

|  |  |
| --- | --- |
|  |  |
| Student Names: | Nitika Gupta  Kanan Garach  Anjali Kumari  Thanh Trung Nguyen |
| Deliverable: | Build Guide\_Group1 |
| Course Name: | SYST8200-22W-Sec1-Capstone Implementation |
| Date Assigned: | Feb 13, 2022 |
| Date Due: | March 26, 2022 |

|  |
| --- |
| **Document Version: Build Guide\_Work Instructions\_Group1\_v1.0** |
| **Document Status: Final Release** |
| **Document Author: Nitika Gupta, Anjali Kumari, Kanan Garach, Thanh Trung Nguyen** |
| **Line of Business: Retail** |
| Project Objective: Build of Cloud Foundation Design |
| **Document Location: Group Teams Channel / SharePoint / Conestoga Submission older** |

Table of Contents

About This Document 4

Revision History 4

Governance & Approval 4

Reviewed By 4

1. Project Overview 5

2. Requirements 6

3. Constraints & Limitations 7

4. Build Requirements 8

4.1 Build Topology Diagram 8

4.2 Azure Account Structure 8

4.3 vNet and Subnets 9

4.4 vNet Peering Requirements 9

4.5 Private Endpoint / Link Requirements 10

4.6 Azure Firewall Requirements 12

5. Rollout Plan 13

5.1 Project Phases 13

5.2 Project Planning 14

6. Project implementation 15

6.1 Network Table 15

6.2 Infrastructure Table 15

6.3 Create Resource Groups (RG) 16

6.4 Creation of vNets and Subnets (NAIKA + KAN) 16

6.5 Configure vNet Peering for Hub Spoke Architecture within same subscription 18

6.6 Configure vNet Peering across different Subscriptions / AD tenants 20

6.7 Configure routing via Azure Firewall 24

6.8 Build Client Desktops in NAIKA-RG-01 26

6.9 Build Client Desktops in KAN Subscription 28

6.10 Build Apache Web server 29

6.11 Build MySQLDB – Private access 31

6.12 Build MySQLDB – Public access 33

6.13 Configure Private Endpoint / Link 35

6.14 Build FTP Server 38

6.14 Install SSL Certificates on FTP Server: 40

7.Project Testing 41

8. Detailed task list 44

9. Roles & Responsibilities 45

References 48

# About This Document

This document describes the Implementation Strategy for Azure Cloud Foundation Design for adopting Azure services. This covers Hub Spoke architecture setup in detail covering the network requirements and the build of different Azure services.

## Revision History

Reviewers are the group members working on this Capstone project.

| Version | Date | Changes | Author | Reviewer(s) |
| --- | --- | --- | --- | --- |
| 0.1 | 12th March’22 | Initial Release | NG, AK, KG, TTN | NG, AK, KG, TTN |
| 0.2 | 22nd Mrach’22 | Document updated with Project Implementation Tasks | NG, AK, KG, TTN | NG, AK, KG, TTN |
| 1.0 | 26th March’22 | Final Release | NG, AK, KG, TTN | NG, AK, KG, TTN |

## Governance & Approval

Active Business Owners should approve all documents. Client must approve each section independently.

| Approver | Title | Version | Date Approved |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reviewed By

| Role | Name |
| --- | --- |
| Project Approver |  |
|  |  |

# Project Overview

This project is based on providing a framework for cloud adoption and future migrations.

Company NAIKA is currently running all their IT infrastructure from their on-premises data center. The future strategy is to migrate all IT services to cloud and follow **“Cloud Only”** principles. To achieve this, Microsoft Azure is the strategic cloud platform that has been selected for now.

Our solution provides core foundation technologies, architecture, and design guidelines that will enable business functions in NAIKA to build and operate in cloud with adequate guardrails around security, access, network, and governance. Company NAIKA has a partner company KAN and the resources across two companies need to be made available over the private network. KAN already has service running in Azure, so cross tenant integration is sufficient to achieve the inter-connectivity.

The document covers the design of the solution architecture that show how cloud adopters could work with data in a secure and supportive environment that provided by Azure. This document is focusing on the implementation phase, including the integration between company NAIKA and the partner KAN over the private network across two domains. We are using VNet peering method and the Private Link connection for this project.

The impact of the network and the conducting of various test cases is showing in this document. We are trying to integrate with two Azure accounts as the NAIKA and KAN, let them transfer data and communicate with each other.

# Requirements

Below list of requirements were captured as part of initial proposal. The design and build are done to meet all the requirements. There were couple of limitations owing to trial subscription used in Azure, those have been outlined in in Section 4 of this document.

**Requirements Table:**

|  |  |
| --- | --- |
| Ref | Business / Technical Requirements |
| REQ01 | Allow business functions to deliver and consume Azure services |
| REQ02 | Enable business platforms to be early adopters of Azure |
| REQ03 | Resource Group level segregation between apps for easy costing and billing |
| REQ04 | Use this foundation design for cloud adoption and to migrate services from On-Premises in future |
| REQ05 | Integration between two different Azure tenants |
| REQ06 | Host shared and central management services in Hub/Foundation subscription |
| REQ07 | Peer Virtual networks for different business functions |
| REQ08 | Isolate applications per business functions and their environments |
| REQ09 | Span subnets across Azure datacenters / zones |
| REQ10 | Secured environments by using cloud native encryption technology |
| REQ11 | Enable network connectivity from Business/Platform function to Hub / Foundation Services |
| REQ12 | Use Azure tenant default AD for any type of authentication requirement |
| REQ13 | Enable connectivity to partner Azure tenant over private network |
| REQ14 | Use of IaaS, PaaS and Marketplaces images in Azure for deployments as much as possible |
| REQ15 | Secure the traffic flow through Hub subscription via Azure Firewall |

# Constraints & Limitations

1. The trial subscription has many limitations, this has led to few changes in the account architecture. They are summarised as below:

* In trial subscription, the VM resources with small VM sizes are only available in South Central US region. Hence, Account 1 resources had to be moved from EAST US to South Central US. East US was the preferred region due to low cost, however no D and E series VM are not available in this region.
* With trial account, only single default subscription is available. Additional subscriptions cannot be created which is only possible with paid accounts. Hence, for this project two separate accounts had to be created which was not the initial idea.
* Microsoft limits the free accounts that can be created with same phone number and credit/debit card details. So, none of the group members were able to create second account. We then figured out to use loaned credit card for second account creation.

1. When setting up the FTP Server, we must enable the FTP inbound rule inside virtual machine with the specific port and port range that matches with the inbound rule from Azure to allow FTP connection.
2. Global peering cannot be established since the project was limited by the trial subscription. All resources had to be built in a single region. Hence, the Global peering has changed to VNet peering across subscriptions.
3. Due to trial subscription, we have used HDD for any type of storage, have not enabled any type of diagnostics and logging, monitoring, and backup. Encryption is disabled at VM level and front-end app is running on Port 80.
4. To disable public IPs from all servers, the solution requires a Bastion host or an RDP/Citrix access. Bastion host cost is very high and cannot be implemented in trial subscription. Creating an RDP or Citrix access farm requires a separate solution architecture. Hence, we have one of the client workstations in NAIKA-RG-01 to connect to different services via different client tools.
5. Since this is a lab environment, there is no additional email server. Hence, all configurations have been done with a single account in each subscription (which is also the Account owner and Global administrator by default). We can add additional users in Azure AD, however, cannot associate email accounts with them.
6. If a vNet has any existing delegation via private access it cannot be used to create a private endpoint.
7. If the PaaS resource is already created with public or private access it cannot be enabled for private endpoint, a new resource will have to be created.

# Build Requirements

This section provides the reader with an overview of pre-requisites to start with build activities. It is critical to have a clear view of the dependencies in terms of account structure, network architecture and requirements associated with each service to be deployed.

## Build Topology Diagram

Below topology reflects the actual implementation diagram. This is the basis for the entire implementation done to achieve the objective.

**Graphical user interface, application

Description automatically generated**

## 4.2 Azure Account Structure

For this project, two azure accounts have been created to provide two separate subscriptions with their respective AD tenants.

