KSHEMA – iAgri

Azure Cloud Deployment Physical Architecture

Contents

List of Tables & Figures

1 Architecture – Cloud Infrastructure (Physical) View

This architecture document represents the most optimum options designed to maximise the performance, provide high availability as well as Regional Disaster Recovery in the Azure cloud.

This design utilizes the maximum built-in provisions with configurations that Azure provides in most advanced, managed setup. This will minimize the operational resources (staff as well as associated devices) and effectively utilize the development resources to enable rapid feature releases in future.

Production Region shall be Central India region of Azure that translates to the Azure Datacenter in Pune and the Disaster recovery shall be in South India.

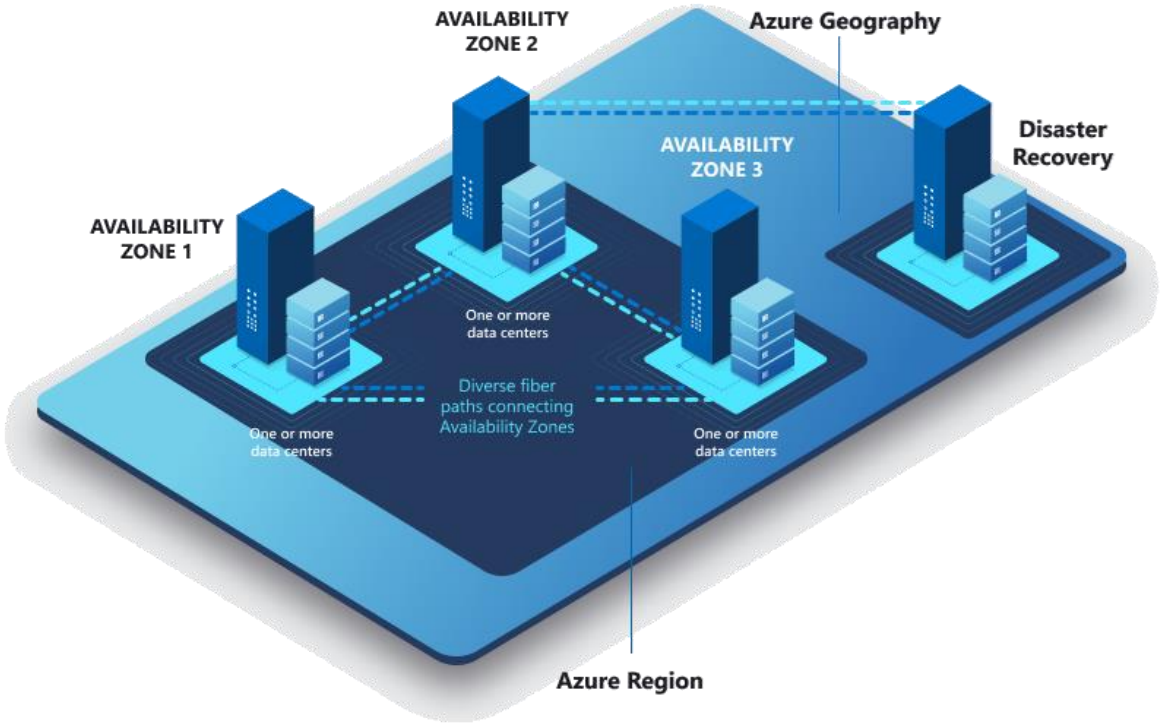


Figure 1: MicroSoft Azure Regions and Availability Zones

The selection and design of the components and setup to provide the platform for Kshema's Phase I services that can be extended easily for future is expanded and developed over the rest of this section and can be viewed in the diagram below.

