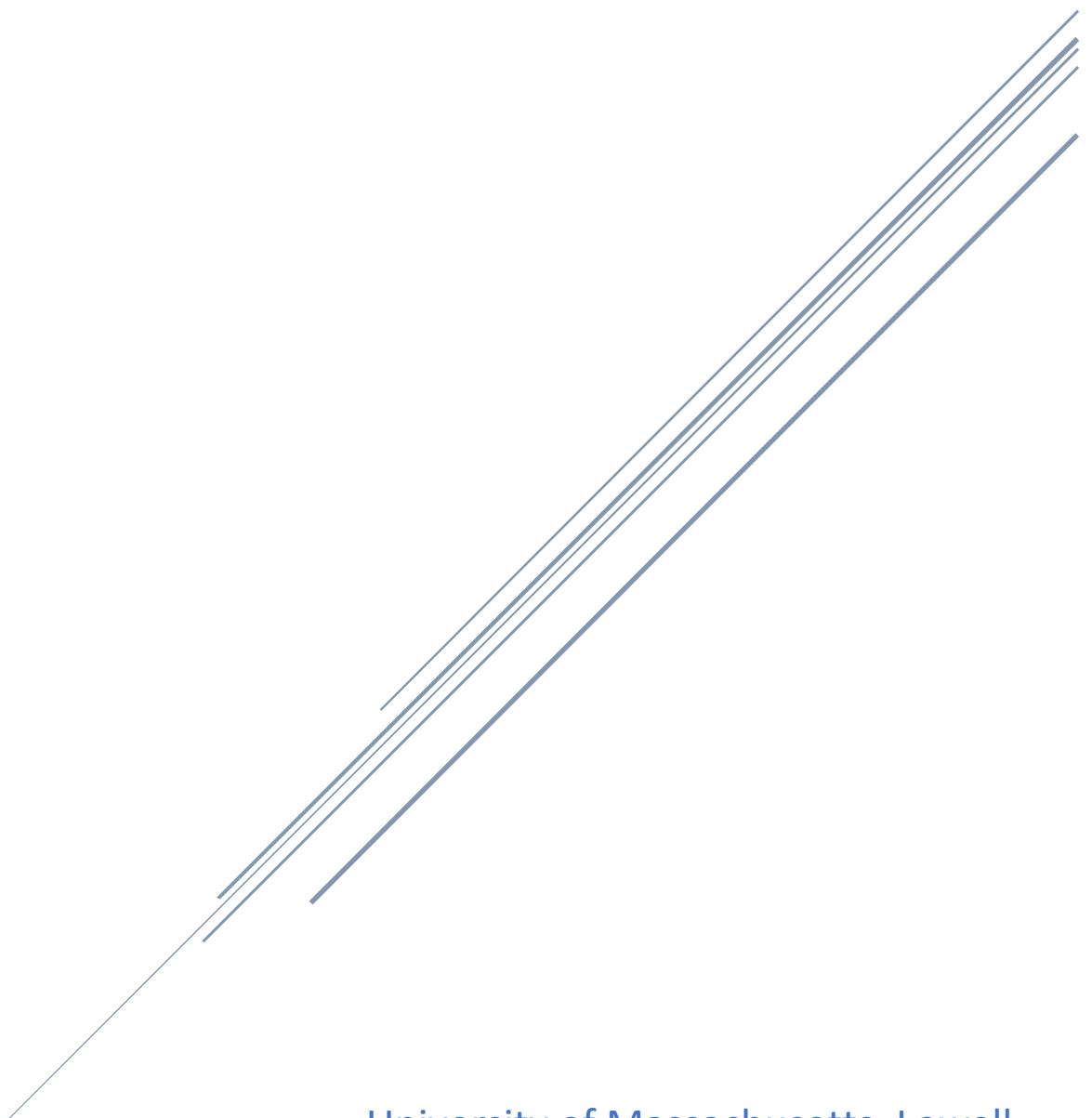


AZURE PROJECT

MSIT Fall 2020

By Kamalakar Borlakunta



University of Massachusetts, Lowell
Cloud Computing by Prof. Robert Bell

Kamalakar Borlakunta
MSIT 5650 Fall 2020 – Azure Project

INTRODUCTION:.....	2
PROJECT:	2
PROJECT DIAGRAM.....	3
PROJECT IMPLEMENTATION	4
RESOURCE GROUP:.....	4
VIRTUAL NETWORK INFRASTRUCTURE:	4
<i>Virtual Network (VNet) and sub networks (subnet):.....</i>	4
<i>Network security Groups.....</i>	6
LOAD BALANCER	8
VIRTUAL MACHINES.....	11
AZURE BASTION	12
<i>Using Bastion to connect and configure Virtual Machines</i>	13
<i>Testing out Nginx and load balancer before deploying demo app</i>	15
COSMOS DATABASE - MONGO DB API.....	18
DEPLOYING NODE.JS APP.....	20
VERIFYING DATA IN COSMOS DB.....	23
BACKUP VMs USING RECOVERY SERVICES VAULT.....	26
BILLING ALARM	32
PERFORMANCE ALERTS – CPU UTILIZATION	34
ACCESS CONTROL	36
CONCLUSION	36
REFERENCES.....	38

Introduction:

Azure is one of the leading cloud providers on the market offering a variety of IaaS, PaaS and SaaS solutions. They are rapidly increasing their services with many of the services in public and private preview.

In this paper I explored various services of Azure platform with a goal of getting hands on experience on Azure cloud platform. I have explored creating virtual networks, subnets and configuring network rules using network security groups (NSG), provisioning virtual machines as an Infrastructure as a service (IaaS), using Azure Bastian services to securely connect to virtual machines for configuration and maintenance from within Azure local network without having to expose them to world wide web for security reasons, deploying a simple web application on two virtual machines and configuring load balancer for web traffic, using Azure Cosmos database as a Platform as a service (PaaS) solution to persist application data with Mongo DB API, using Azure Recovery services Vault to back up and restore VMs and finally configuring alerts and monitoring.

Project:

For this project I have used Azure Cloud CLI commands to create and configure resources. After creating the resources, CLI provides a JSON response. However, I have not included the response in this document. I have provided the screen shots of the resources created from the azure portal instead.

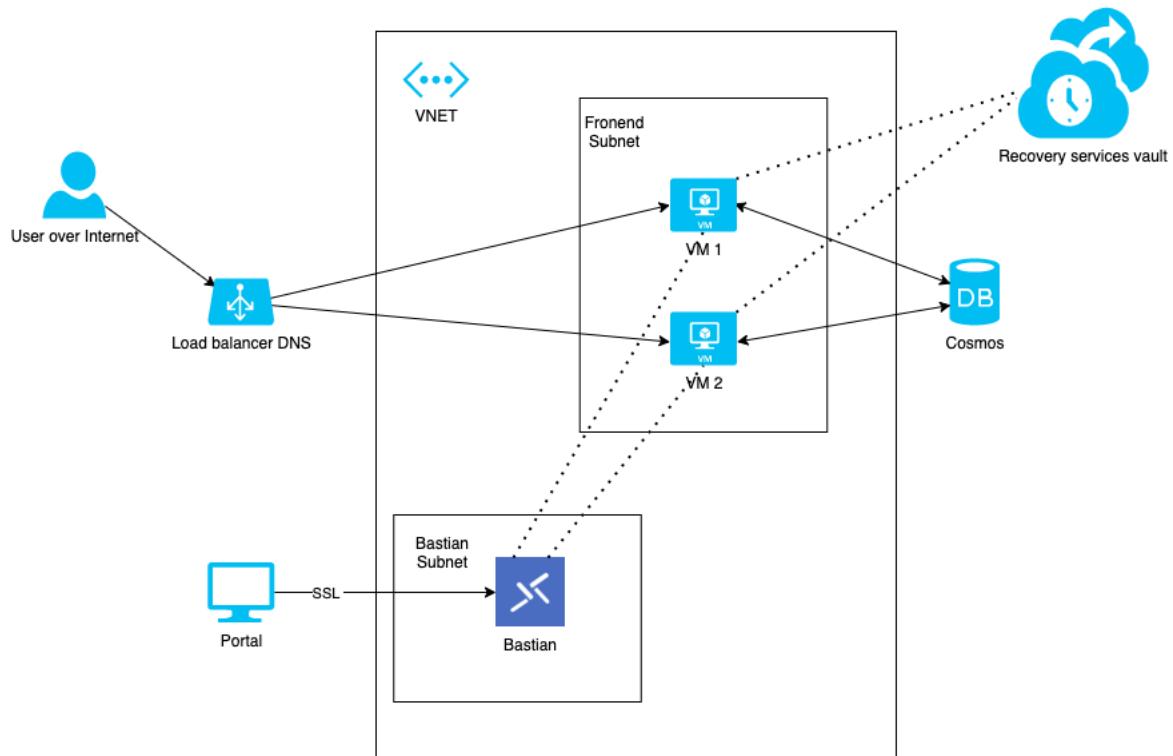
In this project I first created a virtual network, subnets and Network interface cards (NICs). These are long living resources and are not changed very often in an enterprise setting. These resources will usually be maintained by a team separate than the team building the application solution. I went on to build the application after the network is in place.

