IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 5 - V7I5-1191)
Available online at: https://www.ijariit.com

Restoring data from one cloud to another cloud in case of disaster

Prashant Biradar
biradarpd19.comp@coep.ac.in
College of Engineering, Pune, Maharashtra

Abstract—Nowadays, a tremendous amount of information is produced, and it needs a data recovery management system so that the data can be recovered in case of a disaster. Different cloud service providers offer security to the users and clients despite the risk that their systems may go down due to disaster. So, there is a need to recover data services in the developing stage, which requires evolution and organized powerful data rescue methods if data is lost in a disaster. Recovery methods aim to support user's sensitive data that should be protected in a disaster. There are many reasons for it. Various procedures have been submitted in an environment like security breach (ransomware attack), water flood, sudden fire, earthquake, device mistake, or non-expected reason for deletion of information which might not be available permanently. Recovery aims to eliminate massive data recovery from one cloud platform to another cloud platform, which can be considered a part of cloud computing.

Keywords— *Component, Formatting, Style, Styling, Insert*

1. INTRODUCTION

Cloud Computing can be defined as storing and accessing data through the internet rather than self-computer system hardware peripherals and like webform process of organization combined with the distribution of resources. Cloud contains different servers where customers store their private data information with the help of the web, and they can receive their private data from anywhere. We define it as a real method of communication. Distributed computing systems are certainly known for the large registering scale today due to their volume to share with suitable assets. Moreover, many small businesses, the industry today, are based on the internet. Balance in business gives a robust demand of maximum large- scale business, and unpredictable interrupt may directly affect to industry, which causes massive financial loss, commercial reputation, and retail reputation. Many organizations might have solutions to detect disaster situations. The cause of disasters may be natural or artificial. It causes tremendous data loss. The User or organization has to protect their data from this type of attack.

2. LITERATURE SURVEY

R.V. Gandhi proposed a technique for taking back up of data and recovering. It was developed for Cloud computing such as Linux Box, Cold, and Hot backup methods, in which a large amount of data is converted into electronic format for metadata and needs data recovery services [1]. Ms. Kriti Sharma proposes the techniques used in cloud computing for powerful data backup and recovery, All these techniques capable of providing better performances under non-controlled conditions. In this, some techniques which can be cost increases gradually as data increases [2]. A parity cloud services study was done, which is reliable and straightforward and has less recovery cost. [3]. Shared Backup Router Resources (SBRR) is related to cost reduction, and it works even if there is router failure, discrepancy problem degrades the performance [4].

A comparison between the Amazon web services and Microsoft AZURE cloud providers through their services was made and which service gives the more cost-competitive solution based on applications that are needed to run processes [5]. A comparative study of Amazon Web Services (AWS) and Microsoft Azure, both cloud service providers, was done. which provides backup protection [5]. Kailas Podhale studied different data recovery and data techniques, and with their comparison, they explained the recovery time objective and recovery point objectives [6].

3. PROPOSED WORK

SOURCE ENV: In this project, we are taking AWS cloud service provider. We will create a virtual machine with data volumes attached to it. These volumes have to read by our staging env once a disaster gets trigger or happens.

STAGING ENV: In staging env, we are going to read AWS data volume by mounting it. To read the data volumes programmatically, we must convert them to AWS compatible data format. Once data is read from AWS data volumes, it must be converted to the azure supported data format to push the data in the Azure data disk. Staging env will play a crucial role in format conversion of data & migrating it from source to destination cloud vendor. Our futuristic approach is that staging env should be flexible enough to convert all different cloud provider's data formats, not limited to AWS and Azure.

TARGET DR ENV: This env is our target to restore our source DR data into our target env that is AZURE.

