`

High Availability & Disaster Recovery in Winning Practice cloud application

Vishal Dasa
201601168@daiict.ac.in
Dhirubhai Ambani Institute of Information and Communication technology
Gandhinagar, India
Mentor: Prof. Lavneet Singh

Abstract – I worked on 'Winning Practice' project as part of my B. Tech Project in the Junept India. Winning Practice is six month long live industry project.

In this project I worked in the IT services team and in the latter part of my internship I worked in the backend team. As a part of the IT services team, I have explored various ways to backup critical business data, understand different types of backup, compared different solutions and finally suggested the best one based on my findings. As a part of the backend team, I have created an API to fetch details of backup related operations from azure and used it to display that information on the dashboard of our website.

Keywords – Azure, Cloud intern, Backup, Always on, Rest API

I. INTRODUCTION

Right now, we are living in the digital era. All types of companies are moving towards digitization. All of the things which we were traditionally doing offline like paying different bills, buying different things, buying tickets etc. has become an integral part of the online market. It's the same case with the healthcare sector also. Junept India provides different types of services to dental practices so that they can start their businesses easily and once that business is established, they only have to worry about the clinical work and we take care of the rest of the things. These things include scheduling appointments of the patients through our portal, providing best sellers to dentists so that they can buy required things for their practice, marketing of these practices, handling their finances etc. By proving all of the above mentioned things, we are saving a huge time of dentists. Our web platform will take care of all the non-clinical things so that dentists can focus on the thing which they do the best i.e. dentistry. We had three different teams, 1) Frontend team ,2) Backend team and 3) IT-services team. First I worked in IT-services team .For any IT company who is developing and providing solutions to the customers, disaster recovery is a must have feature because if primary server lost data because of any reason like failure of some disk or because of natural disaster then valuable data of our customers will be lost. I have explored various solutions for disaster recovery, implemented them and after that compared them based on the cost and efficiency of the solution while working in the IT services team. I created scripts or batch files to take backup in Microsoft SQL server and SQL Anywhere server 17.I explored online backup solutions provided by Atera and Azure. I also learned C# language and how to create APIs to fetch data and that using that to show that data in the dashboard of our website which we have created in power-bi. Details of the different tasks which I had completed are given below.

II. PRIMARY WORK

A) Compared Backup solutions provided by Azure and Atera:

1) *Overview*: Azure is Microsoft's public cloud computing platform. It provides different type of services like Software as a Service, Platform as a Service and Infrastructure as a service. These services include different types of solutions related to virtual computing, cloud storage, Backup and disaster recovery solution, identity management solutions, remote monitoring solutions, centralized monitoring etc. Whether you are an individual or a corporate company, small organization or very large organization, Microsoft Azure has something for you. In these solutions, they provide so many options to choose from in which there is a tradeoff between price and performance.

We can use azure backup service for our backup solution. To use azure backup services, First we have to create a recovery service vault in which all configured backups will be stored. Azure provides 2 redundancy options for this recovery service vault. By default it is Geo-redundant storage (GRS).GRS copies our data synchronously three times within a single physical location (Same data center) in the primary region using LRS and after that it copies our data asynchronously to a single physical location in the secondary region. When this asynchronous write operation is completed, it is replicated within that second region using LRS. You can change this option to Locally-redundant storage(LRS).LRS replicates our data three times within a single data center(single physical location) in the primary region.LRS protects our data from disk failure or server failure within a data center but if that whole data center is down, whether because of some technical issue or natural disaster like earthquake or flood then our data will be lost in case of LRS but in GRS it will be there if secondary region 's data center is up and running (These regions are far away from each other). As we can see, the durability of GRS is higher than LRS. Because in GRS there is more redundancy, Price of GRS is also higher than LRS so there is a tradeoff between cost and data redundancy.

Once we have created our recovery service vault, We have to go to there and click on backup and after selecting options like types of workload we want to backup(These includes files and folder, SQL servers, Virtual machines etc.) and where is these workload running (This includes azure, Azure stack, On-premise) and after this we have click on configure and azure redirects us to appropriate solution. For example if we only want to backup files and folders then there is Microsoft azure recovery service agent(MARS), If we want to backup files and folders and SQL server then there is Microsoft azure backup server (MABS).MABS is basically superset of MARS. We can configure these solutions in our machine and after that configure our backup.

On the other hand Atera is a cloud based services automation software solution which is aimed at small to medium size IT companies. Atera includes features like remote monitoring and management, different third party integration like splash top, team viewer, screen connect for remote access, cloud backup solution etc.