**Azure accounts:**

* [CapstoneAcc2@outlook.com](mailto:CapstoneAcc2@outlook.com)
  + Domain Name: Capstoneacc2outlook.onmicrosoft.com
* [Syst8200Proj@outlook.com](mailto:Syst8200Proj@outlook.com)
  + Domain Name: Syst8200Projoutlook.onmicrosoft.com

## 4.3 vNet and Subnets

Project’s vNet structure involves the implementation multiple virtual networks in two Microsoft Azure subscriptions. One subscription serves company NAIKA and utilizes three different virtual networks within three distinct resource groups. The other subscription serves company KAN, and this organization has two different virtual networks within two distinct resource groups. ***These have been created with non overlapping IPv4 addressing spaces so that vNet Peering and Private Link connectivity can be successfully established and demonstrated.*** These virtual network configurations will utilize cross region connection capabilities within the Azure environment, while ensuring effective security for the different systems in each of the infrastructure environments.

The below table shows the vNet and subnet address space for the project.

|  |  |  |  |
| --- | --- | --- | --- |
| vNet Name | vNet IP Address Space | Subnet Name | Subnet IP Address Space |
| NAIKA-HubVNET-01 | 10.0.0.0/16 | default | 10.0.0.0/24 |
| NAIKA-AppVNET-02 | 10.1.0.0/16 | default | 10.1.0.0/24 |
| NAIKA-AppVNET-03 | 10.2.0.0/16 | default | 10.2.0.0/24 |
| KAN-AppVNET-01 | 10.3.0.0/16 | default | 10.3.0.0/24 |
| KAN-AppVNET-02 | 10.4.0.0/16 | default | 10.4.0.0/24 |

## 4.4 vNet Peering Requirements

1. For vNet peering to work, virtual networks must be created with Resource manager
2. The foundation high level architecture must be in place to establish network connectivity and peering
3. Cross-subscription vNet Peering requires authorization across the AD tenants. Network Contributor Role are required to the accounts across the AD tenants with which peering is established.
4. Resource ID for vNets across the subscription must be handy to peering to be setup.
5. Peering across AD tenants is unidirectional, so must be set from both accounts to be peered if bi-directional connectivity is required.

**vNet Peering in same subscription in same AD tenant**

Azure Account details:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **vNET Name** | NAIKA-HubVNET-01 | NAIKA-AppVNET-02 | NAIKA-AppVNET-03 |
| **Address space** | 10.0.0.0/16 | 10.1.0.0/16 | 10.2.0.0/16 |
| **Subnet name** | Default | Default | Default |
| **Subnet address range** | 10.0.0.0/24 | 10.1.0.0/24 | 10.2.0.0/24 |
| **Subscription** | Azure Subscription 1 | Azure Subscription 1 | Azure Subscription 1 |
| **Resource group** | NAIKA-RG-01 | NAIKA-RG-02 | NAIKA-RG-03 |
| **Location** | Central US | Central US | Central US |

**Peering will be setup as:**

* ***HubVNET-01 - peered to – AppVNET-02***
* ***HubVNET-01 - peered to – AppVNET-03***
* ***AppVNET-02- peered to – HubVNET-01***
* ***AppVNET-03- peered to – HubVNET-01***

**vNet Peering between different subscriptions and AD tenants:**

Azure Account details:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **vNET Name** | NAIKA-HubVNET-01 | KAN-AppVNET-01 |
| **Address space** | 10.0.0.0/16 | 10.3.0.0/16 |
| **Subnet name** | Default | Default |
| **Subnet address range** | 10.0.0.0/24 | 10.3.0.0/24 |
| **Subscription** | Azure Subscription 1 | Azure Subscription 1 |
| **Resource group** | NAIKA-RG-01 | KAN-RG-01 |
| **Location** | Central US | Central US |

**Peering will be setup as:**

* ***NAIKA-HubVNET-01 - peered to – KAN-AppVNET-01***
* ***KAN-AppVNET-01 - peered to - NAIKA-HubVNET-01***

## 4.5 Private Endpoint / Link Requirements

Private Link is used to integrate Azure services into your private virtual network by mapping it to a private endpoint. No gateways, NAT devices, ExpressRoute or VPN connections, or public IP addresses are required because all traffic to the service is routed through the private endpoint. Traffic is kept on the Microsoft global network by using Private Link.

For our project, we are demonstrating use of private link with two different AD tenants to map an Azure PaaS service to a server in virtual network.

A picture containing text, porcelain

Description automatically generated

**Microsoft Image of Private Link (** (Private Link)

(Private Link documentation), (Private Link)

**Prerequisites of Private Link and Private Endpoint:**

1. An Azure virtual network (vNet) to deploy private endpoint and link.
2. Disable public network access on Azure resources accessed over private link
3. Private Endpoint mapping to one of the accounts of the PaaS Resource to be connected to
4. Automatic DNS registration so that the account accessed via Private link gets the IP from the vNet where endpoint is created.

*PS: This will require additional considerations where organizations are managing their own DNS servers.*

1. Network connectionsare unidirectional and can only be initiated by clients that are connecting to private endpoint.
2. Private endpoint is required for private link and should be deployed in the same subscription and network where the PaaS resource is. Private link can be deployed in the global or regionally peered network.
3. The subscription where private-link resource is created must also be registered with the Microsoft network resource provider.

**SYST8200PROJ Account – Registered Resource Providers**

Graphical user interface, application, email

Description automatically generated

**CAPSTONEACC2 Account - – Registered Resource Providers**

Graphical user interface, text, application, email

Description automatically generated

## 4.6 Azure Firewall Requirements

To successfully implement the Azure Firewall managed service into the cloud environment, there are a few prerequisites that must be configured and implemented.

1. The first requirement that will be needed to build the firewall within Azure is to have an active Azure subscription which will allow for the creation of resources in the public cloud environment.
   1. This will give access required to build all the other necessary components. The next component that is vital to the firewall creation is the configuration of a virtual network (vNet).
2. Within this virtual network, further subnetworks must be created. These networks must have a minimum of /26 as the CIDR value, and the name of the subnet designated for the firewall must follow the exact naming convention.
   1. The naming convention is as follows, “AzureFirewallSubnet” with the value being case sensitive.
3. Once this prerequisite has been successfully configured, the firewall creation can begin. The Azure Firewall can be configured to operate as the Standard or Premium edition. Both provide high level security and enhanced features to the management of the network traffic.
4. A new public IP address resource should be created as this will be used for the Firewall.

# Rollout Plan

The rollout plan shows the basic lifecycle of the project that we did to complete the task and deployed it on given time.

## 5.1 Project Phases

The entire project has been divided into multiple phases to segregate tasks and deliverables. Below table outlines different project phases to achieve the end state.

|  |  |
| --- | --- |
| Project Phase | Project Phase Details |
| Phase 1 | Ideation Phase  Project Proposal  High Level Requirements |
| Phase 2 | Identifying technical limitation of project proposal  Re-submission of Project Proposal with different idea |
| Phase 3 | Submission of Solution Template |
| Phase 4 | Building Infrastructure |
| Phase 5 | Creating Azure Accounts, Resource Groups, Virtual Network, and Subnets to match Hub Spoke architecture  Setup second azure account to setup cross-subscription vNet peering |
| Phase 6 | Creating bidirectional vNet peering between Hub and Spokes in first Azure subscription |
| Phase 7 | Building and configuring servers, databases in each Resource group for access and connectivity in first subscription |
| Phase 8 | Creating bidirectional vNet peering across two subscriptions / AD tenants |
| Phase 9 | Building client desktop machines across two subscriptions for setting up private access and connectivity |
| Phase 10 | Creating and configuring Firewall to secure connectivity between two different subscriptions |
| Phase 11 | Creating private link between two subscriptions to connect to an Azure PaaS service using a private endpoint and private link. |
|  |  |

## 5.2 Project Planning

**Azure Cloud Foundation Design (ACF) –** The objective is to implement a repeatable and re-usable standard ACF for NAIKA.

The architecture in the solution used **Hub-Spoke Topology** along with the shared services for managing the connectivity across the network which addresses the concerns like isolation, cost effectiveness, billing, perimeter security etc. It also includes the **multiple vNets** which represent the different functions. Multiple vNets in a same subscription model can be developed for small businesses to run production and non-production workloads on various platforms. vNet peering will provide low latency, high throughput network with no requirement of any internet or VPN gateways. It’s a direct secure private connectivity.