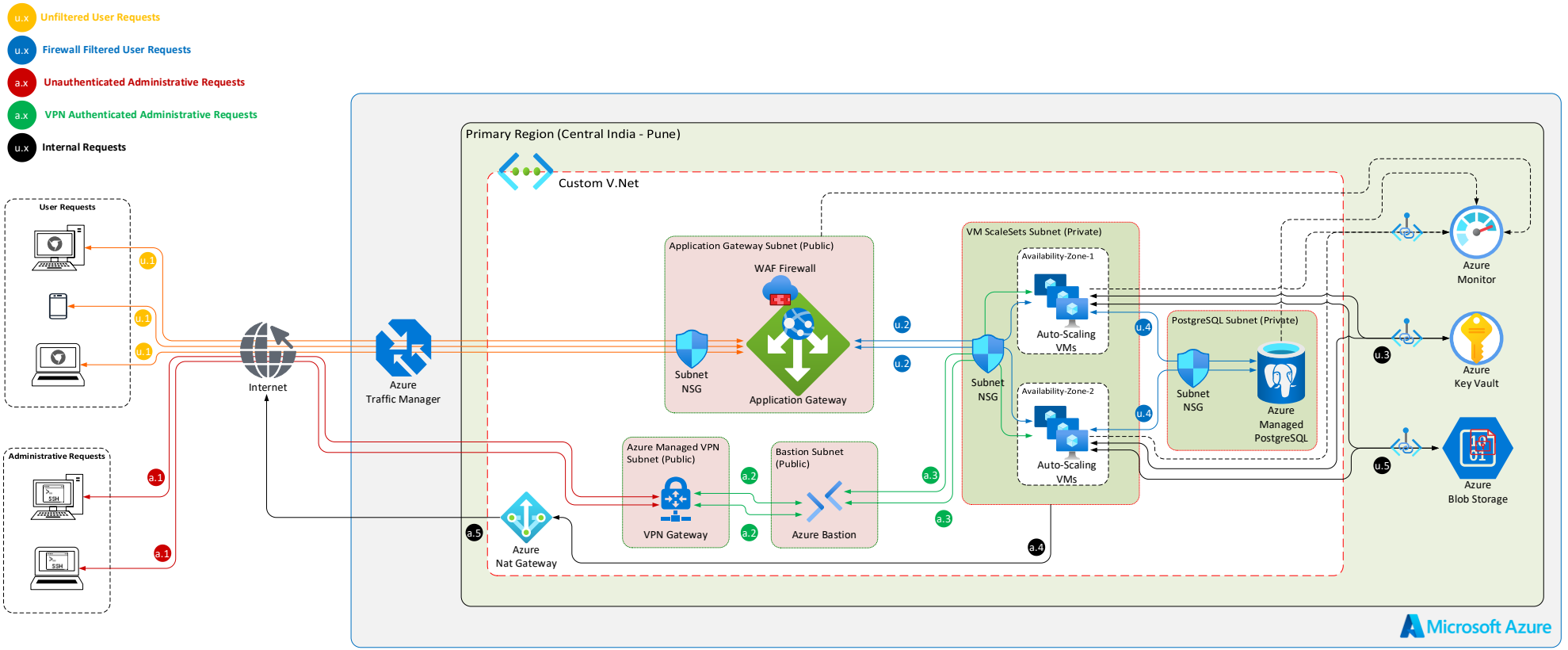


Figure 2: Kshema Physical Architecture

These shall be the components described in the infrastructure view above, and their configurations as implemented for Phase 1. Each of these shall be tweaked for better operability and performance during actual implementation.

1.1 Azure Traffic Manager:

Azure Traffic Manager has been included into the Architecture from the Standpoint of the required Automation in the Disaster Recovery implementation which will be elaborated further under the "Disaster Recovery" section.

Azure Traffic Manager shall automate dynamic routing of the requests from the Production Region's Application Gateway to the DR Region's Application Gateway when the Production Region becomes unavailable due to a disaster.

Resource Name	Prod01
Production Endpoint Name	prod
Production Endpoint Target	20.219.224.109 (Prod. Application Gateway Publi IP)
DR Endpoint Name	dr
DR Endpoint Target	52.172.39.12 (DR Application Gateway Publi IP)

1.2 WAF Firewall:

The WAF Firewall will connect to the Public Application Gateway and play the crucial part of filtering out malicious requests sent to the Application Gateway from the internet by checking the contents of each and every request against the configured security rules like those to prevent SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), blocking and logging these malicious requests. The standard OWASP 3CRS .2 shall be applied /used for Phase-1.

1.2.1 Production Environment

Resource Name	Prod01-apg-waf-policy
Managed Rules Enabled	OWASP_3.2 (185 rules)

1.2.2 DR Environment

Resource Name	Prod01-dr-waf
Managed Rules Enabled	OWASP_3.2 (185 rules)

1.3 Application Gateway

The Application Gateway is the managed and scalable load-balancing solution provided by Azure and will be used here to handle and route requests to the backend VMs after filtering them out through the WAF Firewall mentioned above.

1.3.1 Production Environment

Resource Name	Prod01-apg

Public IP address	20.219.224.109

1.3.2 DR Environment

Resource Name	Prod01-dr-apg
Public IP address	52.172.39.12

1.4 Azure Blob Storage

Azure Blob storage is a cloud-based object storage service provided by Microsoft Azure which will enable you to store and manage massive amounts of unstructured data, such as images, videos, documents, and backups. We propose that with Azure Blob storage, you can securely store your data, access it from anywhere in the world, and scale storage capacity as needed.

1.4.1 Production & DR Environment

Resource Name	Prod01storageaccount

1.5 Azure NAT Gateway

Azure NAT Gateway enables easy outbound connectivity without the need for public IP addresses on individual VMs and since our proposed architecture proposes creating VMs within a Private Subnet to reduce the attack vector, the VMs won't be having a Public IP attached to them and hence will need the NAT Gateway to be able to connect to the outside Internet for downloading OS updates and patches.

1.5.1 Production Environment

Resource Name	Prod01-nat
Public IP address	40.80.82.221

1.5.2 DR Environment

Resource Name	Prod01-dr-nat
Public IP address	20.235.25.172

1.6 Custom Virtual Network (V.Net)

We have designed a Custom Virtual Network to tighten the security of the workload deployed. Custom V.Net is the precursor to creating Custom Public and Private Subnets in which we shall deploy required resources depending on their needing access to the Internet and the Threat model at large.

