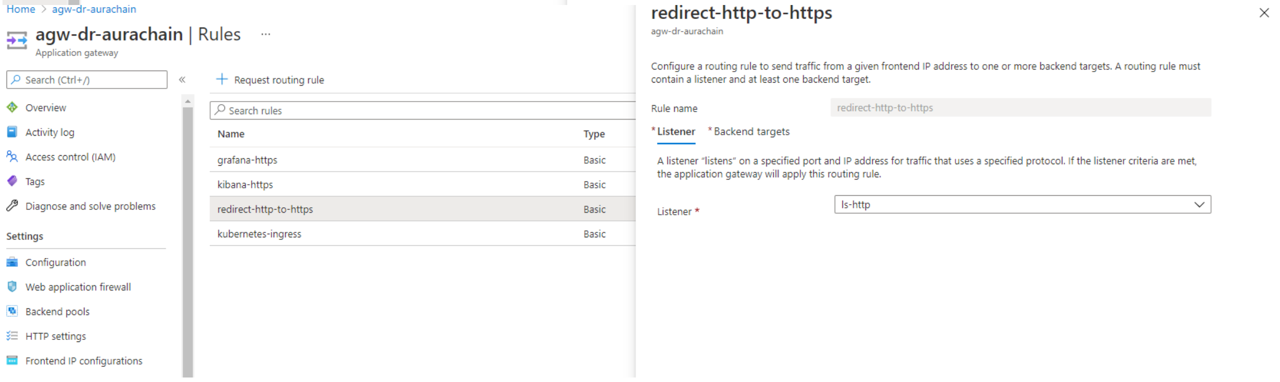
* Listener * Backend targets

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of HTTP settings that define the behavior of the routing rule.

Target type: ● Backend pool ○ Redirection

Backend target * ⓘ: k8s-pool

HTTP settings * ⓘ: ingress-http



Home > agw-dr-aurachain

agw-dr-aurachain | Rules ···

Application gateway

Search (Ctrl+/) «

+ Request routing rule

Search rules

Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

redirect-http-to-https

agw-dr-aurachain

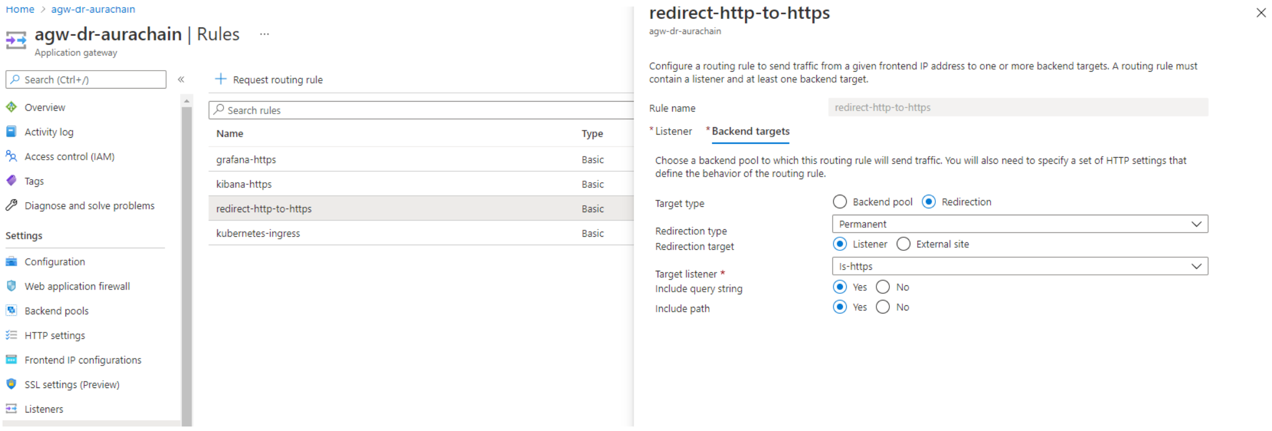
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name: redirect-http-to-https

* Listener * Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener *: ls-http



Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

redirect-http-to-https

agw-dr-aurachain

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.

Rule name: redirect-http-to-https

*Listener *Backend targets

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of HTTP settings that define the behavior of the routing rule.