We have implemented the virtual network peering which help to overcome the issues related to subscription limitation since large workloads will consume more resources and Azure subscriptions have limits and quotas. Different Business functions can adopt one of the spoke segments to run their applications.

The planning involved adopting a **phased approach** wherein multiple tasks were created and distributed amongst the group members. The phased planning was done after finalizing a base topology architecture for deployment.

We have hit multiple roadblocks during the deployment due to trial accounts that were used for the build. With continued focus and diligence, we were able to figure out workarounds and troubleshoot technical issues.

As a result, we have been able to deliver a solution that meets the requirements.

# Project implementation

The project implementation has been done following the below **Network and Infrastructure table.** This section will provide readers with a view of sequence of steps to follow to create this foundation design. This has been done manually for the project, however in production environments ARM templates can be used to automate all builds and deploy all shared components with a single script. There are various sub-sections with detailed build steps for each Azure service.

## 6.1 Network Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource Group | vNet Name | IP Address Space | Subnet Name | Subnet IP Address Space |
| AZURE SUBSCRIPTION 1 | | | | |
| NAIKA-RG-01 | NAIKA-HubVNET-01 | 10.0.0.0/16 | default | 10.0.0.0/24 |
| NAIKA-RG-02 | NAIKA-AppVNET-02 | 10.1.0.0/16 | default | 10.1.0.0/24 |
| NAIKA-RG-03 | NAIKA-AppVNET-03 | 10.2.0.0/16 | default | 10.2.0.0/24 |
| AZURE SUBSCRIPTION 2 | | | | |
| KAN-RG-01 | KAN-AppVNET-01 | 10.3.0.0/16 | default | 10.3.0.0/24 |
| KAN-RG-02 | KAN-AppVNET-02 | 10.4.0.0/16 | default | 10.4.0.0/24 |

## 6.2 Infrastructure Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource Group | Server Role | Server Name | IP Address | Operating System | Application Software |
| NAIKA-RG-03 | FTP Server (Windows) | NAIKA-WFTPS-01 | NAIKA-WFTPS-01-ip | Windows server 2019 Std. Edition | FTP |
| NAIKA-RG-02 | XAMPP WEB/APP Server | NAIKA-LWEB-01 |  | Windows server 2019 Std. Edition | XAMPP Stack |
| NAIKA-RG-02 | MySQL Server | NAIKA-PMYSQL-01 |  | Windows server 2019 Std. Edition | MySQL Server 8 |
| KAN-RG-01 | Windows Client / User machine | KAN-ClientDesktop-01 |  | Windows 10 Pro | MySQL Workbench, Firefox, FileZilla, Putty |
|  | Windows Client / User machine | KAN-ClientDesktop-02 |  | Windows 10 Pro | Firefox |
| NAIKA-RG-01 | HAProxy Server | NAIKA-LPROXY-01 |  | CentOS 7 | HAProxy |
| NAIKA-RG-01 | Azure Firewall | NAIKA-AZFW-01 |  | NA | NA |
| NAIKA-RG-03 | Internal Load Balancer |  |  |  |  |
| NAIKA-RG-01 | Client Desktop | NAIKA-TestVM1 |  | Windows 10 Pro | MySQL Workbench, FileZilla, Firefox, Putty |
|  |  |  |  |  |  |

## 6.3 Create Resource Groups (RG)

The first step to create any resources in Azure is creating Resource Group. To create an RG, follow the below steps:

1. Login to Azure Portal with the respective Account ID. Go to Resource Groups -> Click Create
2. Add the below projects details while creating RG:

**Subscription**: Azure Subscription 1

**Resource Group Name**: NAIKA-RG-01

**Region**: Central US

1. Click “Review + Create”
2. Create other RGs for the project in the same way for First and Second Account

* List of RGs created for the project:
* NAIKA-RG-01
* NAIKA-RG-02
* NAIKA-RG-03
* KAN-RG-01
* KAN-RG-02

## 6.4 Creation of vNets and Subnets (NAIKA + KAN)

The next step is to create vNet to run IaaS resources. Each vNet creates a default subnet. For our project we have used /24 default subnet with /16 default vNet address space.

1. Login to Azure Portal with the respective Account ID. Go to Virtual Networks -> Click Create
2. Add the below projects details while creating vNet:

**Subscription**: Azure Subscription 1

**Resource Group Name**: NAIKA-RG-01

**Name:** NAIKA-HubVNET-01

**Region**: Central US

1. Click Next to configure IP address and subnet:

Leave default IPv4 address space: 10.0.0.0/16

Subnet name: default

Subnet address range: 10.0.0.0/24

1. Click next to Security -> Leave all Security options as default -> Click “Review + Create”
2. Follow the same steps to create other two vNets for first account. Details for vNet are as below:

**Resource Group Name**: NAIKA-RG-02

**Name:** NAIKA-AppVNET-02

Leave default IPv4 address space: 10.1.0.0/16

Subnet name: default

Subnet address range: 10.1.0.0/24

**Resource Group Name**: NAIKA-RG-03

**Name:** NAIKA-AppVNET-03

Leave default IPv4 address space: 10.2.0.0/16

Subnet name: default

Subnet address range: 10.2.0.0/24

1. Login to Azure Portal with the SYST8200Proj Account ID. Go to Virtual Networks -> Click Create
2. Add the below projects details while creating vNet:

**Subscription**: Azure Subscription 1

**Resource Group Name**: KAN-RG-01

**Name:** KAN-AppVNET-01

**Region**: Central US

1. Click Next to configure IP address and subnet:

Leave default IPv4 address space: 10.0.0.0/16

Subnet name: default

Subnet address range: 10.3.0.0/24

1. Click next to Security -> Leave all Security options as default -> Click “Review + Create”
2. Repeat steps 7-10 to create the second vNet located in the KAN domain with the following properties

**Resource Group Name**: KAN-RG-03

**Name:** KAN-AppVNET-02

Leave default IPv4 address space: 10.2.0.0/16

Subnet name: default

Subnet address range: 10.4.0.0/24

## 6.5 Configure vNet Peering for Hub Spoke Architecture within same subscription

To configure vNet peering within the same subscription, below steps have been followed.

The same steps can be followed for setting up vNet peering between different virtual networks in any Azure subscription.

In peered networks, the resources communicated with each other with the same latency as they were in same vLAN / virtual network. This solution is helpful to connect latency sensitive applications across or within subscriptions.

**Create vNet Peering Hub vNet and AppVNET-02**

1. Login to Azure Portal with the account: [Capstoneacc2@outlook.com](mailto:Capstoneacc2@outlook.com) -> Go to Hub vNet: NAIKA-HubVNET-01 -> Click Peerings
2. In the Add Peering window, add the below details:

* Peering link name: **vNetPeer-Hub-App02**
* Traffic to remote virtual network: **Allow**
* Traffic forwarded from remote virtual network: **Allow**
* Virtual network gateway or Route Server: **None**
* Remote Peering Link Name: **vNetPeer-App02-hub**
* Virtual Network Deployment Mode: **Resource Manager**
* Remote Virtual Network: **NAIKA-AppVNET-02**
* Traffic to remote virtual network: **Allow**
* Traffic forwarded from remote virtual network: **Allow**
* Virtual network gateway or Route Server: **None**

**Create vNet Peering Hub vNet and AppVNET-03**

Follow the steps as mentioned above for setting up vNet Peering. Below settings must be changed during the setup.

* Peering link name: **vNetPeer-Hub-App03**
* Remote Peering Link Name: **vNetPeer-App03-hub**
* Remote Virtual Network: **NAIKA-AppVNET-03**

***PS: For peering to work, two peering links must be created. By selecting remote virtual network, Azure will create both peering links.***

**Figure 2:** This figure shows “connected” state of vNet peering setup between Hub and two App vNets. This will allow access to App-02 vNet from Hub vNet.

Graphical user interface, application, Word

Description automatically generated

Figure vNet Peering between Hub and App vNets

**Figure 3:** This figure shows “connected” state of vNet peering setup between AppVNET-02 to Hub. This will allow access to Hub vNet from App-02 vNet.