We shall use an IP range with CIDR – /16 – so that we have an abundant IP address at our disposal for use when required in the future.

1.6.1 Production Environment

Resource Name	Prod01-vnet
IP address range	11.0.0.0/16

1.6.2 DR Environment

Resource Name	Prod01-dr-vnet
IP address range	20.0.0.0/16

1.7 Subnets

We shall create different subnets for different sub-groups of resources to ensure isolation between the different classes of resources – as in business layer, UI layer etc. and control over the ingress-egress of traffic between the resource classes.

1.7.1 Application Gateway Subnet (Public)

This subnet specifically shall host only the Application Gateway with the Web Application Firewall attached to it. The resources running within this subnet shall be provided public IP and explicitly allowed public access to be able to receive and handle requests being sent from the public internet by the end-users.

1.7.1.1 Production Environment

Resource Name	Prod01-apg
IP address range	11.0.2.0/24

1.7.1.2 DR Environment

Resource Name	Prod01-dr-apg
IP address range	20.0.2.0/24

1.7.2 VM Scale Sets subnet (Private)

This subnet shall host the VM spawned through VM ScaleSets. Resources running within this subnet are not provided public IP and shall be denied public access, and only allowed access from select internal azure Services namely: Azure Application Gateway and the Azure Bastion service.

1.7.2.1 Production Environment

Resource Name	Prod01-vmss
IP address range	11.0.3.0/24

1.7.2.2 DR Environment

Resource Name	Prod01-dr-vmss
IP address range	20.0.3.0/24

1.7.3 PostgreSQL subnet (Private)

This subnet shall host the Azure Managed PostgreSQL service. Resources running within this Subnet shall be configured without public IP and explicitly denied public access and only allowed access from select Internal Azure Services namely: the Azure VM ScaleSets.

1.7.3.1 Production Environment

Resource Name	Prod01-pgsql

IP address range	11.0.4.0/24

1.7.3.2 DR Environment

Resource Name	Prod-01-dr-pgsql
IP address range	20.0.4.0/24

1.7.4 VPN Subnet

These are Azure managed subnets to provide VPN gateway into the other private subnets. This is solely meant for deployment, debugging and administration purposes. In subsequent phases, this may be replaced and/or enhanced with additional security/access restrictions.

1.7.4.1 Production Environment

Resource Name	GatewaySubnet
IP address range	11.0.5.0/24

1.7.4.2 DR Environment

Resource Name	GatewaySubnet
IP address range	20.0.5.0/24

1.8 Network Security Groups

1.8.1 Application Gateway Subnet NSG

This shall be the entry point to the whole workload and shall have all the necessary rules required to only allow requests for the ports on which the Application Gateway needs to listen and explicitly deny any other requests sent to any other ports (which are added by default by azure).

- Inbound Rules

Name	Port	Source	Destination	Reason
AllowAnyHTTPInbound	80	Any	Any	Allowing http requests from the outside internet
AllowAnyHTTPSInbound	443	Any	Any	Allowing https requests from the outside internet
AllowAnyCustom65200-65535Inbound	65200-65535	Any	Any	* Required by Azure for infrastructure communication

- Outbound Rules

The default rules will be adequate and shall not be changed.

1.8.2 Production Environment

Resource Name	Prod01-apg-nsg

1.8.3 DR Environment:

Resource Name	Prod01-dr-apg-nsg

1.9 VM ScaleSets Subnet NSG

This NSG again makes sure that only requests from the Application Gateway (which is added by default by azure) and the Azure Bastion service are allowed along with any other IPs required and access from anywhere else is blocked. The rules configured by us as of now are:

1.9.1 Production Environment

Resource Name	Prod01-vmss-nsg

- Inbound Rules

Name	Port	Source	Destination	Reason
AllowCidrBlockCustom8080Inbound	22,80,443	20.198.77.94	Any	Allowing ssh and http/https requests from the public Windows GUI VM being used for manual deployments & testing
AllowCidrBlockCustom80Inbound	80	20.219.224.109	Any	Allowing requests sent from the Application Gateway to the VMs.

- Outbound Rules

The default rules will be adequate and shall not be changed.

1.9.2 DR Environment

Resource Name	Prod01-dr-vmss-nsg

- Inbound Rules

Name	Port	Source	Destination	Reason
AllowCidrBlockCustom80Inbound	80	52.172.39.12	Any	Allowing requests sent from the Application Gateway to the VMs.

- Outbound Rules

The default rules will be adequate and shall not be changed.

1.10 PostgreSQL Subnet NSG

This NSG shall ensure that only requests from the region's Azure VM Scale Sets (which is added by default by Azure) are allowed and rest all are blocked.

1.10.1 Production Environment

Resource Name	Prod01-pgsql-nsg

- Inbound Rules

Name	Port	Source	Destination	Reason
AllowCidrBlockCustom5432Inbound1	5432	20.0.3.0/24	Any	Allowing requests from the DR environment's Postgresql subnet for the Data Sync to be successful between 2 DBs running in 2 different VNets

- Outbound Rules

The default rules will be adequate and shall not be changed.