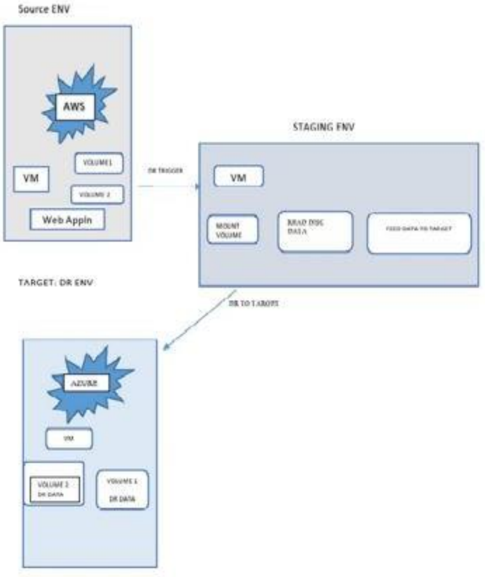


Fig. 1: Detail Diagram For Project Implementation

A. Work Flow of Restoring data from one cloud to another cloud in case of Disaster.

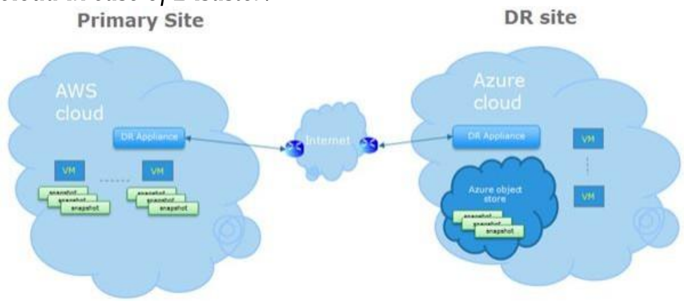


Fig. 2: Primary Site

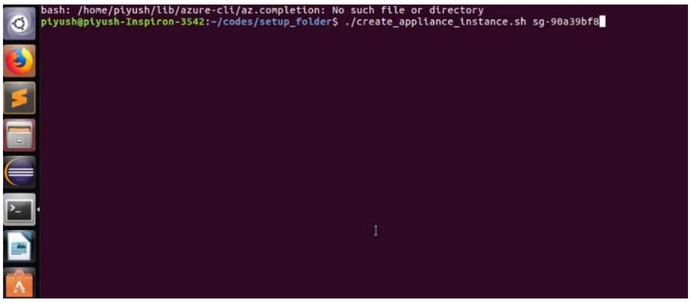
Primary Site:

- VMs with DR appliances are running in the AWS cloud.
- User will register the VM with the appliance for data protection and schedule/trigger the snapshot.
- DR appliance will take a snapshot of VM in AWS.
- Once the snapshot is ready, the DR appliance will copy the snapshot to the DR site, i.e., the Azure object-store.
- Sync the backup details with DR site appliance.

DR Site:

- Trigger DR process for backup VM through DR appliance
- Select the snapshot data for restore.
- Build the volume from a given VM snapshot.
- Create VM using created volume.
- Start the VM.

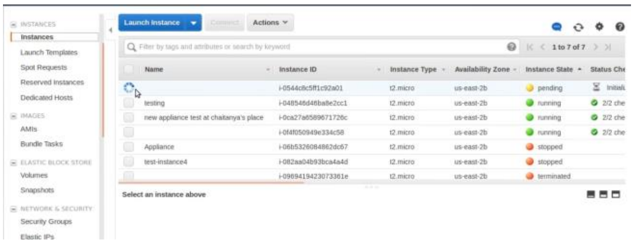
4. RESULTS



```
bash: /home/piyush/lib/azure-cli/az.completion: No such file or directory
piyush@piyush-Inspiron-3542:~/codes/setup_folder$ ./create_appliance_instance.sh sg-90a39bf8
```