For the application, I chose a simple Node.js webapp with MongoDB as backend as I have some beginner level exposure to Node.js. As I want to demonstrate the load balancing between Virtual machines, I went with deploying the webapp on virtual machines instead of using the Azure App services (PaaS). The web application is taken from another git repository and is slightly modified to suit my needs. I have included the git repo in the references section at the end of this paper. Using this application, anyone from the internet should be able to provide a name and email Id

and click on a button to register. The registration details will be saved to the Azure Cosmos DB. There is a password protected admin path from where we can view all the registrations.

In order to configure the virtual machines after provisioning them, I have chosen to use the new bastion services provided by azure. Prior to the introduction of Bastian service, the approach has been to use another Virtual machine as a jump box where users would ssh into (or RDP in case of windows) a single VM on the network that was exposed to internet and then connect to another VMs on the same network as the jump box to configure or maintain them. The Idea is to minimize the attack surface by not exposing all VMs to the internet. Using the Azure bastion service, we can connect to VM directly using the browser from the azure portal to any VM on private network. This is more secure as the VM need not be exposed to internet.

Project Diagram



Project Implementation

Resource Group:

Resource Group is a way to organize azure resources. All resources that belong to a same functionality can be grouped together under one resource group. Here is the azure CLI command to create a resource group called “KbResourceGroup”. We can put network related resources and application related resources in separate resource group. For this exercise, I have chosen to put all resources under one resource group (“KbResourceGroup”) as I wanted to export this project as a single template.

```
az group create --name KbResourceGroup --location centralus
```

Virtual Network Infrastructure:

In this section, I have set up virtual network, created subnets, configured network security groups to filter the inbound and outbound traffic from subnets.

Virtual Network (VNet) and sub networks (subnet):

Creating virtual network is very easy with azure when compared to buying hardware and setting it up in the datacenter.

This azure CLI command is used to create a virtual network with name KbVnet in the resource group KbResourceGroup created earlier. We can split the network into multiple subnetworks (subnets). The address-prefix of the command 10.0.0.0/16 creates a virtual network with ip address range 10.0.0.0 - 10.0.255.255. I have also created a subnet called KbSubnet-FrontEnd along with the vnet with subnet-prefix of 10.0.1.0/24 which translates to an address range of 10.0.1.0 - 10.0.1.255. Azure however uses 5 IP address of each subnet for reserved use.

The intent of using front end subnet is to separate out the subset of network which is facing the internet. Any machine on this subnet can allow the traffic from internet on port 80 and port 443.

```
az network vnet create \
--name KbVnet \
--resource-group KbResourceGroup \
--location centralus \
```

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

```
--address-prefix 10.0.0.0/16 \
--subnet-name KbSubnet-FrontEnd \
--subnet-prefix 10.0.1.0/24
```

The below CLI command is used to create another subnet for backend purposes. The intent of separating the backend subnet is to segregate the network which doesn't need to be exposed to any http traffic. A typical use case is to attach a database server to this subnet. We can only open up the port on which the database listens to. I have ended up not using this subnet as I switched to using a PaaS Azure Cosmos database, but I am including it here for sake of completeness.

```
az network vnet subnet create \
--address-prefix 10.0.2.0/24 \
--name KbSubnet-BackEnd \
--resource-group KbResourceGroup \
--vnet-name KbVnet
```

Here are the screen shots of the Azure Virtual network and Subnets from azure portal.

This screenshot shows the 'KbVnet' virtual network configuration in the Azure portal. The 'Overview' tab is selected, displaying basic information such as Resource group (KbResourceGroup), Address space (10.0.0.0/16), Location (Central US), Subscription (Azure subscription 1), and Tags. The 'Subnets' section shows two subnets: KbSubnet-FrontEnd (10.0.1.0/24) and KbSubnet-BackEnd (10.0.2.0/24). The 'Connected devices' section is currently empty.

This screenshot shows the 'Subnets' list for the KbVnet virtual network. The 'Subnets' tab is selected. It lists two subnets: KbSubnet-FrontEnd (IPv4 range 10.0.1.0/24) and KbSubnet-BackEnd (IPv4 range 10.0.2.0/24). The table includes columns for Name, IPv4, IPv6 (many available), Delegated to, and Security group.

Name	IPv4	IPv6 (many available)	Delegated to	Security group
KbSubnet-FrontEnd	10.0.1.0/24 (251 available)	-	-	-
KbSubnet-BackEnd	10.0.2.0/24 (251 available)	-	-	-

Network security Groups

Network security groups are used to specify rules for allowing or disallowing network traffic. First a network security group is created for frontend subnet and network security rules are added to allow only http and https traffic ie port 80 and 443 respectively. One thing to note is that NSG rules are evaluated on the basis of priority (lower the priority number higher the priority)

To create NSG

```
az network nsg create \
--resource-group KbResourceGroup \
--name KbNsg-FrontEnd \
--location centralus
```

To create a rule to allow http

```
az network nsg rule create \
--resource-group KbResourceGroup \
--nsg-name KbNsg-FrontEnd \
--name Allow-HTTP-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 100 \
--source-address-prefix Internet \
--source-port-range "*" \
--destination-address-prefix "*" \
--destination-port-range 80
```

To create a rule to allow https

```
az network nsg rule create \
--resource-group KbResourceGroup \
--nsg-name KbNsg-FrontEnd \
--name Allow-HTTPS-All \
--access Allow \
--protocol Tcp \
--direction Inbound \
--priority 200 \
--source-address-prefix Internet \
--source-port-range "*" \
--destination-address-prefix "*" \
--destination-port-range 443
```

Then this CLI Command is used to associate front end NSG to front end Subnet

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

```
az network vnet subnet update \
--vnet-name KbVnet \
--name KbSubnet-FrontEnd \
--resource-group KbResourceGroup \
--network-security-group KbNsg-FrontEnd
```

The screenshot shows the Azure portal interface for managing a Network Security Group (NSG). The top navigation bar includes 'Microsoft Azure', 'Upgrade', 'Search resources, services, and docs (G+)', 'Azure Cloud Shell', and a user profile. The main content area displays the 'KbNsg-FrontEnd' NSG settings under the 'KbResourceGroup' resource group. The left sidebar lists navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Inbound security rules, Outbound security rules, Network interfaces, Subnets, Properties, Locks), Monitoring (Diagnostic settings, Logs, NSG flow logs), and Automation. The 'Overview' section provides basic information: Resource group (KbResourceGroup), Location (Central US), Subscription (Azure subscription 1), Subscription ID (11c060c5-e032-4c53-8ddb-a49a129587c0), and Tags (Click here to add tags). The 'Inbound security rules' table lists five rules:

Priority	Name	Port	Protocol	Source	Destination	Action	...
100	Allow-HTTP-All	80	TCP	Internet	Any	Allow	...
200	Allow-HTTPS-All	443	TCP	Internet	Any	Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow	...

The 'Outbound security rules' table lists three rules:

Priority	Name	Port	Protocol	Source	Destination	Action	...
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow	...

Similarly, NSG for backend subnet is created and outgoing traffic to internet is blocked. The incoming traffic for internet is blocked by default. Backend subnet is intended to be used by databases which needs to be accessed from within private network only. As I didn't not end up using a database VM, I did not open any inbound ports yet (e.g... default port 3306 for MySQL)

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Azure portal interface for managing a Network Security Group (NSG). The left sidebar lists navigation options like Home, Resource groups, and Azure Cloud Shell. The main content area displays the details for the 'KbNsg-BackEnd' NSG under the 'KbResourceGroup' resource group. The 'Overview' tab is selected, showing basic information such as Location (Central US), Subscription (Azure subscription 1), and Tags. Below this, two tables show security rules: 'Inbound security rules' and 'Outbound security rules'. Both tables include columns for Priority, Name, Port, Protocol, Source, Destination, and Action (Allow or Deny). The inbound rules allow traffic from VirtualNetwork to VirtualNetwork, while the outbound rules allow traffic from Internet to VirtualNetwork.

Priority	Name	Port	Protocol	Source	Destination	Action	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	Deny	...

Priority	Name	Port	Protocol	Source	Destination	Action	...
100	Deny-Internet-All	Any	TCP	Any	Internet	Deny	...
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow	...
65500	DenyAllOutBound	Any	Any	Any	Any	Deny	...

Since no NICs and VMs are attached yet to the subnets, the interaction with NSG and the network traffic is demonstrated after VM creation.

Load Balancer

In this section I explored creating a load balancer to distribute the load across 2 VMs. I have configured the load balancer to check for the health of application using and html page and route traffic across healthy VMs.