1.10.2 DR Environment

Resource Name	Prod01-dr-pgsql-nsg

- Inbound Rules

The default rules will be adequate and shall not be changed.

- Outbound Rules

The default rules will be adequate and shall not be changed.

1.11 VM Scale Sets (VMSS):

Azure VMSS is the solution that will satisfy the requirement of scaling the application as the traffic surges as well as managing multiple VM with ease in comparison to manual efforts for the same. The VMs that are deployed in this VMSS will not have any public IP attached to them and hence won't be accessible directly from the internet.

1.11.1 Production Environment

Resource Name	Prod01-vmss
Default Instance Count	2
Maximum Autoscalable Instance Count	10
Instance Size	4 vCPUs, 16GiB Memory, 30GiB Permanent Storage Disk & 150GiB Ephemeral Storage Disk
Operating System	Ubuntu 20.04
Tomcat Server	v9.0.74

Java	v11.0.18
Python	v3.8

1.11.2 DR Environment

Resource Name	Prod01-dr-vmss
Default Instance Count	2
Maximum Autoscalable Instance Count	10
Instance Size	2 vCPUs, 8GiB Memory, 30GiB Permanent Storage Disk & 150GiB Ephemeral Storage Disk
Operating System	Ubuntu 20.04
Tomcat Server	v9.0.74
Java	v11.0.18
Python	v3.8

1.12 Autoscaling in VMSS:

The Autoscaling in VMSS has been configured with the following parameters for both the Prod and DR environment:

Scale Out: Monitor the "Average CPU Percentage" metric of the existing VMs in the Scale Sets and if for 10 consecutive minutes the usage is above 75%, 1 new VM will be launched followed by 5 minutes of Cool down period after which the whole process will start again and check whether now the updated "Average CPU Percentage" metric is below 75% or not, if yes then the Scale Out will stay on standby and if no then again a new single VM will be added to the Scale Set and this process will continue till the "Average CPU Percentage" metric gets below 75%.

Scale In: Monitor the "Average CPU Percentage" metric of the existing VMs in the Scale Sets and if for 5 consecutive minutes the usage is below 25%, any 1 of the existing VMs will be terminated followed by 1 minute of Cool down period after which the whole process will start again and check whether now the updated "Average CPU Percentage" metric is above 25% or not, if yes then the Scale Out will stay on standby and if no then again a single VM will be terminated from the Scale Set and this process will continue till the "Average CPU Percentage" metric gets above 25%.

1.13 Azure Managed PostgreSQL:

Azure Managed PostgreSQL is a fully managed database service provided by Microsoft Azure which enables you to deploy and operate PostgreSQL databases in the cloud without the need for infrastructure management. With Managed PostgreSQL, Azure will handle tasks such as backups, patching, and high availability, freeing you to focus on your application. It will provide built-in scalability, security features, and

integration with other Azure services, making it an efficient and reliable solution for running PostgreSQL workloads in the cloud.

1.13.1 Production Environment

Resource Name	prod01-pgsql
Instance Size	2 vCPUs, 8GiB Memory & 64GiB Permanent Storage Disk
PostgreSQL	v14.7

1.13.2 DR Environment

Resource Name	prod01-dr-pgsql
Instance Size	2 vCPUs, 8GiB Memory & 64GiB Permanent Storage Disk
PostgreSQL	v14.7

1.14 Azure Managed PostgreSQL Maintenance Window:

The Maintenance Window for security patches and updates to the Managed DB has been allowed to be chosen by Azure to choose any day in the week and during that particular day the security patches and updates will be allowed to happen only between 11:00PM to 07:00AM, also Azure will notify 5 days prior to scheduling this activity providing enough time to change the timeslots manually to a more preferred time if required.

https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-maintenance

1.15 Private Endpoints

With Private Endpoints, you can connect to Azure services using private IP addresses, ensuring a secure and isolated connection. The design shall use individual Private Endpoints for enabling communication of the private resources deployed within our custom V.Net to select Azure Managed Services like Azure Monitor, Azure Key Vault and Azure Blob Storage.

1.15.1 Production Environment

Resource Name	Prod01-storage-private-endpoint
Private IP address	11.0.3.8
FQDN	prod01storageaccount.blob.core.windows.net
Private Link Resource	prod01storageaccount

1.15.2 DR Environment

Resource Name	Prod01-dr-storage-private-endpoint
Private IP address	20.0.3.7
FQDN	prod01storageaccount.blob.core.windows.net
Private Link Resource	prod01storageaccount

1.16 VPN

Point to Site VPN with root certificate authentication will be configured for remote access to the private VNETS for administration purposes. The VPN end point is public and not restricted by NSG – advanced configurations shall be considered for Phase-2.

1.16.1 Production Environment

Resource Name	Prod01-VPN
First Public IP Address	20.204.68.97
Second Public IP Address	20.204.68.89
VPN Type	Route-based
Authentication Type	Certificate
User VPN Config. Public IP	20.204.51.194

1.16.2 DR Environment

Resource Name	Prod01-dr-VPN
First Public IP Address	20.235.29.15
Second Public IP Address	20.235.30.10
VPN Type	Route-based
Authentication Type	Certificate
User VPN Config. Public IP	104.211.224.15

2 Disaster Recovery

The Disaster Recovery (DR) design for the cloud infrastructure is scoped to a regional failure. In keeping with the overall principles, DR design leverages to the maximum the Azure built-in provisions for DR – both directly as managed services and also as specific DR capabilities built into the individual Azure services.