Fig. 3: Creating VM on AWS

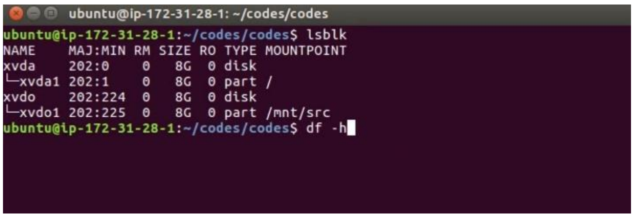
1) *Creating volume on AWS.:* Firstly, we have created a virtual machine in AWS. This VM contains the volume, which is to be stored in Amazon EBS storage. In this research, we are using our scripted instance on the virtual machine. We have written a script for creating an instance, and this instance has its own data volume stored in Amazon EBS. The script is run by using an API call 'Create Instance' in the command prompt. This creates a virtual machine in our AWS.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Che
	i-0544c8c5ff1c92a01	t2.micro	us-east-2b	pending	Initiali
testing	i-048546d46ba8e2cc1	t2.micro	us-east-2b	running	2/2 che
new appliance test at chaitanya's place	i-0ca27a6589671726c	t2.micro	us-east-2b	running	2/2 che
	i-0f4f050949e334c58	t2.micro	us-east-2b	running	2/2 che
Appliance	i-06b5326084862dc67	t2.micro	us-east-2b	stopped	
test-instance4	i-082aa04b93bca4a4d	t2.micro	us-east-2b	stopped	
	i-0969419423073361e	t2.micro	us-east-2b	terminated	

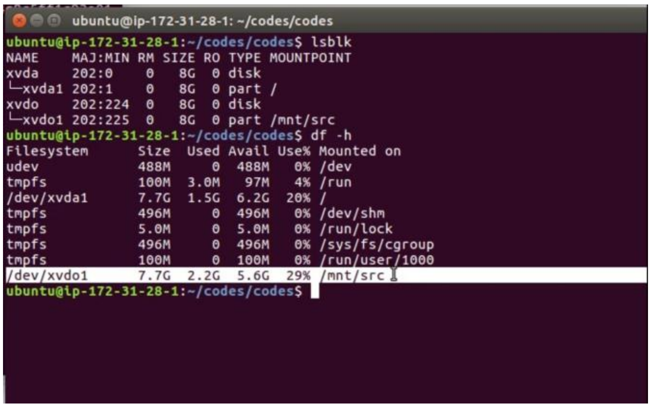
Fig. 4: Create VM

2) *Create VM on AWS:* Once we have executed our in- stance, the script creates an instance in AWS. We can check whether this scripted instance is running or not by AWS console. Once the instance is created successfully, then the instance data volume is also being created on AWS. This data volume which is created on AWS can be successfully restored to our azure.



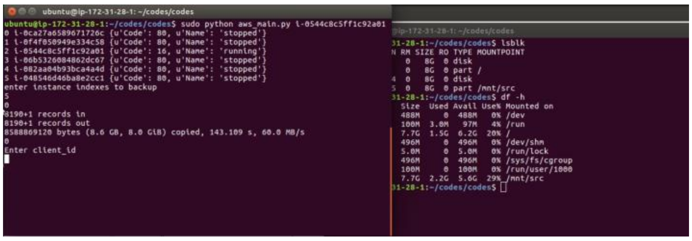
```
ubuntu@ip-172-31-28-1: ~/codes/codes
ubuntu@ip-172-31-28-1:~/codes/codes$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdo    202:224  0   8G  0 disk
└─xvdo1 202:225  0   8G  0 part /mnt/src
ubuntu@ip-172-31-28-1:~/codes/codes$ df -h
```

Fig. 5: Backing AWS VM EBS not mounted



```
ubuntu@ip-172-31-28-1: ~/codes/codes
ubuntu@ip-172-31-28-1:~/codes/codes$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdo    202:224  0   8G  0 disk
└─xvdo1 202:225  0   8G  0 part /mnt/src
ubuntu@ip-172-31-28-1:~/codes/codes$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            488M     0  488M   0% /dev
tmpfs           100M  3.0M   97M   4% /run
/dev/xvda1      7.7G  1.5G  6.2G  20% /
tmpfs           496M     0  496M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           496M     0  496M   0% /sys/fs/cgroup
tmpfs           100M     0  100M   0% /run/user/1000
/dev/xvdo1      7.7G  2.2G  5.6G  29% /mnt/src
ubuntu@ip-172-31-28-1:~/codes/codes$
```