Graphical user interface, application

Description automatically generated

Figure vNet Peering between remote App vNet-02 and Hub

**Figure 4:** This figure shows “connected” state of vNet peering setup between AppVNET-03 to Hub. This will allow access to Hub vNet from App-03 vNet.

Graphical user interface, text, application, email

Description automatically generated

Figure vNet Peering between remote App vNet-03 and Hub

## 6.6 Configure vNet Peering across different Subscriptions / AD tenants

To configure vNet peering across two different AD tenants require authorization to be setup. Hence, a user from other AD must be invited as Guest user

1. Open the Browser incognito window and Login to Azure Portal with the account: [Capstoneacc2@outlook.com](mailto:Capstoneacc2@outlook.com)
2. Open another Browser incognito window and Login to Azure Portal with the account: [Syst8200proj@outlook.com](mailto:Syst8200proj@outlook.com)

Cross-subscription /AD tenant peering requires peering to be setup from each side respectively. As first step, to initiate the peering from ***Capstoneacc2 to*** [**Syst8200Proj**](mailto:SYST8200proj@outlook.com)**,** below steps were performed:

1. Go to Active Directory -> Users -> Invite User -> Provide details of the account which peering will be setup and that can be granted Network Contributor rights, which is [Capstoneacc2@outlook.com](mailto:Capstoneacc2@outlook.com) -> Click Invite.

Graphical user interface, application

Description automatically generated

1. Go to the mailbox of Capstoneacc2 accounts and accept the invite. This will allow “capstoneacc2” user to be visible in SYST8200Proj AD as Guest. When you switch active directory, the target account AD will be visible.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Now go to the Syst8200Project KAN AD to which the peering has to be setup -> Go to Access Control (IAM) -> Add Role Assignment -> Select Network Contributor -> Click Next.
2. Under Members, assign access to Capstoneacc2 account (Guest) -> Click Review + Assign

Graphical user interface, application

Description automatically generated

1. Now connect to CapstoneAcc2, got to NAIKA-HubVNET-01, Under HubVNET-01 settings, click Peerings -> Click Add -> Provide peering name

In the remote network, provide Resource ID for KAN VNET which is to be peered:

KAN-AppVNET-01 resource ID: */subscriptions/45459a97-366d-479e-871b-c82607eda9c0/resourceGroups/KAN-RG-01/providers/Microsoft.Network/virtualNetworks/KAN-AppVNET-01*

Graphical user interface, text, application, email

Description automatically generated

Under Directory, select SYST8200Proj directory for authentication from this account -> Click Add.

Graphical user interface, text, application, Word, email

Description automatically generated

1. The peering will be setup successfully and will show under Peerings in HUBVNET-01.

Graphical user interface, text, application, email

Description automatically generated

1. Follow the same steps from Step 3 to 9 to set up peering from Syst8200Proj to Capstoneacc2 network.

NAIKA-HubVNET-01 Resource ID: */subscriptions/e27b2238-1465-409d-9d2a-824d64998a1f/resourceGroups/NAIKA-RG-01/providers/Microsoft.Network/virtualNetworks/NAIKA-HubVNET-01*

The below screenshots show successful AD configuration and peering established in reverse direction.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Following above steps, bi-directional peering is successfully setup. Connectivity Testing was done using RDP protocol over private network between two subscriptions:

* Disassociate public IP from KAN-ClientDesktop-01 machines
* RDP from TestVM1 in NAIKA domain to KAN-ClientDesktop-01 in KAN domain - Successful

(Create a virtual network peering - Resource Manager, different subscriptions and Azure Active Directory tenants, 2022)

*PS: Network Security Groups (NSGs) have to be added either at subnet or NIC level to allow point to point to connectivity even after peering is connected.*

## 6.7 Configure routing via Azure Firewall

1. Navigate to the Azure portal and search for the Firewall resource. From this section, a new standard Azure firewall will be created. *It is important to note that the region in which this resource resides should be the same as the other resources created in the Azure environment.* The IP address space assigned should also match the respective virtual network that it will reside in.

Graphical user interface, application

Description automatically generated

1. Ensure that a public IP has been created and is assigned to the Azure Firewall service.
2. After the firewall has been created, network rules must be added to filter traffic and deliver connectivity. A rule titled KAN-RDP has been created with a priority of 100 allowing access to TCP port 3389. This rule specifies the source IP address as the private IPv4 address of the Firewall, and the destination is set as the subnet located in the KAN-RG-01 resource group.

Graphical user interface, application

Description automatically generated

1. The next step required for the firewall creation is to navigate to the Route Table section within the Azure resource portal. In this section a new route titled Hub-Route-KAN was created to establish a greater level of security for the NAIKA-HubVNET-01.
2. Once the route table entry has been created, a route must be created within this resource. This route must specify virtual network subnet prefix as well as the next hop type in the route.
   1. The address prefix has been set to the 10.0.0.0/24 subnet and next hop type has been specified as VirtualAppliance to refer to the Azure Firewall.

Graphical user interface, application

Description automatically generated

1. The next step during the Azure Firewall configuration process is to select the subnets category within the route table resource and associate the subnet that will be routed through the firewall. The default subnet of 10.0.0.0/24 located in the virtual network NAIKA-HubVNET-01

Graphical user interface, text, application, email

Description automatically generated

1. Once routing propagation has been completed, Azure Firewall should manage traffic as per outlined configuration. (Deploy and configure Azure Firewall using the Azure portal, 2021)

## 6.8 Build Client Desktops in NAIKA-RG-01

1. Login to Azure Portal with the account: [*Capstoneacc2@outlook.com*](mailto:Capstoneacc2@outlook.com) and navigate to the virtual machine service. Click on ->Create -> Select Azure Virtual Machine

2. Update the Project Details as below for VM configuration:

Subscription: CapstoneAzure

Resource Group: NAIKA-RG-01

Virtual Machine Name: TestVM1

3. Update the Instance details as below for VM configuration:

Virtual machine name: TESTVM1

Region: Central US

Availability Options: None (in Production scenario, one of the available options will be chosen)

Security Type: Standard

Azure Image: Windows 10 Pro

Size: Standard\_DS1 – 1 vCPU, 3.5 GB RAM

4. Update the Administrator account as below for VM configuration:

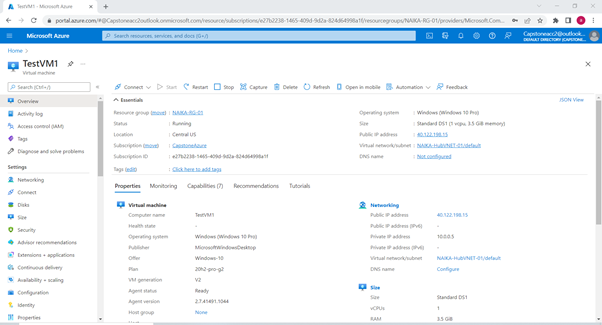
Username: CapstoneUser

Password: Secret55Secret55

Confirm password: Secret55Secret55

5. Options like Encryption, Backup, Boot Diagnostics must be setup based on organization policies. For our lab environment, we have not enabled either of these options.

6. Click on “Review + Create”. Once the validation tests have passed, click on Create.



1. Verify the RDP connection is successful. Install the XAMMP on this machine for launching the web application– (hospital App) connected to the My SQL database.
2. Install MY SQL Workbench to connect the application to the database.
3. To ensure connectivity, make sure that the network security groups have been modified either at the NSG or VM level to allow the services. The image below shows the following inbound rules that have been configured.

Table

Description automatically generated

## 6.9 Build Client Desktops in KAN Subscription

1. Navigate to the Azure Portal and click the “create” option under the virtual machine resource.
2. Select the region as US Central as this is where all other resources are created in.
3. Set Windows 10 Pro as the image for the deployment of the client virtual machine.
4. Establish a username and password that will be used for the connection to the machine. Three client desktops have been created between the two environments for our solution, the VM properties are listed below.
   1. Properties for the Client Desktop located in KAN-AppVNET-01: Username – Kg3477, Password – Secret55Secret55
   2. Properties for the Client Desktop located in KAN-AppVNET-02: Username – KANUSER02, Password – Secret55Secret55
5. Ensure that the RDP connection is selected, and port is open so that connection to the machine can be established
   1. Network security group configuration should be configured at the NSG or VM level with the following properties to allow connectivity (Both Client Desktops will have this configuration).