First, I created a public IP.

```
az network public-ip create \
--resource-group KbResourceGroup \
--name publicip \
--sku standard
```

Next, I created a load balancer and associated the public ip created in previous step to the front-end pool during creation

```
az network lb create \
--resource-group KbResourceGroup \
```

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

```
--name loadbalancer \
--public-ip-address publicip \
--frontend-ip-name frontendpool \
--backend-pool-name backendpool \
--sku standard
```

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes 'Microsoft Azure', 'Upgrade', 'Search resources, services, and docs (G+)', 'Azure Cloud Shell', and the user's name 'Kamalakar_Borlakunta@...' with 'UNIVERSITY OF MASSACHUSETTS' below it. The main content area is titled 'KbResourceGroup - Microsoft Azure' and shows the 'Resource groups' blade. On the left, there's a sidebar with 'Resource groups' (containing 'cloud-shell-storage-southcentralus', 'KbResourceGroup', 'kbTestRG', and 'NetworkWatcherRG'), 'Quickstart', 'Deployments', 'Policies', 'Properties', 'Locks', 'Cost Management' (with 'Cost analysis', 'Cost alerts (preview)', 'Budgets', and 'Advisor recommendations'), and 'Monitoring' (with 'Insights (preview)'). The right side displays the 'Overview' tab for the 'KbResourceGroup' resource group. It shows the following details:

Subscription (change)	Deployments
Azure subscription 1	No deployments
Subscription ID 11c060c5-e032-4c53-8ddb-a49a129587c0	
Tags (change)	
Click here to add tags	
Filter by name...	Type == (all) X Location == (all) X Add filter
Name ↑	Type ↑↓ Location ↑↓
KbNsG-BackEnd	Network security group Central US
KbNsG-FrontEnd	Network security group Central US
KbVNet	Virtual network Central US
loadbalancer	Load balancer Central US
publicip	Public IP address Central US

Next, a health probe is created so it can be associated with load balancer rule later. This health probe is used by azure load balancer to determine if traffic can be routed to a VM or not. Health check can be configured at networking level to check for a TCP port response or at application level. Here I chose to use application level by checking if a static html page called health.html is up.

```
az network lb probe create \
--resource-group KbResourceGroup \
--lb-name loadbalancer \
--name healthprobe \
--protocol http \
--port 80 \
--path health.html \
--interval 10 \
--threshold 3
```

Kamalakar Borlakunta
MSIT 5650 Fall 2020 – Azure Project

Next, load balancer rules are created to use health probe. Also, the traffic from front end IP on port 80 is routed to port 80 on the nics/vms associated with the backend pool.

```
az network lb rule create \
--resource-group KbResourceGroup \
--lb-name loadbalancer \
--name httprule \
--protocol tcp \
--frontend-port 80 \
--backend-port 80 \
--frontend-ip-name frontendpool \
--backend-pool-name backendpool \
--probe-name healthprobe
```

Network Interface Cards: Network Interface cards (NICs) are used to connect virtual machines to the network. Here NICs are created and are associated with the load balancer backend pool. No VMs are created yet. VMs will be created later and attached to these NICs.

```
az network nic create \
--resource-group KbResourceGroup \
--name webnic1 \
--vnet-name KbVnet \
--subnet KbSubnet-FrontEnd \
--lb-name loadbalancer \
--lb-address-pools backendpool
```

```
az network nic create \
--resource-group KbResourceGroup \
--name webnic2 \
--vnet-name KbVnet \
--subnet KbSubnet-FrontEnd \
--lb-name loadbalancer \
--lb-address-pools backendpool
```

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Azure portal interface for managing a Load Balancer named 'loadbalancer'. The left sidebar lists navigation options such as Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Frontend IP configuration, Backend pools, Health probes, Load balancing rules, Inbound NAT rules, Outbound rules, Properties, Locks, Monitoring, Alerts, Metrics, and Insights (preview). The main content area displays essential details about the load balancer, including its resource group (KbResourceGroup), location (Central US), subscription (Azure subscription 1), and SKU (Standard). It also shows the backend pool (backendpool), health probe (healthprobe), and load balancing rule (httprule). A section titled 'Configure high availability and scalability for your applications' provides links to balance IPv4 and IPv6 addresses, build highly reliable applications, and secure your networks.

Virtual Machines

I have chosen to use Linux VM for this project. Operating system Ubuntu and VM size of Standard_B1ms is used. It has 1 CPU core only, but it is enough for our demo app. For the first VM, availability zone of 1 is used and for the second VM availability zone of 2 is used. The VMs are attached to network interface cards (NICs) webnic1 and webnic2 respectively.

The generate ssh option creates the RSA public and private keys during VM creation in '.ssh' folder of home directory of cloud CLI storage. If the key already exists from a previous time, the existing keys will be used.

```
# first VM - assign it to availability zone 1
az vm create \
--resource-group KbResourceGroup \
--name webvm1 \
--image ubuntults \
--size Standard_B1ms \
--admin-username kamal \
--generate-ssh-keys \
--zone 1 \
--nics webnic1
```

```
#second VM
az vm create \
--resource-group KbResourceGroup \
--name webvm2 \
--image ubuntults \
--size Standard_B1ms \
--admin-username kamal \
--generate-ssh-keys \
--zone 2 \
--nics webnic2
```

As the Virtual machines are connected to NICs that are already configured for the load balancer, the VMs should be automatically on the load balancer.

Azure Bastion

Azure bastion is used to securely connect to Virtual machines right from azure portal without exposing them to internet. We need not expose the VMs to internet for SSH or RDP access.

To create azure bastion there are some prerequisites. First a subnet with exact name ‘AzureBastionSubnet’ needs to be created in the same Vnet to which the virtual machines belong to. Azure will use this subnet to create Bastian resources.

Here is the CLI Command to create the subnet

```
az network vnet subnet create \
--address-prefix 10.0.0.0/24 \
--name AzureBastionSubnet \
--resource-group KbResourceGroup \
--vnet-name KbVnet
```

Next a public Ip needs to be created for Bastion service. This IP should be created in the same region as the vnet.

```
az network public-ip create \
--resource-group KbResourceGroup \
--name KbBastionIp \
--sku Standard \
--location centralus
```

After the subnet and IP address are created, Bastian can be created with the following command. It takes about 5 minutes to create the Bastion resources.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

```
az network bastion create \
--name KbBastion \
--public-ip-address KbBastionIp \
--resource-group KbResourceGroup \
--vnet-name KbVnet \
--location centralus
```

Using Bastion to connect and configure Virtual Machines

To connect a VM using bastion, Go to VM overview tab from azure portal, click on connect and select Bastion from the drop down. On the next screen click on the button “Use Bastion”. This will prompt for user id and the SSH private key. Click on connect after entering these details.

Azure logs the user in and opens up the session in another browser. Internally Azure opens up a SSH session on the internal network on port 22.

The screenshot shows the Azure portal interface for a virtual machine named 'webvm1'. The 'Connect' button is visible in the top navigation bar. A dropdown menu is open over the 'Connect' button, with 'Bastion' selected. The main content area displays the following VM details:

Setting	Value
Operating system	Linux (ubuntu 18.04)
Size	Standard B1ms (1 vcpus, 2 GiB memory)
Public IP address	52.143.244.202
Virtual network/subnet	KbVnet/KbSubnet-FrontEnd
DNS name	kbdemo.centralus.cloudapp.azure.com

Below these details, there are sections for 'Virtual machine' and 'Networking' with their respective properties listed.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Search (Cmd+)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Networking

Connect

Disk

Size

Security

Advisory recommendations

Extensions

Continuous delivery

Availability + scaling

Configuration

Identity

Properties

SSH Private Key * (RSA PRIVATE KEY)

Open in new window

Username * kamal

Authentication Type * (Password, SSH Private Key, SSH Private Key from Local File)

SSH Private Key *

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEWdJHPixue8KeNA1TmYyfdnv7Ritqfp1HN4kFIGY7RG/
ljTgNzD/7V+7ZTLMUFSKczs99C1WqgFAGCr4r0BbfvzG7rzVhm7heavM1xBSLE
hr1mQgMN22GsmAGYbcYjlm/1LSYk6ZRSOpkt57wJBzE4fe8WlkLPBzXUlrZRS
dadYmmVWzDU1gbS+C+dth+vnUeo/y4LqRv85ySiCnNht+lxboXcdkrueh
VddjbzUOjL96EZ62hPU+xL4udyDeqOWp99UKmy7CeWdU4lpWM+hLmw9wQ/WuYA
9EBG8Psd8+moDmMcjyVH1ZG+XybAW3lnN+K4wiDAQABAoIBAkwwOdPmylk/2kf

Advanced

Connect

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1026-azure x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sat Oct 3 22:56:16 UTC 2020

System load: 0.0 Processes: 110
Usage of /: 4.4% of 28.90GB Users logged in: 0
Memory usage: 14% IP address for eth0: 10.0.1.4
Swap usage: 0%

* Kubernetes 1.19 is out! Get it in one command with:
  sudo snap install microk8s --channel=1.19 --classic
  https://microk8s.io/ has docs and details.