Fig. 6: Backing AWS VM EBS mounted with data

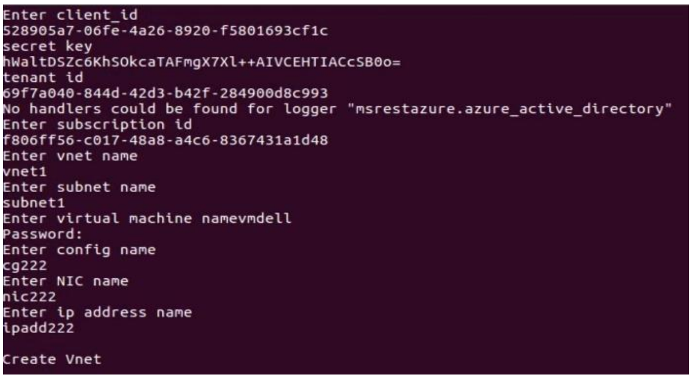


```
ubuntu@ip-172-31-28-1: ~/codes/codes
ubuntu@ip-172-31-28-1:~/codes/codes$ sudo python aws_main.py i-0544c8c5ff1c92a01
0 i-0ca27a6589671726c {u'Code': 80, u'Name': 'stopped'}
1 i-0f4f050949e334c58 {u'Code': 80, u'Name': 'stopped'}
2 i-0544c8c5ff1c92a01 {u'Code': 16, u'Name': 'running'}
3 i-06b5326084862dc67 {u'Code': 80, u'Name': 'stopped'}
4 i-082aa04b93bca4a4d {u'Code': 80, u'Name': 'stopped'}
5 i-048546d46ba8e2cc1 {u'Code': 80, u'Name': 'stopped'}
enter instance indexes to backup
5
0
8190+1 records in
8190+1 records out
8588869120 bytes (8.6 GB, 8.0 GiB) copied, 143.109 s, 60.0 MB/s
0
Enter client_id
```

```
@ip-172-31-28-1: ~/codes/codes
31-28-1:~/codes/codes$ lsblk
N RM SIZE RO TYPE MOUNTPOINT
   0   8G  0 disk
   0   8G  0 part /
4  0   8G  0 disk
5  0   8G  0 part /mnt/src
31-28-1:~/codes/codes$ df -h
 Size  Used Avail Use% Mounted on
 488M     0  488M   0% /dev
 100M  3.0M   97M   4% /run
 7.7G  1.5G  6.2G  20% /
 496M     0  496M   0% /dev/shm
 5.0M     0  5.0M   0% /run/lock
 496M     0  496M   0% /sys/fs/cgroup
 100M     0  100M   0% /run/user/1000
 7.7G  2.2G  5.6G  29% /mnt/src
31-28-1:~/codes/codes$
```

Fig. 7: Backing AWS VM EBS

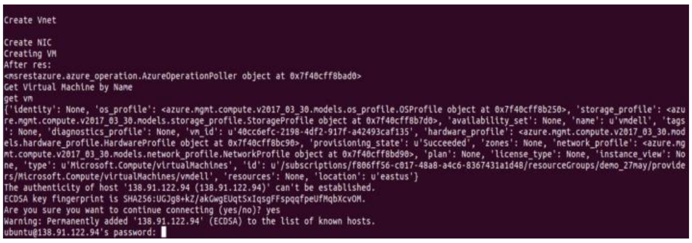
For our research, we restore the AWS data to azure. So we have taken a backup of AWS volume data. For taking the backup, we select the desired AWS data volume from AWS EBS by using its instance index. After taking the backup, we can restore it in azure. It is shown in fig.5,6 and 7



```
Enter client_id
528905a7-06fe-4a26-8920-f5801693cf1c
secret key
hWaltDSZc6KhSOkcaTAFmgX7Xl++AIVCEHTIACcSB0o=
tenant id
69f7a040-844d-42d3-b42f-284900d8c993
No handlers could be found for logger "msrestazure.azure_active_directory"
Enter subscription id
f806ff56-c017-48a8-a4c6-8367431a1d48
Enter vnet name
vnet1
Enter subnet name
subnet1
Enter virtual machine namevmdell
Password:
Enter config name
cg222
Enter NIC name
nic222
Enter ip address name
ipadd222

Create Vnet
```