Graphical user interface, text, application

Description automatically generated

1. Navigate through the different options associated with the virtual machine creation. When on the Disks selection criteria, change the OS Disk type to the Standard HDD.
2. On the Networking section, make sure that the correct VirtualNetwork is selected based on the VM’s associated domain AppVNET.
   1. Example – NAIKA-TestVM1 will require the NAIKA-HubVNET-01.
   2. Example – KAN-ClientDesktop-01 will require KAN-AppVNET-01
3. Leave the other options on the default setting and select the “review and create” option for the virtual machine in development, if all settings look good, deploy VM.
4. Once VM has been deployed, install the respective application required in solution to deliver functionality to the respective client desktop.
5. Web server functionality will be utilized through the use of XAMPP web application tools, therefore the inbound rule for HTTP service must be added on the VM network connections

Graphical user interface, application, email

Description automatically generated

**Client Desktop Chart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Client Desktop Name | vNet Name | IP Address Space | Subnet Name | Subnet IP Address Space | Applications Installed |
| NAIKA-TestVM1 | NAIKA-HubVNET-01 | 10.0.0.0/16 | default | 10.0.0.0/24 | MySQL Workbench, Firefox, Putty, FileZilla |
| KAN-ClientDesktop-01 | NAIKA-AppVNET-02 | 10.1.0.0/16 | default | 10.3.0.0/24 | MySQL Workbench, Firefox, Putty, FileZilla |
| KAN-ClientDesktop-02 | NAIKA-AppVNET-03 | 10.2.0.0/16 | default | 10.4.0.0/24 | Firefox |

## 6.10 Build Apache Web server

1. Create a Windows 2019 virtual server following the steps for VM creation already outlined in Section 6.8
2. The server name is: **NAIKA-LWEB-01**
3. Install XAMPP stack on this server, this is downloaded from internet and installed directly. Once successfully installed, follow the below steps for further configuration. The application code has been taken from one of the other course’s assignments. This will help us to demonstrate the web access to a running application.
4. Open the XAMMP server and start the Apache and My SQL server service.

Graphical user interface

Description automatically generated

1. Navigate to the C Drive and locate the XAMMP folder. Copy the Hospital application folder in the htdocs folder.
2. Edit the dbfuns and pxcore file under the libs folder of hospital app: replace the hostname name with the fully qualified sever name: naika-pmsql-02.mysql.database.azure.com, username: NAIKAMYSQLDB02 password: Secret55 for rk\_hospital database. (The detailed steps for DB build are called out in Section 6.11)
3. Navigate to the Browser and do local host: <http://localhost/hospital> t*o* access the webapp.

Graphical user interface, application

Description automatically generated

## 6.11 Build MySQLDB – Private access

This PaaS DB has been created as back-end database for the web front server created under Step 6.10

1. Navigate to the Azure Portal and search the Azure Database for SQL Server service.
2. Click on the “create” option to create the Azure Database for SQL Server.
3. Deploy the server using the flexible option and then click on create.
4. Update the Project Details as below for Azure Database configuration:

Subscription: CapstoneAzure

Resource group: NAIKA-RG-02

1. Update the Server details as below for Azure Database configuration:

Server name: Central US

MySQL version: naika-pmsql-02

Workload type: For development or hobby projects Central US

Compute +storage: Burstable, B1ms - vCores, 2 GiB RAM, 20 GiB storage, 360 IOPS

1. Update the Server details as below for Azure Database:

Enable high availability: Do not enable the high availability

1. Update the Administrator account as below for Azure Database

Admin username: NAIKAMYSQLDB02

Password: Secret55

Confirm password: Secret55

8. On networking section select the networking connectivity method as private access (VNet integration)

9. Click on “Review + Create”. Once the validation tests have passed, click on Create.

Graphical user interface, text, application, email

Description automatically generated

10: Verify that the Private DNS Zone is created: naika-pmsql-02.private.mysql.databse.azure.com

Graphical user interface, text, application, table, email

Description automatically generated

Establish Database connection:

1. Navigate to the Virtual machine- TESTVM1, connect to the TESTVM1 using the RDP with username as CapstoneUser and Password as Secret55Secret55.

2. Open My SQL Workbench server. Click on “+” icon to establish the new connection.

3. Enter the following details for establishing the connection to the database:

1. Connection Name: AzureDB2
2. Connection Method: Standard (TCP/IP)
3. Hostname: naika-pmsql-02.mysql.database.azure.com
4. Post: 3306 to connect to the database
5. Username: NAIKAMYSQLDB02
6. Password: Secret55

4. Click on Test Connection to test the connection.

5. Once the Connection is established, open it, and click on the database icon, create the new schema: rk\_hospital.

6. Navigate to the “server” tab and select the Data Import option form the dropdown.

7. Import the rk\_hospital.sql file from the “Import from Self-Contained File” and click on “Import” to import the database successfully.

Graphical user interface, text, application

Description automatically generated

8: Fire the Query on the doctors table and verify the data is populating successfully

Graphical user interface, text, application

Description automatically generated

## 6.12 Build MySQLDB – Public access

This second database has been created to demonstrate PaaS db with public access which can be accessed by a virtual server in a network by permitting the IP of the server in network settings of the database.

1. Navigate to the Azure Portal and search the Azure Database for SQL Server service.

2. Click on the “create” option to create the Azure Database for SQL Server.

3. Deploy the server using the flexible option and then click on create.

4. Update the Project Details as below for Azure Database configuration:

Subscription: CapstoneAzure

Resource group: NAIKA-RG-01

5. Update the Server details as below for Azure Database configuration:

Server name: Central US

MySQL version: naika-pmsql-01

Workload type: For development or hobby projects Central US

Compute +storage: Burstable, B1ms

vCores, 2 GiB RAM, 20 GiB storage, 360 IOPS

6. Update the Server details as below for Azure Database:

Enable high availability: Do not enable the high availability

7. Update the Administrator account as below for Azure Database

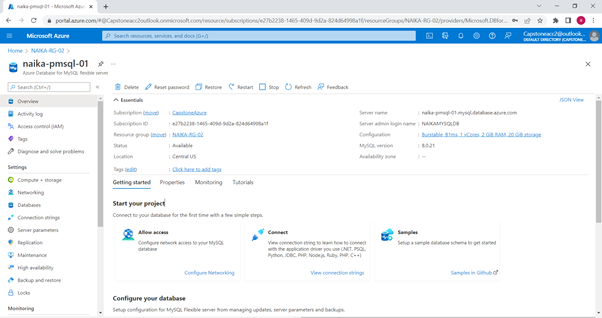
Admin username: NAIKAMYSQLDB

Password: Secret55

Confirm password: Secret55

8. On networking section select the networking connectivity method as public access.

9. Click on “Review + Create”. Once the validation tests have passed, click on Create.



Establish Database connection:

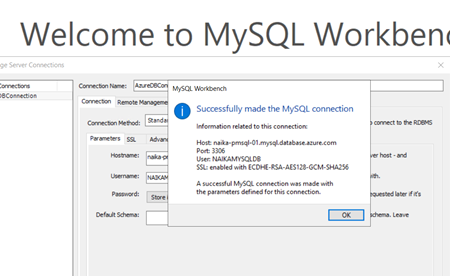
1. Navigate to the Virtual machine- TESTVM1, connect to the TESTVM1 using the RDP with username as CapstoneUser and Password as Secret55Secret55.