0 packages can be updated.
Updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

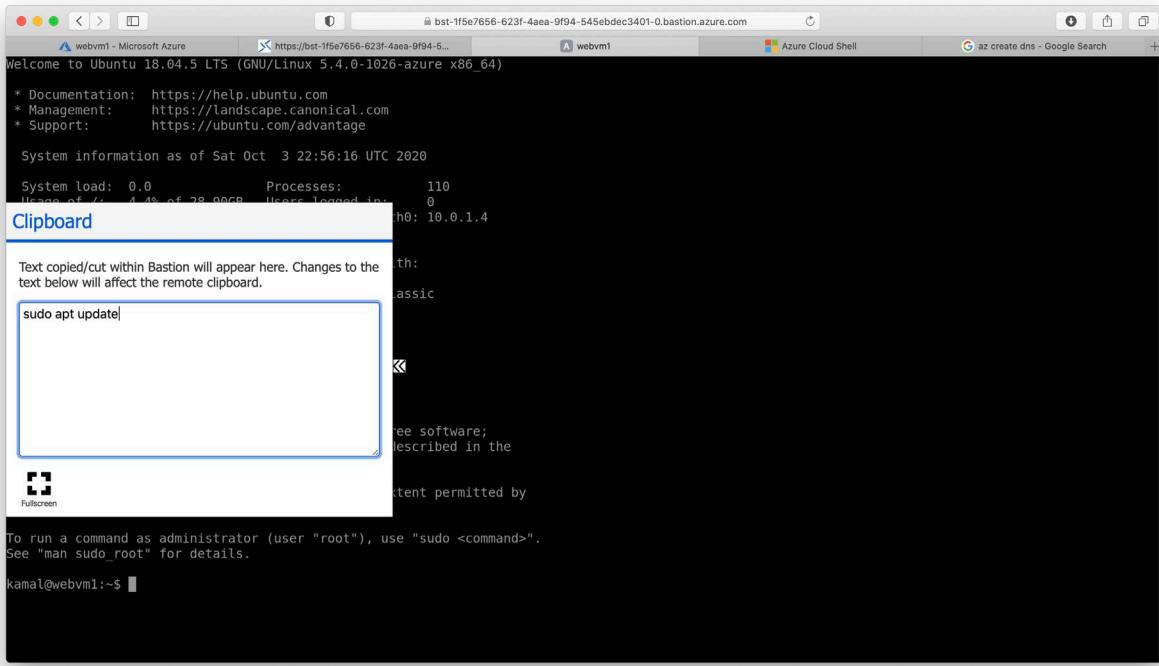
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kamal@webvm1:~$
```

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project



To configure the VM to use demo web application, I updated apt package manager and installed the packages – git, node js , npm , and ngnix using below shell commands.

```
sudo apt update
sudo apt install git
sudo apt install nodejs
sudo apt install npm
sudo apt install nginx
```

One thing to note is that we cannot copy and paste text directly into the vm. Instead click on the small arrow mark icon on the left edge of browser which opens up a clipboard. We can paste text here which can be used in the VM.

Testing out Nginx and load balancer before deploying demo app

In order to test out the configurations made up to this point, I went ahead and added DNS entry for the public IP resource from azure portal.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

publicip | Configuration

Public IP address

Assignment: Static
IP address: 52.143.244.202
Idle timeout (minutes): 5

DNS name label (optional): kbdemo
URL: .centralus.cloudapp.azure.com

So, the app should be available on <http://kbdemo.centralus.cloudapp.azure.com>. However, it shouldn't work yet as the health probe is set to html page which doesn't exist yet. I created another health probe name healthprobe2 to use TCP port 80 and modified the load balancer to use this health probe instead. This is a temporary change and will be reverted back after testing.

loadbalancer | Health probes

Name	Protocol	Port	Path	Used By
healthprobe	HTTP	80	health.html	-
healthprobe2	TCP	80	-	httprule

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Azure portal interface for configuring an HTTP rule. The top navigation bar includes 'Microsoft Azure', 'Upgrade', 'Search resources, services, and docs (G+)', and 'Welcome to nginx!'. The main content area is titled 'httprule' under 'loadbalancer'. The configuration fields include:

- Name:** httprule
- IP Version:** IPv4 (selected)
- Frontend IP address:** 52.143.244.202 (frontendpool)
- Protocol:** TCP (selected)
- Port:** 80
- Backend port:** 80
- Backend pool:** backendpool (2 virtual machines)
- Health probe:** healthprobe2 (TCP:80)
- Session persistence:** None
- Idle timeout (minutes):** 4

After this change hitting <http://kbdemo.centralus.cloudapp.azure.com> in browser show the Nginx screen. This will be changed in next steps where a Node.js app will be deployed that persists data into Cosmos DB.

The screenshot shows a web browser window with the URL 'Not Secure — kbdemo.centralus.cloudapp.azure.com'. The page content is the standard Nginx welcome message:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

As of this step, load balancer routes traffic only to webvm1 virtual machine as Nginx is configured only on this machine.

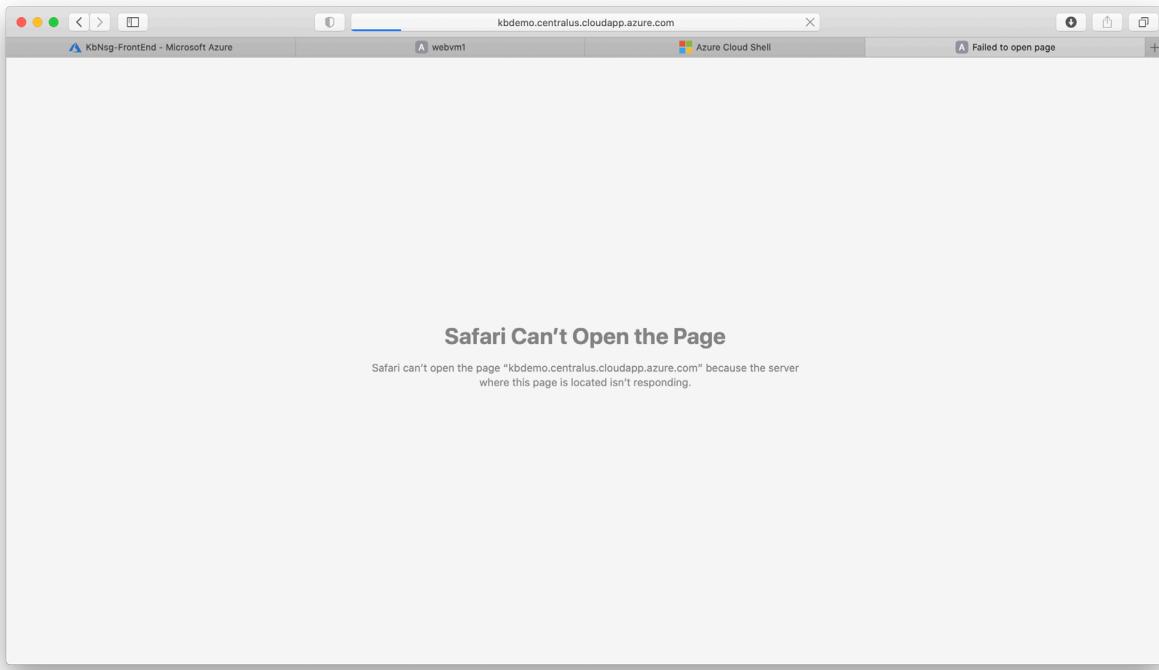
To test the NSG rules created earlier, I temporarily deleted the inbound security rule which allows all traffic on port 80.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Inbound security rules							
Priority	Name	Port	Protocol	Source	Destination	Action	
200	Allow-HTTPS-All	443	TCP	Internet	Any	Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	Deny	...

Hitting the load balancer URL timed out after removing the rule to allow http traffic. During testing I found that it took about 30 to 60 seconds for the NGS rule changes to take effect.



Cosmos Database - Mongo DB API

The Node.js application that is yet to be installed on VMs uses MongoDB document storage to persist data. Therefore, I installed the Azure Cosmos DB with Mongo DB API first. In the backup policy, I have left the defaults setting of 240-minute backup interval and 8 hours of back up retention.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Validation Success

Azure Cosmos DB is a globally distributed, multi-model, fully managed database service. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

Project Details

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

Subscription * Resource Group * [Create new](#)

Instance Details

Account Name * API * Notebooks (Preview) Location * Capacity mode [Learn more about capacity mode](#)

With Azure Cosmos DB free tier, you will get 400 RU/s and 5 GB of storage for free in an account. You can enable free tier on [Review + create](#) [Previous](#) [Next: Networking](#)

Basics Networking **Backup Policy** Encryption Tags Review + create

Azure Cosmos DB provides two different backup policies. You will not be able to switch between backup policies after the account has been created.

Backup policy [Sign up for enabling continuous backup policy](#)

Backup interval 60-1440

Backup retention 8-720

Copies of data retained 2

For additional pricing details, please check [here](#)

It took about 10 minutes to create the database.

Connection string can be found in the settings. This will be used by the Node.js application to connect to Cosmos DB.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Azure portal interface for the 'kb-cosmos-db' database account. The left sidebar lists various options like Tags, Diagnose and solve problems, Quick start, Notifications, Data Explorer, Settings, Connection String (which is selected), Features, Replicate data globally, Default consistency, Backup & Restore, Firewall and virtual networks, Private Endpoint Connections, Preview Features, Locks, Collections, and Browse. The main content area displays the 'Connection String' configuration page. It includes fields for HOST (kb-cosmos-db.mongo.cosmos.azure.com), PORT (10255), USERNAME (kb-cosmos-db), PRIMARY PASSWORD (X2uTyHAcRP9S8gVPSKaA3vVB8rufxsKbg4moy6XBvyxo33NdRWcm3W6qxSxEpY0he8IM60MSh06KP2Num3mz0A==), SECONDARY PASSWORD (XuXDXDaZnQac5fgr9xHKmW8GsaSKA4IpgJ0kDM1TpLsFofksPGDS8ID81zsHZOyGbs0DBokKxaSW78NghnuOw==), PRIMARY CONNECTION STRING (mongodb://kb-cosmos-db:X2uTyHAcRP9S8gVPSKaA3vVB8rufxsKbg4moy6XBvyxo33NdRWcm3W6qxSxEpY0he8IM60MSh06KP2Num3mz0A==@kb-cosmos-db.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=@kb-cosmos-db@), SECONDARY CONNECTION STRING (mongodb://kb-cosmos-db:XuXDXDaZnQac5fgr9xHKmW8GsaSKA4IpgJ0kDM1TpLsFofksPGDS8ID81zsHZOyGbs0DBokKxaSW78NghnuOw==@kb-cosmos-db.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=@kb-cosmos-db@), and SSL (true). A note at the bottom states: 'Azure Cosmos DB has strict security requirements and standards. Azure Cosmos DB accounts require authentication and secure communication via SSL.'

Deploying Node.js app.

From azure bastion connection for VM1, I cloned the app into home folder from below git repository. I am not the creator of this repository.