Target type: ○ Backend pool ● Redirection

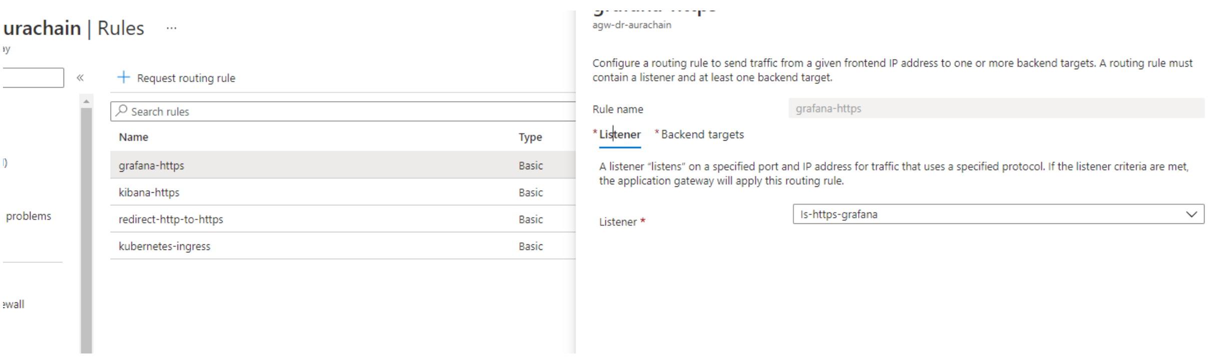
Redirection type: Permanent

Redirection target: ● Listener ○ External site

Target listener *: ls-https

Include query string: ● Yes ○ No

Include path: ● Yes ○ No



Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

agw-dr-aurachain

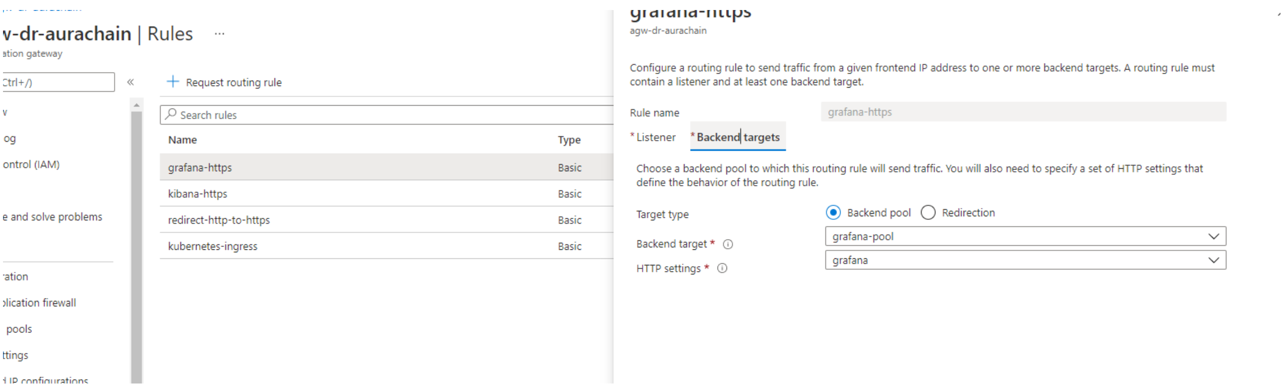
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name: grafana-https

*Listener *Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener *: ls-https-grafana



Name	Type
grafana-https	Basic
kibana-https	Basic
redirect-http-to-https	Basic
kubernetes-ingress	Basic

grafana-https

agw-dr-aurachain

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.

Rule name: grafana-https

*Listener *Backend targets

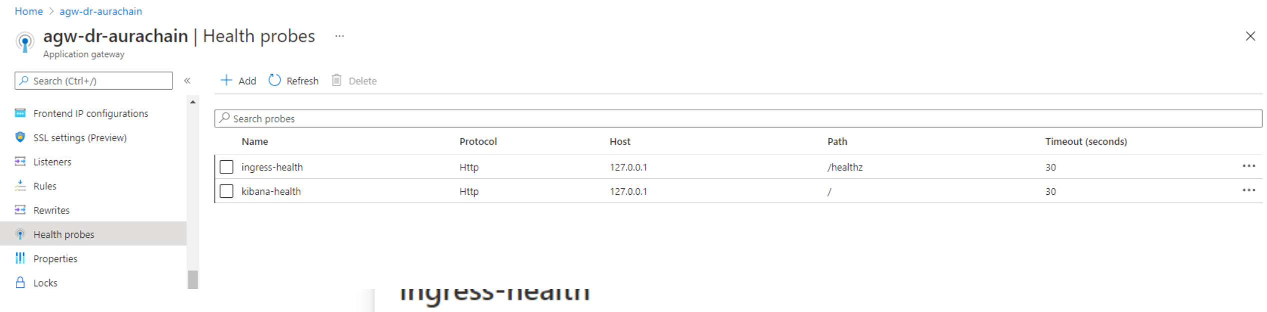
Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of HTTP settings that define the behavior of the routing rule.