2. Open My SQL Workbench server. Click on “+” icon to establish the new connection.

3. Enter the following details for establishing the connection to the database:

1. Connection Name: AzureDBConnection
2. Connection Method: Standard (TCP/IP)
3. Hostname: naika-pmsql-02.mysql.database.azure.com
4. Post: 3306
5. Username: NAIKAMYSQLDB
6. Password: Secret55

4. Click on Test Connection to test the connection.

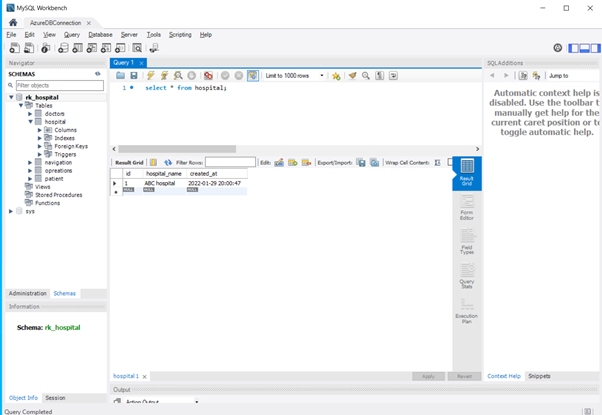


5. Once the Connection is established, open it and click on the database icon, create the new schema: rk\_hospital.

6. Navigate to the “server” tab and select the Data Import option form the dropdown.

7. Import the rk\_hospital.sql file from the “Import from Self-Contained File” and click on “Import” to import the database successfully.

8: Fire the Query on the doctors table and verify the data is populating successfully.



## 6.13 Configure Private Endpoint / Link

The diagram below is taken from one of the Microsoft YouTube videos on how to connect a storage account via Private link. We will be demonstrating the same in our project. (Tutorial: Connect to a storage account using an Azure Private Endpoint, 2021)

https://www.youtube.com/watch?v=AZ0iFcyPDkc

Graphical user interface

Description automatically generated

Private Link Image (https://docs.microsoft.com/en-us/azure/private-link/tutorial-private-endpoint-storage-portal)

**Create storage account with a private endpoint**

1. Go to Azure Portal, login with [capstoneAcc2@outlook.com](mailto:capstoneAcc2@outlook.com) -> Search Storage Accounts -> Click Storage accounts -> Click Create to create a new storage account

Use the following projects details for setting up storage account:

**Subscription**: Azure Subscription 1

**Resource Group Name**: NAIKA-RG-01

**Region**: Central US

**Storage Account Name:** naikastr01

**Performance:** Standard

**Redundancy: LRS**

1. Leave Security as Default for Blob access, click Networking. Under **Connectivity method** select **Private endpoint**.

Graphical user interface, application

Description automatically generated

Click “Add Private Endpoint” and update the following details:

**Private endpoint name: StoragePrivateEndpoint**

**Storage sub-resource: Blob**

**Virtual Network: NAIKA-AppVNET-03**

**Subnet: 10.2.0.0/24**

Graphical user interface, text, application, email

Description automatically generated

This will create a private mapping in DNS with a private IP. This will block access from Internet and only access from the selected network is allowed.

Click OK -> Click Review + Create

1. Go to the storage account: naikastr01 -> Click Settings -> Access Keys -> Copy the connection string for Key 1. This will be used for connecting to Storage account.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application, email

Description automatically generated

**Test connectivity to private endpoint**

We will use the server created in KAN network – KAN-ClientDesktop-02 to connect to this storage account using private link.

Do nslookup to the private DNS name for the storage container created, it returns the private IP address. This is the reason VM will be able to connect to storage account.

## 6.14 Build FTP Server

1. Log into the account on Azure portal. Click on NAIKA-RG-03 and create new virtual machine (NAIKA-WFTPS-01).
2. Connect to the VM using RDP and download the RDP file.
3. In Windows Server Manager go to Dashboard and click “Add Roles and Features”.
4. In "Add Roles and Features" wizard select the "Proceed to Installation" option and confirm “Role-based or feature-based installation”, and then click “Next”.
5. Click the “Web Server (IIS)” role. Install the FTP Server with FTP Extensibility and Service, Management Tools: IIS Management Console.
6. Open IIS Manager and create a new FTP Site. The name of the FTP Site: NaikaFTP; physical path: C:\Data.
7. In Binding and SSL Setting: IP Address: Choose All Unassigned; Port: by default, it’s 21; SSL: Click No SSL
8. In the Authentication and Authorization Information: Authentication: Basic; Authorization: Select all users; Permission: select read and write.
9. After successfully deployed the FTP site. Click the “NAIKA-FTP” site, and then Open “FTP Firewall Support”.
10. In the External IP Address of Firewall Enter your Azure Server Public IP Address. Click “Apply”.
11. For configuring the “Data Channel Port Range”, select the NAIKA-WFTPS-01 and select “FTP Firewall Support”. In the Data Channel Port Range, enter “3000 to 3005”, and Click “Apply”, in this section we don’t need to enter the External IP Address of Firewall.

A screenshot of a computer

Description automatically generated

1. After successful, we must restart the “FTP Service”. Go the Services and selects Microsoft FTP Services. Right click and restart.
2. Configure Inbound Port Rule. On Azure portal, open the VM and click on Networking, then choose “Add Inbound Port Rule”. In add inbound security rule, change Basic to Advance.
3. The Inbound Security Rule: Click Add

* Service: FTP
* Port Ranges: 3000-3005 (Data Port Range)
* Priority: It will automatically be assigned
* Name: FTP-Ports

Graphical user interface, text, application, email

Description automatically generated

1. Install FileZilla to test the FTP Server. **Run as administrator.** In the Host, enter our Azure Virtual machine Public IP Address, Username our virtual machine Username and the Password, click “Quick connect”. The pop-up message will display, so click OK to continue. Now the Azure FTP Server is connected successful.

A screenshot of a computer

Description automatically generated

(https://getanadmin.com/azure/configure-ftp-server-on-azure-virtual-machine/, 2020)

## 6.14 Install SSL Certificates on FTP Server:

1. Navigate to the Azure Portal and open the FTP Server machine that we created.
2. Make sure the IIS Manager is set up properly and the FTP Site is installed.
3. Click on the NAIKA-WFTPT1 and choose the Server Certificate.
4. Click on create Self-Signed Certificate.
5. Specify the name for the Certificate and click OK.

Graphical user interface, application

Description automatically generated

# 7.Project Testing

**Use Case 1 - Connectivity Testing between peered networks in the same subscriptions**

Tested connectivity between the Client Desktop located in the NAIKA-RG-01 resource group and the FTP server located in the NAIKA-RG-03 resource group. A connection from FileZilla to the FTP server was established to demonstrate successful integration of the peering across virtual networks.

**Use Case 2 – Tested cross connectivity between Azure vNets located in the** [**CapstoneAcc2@outlook.com**](mailto:CapstoneAcc2@outlook.com) **subscription**

Established connection between virtual networks using application tools such as Putty, FileZilla, MySQL Workbench.