The DR shall be an Active – Warm setup to optimize business continuity needs with costs, without compromising on RPO by keeping it near zero and RTO at less than 15 minutes.

The VMs in the DR environment will initially provide for no more than 25% of the performance and will auto scale through Azure VMSS autoscale functionality in relation to the traffic being handled by the DR environment at that given point of time.

ⓘ This is as required for Phase 1 but will need to be revisited for subsequent phases as the demands and infrastructure components increase/expand.

The Production Region for these workloads is in Central India, which is Azure's Datacenter in Pune, and for the Disaster Recovery (DR) Workload, South India is the Azure Region where it is set up which translates to Azure's Datacenter in Chennai.

The DR Region is a replica of the Primary region.

1. Azure Managed PostgreSQL Replication
2. Azure blob storage replication

2.1 Failover

In case of Primary region failure, Azure Traffic Manager (Azure managed service) service will automatically detect the primary region outage and requests will be routed to the DR region. The setup will have a heartbeat of thirty (30) secs and two consecutive failures to trigger automatic DR switchover.

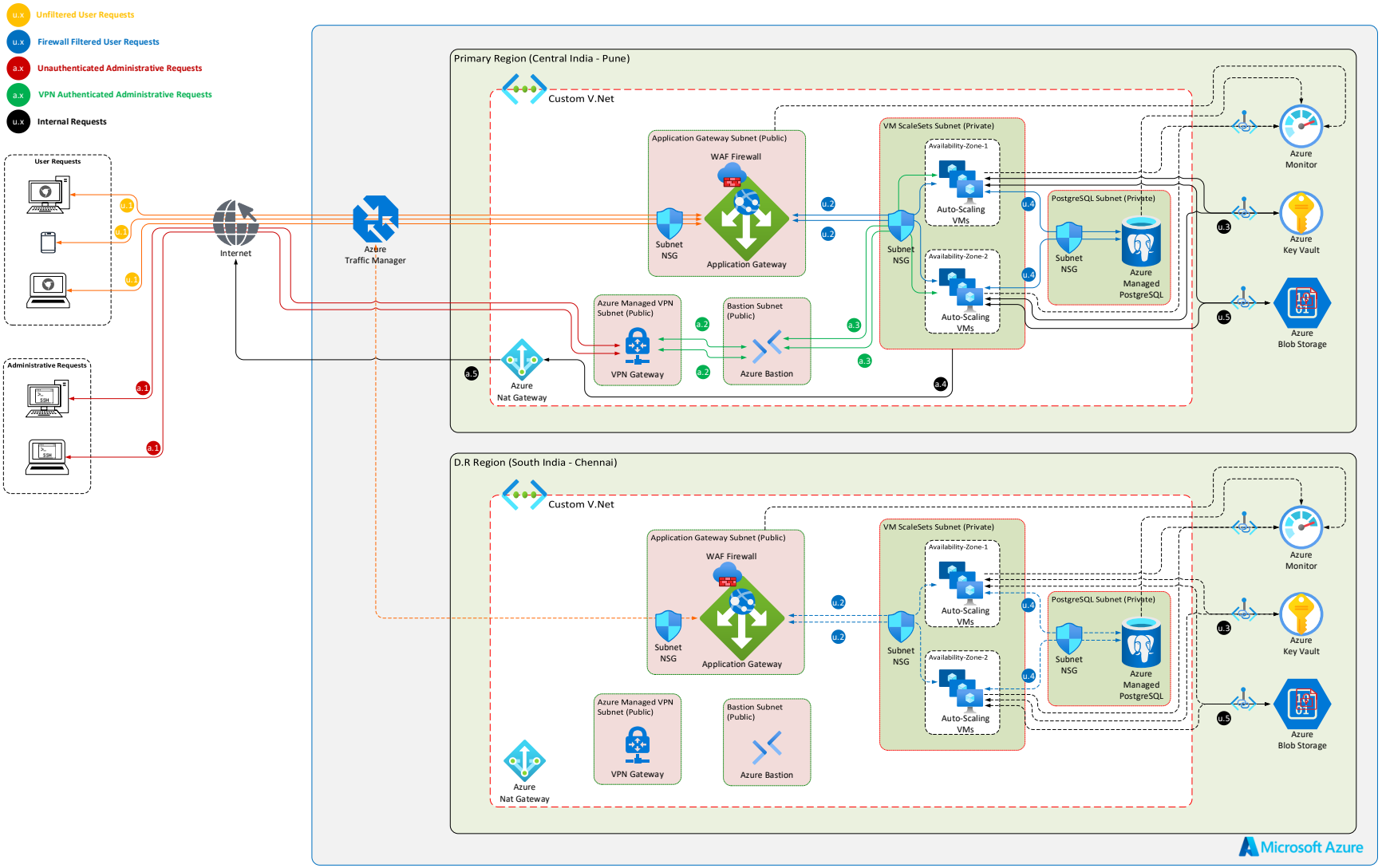
DR Region's VM ScaleSets service will automatically trigger the autoscaling of the Virtual Machine(VM) in DR region after the requests start getting routed to the DR Region.