Target type: ● Backend pool ○ Redirection

Backend target *: grafana-pool

HTTP settings *: grafana

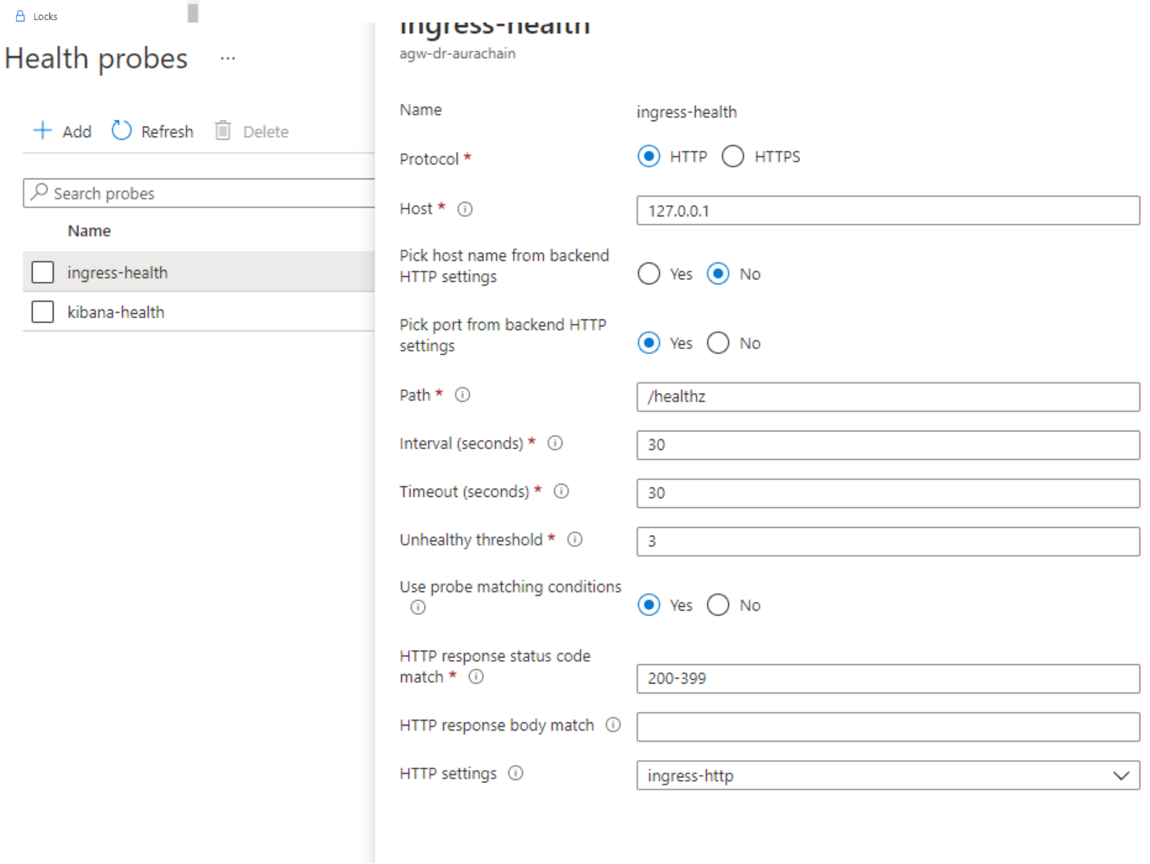
Additional we can create custom probes or health probes that will be added to HTTP settings:



agw-dr-aurachain | Health probes ···

Application gateway

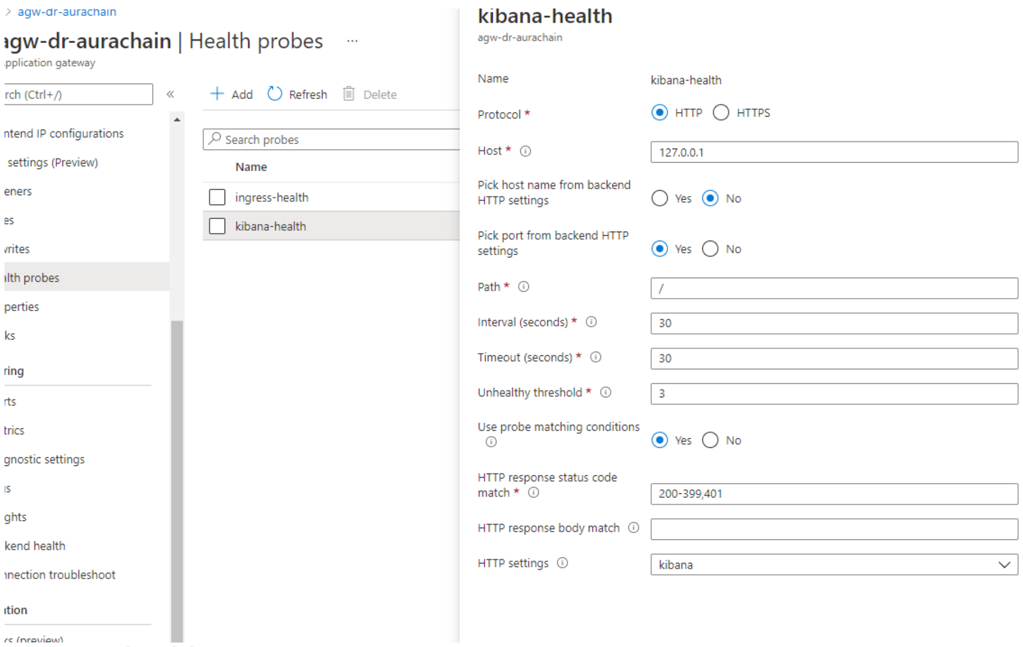
	Name	Protocol	Host	Path	Timeout (seconds)
□	ingress-health	Http	127.0.0.1	/healthz	30
□	kibana-health	Http	127.0.0.1	/	30



ingress-health

agw-dr-aurachain

Name	ingress-health
Protocol *	● HTTP ○ HTTPS
Host * ⓘ	127.0.0.1
Pick host name from backend HTTP settings	○ Yes ● No
Pick port from backend HTTP settings	● Yes ○ No
Path * ⓘ	/healthz
Interval (seconds) * ⓘ	30
Timeout (seconds) * ⓘ	30
Unhealthy threshold * ⓘ	3
Use probe matching conditions ⓘ	● Yes ○ No
HTTP response status code match * ⓘ	200-399
HTTP response body match ⓘ	
HTTP settings ⓘ	ingress-http