```
git clone https://github.com/jameshibbard/demo-node-app.git
```

From the home folder of the project, I ran the command

```
npm install
```

to install all the node module dependencies. Next I created a file named ‘.env’ using nano editor and added the below line to configure the Cosmos DB connection string. This configuration is specific to the design of the application I am using.

```
DATABASE= mongodb://kb-cosmos-
db:X2uTyHAcRP9S8gVPSKaA3vVB8rufxsKbg4moy6XBvyxo33NdRWcm3W6qxSxEpY0he8IM60MSh06KP2Num3mz0A==@kb
-cosmos-
db.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=
120000&appName=@kb-cosmos-db@
```

Then, based on the entry point of the application as defined in ‘package.json’ file, I ran the app using `node start.js` command. The app started running on port 3000.

```
npm WARN npm-demo-app@1.0.1 No repository field.  
kamal@webvm1:~/demo-node-app$ node start.js  
Express is running on port 3000  
Mongoose connection open
```

The node run time serves the webpage on port 3000. Next I configured Nginx to listen to node app on port 80.

For this I replaced the contents of the file `/etc/nginx/sites-available/default` as below

```
server {  
  listen 80;  
  location / {  
    proxy_pass http://localhost:3000;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection keep-alive;  
    proxy_set_header Host $host;  
    proxy_cache_bypass $http_upgrade;  
  }  
}
```

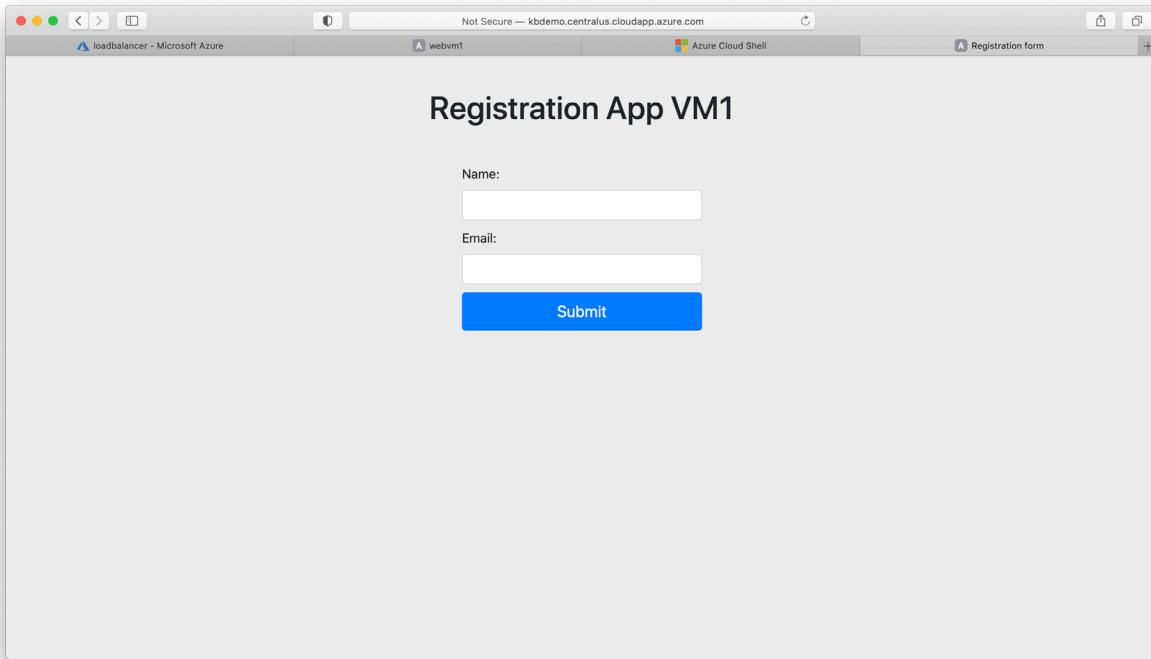
Then restarted Nginx webserver with `sudo systemctl restart nginx`

After about 30 seconds I could see the app load up correctly. I have also modified the html of the webpage at ‘/views/layout.pug’ using nano editor to display VM1 on the UI to easily identify during load balancing testing

I have also installed a process manager called PM2 to run my node app instead of directly running from terminal. First, I installed pm2 globally and then ran the node app using pm2. Here are the commands.