The RPO is almost NIL and RTO is well under 15 minutes.

2.2 Fail Back

Once the Primary region is back online, Azure Traffic Manager will detect the same and again reroute the requests to the primary region.

Figure 3: Kshema DR Physical Architecture



3 Security

In the Kshema Architecture, many Security measures have been considered to make sure that security is best possible as described in this section.

- Deploying the application within a Custom V.Net will give more control regarding isolating different resources of the workload into their own individual Subnets.
- Creating limited Public Subnets within the Custom V.Net and only deploying resources that specifically need Public Access and the rest of all resources being deployed in their respective Private Subnets.
- Creating NSGs for every individual Subnet so that traffic into and out of that particular Subnet can be tightened and filtered to keep any possible data leaks or access by malicious intent to the minimum.
- Only HTTPS (SSL/TLS) requests in the Application Gateway and redirecting any and all HTTP (NON-SSL/TLS) requests to https to be handled further as https.
- Integrating WAF Firewall with the Public Application Gateway so that any request handled by the Application Gateway is analyzed and filtered out by the WAF Firewall by analyzing the content of the requests and using a set of security rules as below –
    - and algorithms to identify common attack patterns, such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), etc. [OWASP rules set will be imported]
- Azure Managed Bastion service for administrative access to the Azure VMs instead of directly accessing the Azure VMs for tightly controlled high risk, administrator accesses.
- Deploying Azure VMs in a Private Subnet so that the Azure VMs are never accessible directly through the internet – they will never have a Public IP assigned to them and attack surface reduces further.
- Deploying the Azure Managed PostgreSQL in a Private Subnet with no direct/public access to the Database.
- Enabling "Data-In-Transit" encryption for the requests happening between the Azure VMs and the Azure Managed PostgreSQL (optional).
- Using Private Endpoint will enable private access to supported Azure services from within a custom Azure V.Net. So, it will allow private resources, such as the private VMs in our case, to securely access other Azure services, such as Azure Blob Storage and Azure Key Vault and Azure Monitor, using private IP addresses. By establishing a private connection, Private Endpoints keep data traffic within the Azure network, reducing exposure to the public internet and enhancing security. It enables seamless connectivity and integration between private resources and supported Azure services while maintaining data privacy and network isolation.
- Enabling "Data-At-Rest" encryption for all data stored across all the Azure Services being used in this Proposed Architecture
    - Azure VM Volumes,
    - Azure Managed PostgreSQL Volumes,
    - Azure Key Vaults,

- Azure Blob Storage

4 Backup & Archival Policy

This shall be established in subsequent Phases, as it is not anticipated to be needed in Phase 1. This will be integrated with application architecture/design and enabled/supported in cloud/Azure infrastructure.

Keeping in view the Phase 1 constraints, PostgreSQL Flexi Server Backup shall be configured for standard settings as below:

1. Retention period – 7 days
2. Snapshots – Daily, On creation full backup
3. Log Files backed up constantly for near-zero RPO (snapshot restore to PIT using logs)

All Azure Database for PostgreSQL data, backups and temporary files are created during query execution are encrypted using AES 256-bit encryption.

5 TCO estimations

5.1 Production

Production Environment				
Resource	**Usage type**	**Estimated Usage**	**Estimated cost/mo**	**Note**
Bastion - Standard SKU	Azure Bastion pricing is a combination of hourly pricing based on SKU and instances (scale units), plus data transfer rates. - ₹11.560 Per unit/hour(Central): continuosly on - 5 GB Outbound Data Transfer	730 hrs 5GB IO	₹ 17,480.33	₹23.946 per hour
Application Gateway, Web Application Firewall V2	₹41.2128 per gateway-hour ₹1.1776 per capacity unit-hour	730 hrs	₹ 31,247.60	₹41.6160 per Gateway-hour ₹1.1891 per capacity unit-hour
VNet Peering	₹7.43 per GB		₹ 1,486.28	Inbound Data Transfer &Outbound Data Transfer (₹7.43Per GB)
NAT Gateway	₹3.71571 per hour	730 hrs	₹ 2,898.25	₹3.72 Per hour & Data Processed ₹3.72Per GB
VMSS - 2vms with 10autoscaling, Standard_D4ds_v (4 vcpus, 16 GiB memory)	₹20.147 per hour	730 hrs	₹ 29,415.18	Taken only 2 vm's if the scale set incerases the amount increases based on how many vm's scales up and hours running.
Jump Server (D2as v4 (2 vCPUs, 8 GB RAM)	₹13.575 per hour	730 hrs	₹ 9,909.54	
Public IP	7 Public IP (₹0.33029/hour)	730 hrs * 7	₹ 1,687.78	4 Public IP are created &3 Public IP are created for VPN Gateway.
VPN Gateway with P2S	P2S ₹0.01 Per hour/tunnel	730 hrs	₹ 2,20,011.10	₹301.3851/hour &P2S Max 10,000 1-128: Included 129-10,000: ₹0.826/hour per connection
PostgreSql Flexible Server Deployment, General Purpose Tier, 1 D2ds v4 (2 vCores) x 730 Hours (Pay as you go), 64 GiB Storage, 100 GiB Additional Backup storage - GRS redundancy, with High Availability	Instance -₹20.7254 per hour Storage -₹10.817 per Gib Backup storage -₹7.844 per Gib	730 hrs	₹ 32,428.04	
Blob Storage: StorageV2 (general purpose v2)-100GB(assuming)	Storage -₹7.844 per Gib	100GB	₹ 2,978.05	Additional Charges are applicable
Azure DNS	₹41.286 per zone per month	730 hrs	₹ 115.60	
Private endpiont	₹0.826 Per unit/hour	730 hrs	₹ 1,370.68	
Traffic Manager	DNS -₹44.589 per million queries Endpoint -₹44.59 per month	730 hrs	₹ 535.06	
Total cost			**₹ 3,51,563.49**	