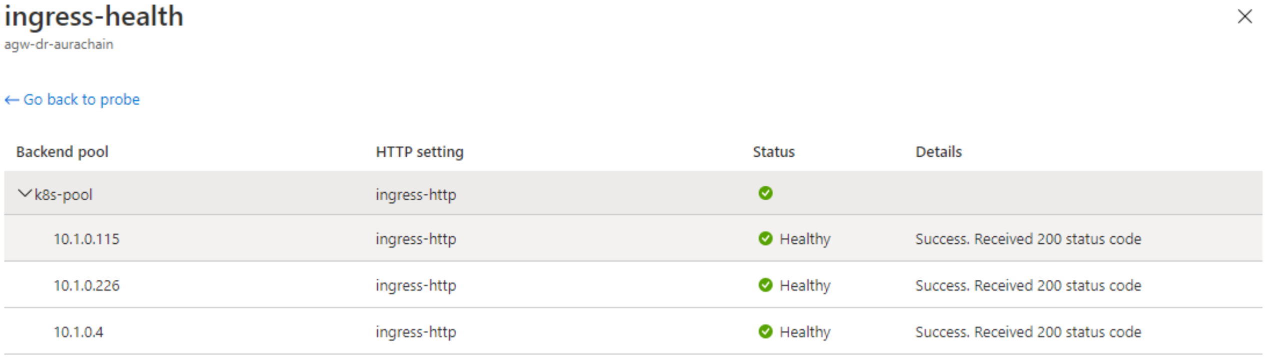


Name
□ ingress-health
□ kibana-health

kibana-health

agw-dr-aurachain

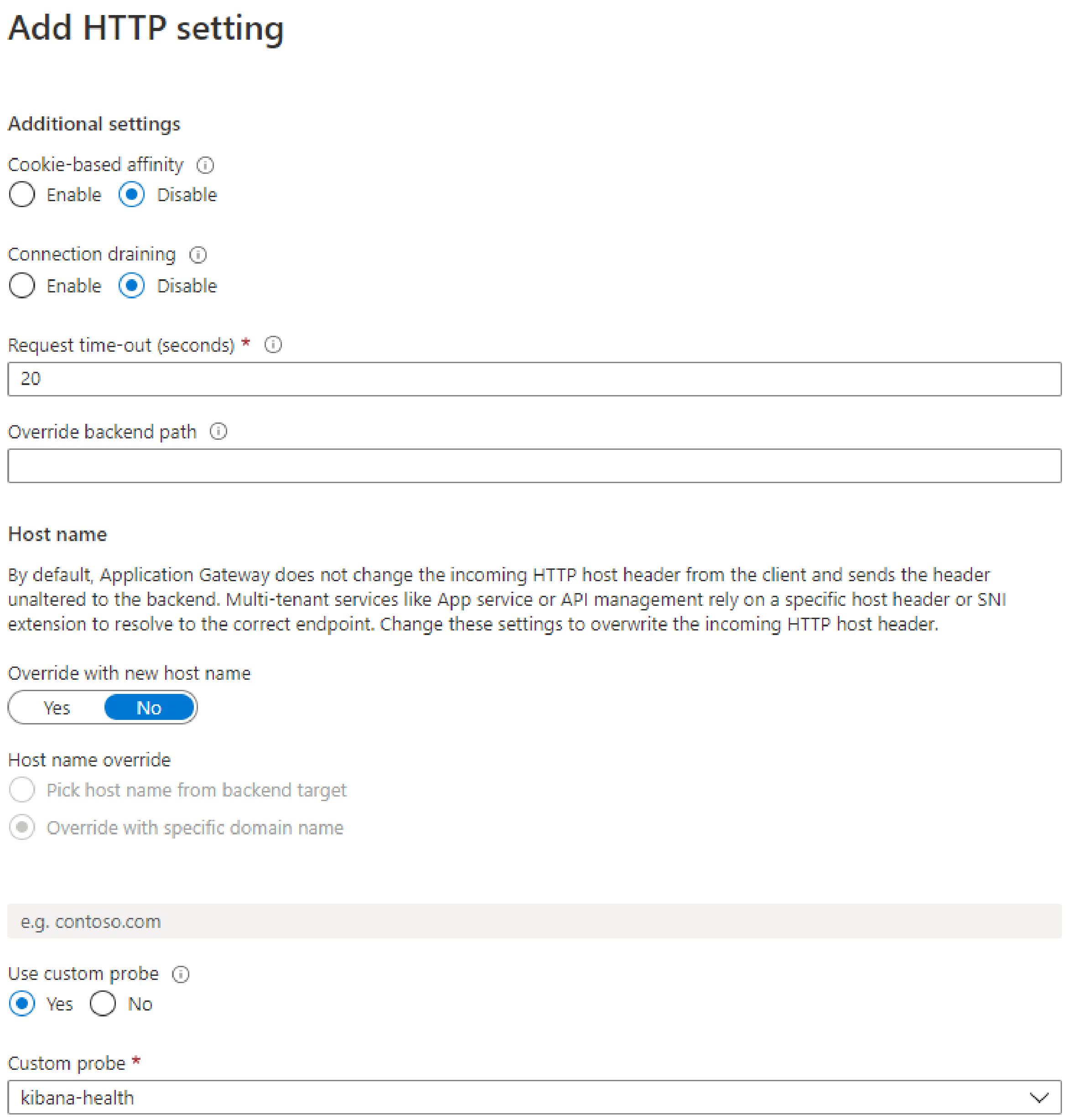
Name	kibana-health
Protocol *	● HTTP ○ HTTPS
Host *	127.0.0.1
Pick host name from backend HTTP settings	○ Yes ● No
Pick port from backend HTTP settings	● Yes ○ No
Path *	/
Interval (seconds) *	30
Timeout (seconds) *	30
Unhealthy threshold *	3
Use probe matching conditions	● Yes ○ No
HTTP response status code match *	200-399,401
HTTP response body match	
HTTP settings	kibana



ingress-health

agw-dr-aurachain

Backend pool	HTTP setting	Status	Details
k8s-pool	ingress-http	✓	
10.1.0.115	ingress-http	✓ Healthy	Success. Received 200 status code
10.1.0.226	ingress-http	✓ Healthy	Success. Received 200 status code
10.1.0.4	ingress-http	✓ Healthy	Success. Received 200 status code



Add HTTP setting

Additional settings

Cookie-based affinity ⓘ

○ Enable ● Disable

Connection draining ⓘ

○ Enable ● Disable

Request time-out (seconds) * ⓘ

20

Override backend path ⓘ

Host name

By default, Application Gateway does not change the incoming HTTP host header from the client and sends the header unaltered to the backend. Multi-tenant services like App service or API management rely on a specific host header or SNI extension to resolve to the correct endpoint. Change these settings to overwrite the incoming HTTP host header.

Override with new host name

Yes [✓ No]

Host name override

○ Pick host name from backend target

● Override with specific domain name

e.g. contoso.com

Use custom probe ⓘ

● Yes ○ No

Custom probe *

kibana-health

ⓘ Do not forget to delete the dummy configuration build at the initial deployment.