To configure backup using Atera, First we have to download the backup agent of Atera. After installation, we have to provide username and password which we can find from the website of Atera and create a secret key which we'll need when we want to perform the restore operation.

2*) Approach*: I downloaded a sample database of 350 GB to compare these two solutions. After that I performed multiple backups and restores to compare the performance of these two solutions .Both of them use volume

`

shadow copy snapshot service (VSS) provided by Microsoft to create point in-time copy of the data which we want to backup. One of the most important features of VSS is that we can use it to take backup of data while the application is still running. VSS temporarily freezes all the write I/O operations to the application which data we want to backup. This application freeze time cannot exceed than 60 seconds otherwise this snapshot creation task will be aborted. In this 60 seconds also, Creation of snapshot can take at max 10 seconds otherwise it will be aborted. Generally, the creation of a snapshot only takes a few seconds. Other time is used for prerequisite checks, setting up the required environment for this task and once this operation is completed, Post operations happen to make everything back to normal and resume write I/O requests. All of these processes occur in less than 60 seconds and we get a consistent point in time snapshot of our data. Almost all the backup solutions which are available use this service to backup data.

When we perform backup for the first time, both of them take full backup. Once this full backup is done, All subsequent backups will be incremental which means that only data which is changed after the last backup will be sent because all other data is the same and is already backed up. Both of them have automated agents to perform backup tasks which means that once we configured them to take backups after specified intervals, they will do their job automatically. No user action is required. When I tried to perform backup operation of the sample database of size 350GB, in azure it took only 4 hours compared to 9 hours in Atera. Restore time is also very low in azure.

Pricing of solution is one of the most important factor that took part in decision making for business .For 500 GB data, azure costs us $32 compared to $179 in Atera (costs are given in per month basis) which means pricing in azure is 1/10th of Atera .Please that for using backup solution of Atera, we have to buy a whole pack starting from $79 per month which means that we will get another service apart from cloud backup like update management, remote monitoring and management etc. Later we found out from our clients that we don't require other services from Atera which means that for only our backup solution, Azure is better than Atera in both performance and pricing.

B) Developed a backup solution to take local backup of SQL anywhere server :

1) *Overview*: Our clients are using SQL anywhere 17 database to store and query their business data. Previously they were shutting down their SQL anywhere server daily to perform local backup of their server data. I was assigned a task to develop and implement a backup solution to take a local backup while the server is still running.

2) *Approach*: I created backup scripts for Microsoft SQL server at the start of this internship. After creating these scripts, I scheduled them using a job scheduler of the SQL server so I was familiar with the whole process but since I don't have any prior knowledge of the SQL anywhere server I have to acquire more knowledge about it. After some research, I found that SQL anywhere has a maintenance plan facility to configure and automate backups but while I was working with it, I ran into error. I first tried to resolve it on my own but failed to do so so I asked the question related to this in SQL anywhere forum. Some senior developer told me that the maintenance plan of SQL anywhere is not good enough to use for our backup solution and I only use it for understanding the process. I should develop my own backup batch file using different utilities provided by SQL anywhere and in that forum only I found an article in which the author already created the backup batch files for full backup, incremental backup and restore so now I just have to understand these scripts and modify them according to our needs. Author also posted scripts for rapid insertion of data in table so that we can verify that our backup scripts are working also when database is constantly updating .These scripts also generate log file so that we look at that to confirm that it's executed successfully. These scripts included different backup utilities of SQL anywhere. These utilities are nothing but executable programs. We have to provide appropriate inputs to these utilities and we will get the desired result as output. These scripts use " errorlevel" (environmental variable of command line) to keep track of output of these utilities .Errorlevel keeps return status of last executed file in batch file so if there is some error in executing any of these utilities then it's logged into the log file using errorlevel. We will check this variable every time after executing utilities. The utilities used in these script files are dbsrv (To start database server), dbspawn (To start a database server in background), dbbackup (To take backup), dbvalid (For validating backup once it's taken), dbstop (To stop server).

We have two different types of scripts for backup. First one is for full backup. We'll run it daily at a fixed time. It will take full backup of our database and clear its log file. Once the backup process is started, if there are any incoming write requests to the database, they will be written in the log file. Once this process is completed, the log file will be copied to the destination of the backup file. Once this backup process is completed, a backed up copy of the database will be started on the new server and the dbvalid utility will check that this new copy is valid. It checks that there is not any inconsistency in the database and it also validates indexes in the database. Once this process is completed it will stop that new server and our full backup script is executed successfully. Incremental backup scrip will only backup data which is changed after the last full backup or incremental backup. Incremental backup also clears the log which means that for our restore operation, we have to apply all these incremental backups to the database if we want to restore our database to till last incremental backup. In these two scripts, there is a difference only in input parameters for the dbbackup utility rest all the things are same. We have to take daily full backup of our data because if we only take incremental backup then when we want to do our restore operation then it'll take much higher time so it's advisable to take daily full backups and incremental backups in between them. For our solution, I decided to take 3 incremental backups between daily full.