```
npm install pm2 -g  
pm2 start start.js
```

Kamalakar Borlakunta
MSIT 5650 Fall 2020 – Azure Project



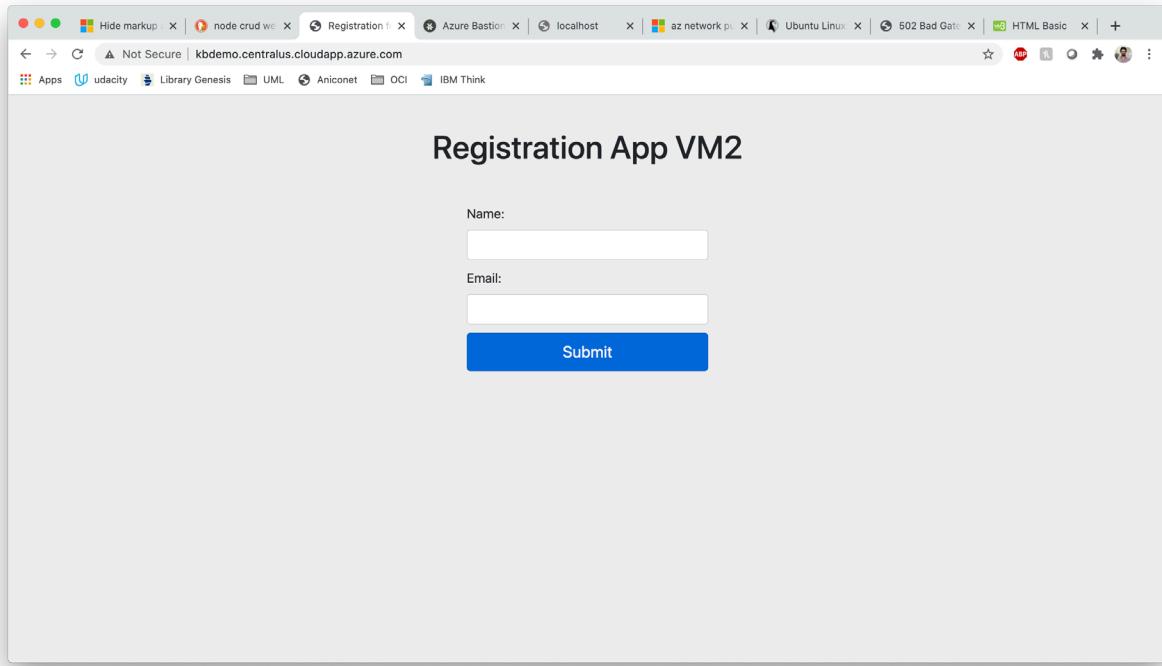
After successfully testing out that web app on VM webvm1, I logged into another VM webvm2 via Azure Bastian and repeated the same configurations.

After configuring both VMs and hitting the load balancer DNS <http://kbdemo.centralus.cloudapp.azure.com/> multiple times. I am able to hit both VMs at random successfully.

Here is the screen shot where load balancer routed the traffic to VM2

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project



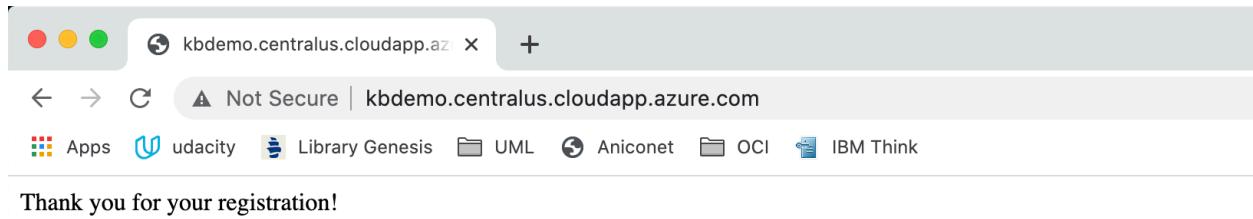
Verifying data in Cosmos DB

Here I have navigated to my app URL and added some test data by using name as John and Email as John@test.com and hitting submit.

The screenshot shows the "Registration App VM1" page. It contains a "Name:" field with "John" entered, an "Email:" field with "John@test.com" entered, and a blue "Submit" button at the bottom.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

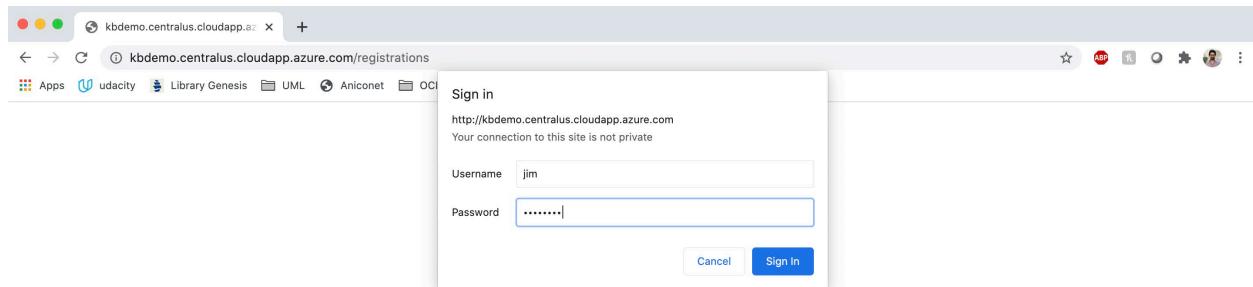


The app saves the data to Cosmos collection and shows a thank you message. There is another route “/registrations” for the app where we can see the registrations. This route is password protected you can use the user as **jim** and password as **password** to view existing registrations.

<http://kbdemo.centralus.cloudapp.azure.com/registrations>

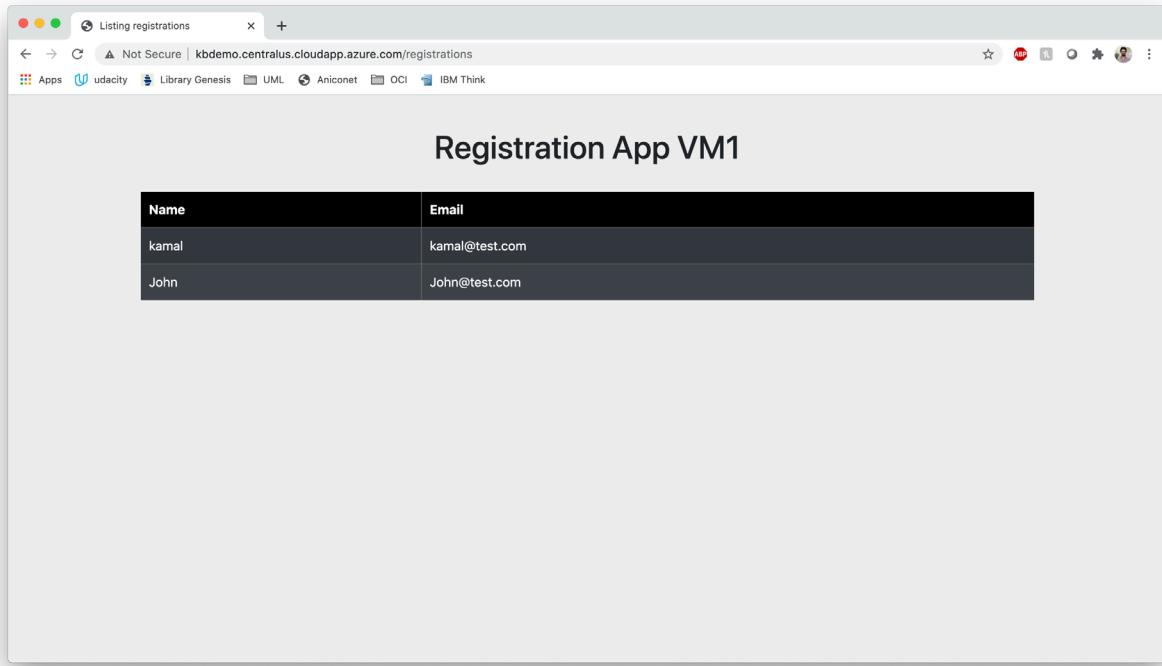
user: jim

password: password



Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project



The data can be viewed and verified directly from azure portal using azure data explorer.

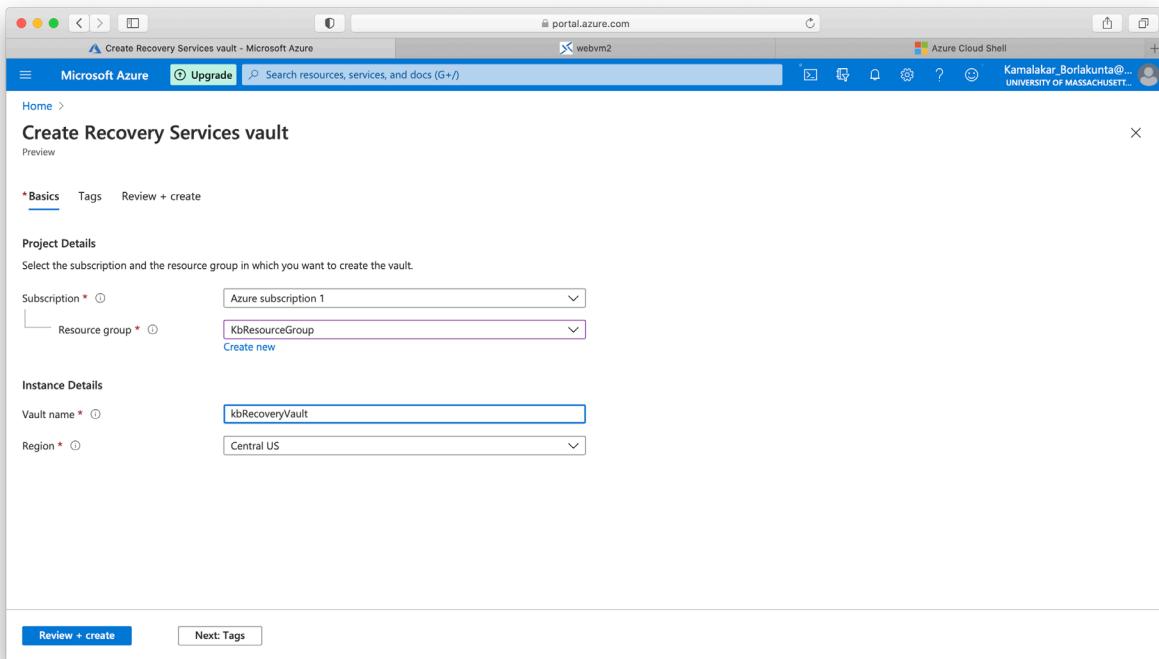
```
_id : ObjectId("5f7941de1d6f3160677d6f99"),
"name" : "John",
"email" : "John@test.com",
"_v" : 0
```

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Backup VMs using Recovery Services Vault

Azure provides way to back up and restore virtual machines and other protected resources for disaster recovery purposes. By default, azure sets the vault for geo-redundant storage, but it can also be configured for local redundancy if needed. The vault location should be in the same location of the resources that need backing up. I chose Central US as location for the Vault as my VMs are in that location.



Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Azure Recovery Services vault 'kbRecoveryVault'. The left sidebar includes sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Identity, Private endpoint connections, Properties, Locks), Getting started (Backup, Site Recovery), Protected items (Backup items, Replicated items), and Manage. The main content area displays a 'What's new' section with links to various Azure features like Backup Center for Azure VM and Azure Database for PostgreSQL Server, SAP HANA databases, Azure File shares snapshot management, and more. At the bottom are two decorative cloud icons.

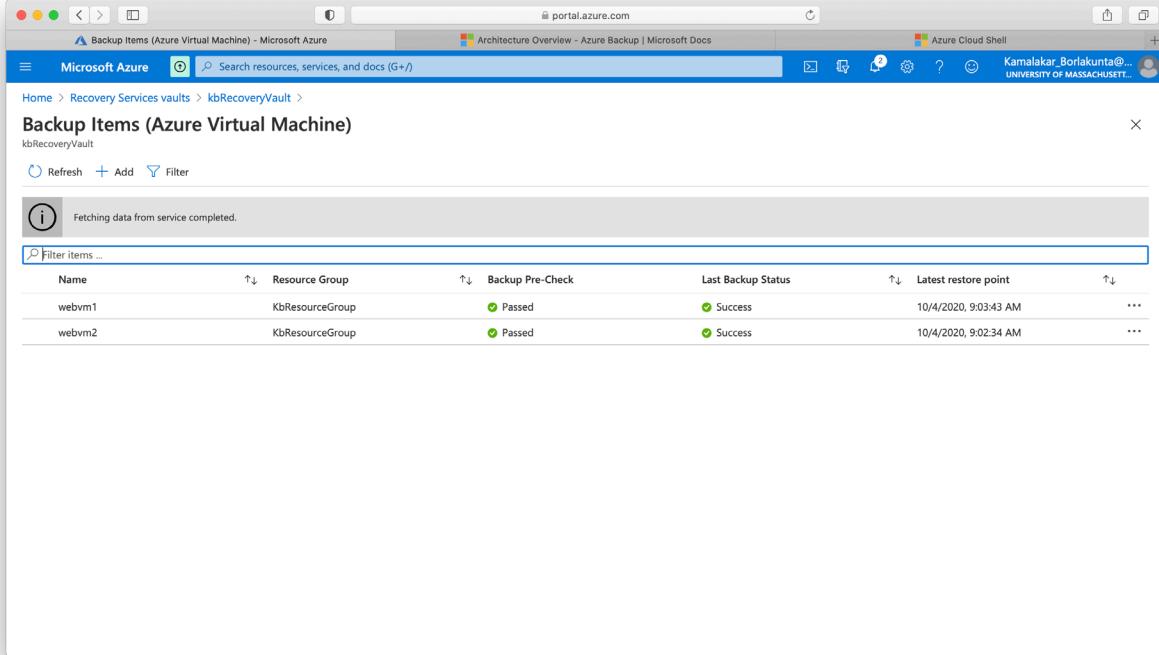
After creating recovery vault, I added VMs webvm1 and Webvm2. I left the default configuration in and the backup is scheduled daily at 2:PM UTC

The screenshot shows the 'Backup Goal' configuration for the 'kbRecoveryVault'. It includes settings for Policy (DefaultPolicy), BACKUP FREQUENCY (Daily at 2:00 PM UTC), Instant Restore (Retain instant recovery snapshot(s) for 2 day(s)), and RETENTION RANGE (Retention of daily backup point: Retain backup taken every day at 2:00 PM for 30 Day(s)). The 'Virtual Machines' section lists 'webvm1' and 'webvm2' under 'Resource Group' 'KbResourceGroup' with the 'OS Disk Only' option selected. A note explains the 'OS Disk only backup' feature. A blue 'Enable Backup' button is at the bottom.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The next day, I was able to see the VMs Backed up.



Backup Items (Azure Virtual Machine)

kbRecoveryVault

Name	Resource Group	Backup Pre-Check	Last Backup Status	Latest restore point	...
webvm1	KbResourceGroup	Passed	Success	10/4/2020, 9:03:43 AM	...
webvm2	KbResourceGroup	Passed	Success	10/4/2020, 9:02:34 AM	...

Before attempting to restore VM, I created a test file in VM1 after the backup was stored to recovery services vault.