Fig. 8: Restoring to azure

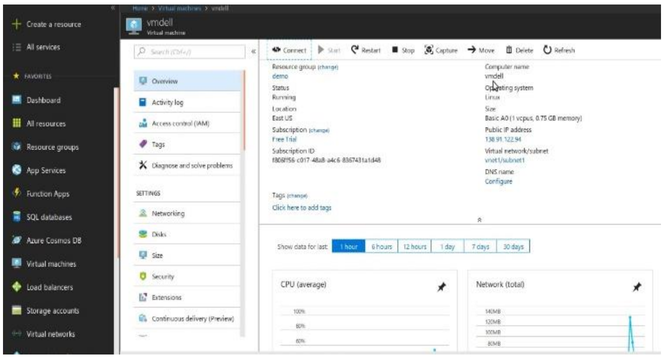


```
Create Vnet

Create NIC
Creating VM
After res:
<msrestazure.azure_operation.AzureOperationPoller object at 0x7f40cff8bad0>
Get Virtual Machine by Name
get vm
{'identity': None, 'os_profile': <azure.mgmt.compute.v2017_03_30.models.os_profile.OSProfile object at 0x7f40cff8b250>, 'storage_profile': <azu
re.mgmt.compute.v2017_03_30.models.storage_profile.StorageProfile object at 0x7f40cff8b7d0>, 'availability_set': None, 'name': u'vmdell', 'tags
': None, 'diagnostics_profile': None, 'vm_id': u'40cc6efc-2198-4df2-917f-a42493caf135', 'hardware_profile': <azure.mgmt.compute.v2017_03_30.mod
els.hardware_profile.HardwareProfile object at 0x7f40cff8bc90>, 'provisioning_state': u'Succeeded', 'zones': None, 'network_profile': <azure.mg
mt.compute.v2017_03_30.models.network_profile.NetworkProfile object at 0x7f40cff8bd90>, 'plan': None, 'license_type': None, 'instance_view': No
ne, 'type': u'Microsoft.Compute/virtualMachines', 'id': u'/subscriptions/f806ff56-c017-48a8-a4c6-8367431a1d48/resourceGroups/demo_27may/provide
rs/Microsoft.Compute/virtualMachines/vmdell', 'resources': None, 'location': u'eastus'}
The authenticity of host '138.91.122.94 (138.91.122.94)' can't be established.
ECDSA key fingerprint is SHA256:UGJg8+kZ/akGwgEUqtSxIqsgFFspqqfpeUfMqbXcvOM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '138.91.122.94' (ECDSA) to the list of known hosts.
ubuntu@138.91.122.94's password: █
```

Fig. 9: Restoring to azure progress

Now, as we have successfully backed up AWS data volume in AWS, we are restoring this backed up data volume to azure by connecting our azure to AWS using the rsync algorithm. Figure 9 shows the restoration process.

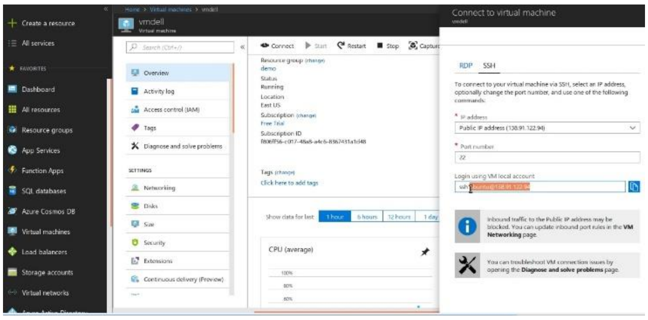


Resource group (change)
demo
Status
Running
Location
East US
Subscription (change)
Free Trial
Subscription ID
f806ff56-c017-48a8-a4c6-8367431a1d48
Tags (change)
Click here to add tags

Computer name
vmdell
Operating system
Linux
Size
Basic A0 (1 vcpus, 0.75 GB memory)
Public IP address
138.91.122.94
Virtual network/subnet
vnet1/subnet1
DNS name
Configure

Fig. 10: Restore done on azure

In the restoration process, the azure connect console shows both AWS and Azure's configurations. It describes the details of our source and target services. It is shown in fig. 11. We can see restore completion progress in Fig. 10.