After I created these scripts, I executed a batch script for rapid insertions which will insert data in the table and there is another script for displaying data in that table. That script will show the maximum value of the primary key of that table and number of rows in that table. Every time when the insertion script inserts a new row in the table, it will increase the value of the primary key by 1.By comparing both of the above two values, we can make sure that table has all the data which we entered. We will use this script once we restore our data from the backup file. I performed full backups and incremental backups of the database while the rapid insertion script was running and after every backup, I executed the display script so that I can know how many rows are inserted in the table when backup was taken. I have noted down values for the number of rows after each backup. After that I have performed restore operation on various different combinations of backups like restore for only the first full backup, third full backup, third full backup plus two incremental backups etc. After restoring these backups, I once again executed a display script on that database and I compared the values which I got from these scripts to the values I recorded earlier and they were the same. In this way I made sure that all scripts are working correctly and we can take a backup of our SQL anywhere database while it's running.

C) Explored Always on availability features of Microsoft SQL server and developed a solution for High availability (HA) and Disaster recovery (DR) of our SQL server :

1) *Overview*: There are two main availability features of the SQL server: 1) Always on availability group and 2) Always on failover cluster instance. We can combine both of these features to develop a solution for HA and DR.

1.1) An availability group (AG) allows us to create duplicate copies of user databases. In the availability group, we have a primary replica and we can have 1 to 8 secondary replicas based on our needs. When we are creating our availability group, we have to give all this information like which one is our primary server in which we have the primary database, which servers will be the secondary server, which databases we want to include in our AG etc. Once we have created our AG after that whatever changes we do to our primary databases of AG will be replicated to secondary databases. Primary database sends its log to the secondary database and then secondary databases use that log to make changes in the database. There are two modes to send this data from primary to secondary, first one is synchronous commit mod. In this mode the secondary database sends acknowledgement to the primary database that it has applied the received log to it and after getting this confirmation, the primary database will mark that transaction as committed which means that it waits for that acknowledgement. Second one is asynchronous commit mode. In this mode the primary database doesn't wait

`

for acknowledgement from the secondary database. Main difference between these two modes is that in synchronous commit mode , If we do failover from primary to secondary database, there won’t be any data loss because primary and secondary databases are in consistent state all the time but in asynchronous commit mode there will be some data loss.

When our primary database is down for whatever reason, we can failover to any secondary database. There are three ways in which we can do this failover,1) Automatic failover, 2) Planned manual failover and 3) forced failover. First two modes require synchronous commit mode for data synchronization and in these two modes, there won’t be any data loss after failover. If our data synchronization mode is asynchronous commit mode then we can do failover only by third mode which is forced failover in which there might be data loss after failover.

Another important component which AG provides is the listener. Listener allows client applications and end users to connect to SQL database without knowing which SQL instance is currently hosting the primary database.

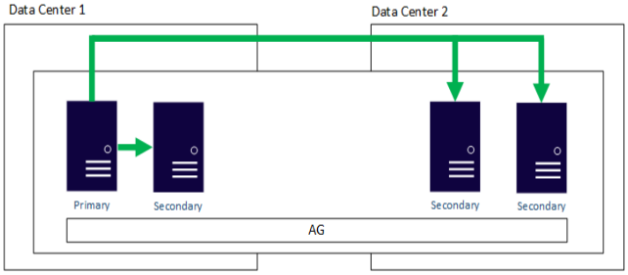

Fig. 1. Example for availability group

In the above figure, we can see a configuration for an AG. In this AG there are three secondary replicas. If data center 1 is down because of any unexpected event than we can failover to one of the secondary databases of data center 2 or if something happen to only primary of datacenter one then we can failover to secondary of data center 1 because failover in same data center will happen quickly than different data center. Apart from failover, Another benefit of AG is that we can configure secondary databases for read operations. This will reduce workload on the primary database and increase the overall performance of our system. If we have customers in a region in which data center 2 resides then we can configure its secondary databases to give information to them via readable secondary in data center 2. We can also perform backup operations via our secondary databases so it will further reduce load on primary.