**Use Case 3 - Connectivity Testing between peered networks across different AD tenants with active Firewall**

**Graphical user interface

Description automatically generated**

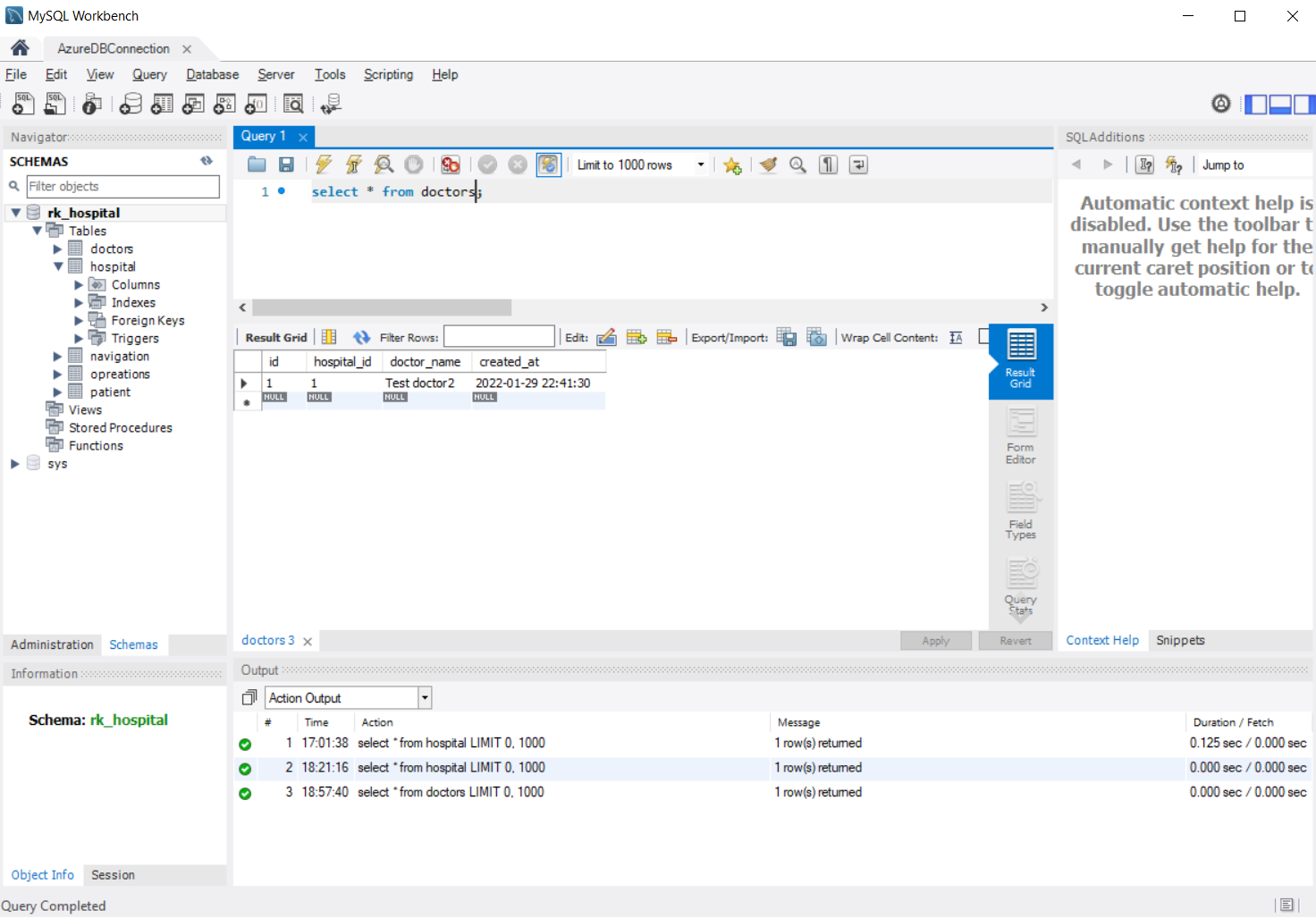
**Use Case 4** **– Build XAMPP app with MySQL DB (public access). Access the database using MySQL Workbench tool running on one of the client machines**

1. Connect to the TESTVM1 machine in NAIKA-RG-02 resource group through the RDP connection.

Once the connection successfully completed, launch the XAMMP and start the Apache and MySQL service.

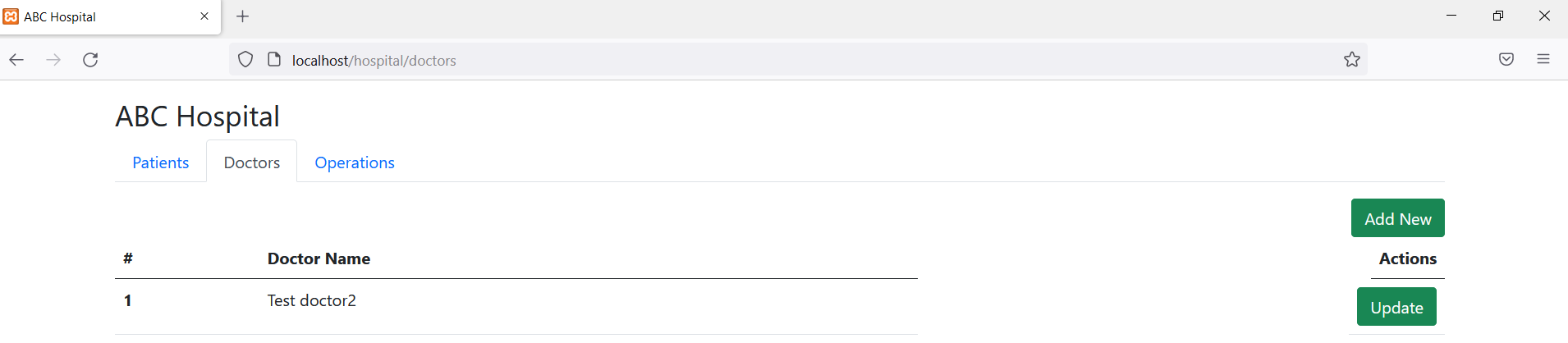
1. Launch the MySQL Workbench and select the AzureDBConnection which has public access.

3. Select the rk\_hospital database from the left pane and in the query section, fire the query on the hospital table. Query: Select\* from hospital.



6. Once the database connection is tested successfully, launch the browser.

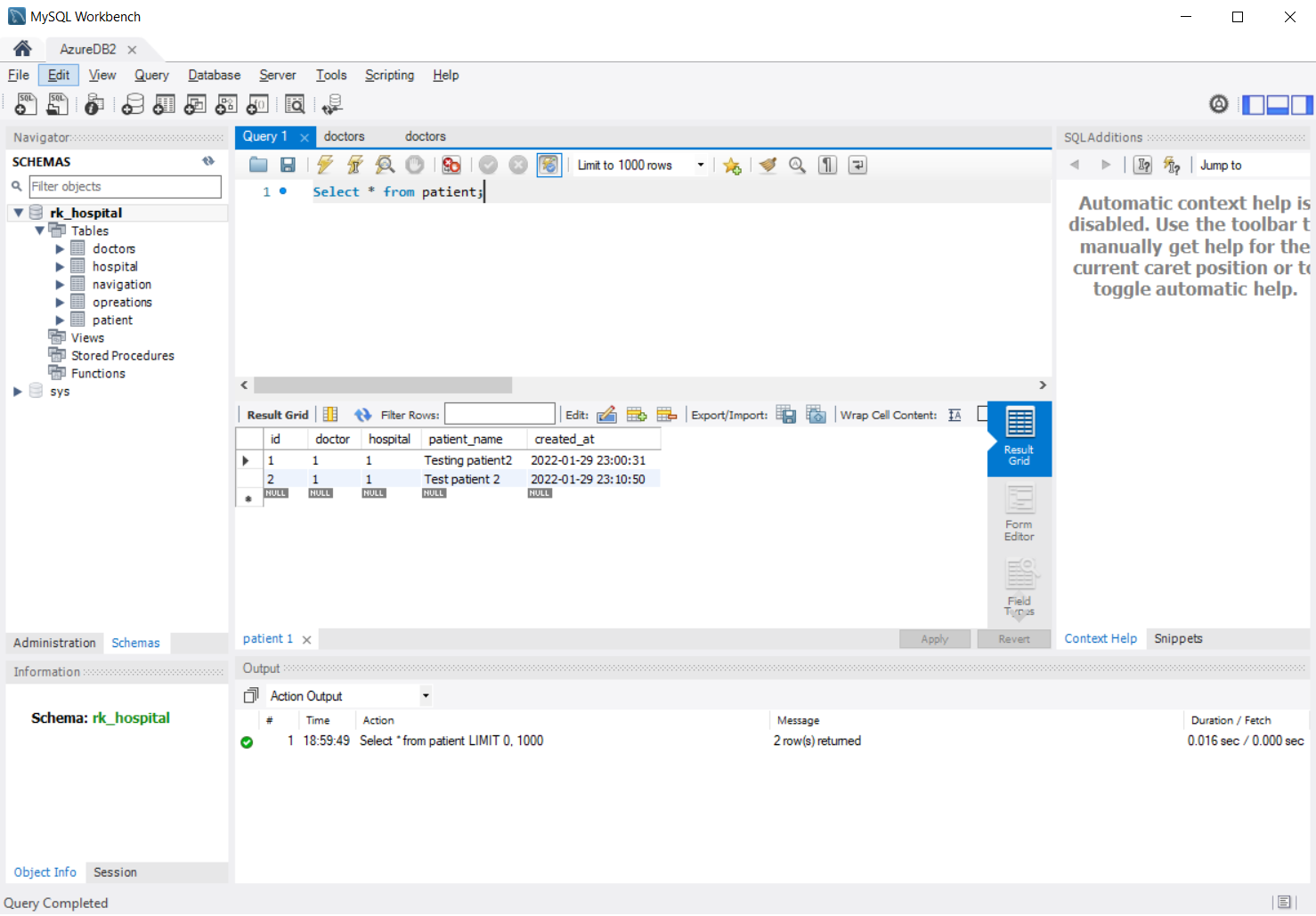
7. Do localhost: <http://localhost/hospital>