Resource group (change)
demo
Status
Running
Location
East US
Subscription (change)
Free Trial
Subscription ID
f806ff56-c017-48a8-a4c6-8367431a1d48
Tags (change)
Click here to add tags

Connect to virtual machine
vmdell

To connect to your virtual machine via SSH, select an IP address, optionally change the port number, and use one of the following commands:

* IP address
Public IP address (138.91.122.94)

* Port number
22

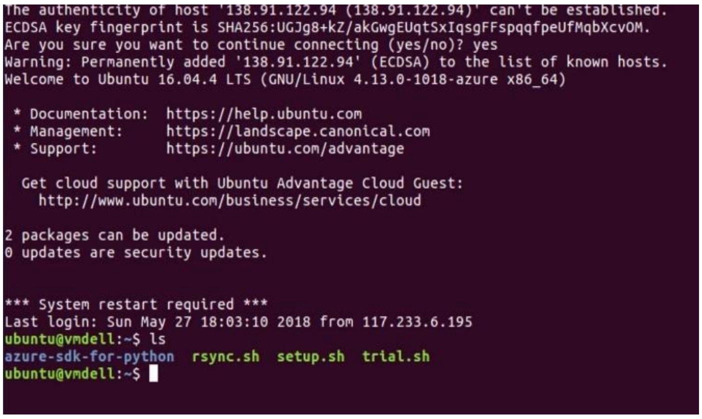
Login using VM local account
ssh ubuntu@138.91.122.94

Inbound traffic to the Public IP address may be blocked. You can update inbound port rules in the **VM Networking** page.

You can troubleshoot VM connection issues by opening the **Diagnose and solve problems** page.

Fig. 11: Connect restore done from azure

A connection between AWS with azure is to be established, so we have to use ssh to connect both services to establish this connection between these different services. Using the AWS IP, we can now connect to the azure account by SSH.



```
The authenticity of host '138.91.122.94 (138.91.122.94)' can't be established.
ECDSA key fingerprint is SHA256:UGJg8+kZ/akGwgEUqtSxIqsgFFspqqfpeUfMqbXcvOM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '138.91.122.94' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-1018-azure x86_64)

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

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

2 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Sun May 27 18:03:10 2018 from 117.233.6.195
ubuntu@vmdell:~$ ls
azure-sdk-for-python  rsync.sh  setup.sh  trial.sh
ubuntu@vmdell:~$ █
```

Fig. 12: Connect restored vm from azure

5. CONCLUSION

We successfully restored the AWS data volume in Azure. While migrating the backup from AWS to Azure, it must sync with the Resync algorithms continuously. If there will be any update in the AWS data, Resync will automatically update it in the Azure account. If anything, malicious or disastrous happens in AWS services, then the azure data is safe. If, unfortunately, there was a loss of data volume, it would be available on the Azure account. In this way, it can get critical or lost data from Azure.

REFERENCES

[1] RV Gandhi, M Seshaiah, A Srinivas, and C Reddi Neelima. Data back- up and recovery techniques for cloud server using seed block algorithm. *Gandhi et al. Int. Journal of Engineering Research and Applications*, 5(2 (Part 3)), 2015.

[2] Kruti Sharma and Kavita R Singh. Seed block algorithm: a remote smart data back-up technique for cloud computing. In *2013 International Conference on Communication Systems and Network Technologies*, pages 376–380. IEEE, 2013.

[3] Chi-won Song, Sungmin Park, Dong-wook Kim, and Sooyong Kang. Parity cloud service: a privacy-protected personal data recovery service. In *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 812–817. IEEE, 2011.

[4] Eleni Palkopoulou, Dominic A. Schupke, and Thomas Bauschert. Re- covery time analysis for the shared backup router resources (sbrr) architecture. In *2011 IEEE International Conference on Communications*, 2011.

[5] BV Rajeev, Vinod Baliga, et al. A comparative study of amazon web service and windows azure. *International Journal of Advanced Computer Research*, 3(3):80, 2013.

[6] Kailas Pophale, Priyanka Patil, Rahul Shelake, and Swapnil Sapkal. Seed block algorithm: remote smart data-backup technique for cloud computing. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(3):105–107, 2015.