AG gives us database level protection. If there is any server level information like SQL server agent jobs, logins at instance level, all this data needs to be taken care of manually.

1.2) Always on failover cluster instance (FCI) provides server level protection. FCI provides local high availability through redundancy at the server instance level. Suppose we have two server level nodes which we configured as FCIs. Only one of them will be up and running at a time. Other one which is not running act as a standby. They’ll share the same storage. If anything happens to the node which is up and running and it went down then the other node of FCI will start running and the user will not experience any down time.

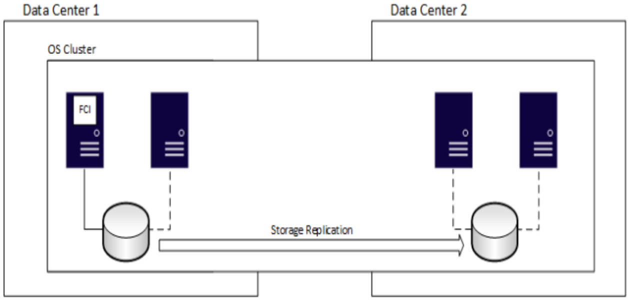

Fig. 2 Example of failover cluster instance

As shared storage is a bottleneck link for FCIs generally people replicate that shared storage to make sure that FCI will be up and running even if something happens to the shared storage as we can see in the above diagram. Another solution to this problem is that we can combine AG and FCI into a single solution to get very high availability and disaster recovery. We can configure different topologies using both AG and FCIs based on our requirements.

2) *Approach*: First, I looked at the official Microsoft document related to availability features. I also read from other sources and watched youtube videos to get more information. After reading all the details about AG and FCI, I discussed them with my mentor so that I can start creating a solution for our requirements. After the discussion, we decided that we only need AG for our scenario .We want both our master and failover server in the cloud. After this I started looking at implementation details for AG. First of all, I have to create a Windows server failover cluster (WSFC) which is a failover cluster facility provided by windows. AG uses this as an underlying architecture. I created this WSFC and after that I created an AG and configured a database in this AG. After creation I inserted various data in the primary database and by checking the last_harden_lsn number of primary and secondary databases (We can get this information in the primary server), I made sure that these changes are happening in secondary databases. We have used synchronous commit mode for data synchronization which means that we have to use forced manual failover as our failover mode.

Initially I created Virtual machines which we used for primary and secondary database in the same virtual network but later I found that we want our failover server in different regions which means I have to create another virtual network in that region and after that connect these two virtual networks so that data synchronization can happen. There are two ways to connect virtual networks in azure 1) Vnet peering and 2) VPN gateways both offer different features with different pricing. I researched about both of them and I found that vnet peering facilitates a low latency and high bandwidth connection which is appropriate for cross region disaster recovery solution so by using vnet peering I connected these two virtual networks and after that I once again created and configured AG and tested that. Finally I gave this as a final solution to the task which was assigned to me.

D) Design a dashboard in power-bi and integrate it with web application:

1) *Overview*: We have decided to use an azure recovery service agent to take the backup of our data. We want to show information related to backup like when last backup was taken, what was the end result of that operation and size of the data that was backed up last time so for this first we have fetched all the information related to backup from azure. For this I have to create an API which can give us all required data from azure. After that we can enter this API as a data source in our power-bi and show that data in our power-bi dashboard. Once that dashboard is created, we can integrate it into our web application.

2) *Approach*: First of all, I looked how can I get all details related to backup and recovery service vault (In which our backup data will be stored) so I fund that we can push logs related to recovery service vault in an azure service called log analytics and after we can query data from there but later sir told me that it is not required and there are APIs provided by azure to get all these information and log analytics was overkill for it. I found azure rest APIs which we can use to authentication our API requests to azure and get our required data from there so I started building my micro service which will authenticate my request to azure and once I’ll get bearer token after that I can use that for getting requests to get backup details for specific account. We have to register this in app registration in azure to give this application required permission so that it can authenticate against Azure and get required information related to backups. We will get clientid and client secret from app registration which we have to enter in the app.setting file in our API. We also have to provide tenant id and authority url which we will use for authentication. After the creation of this API, we have to publish it in Azure so that we can call it from power-bi. After I published it, I went to power-bi and entered it as a data source. Once I got required data in power-bi, I turned that into tables from data which was in Jason form. After that we have to publish it to a workspace. Once it’s published from power-bi dashboard to power service, we can create a dashboard from it. We just have to go to power-bi service and select our report. After that we have to click on ”pin this on dashboard” and our dashboard is ready for integration in our web application.