5.2 DR

DR Environment				
Resource	**Usage type**	**Estimated Usage**	**Estimated cost / mo**	
Bastion - Standard SKU		730 hrs	₹ 17,480.33	
Application Gateway	Web Application Firewall V2	730 hrs	₹ 31,247.60	
NAT Gateway	₹3.71571 per hour	730 hrs	₹ 2,898.25	
VNet Peering	₹7.43 per GB		₹ 1,486.28	
Virtual Machine - Standard D2s_v3 (2vcpus, 8 GiB memory)	₹8.670 per hour	730 hrs	₹ 12,658.17	
Public IP	6 Public IP (₹0.33029/hour)	730 hrs*6	₹ 1,446.67	3 Public IP are created &3 Public IP are created for VPN Gateway.
VPN Gateway with P2S	P2S ₹0.015 Per hour/tunnel	730 hrs	₹ 2,20,011.10	
Private endpiont	₹0.826 Per unit/hour	730 hrs	₹ 1,370.68	
Total Cost			**₹ 2,88,599.08**	

Already included with PROD				
Azure Database for PostgreSQL flexible server	General Purpose, D2ds_v4, 2 vCores, 8 GiB RAM, 64 GiB storage		₹ 33,206.85	
Blob Storage	StorageV2 (general purpose v2)- 5tb(assuming)		₹ 2,978.05	
Traffic Manager	$0.54 per million queries		₹ 535.06	
Azure DNS	$0.50 per Zone per month		₹ 115.60	

6 Phase-2: Suggestions

- The SSL requests are being offloaded at the Application Gateway and the requests between the Application Gateway and the VMs (Tomcat application) are unencrypted. These requests also be encrypted for increased security during intercommunication of services.
- All the requests between the VMs (Tomcat application) and the PostgreSQL database are unencrypted for now, these will be encrypted as well to achieve higher security in intercommunication of services.
- The deployment is being done manually for now through a Jump Server with Windows GUI in addition to a Bastion. A expanded DevOps design and setup including automated deployment by implementing CICD so that the Windows GUI Jump Server is then no longer required.

7 Appendices

7.1 Tomcat production environment deployment standards

- TomCat runs as non-root user the production "tomcat" user which is a user created by the tomcat service is the process owner and executor. This user is a special, no-login user, but owns all the tomcat processess and resources.
- Error response configuration Custom 404, 500 error pages shall be created and configured into the tomcat ROOT environment.
- Standard REST service post Kshema's services shall be served on the standard HTTP/S ports, re-configuring tomcat's default setup to these standard ports (HTTP: 80).
- Tomcat service *shall be* created and deployed on the production, to enable the standard Linux service sub-system shall manage and handle the tomcat base process. This includes the tomcat processes being started up on system boot, stopped on system shutdown as well as restarted on unscheduled termination.

7.2 iAgri Application Deployment procedure

This section details out a sample application deployment steps/description that would be provided by the development team for production deployments. It may be required of such a deployment to include some security hardening steps.

Step 1: Startup a VM based on the latest/current base VM and open an SSH (Putty or like) connection to it.

Step 2: Shutdown tomcat (*systemctl stop tomcat*)

Step 3: Copy the new WAR file to the tomcat applications path /opt/tomcat/webapps/

Step 4: Copy any additional core libraries or property files to /opt/tomcat/lib

Step 5: Copy th SQL configuration file iAgri.XML to /opt/tomcat/conf/Catalina/localhost/ and update the SQL server details to match production PostgreSQL.

Note: confirm and ensure all resources under /tomcat are owned by *tomcat* user, else change ownership

Step 6: Start the tomcat service (*systemctl start tomcat*)

Step 7: Check the URL https://<<VM IP address>> at the REST service /application paths (e.g. https://162.167.67.11/iAgri/REST/testconnection) and confirm application responses are as expected.