```
Last login: Sat Oct  3 22:56:17 2020 from 10.0.0.4
kamal@webvm1:~$ ls
demo-node-app
kamal@webvm1:~$ touch this-is-a-test-file
kamal@webvm1:~$ ls
demo-node-app  this-is-a-test-file
kamal@webvm1:~$ pwd
/home/kamal
kamal@webvm1:~$
```

The restore process also needs a storage account to replace current VM with backup, So I went ahead and created it.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Validation passed

Basics

Subscription	Azure subscription 1
Resource group	KbResourceGroup
Location	Central US
Storage account name	kbstorageaccount2
Deployment model	Resource manager
Account kind	StorageV2 (general purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Performance	Standard
Blob access tier (default)	Hot

Networking

Connectivity method	Public endpoint (all networks)
Default routing tier	Microsoft network routing (default)

Data protection

Point-in-time restore	Disabled
-----------------------	----------

Create < Previous Next > Download a template for automation

For recovering the backup, I selected webvm1 from backup items list and selected restore VM

Name	Resource Group	Backup Pre-Check	Last Backup Status	Latest restore point
webvm1	KbResourceGroup	Passed	Success	10/4/2020
webvm2	KbResourceGroup	Passed	Success	10/4/2020

Pin to dashboard ⌂
Backup now
Restore VM
File Recovery
Stop backup
Delete backup data
Undelete

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

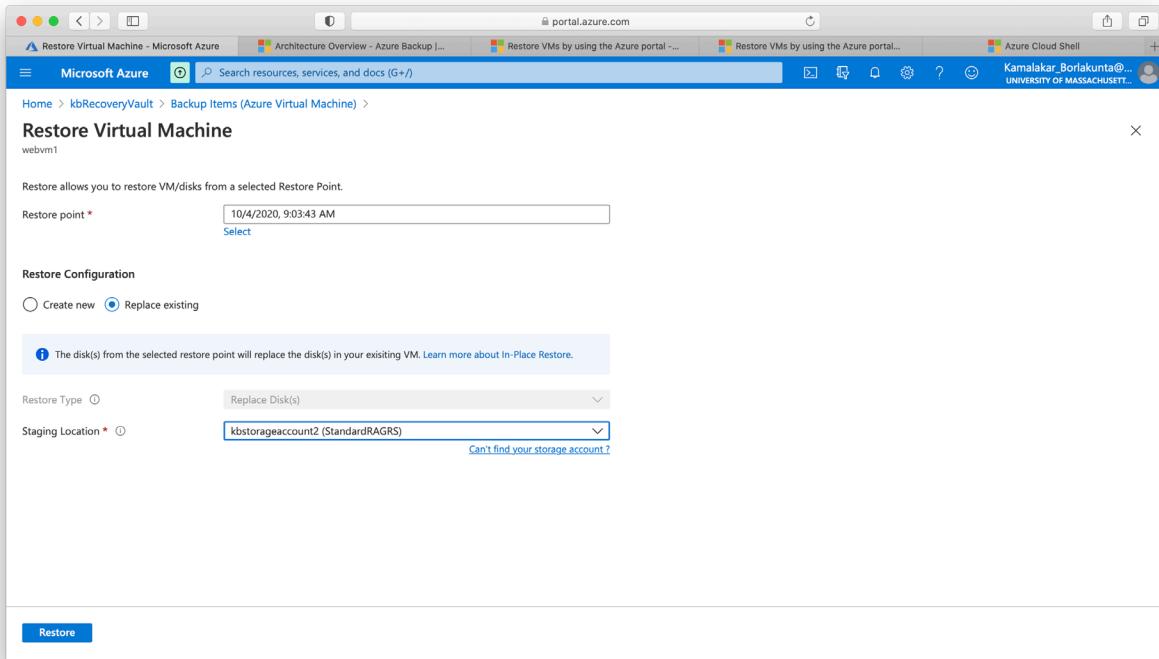
The screenshot shows the Azure portal interface for restoring a virtual machine. The main page displays the 'Restore Virtual Machine' section for 'webvm1'. A modal window titled 'Select restore point' is open, allowing the user to choose between two available restore points based on time, consistency, and recovery type.

Time	Consistency	Recovery Type
10/4/2020, 9:03:43 AM	File-system Consistent	Snapshot
10/4/2020, 8:37:51 AM	File-system Consistent	Snapshot and Vault

There were two options for recovery – either to create a new VM or replace an existing one. For this exercise I went with replacing the existing VM. Azure uses a storage account for staging the data. I provided the details of the storage account. The VM needs to be in deallocated state to use the replace option so I shut down the VM for restore process and click on restore.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project



Restore allows you to restore VM/disks from a selected Restore Point.

Restore point *

Restore Configuration

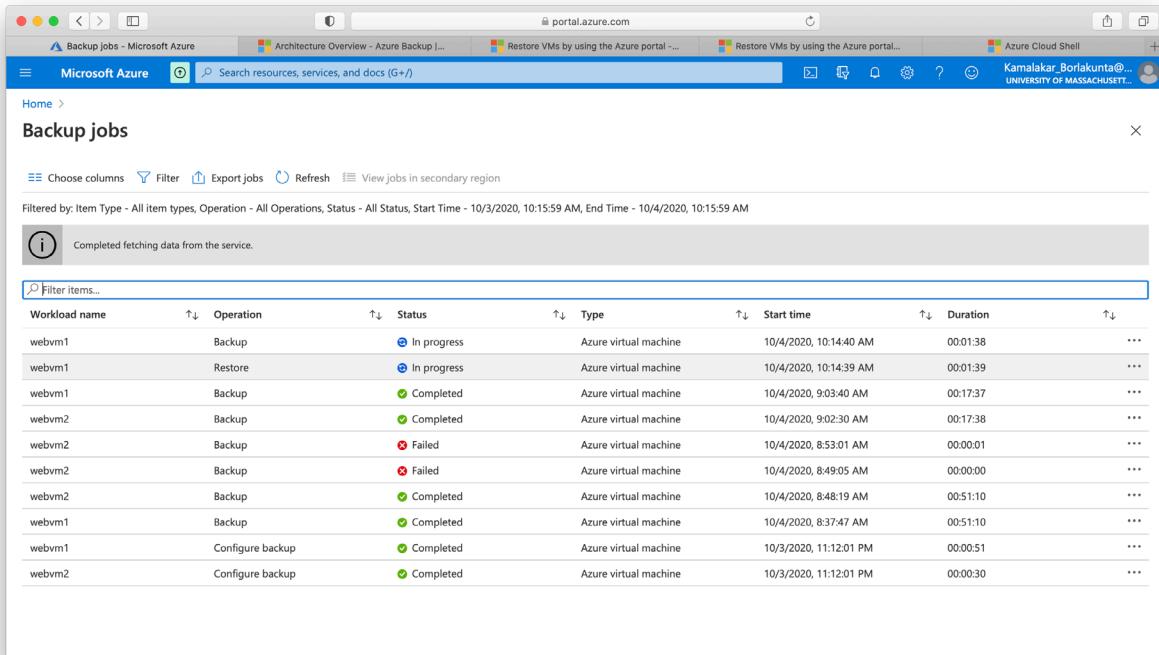
Create new Replace existing

The disk(s) from the selected restore point will replace the disk(s) in your existing VM. Learn more about In-Place Restore.

Restore Type Replace Disk(s)

Staging Location * Can't find your storage account?

From Backup jobs page, I could see that the restore is in progress



Workload name	Operation	Status	Type	Start time	Duration	...
webvm1	Backup	In progress	Azure virtual machine	10/4/2020, 10:14:40 AM	00:01:38	...
webvm1	Restore	In progress	Azure virtual machine	10/4/2020, 10:14:39 AM	00:01:39	...
webvm1	Backup	Completed	Azure virtual machine	10/4/2020, 9:03:40 AM	00:17:37	...
webvm2	Backup	Completed	Azure virtual machine	10/4/2020, 9:02:30 AM	00:17:38	...
webvm2	Backup	Failed	Azure virtual machine	10/4/2020, 8:53:01 AM	00:00:01	...
webvm2	Backup	Failed	Azure virtual machine	10/4/2020, 8:49:05 AM	00:00:00	...
webvm2	Backup	Completed	Azure virtual machine	10/4/2020, 8:48:19 AM	00:51:10	...
webvm1	Backup	Completed	Azure virtual machine	10/4/2020, 8:37:47 AM	00:51:10	...
webvm1	Configure backup	Completed	Azure virtual machine	10/3/2020, 11:12:01 PM	00:00:51	...
webvm2	Configure backup	Completed	Azure virtual machine	10/3/2020, 11:12:01 PM	00:00:30	...

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

After about 30 minutes, I could see the restore process as complete. Now I booted up my webvm1 VM and connected via Bastian to view the home directory. All this time my site was up as the load balancer was routing traffic to webvm2.

As expected, the test file created after the backup was taken could not be found but the project folder is still there.