E) Some other small tasks which I have completed:

1)

1) I have created a batch file which copies data from source file or folder and paste it to destination folder. Our requirement was to run this script on a daily basis so after creating this we scheduled it with the windows task scheduler. I have created this script in a way that it will automatically delete the copy of source folder or file which was generated 10 days ago by this

`

script so in this way at any given time we will have copies of source file or folder of last 10 days.

2) *2)* I also got some basic knowledge about active directories. I have learned to create an on-premise active directory. I was assigned the task to find and compare different solutions to synchronize on-premise active directory and cloud active directory (Azure active directory in our case) for identity management. These solutions create a common user identity which will be used for authorization and authentication to all the resources (On-premise resource plus cloud resource). There are three different ways in which we can do this task,

2.1) **Password hash synchronization (PHS)**: PHS uses AD connect to extract hashes of passwords which are stored in on-premise active directory and it'll send these hashes securely to the cloud so whenever any cloud resource requires any authentication, they will authenticate hash of the password entered by user with these synchronize hash value s. AD connect will synchronize them per user basic and whenever any user changes his on-premise password, it'll sync that to hashes stored in the cloud so that it'll remain consistent.

2.2) **Pass through authentication (PTA)**: In this method, we have configured an authentication agent on an on-premise server in which our on-premise active directory resides. Whenever there is any access request for the resource for on-premise active directory in the cloud, it'll be redirected to that agent and this agent will authorize it.

2.3) **Active directory federation services (ADFS)**: In this method, we have a specific server related to federation service which will handle all of the authentication requests. Organizations with advanced requirements which are not satisfied through the first two methods will use this .For example, if some organization wants more control on their passwords and active directory objects will prefer to use this method.

All of the above methods are different from each other. We can choose it based on our requirements. Like if you don't want your on-premise passwords to synchronize in the cloud, you can use one of the last two methods because in the first method, we are sending hashes of hashes of passwords to the cloud which can be dangerous for us in future.

III. TOOLS AND TECHNOLOGIES

- Azure
- C#
- Power-bi
- Microsoft SQL server
- SQL Anywhere 17
- Rest APIs
- Visual Studio
- Microsoft SQL server management studio
- AD connect
- Postman

IV. LEARNING

- We had a daily scrum meeting in which we can discuss our problems and we'll get tasks for the next day. By attending these scrum meetings, I got so much information about how things should be done in a business environment.
- I learn about how to use online forums to get required information and how to ask questions in those forums and get help from unknown people whenever you feel stuck. I used online forums like Stack overflow, Microsoft forum, SQL anywhere forum for asking different questions and getting help from more experienced people.
- I was introduced to so many new technologies and features like Microsoft Azure, Always on availability features of SQL server, power-bi etc.
- I learned about how to work in a team environment and get help from my colleagues.
- I got to know about how the concepts which we have learned in class are used in real life scenarios. For example, password hash synchronization method which is used as hybrid identity management solution for on-premise active directory and azure active directory uses hashes of hashes of passwords instead of just plain passwords for authentication so in this way I got to know about how we can use hashes to secure very sensitive information.

ACKNOWLEDGMENT

I would like to thank Prof. Lavneet Singh (On-campus mentor) as well as Mr. Harjinder Chandi (Director of Junept India) for providing me this wonderful opportunity to work and acquire knowledge from a live project. Their constant support and guidance throughout this project was remarkable. Appart from them, I would like to thank all my team members and all other people who have helped me throughout this internship. The learning and experience which I had got from this internship will surely help me to grow as a software developer.

REFERENCES

[1] https://docs.microsoft.com/en-us/azure/backup/
[2] https://docs.microsoft.com/en-us/windows-server/storage/file-server/volume-shadow-copy-service
[3] http://dcx.sap.com/SQLa170/en/html/822e707dc8624445a615b7180321d900.html
[4] http://SQLanywhere.blogspot.com/2011/04/demonstrating-backup-validation-and.html
[5] https://docs.microsoft.com/en-us/SQL/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-SQL-server?view=SQL-server-ver15
[6] https://docs.microsoft.com/en-us/SQL/SQL-server/failover-clusters/windows/always-on-failover-cluster-instances-SQL-server?view=SQL-server-ver15
[7] https://azure.microsoft.com/en-in/blog/vnet-peering-and-vpn-gateways/
[8] https://docs.microsoft.com/en-us/rest/API/azure/
[9] https://docs.microsoft.com/en-us/power-bi/
[10] https://docs.microsoft.com/en-us/azure/active-directory/hybrid/