Step 8: Stop the base VM updated above, capture it and store the image in the Azure compute galleries in Prod01_production_gallery with new version.

Step 9: Update the newly created image above from compute gallery to the production and DR VMSS. Restart the VMSS. [Note: update and restart the DR VM at least fifteen minutes after production VM has started up]

Step 10: Check the URL https://prod01.kshema.co at the REST service /application paths and confirm application responses are as expected.

7.3 Azure Bastion

The Azure Bastion is the managed Bastion Host service provided by Azure and which provides the feature of accessing resources which do not have direct internet access and hence cannot be directly connected into from the outside internet for administrative purposes. In this case, it will be available for use by Infrastructure Administrators to access the Private Subnet VMs to be able to configure/tweak the environment of the VMs and also will be available for Developers to be able to connect to the Private VMs for purposes of debugging the code and any other activity falling under Administration, Maintenance and Debugging at large.

7.3.1 Production Environment

Resource Name	Prod01-bastion
Public IP address	20.235.103.193

7.3.2 DR Environment

Resource Name	Prod01-dr-bastion
Public IP address	13.71.83.5

7.3.3 Bastion Subnet (Public)

This subnet specifically hosts only the Azure Managed Bastion service. Also, the resources running within this Subnet are provided Public IPs and explicitly allowed Public Access to be able to receive and handle Requests being sent through the Public Internet by the Infrastructure Administrative users.

7.3.3.1 Production Environment

Resource Name	AzureBastionSubnet
IP address range	11.0.1.0/26

7.3.3.2 DR Environment

Resource Name	Prod01-dr-apg
IP address range	20.0.1.0/26

7.3.4 Azure Bastion Subnet NSG

This shall be the Administrative/Debugging entry point to the VM workload by the Administrators and shall have all the necessary rules required to only allow requests from specific IPs for the ports through which the cli access to the VMs can occur and explicitly deny any other requests sent to any other ports (which are added by default by azure).

- Inbound Rules

Name	Port	Source	Destination	Reason
AllowMyIpAddressHTTPSInbound	443	125.211.115.190	Any	Allowing encrypted requests from Kshema's On-Premises Public IP.
AllowGatewayManagerInbound	443	GatewayManager	Any	* Required by Azure for internal infrastructure communication
AllowAzureLoadBalancerInbound	443	AzureLoadBalancer	Any	* Required by Azure for internal infrastructure communication
AllowBastionHostCommunication	8080,5701	VirtualNetwork	VirtualNetwork	* Required by Azure for internal infrastructure communication

- Outbound Rules

Name	Port	Source	Destination	Reason
AllowSshRdpOutbound	22,3389	Any	VirtualNetwork	* Required by Azure for internal infrastructure communication
AllowAzureCloudOutbound	443	Any	AzureCloud	* Required by Azure for internal infrastructure communication
AllowBastionCommunication	8080,5701	VirtualNetwork	VirtualNetwork	* Required by Azure for internal infrastructure communication

Name	Port	Source	Destination	Reason
AllowHttpOutbound	80	Any	Internet	* Required by Azure for infrastructure communication

7.3.4.1 Production Environment

Resource Name	Prod01-bastion-nsg

7.3.4.2 DR Environment:

Resource Name	Prod01-dr-bastion-nsg

7.4 Self-Hosted Windows Jump Server VM

This is a temporary VM that has been created for the Developers to initially have some ease in deploying consecutive versions of the code into the VMs running under Private VMSS. We already have an Azure managed Bastion service proposed and deployed in the environment for the same purpose but since it provides only a CLI interface, we had to deploy a self-hosted Windows VM in the "default" subnet of the Custom V.Net in the Production environment which has public access enabled but the access is narrowed down to be possible only from within the Kshema Network. This resource is supposed to be deleted as soon as the developers get comfortable with the CLI Interface of the Azure Managed Bastion Service.

Resource Name	Prod01-jump-server-vm
Public IP address	20.198.77.94
Instance Size	2 vCPUs, 8GiB Memory & 127GiB Ephemeral Disk
Operating System	Windows 10 Pro

7.4.1 Windows Jump Server NSG[1]

This NSG shall be created to secure access to the *temporary* GUI Jump Server that has been created for the developers to be able to upload the new versions of the WAR files into the private VM resources.

- Inbound Rules:

Name	Port	Source	Destination	Reason
RDP	3389	125.21.115.190	Any	Allowing RDP access from within the Kshema Public IP to the Windows Jump Server

- Outbound Rules

The default rules will be adequate and shall not be changed.

[1] This is on-request from Kshema team for enhancing their comfort in accessing the production systems, and was not designed nor deemed essential

8 Version History

Version #	Date of publication	Author(s)	Approvers(s)	Publication Notes
1.0	2023 May 26	Srinivas Kamath, Pardhasaradhi Vasamsetty	Kannan Rajaram	Final version – Phase 1, including updates from analysis and reviews
1.1	2023 Jun 15	Srinivas Kamath, Pardhasaradhi Vasamsetty	Kannan Rajaram	Updated with changes requested by Kshema team (i) Addition of VPN and (ii) Database Backup configuration