```
Last login: Sat Oct  3 22:56:17 2020 from 10.0.0.4
kamal@webvm1:~$ ls
demo-node-app
kamal@webvm1:~$ █
```

Billing Alarm

Using Cloud technologies can easily build up costs if the resources are not managed properly. Azure has provided easy way to view the billing by resources, set up budgets and set up alarm when a certain percentage of budget is met. For demonstrating billing alerts, I created a cost budget of \$20 from Cost Management and configured alerts for 10%, 20% and 50% of budget utilization.

The screenshot shows the Microsoft Azure portal interface with the URL portal.azure.com in the address bar. The user is navigating through the 'Create budget' wizard. The left sidebar shows navigation links for Overview, Access control (IAM), Cost Management (Cost analysis, Cost alerts, Budgets, Advisor recommendations), Billing (Invoices, Reservation transactions, Recurring charges, Azure subscriptions, Billing profiles), and Settings (Billing scopes, Properties, Payment methods). The main content area is titled 'Kamalakar Borlakunta | Create budget' under 'Cost Management + Billing | Billing account'. It includes sections for 'Create a budget' and 'Set alerts'. A 'Budget scoping' section allows setting the scope to 'Kamalakar Borlakunta' or changing it. The 'Budget Details' section requires entering a name ('kbbudget'), selecting a reset period ('Monthly'), and specifying creation and expiration dates ('2020 October 1' to '2022 September 30'). At the bottom, there are 'Previous' and 'Next >' buttons.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows the Microsoft Azure portal interface. The title bar reads "Kamalakar Borlakunta - Microsoft Azure". The main content area is titled "Kamalakar Borlakunta | Create budget". On the left, there's a sidebar with sections like "Overview", "Access control (IAM)", "Cost Management" (with "Cost analysis", "Cost alerts", "Budgets" selected, and "Advisor recommendations"), "Billing" (with "Invoices", "Reservation transactions", "Recurring charges", "Azure subscriptions", and "Billing profiles"), and "Settings" (with "Billing scopes", "Properties", and "Payment methods"). The main panel shows "Alert conditions" where a budget of 50% has been set with a threshold of \$10. It also shows "Alert recipients (email)" with three entries: "kamalakar_borlakunta@student.uml.edu", "kamalakar07@gmail.com", and "example@email.com". A note at the bottom says "It is recommended to add azure-noreply@microsoft.com to your email white list to ensure alert mails do not go to your spam folder." At the bottom are "Previous" and "Create" buttons.

Here is the screen shot Azure sent me when notification threshold hit.

The screenshot shows an email from "Microsoft Azure" to "Kamalakar Borlakunta". The subject is "Important notice: You have an Azure budget alert for 'kbbudget'". The email body starts with "You have an alert for budget 'kbbudget'". It states: "Your total spend for budget 'kbbudget' is now \$18.38, exceeding your specified threshold value of \$4.00." Below this, a table provides details about the budget:

Budget name	kbbudget
Budget start date	October 1, 2020
Budget type	Cost
Budget value	\$20.00
Actual value	\$18.38
Notification threshold	\$4.00

At the bottom, there's a blue button labeled "View in Azure portal >" and a note: "Contact Kamalakar_Borlakunta@student.uml.edu if you no longer want to be notified of this budget. If you need additional help, [contact support](#)".

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Performance alerts – CPU Utilization

From the VM monitoring section, I created an alert rule to send email to be sent whenever total percentage CPU is greater than 5%

The screenshot shows the Microsoft Azure portal interface for creating a new alert rule. The URL in the address bar is `portal.azure.com`. The page title is "cpu alert - Microsoft Azure". The breadcrumb navigation shows "Home > Virtual machines > webvm1 > Rules > cpu alert". The main content area is titled "cpu alert" and "Rules management". It includes sections for "Scope", "Condition", and "Action group".

Scope: Selects the target resource "webvm1, webvm2" under "Resource" and shows the "Hierarchy" path: "Azure subscription 1 > kbsourcegroup, KbResourceGroup".

Condition: Configures the alert rule to trigger whenever the total percentage CPU is greater than 5%. The condition is listed as "Whenever the total percentage cpu is greater than 5 %".

Action group: Sets up an action group named "cpuation" which contains one action: "1 Email".

Here is the screen shot of the email I received when the VM hit CPU threshold. For simulating cpu stress, I installed an app called **stress** using '`sudo apt install stress`' and used '`sudo stress -c 1`' command to simulate stress on single core of my VM.

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

The screenshot shows an email from Microsoft Azure. The subject is "Azure: Activated Severity: 3 cpu alert". The body of the email contains the following information:

Rule ID: /subscriptions/11c060c5-e032-4c53-8ddb-a49a129587c0/resourceGroups/KbResourceGroup/providers/microsoft.insights/metricAlerts/cpu%20alert
[View Rule >](#)

Resource ID: /subscriptions/11c060c5-e032-4c53-8ddb-a49a129587c0/resourceGroups/KbResourceGroup/providers/Microsoft.Compute/virtualMachines/webvm1
[View Resource >](#)

Alert Activated Because:

Metric name	Percentage CPU
Metric namespace	virtualMachines/webvm1
Dimensions	microsoft.resourceId = /subscriptions/11c060c5-e032-4c53-8ddb-a49a129587c0/resourceGroups/KbResourceGroup/providers/Microsoft.Compute/virtualMachines/webvm1

The screenshot shows the Azure portal page for the triggered alert. It includes the following details:

Alert Activated Because:

Metric name	Percentage CPU
Metric namespace	virtualMachines/webvm1
Dimensions	microsoft.resourceId = /subscriptions/11c060c5-e032-4c53-8ddb-a49a129587c0/resourceGroups/KbResourceGroup/providers/Microsoft.Compute/virtualMachines/webvm1 microsoft.resourceType = Microsoft.Compute/virtualMachines

Time Aggregation: Total
Period: Over the last 5 mins
Value: 5.93
Operator: GreaterThan
Threshold: 5
Criterion Type: StaticThresholdCriterion

[See in the Azure portal >](#)

Kamalakar Borlakunta

MSIT 5650 Fall 2020 – Azure Project

Access Control

Access control (IAM) is used to create roles and grant access to azure resources. For this project, I added Prof. Bob Bell as a Contributor at resource group level. Contributor is one of the predefined roles created by Azure and it has same access as Owner except that a contributor cannot assign roles.

The screenshot shows the Azure portal interface for managing role assignments in a resource group. The top navigation bar includes 'Home > kbResourceGroup' and a search bar. The main title is 'kbResourceGroup | Access control (IAM)'. Below the title, there are tabs for 'Overview', 'Activity log', 'Access control (IAM)', and 'Tags'. The 'Access control (IAM)' tab is selected. A summary bar indicates 'Number of role assignments for this subscription: 2' and '2000'. A search bar allows filtering by name or email, with filters set to 'Type: All', 'Role: All', 'Scope: All scopes', and 'Group by: Role'. The results table lists two items:

Name	Type	Role	Scope
Bell1, Robert J Robert_Bell1@uml.edu	User	Contributor	This resource
Borlakunta, Kamalakar Kamalakar_Borlakunta@student.u...	User	Owner	Subscription (Inherited)

Conclusion

Although I have barely touched the surface of Azure platform and its offerings, I gained enough hands-on experience during this project to feel confident that I can explore the vast options of its services by looking at the documentation and by playing around by creating proof of concept applications. Before starting this course, I had no idea about the capabilities and scope of any cloud computing platforms but after exploring Azure for this project, I am confident that cloud technologies will continue to increase in importance and adoption for many industries.

Kamalakar Borlakunta
MSIT 5650 Fall 2020 – Azure Project

p.s. I removed Bastian services on 10/7/2020 as my free tier expired and I switched to pay as you go subscription as Bastian services is expensive and is costing about \$5 per day. I have used an alternative implementation of using jump box and ssh agent as I can switch off my VMs when not needed. As of now I am running with only 1 VM to save subscription costs. I can set the bastion services again when prof Bob is ready to view my project in Azure portal.

Here are the private and public RSA keys.

<removed by Prof Bob>

References

<https://docs.microsoft.com/en-us/azure/virtual-network/cli-samples>

<https://docs.microsoft.com/en-us/azure/virtual-network/scripts/virtual-network-cli-sample-filter-network-traffic>

<https://docs.microsoft.com/en-us/azure/bastion/create-host-cli>

<https://docs.microsoft.com/en-us/azure/bastion/bastion-faq>

<https://www.sitepoint.com/build-simple-beginner-app-node-bootstrap-mongodb/>

<https://github.com/jameshibbard/demo-node-app.git>