**Use Case 5** **– Build XAMPP app with MySQL DB (private access). Access the database using MySQL Workbench tool running on one of the client machines**

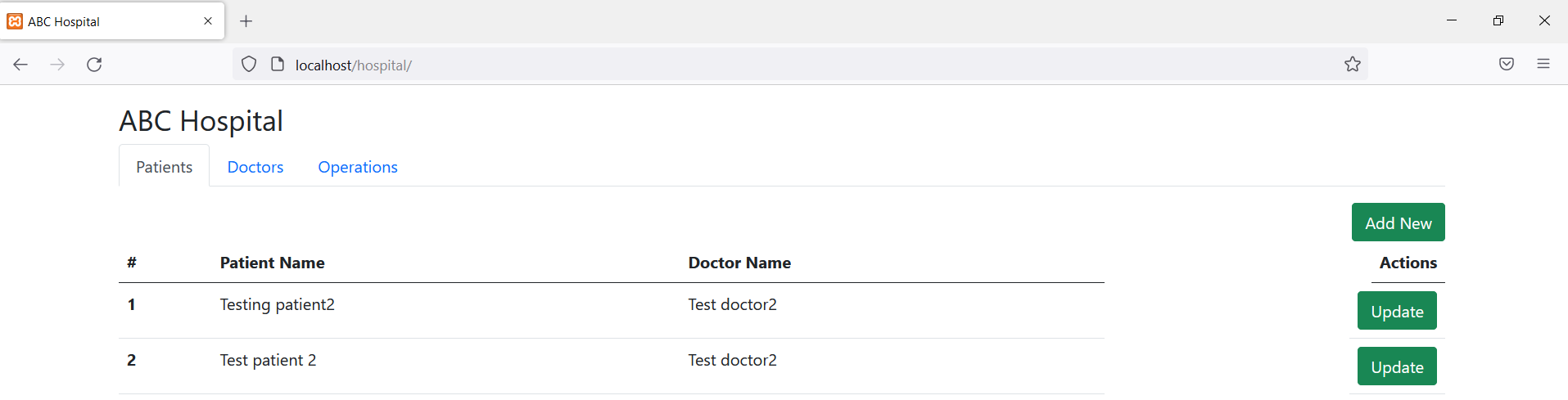
1. Now again navigate to the My SQL Workbench and select another connection which is associated with the database with private access: AzureDB2

2. Select the rk\_hospital database from the left pane and in the query section, fire the query on the hospital table. Query: Select\* from patient.



3. Once the database connection is tested successfully, launch the browser.

4. Do localhost: <http://localhost/hospital>



**Use Case 6** **–** Connect to storage account using Private Endpoint. Connection established successfully.

**Use case 7 –** Tested file transfer from FileZilla in RG-01 to FTP server in RG-03. Successfully completed.

A screenshot of a computer

Description automatically generated

# Detailed task list

|  |  |  |
| --- | --- | --- |
| Milestones | Date | Comments/Assumptions |
| Creation of Resource Groups, vNets and Subnets - NAIKA | 13th March, 2022 | Successful |
| Creation of Resource Groups, vNets and Subnets - KAN | 13th March, 2022 | Successful |
| vNet Peering creation and configuration between vNets in same subscription | 14th March,2022 | Objective achieved |
| Created and configured client workstation for testing and connectivity | 18th March, 2022 | Objective achieved |
| Build MySQL database and test the connection (Public and Private) | 22nd March, 2022 | Tested successfully |
| Create XAMPP Server, configure the application and connect to MySQL DB | 23rd March, 2022 | Tested successfully |
| Build FTP server website, install FileZilla on client machine, test file transfer across | 23rd March, 2022 | Objective achieved |
| VNet Peering across different subscription | 25th March, 2022 | Objective achieved |
| Create Azure Firewall with subnet and route | 26th March, 2022 | Objective achieved |
| Configure the Private Endpoint | March 26, 2022 | Tested successfully |

# Roles & Responsibilities

Detailed resource plan is captured as part of the Detailed Implementation Task List. A matrix is captured here to identify key stakeholders and their responsibility with respect to this implementation.

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nitika Gupta | Project Manager/ Leader | * Lead the team. * Assigned tasks, Kept track on task complication. * Troubleshooting technical issues. * Created Topology design to use for build activities. Created NAIKA network architecture in Azure, including Resource groups, vNets and Subnets. * Created and configured peering between different vNets in NAIKA (CapstoneAcc2 Azure account). * Tested connectivity between all peered networks. * Created and configured bi-directional peering between CapstoneAcc2 and SYST8200Proj Azure accounts. * Test connectivity with and without Firewall with Kanan across the peered networks. * Create a storage account with private endpoint and test connectivity from private network. * Created the Build document template for the Group. * Updated sections 2, 3, 4.3, 4.4, 5.1, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.13, 8 of the Build guide for-build activities * Conducted regular meetings and connects to discuss the limitations, constraints, discuss workarounds and track build and guide update progress. * Final review of the document and inspection of the project. |
| Kanan Garach |  | * Created the client desktop VMs and installed required applications * Monitored Azure accounts to ensure that resource usage is as needed as the credits are a part of the free tier constraints we must manage in our solution * Created and configured Azure Firewall to deliver enhanced security to the solution * Created and configured the Azure Firewall and its related components * Updated sections 4.5, 6.1, 6.4, 6.7, and 6.9 of the build guides * Troubleshooting database connectivity issues * Troubleshoot firewall connectivity issues * Regularly attended the group discussions and meetings involving the build guide * Actively participated in the troubleshooting and workaround discussions for limitations and constraints experienced during build activities |
| Anjali Kumari |  | * Created and configured the web server and set up the environment to run the application on it. * Troubleshooting the technical issues for database connectivity and deploying the application. * Created and configured the public and private Azure My SQL database server. * Configured the connectivity between the application and database. * Updated the section 5.2, 6.10,6.11,6.12,14 in the build book. * Tested the connectivity between the application and webserver with the databases. * Attended all the meetings regularly, discussed the workaround with the teams for the issues and constraints together. |
| Thanh Trung Nguyen |  | * Created Detailed implementation task for the FTP Server machine. * Configure Firewall rules and other network configuration for FTP Server. * Tested file transfer from FileZilla to FTP server across different networks. * Configure the SSL Certificate and test on FTP Server * Test the connection from the user and VNet 2 to FTP Server with the private IP. * Updated the section: 1, 6.14, 6.15 in the build book. * Attended all the meetings regularly, discussed the workaround with the teams for the issues and constraints together. |

# References

*Create a virtual network peering - Resource Manager, different subscriptions and Azure Active Directory tenants.* (2022, February). Retrieved from Azure Documentation: https://docs.microsoft.com/en-us/azure/virtual-network/create-peering-different-subscriptions

*Deploy and configure Azure Firewall using the Azure portal.* (2021, October). Retrieved from Microsoft Documentation: https://docs.microsoft.com/en-us/azure/firewall/tutorial-firewall-deploy-portal

*https://getanadmin.com/azure/configure-ftp-server-on-azure-virtual-machine/.* (2020, January). Retrieved from https://getanadmin.com/azure/configure-ftp-server-on-azure-virtual-machine/

*Private Link.* (n.d.). Retrieved from https://azure.microsoft.com/en-us/services/private-link/#features

*Private Link documentation.* (n.d.). Retrieved from Microsoft Docs: https://docs.microsoft.com/en-us/azure/private-link/

*Tutorial: Connect to a storage account using an Azure Private Endpoint.* (2021, April 13). Retrieved from Microsoft Docs: https://docs.microsoft.com/en-us/azure/private-link/tutorial-private-endpoint-storage-portal