

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Network Function Virtualization over Cloud-Cloud Computing as Business Continuity Solution

Wagdy Anis Aziz, Eduard Babulak and David Al-Dabass

Abstract

Cloud computing provides resources by using virtualization technology and a pay-as-you-go cost model. Network Functions Virtualization (NFV) is a concept, which promises to grant network operators the required flexibility to quickly develop and provision new network functions and services, which can be hosted in the cloud. However, cloud computing is subject to failures which emphasizes the need to address user's availability requirements. Availability refers to the cloud uptime and the cloud capability to operate continuously. Providing highly available services in cloud computing is essential for maintaining customer confidence and satisfaction and preventing revenue losses. Different techniques can be implemented to increase the system's availability and assure business continuity. This chapter covers cloud computing as business continuity solution and cloud service availability. This chapter also covers the causes of service unavailability and the impact due to service unavailability. Further, this chapter covers various ways to achieve the required cloud service availability.

Keywords: Cloud Computing, Business Continuity (BC), Disaster Recovery (DR), Network Functions Virtualization (NFV), Virtual Machine (VM), High Availability, Failover, Private Cloud, Hybrid Cloud, Public Cloud, Recovery Time Objective (RTO), Recovery Point Objective (RPO), IaaS, Paas, Saas, DRaaS

1. Introduction

Cloud computing service outage can seriously affect workloads of enterprise systems and consumer data and applications [1, 2]. Several (e.g. the VM failure of Heroku hosted on Amazon EC2 in 2011) or even human (human error or burglary). It can lead to significant financial losses or even endanger human lives [3]. Amazon cloud services unavailability resulted in data loss of many high-profile sites and serious business issues for hundreds of IT managers. Furthermore, according to the CRN reports, the 10 biggest cloud service failures of 2017, including IBM's cloud infrastructure failure on January 26, Facebook on February 24, Amazon Web Services on February 28, Microsoft Azure on March 16, Microsoft Office 365 on March 21 and etc., caused production data loss, and prevented customers from accessing their accounts, services, projects, and critical data for a very long duration. In addition, credibility of cloud providers took a hit because of these service failures [4].

The business continuity (BC) is important for service providers to deliver services to consumers in accordance with the SLAs. When building cloud

infrastructure, business continuity process must be defined to meet the availability requirement of their services. In a cloud environment, it is important that BC processes should support all the layers; physical, virtual, control, orchestration, and service to provide uninterrupted services to the consumers. The BC processes are automated through orchestration to reduce the manual intervention, for example if a service requires VM backup for every 6 hours, then backing up VM is scheduled automatically every 6 hours [5].

Disaster Recovery (DR) is the coordinated process of restoring IT infrastructure, including data that is required to support ongoing cloud services, after a natural or human-induced disaster occurs. The basic underlying concept of DR is to have a secondary data center or site (DR site) and at a pre-planned level of operational readiness when an outage happens at the primary data center. Expensive service disruption can result from disasters, both manmade and natural. To prevent failure in a Cloud Service Provider (CSPs) system, 2 different disaster recovery models (DR) have been proposed: the first being the Traditional model, and 2nd being cloud-based, – where the first is usable with both the dedicated and shared approach. The relevant model is chosen by customers using cost and speed as the determining factors. On the other hand, in the dedicated approach a customer is assigned an infrastructure leading to a higher speed and therefore cost. At the other end of the spectrum the shared model, often referred to as the Distributed Approach, multiple users are assigned a given infrastructure, which results in a cheaper outlay but leads to a lower recovery speed.

2. Background

2.1 Cloud computing

Cloud computing is storing, accessing, and managing huge data and software applications over the Internet. Access to data is protected by firewalls. Users are still using their computers to access the cloud-hosted data/applications. The difference lies in the fact that these data/applications use no or little the storage and compute resources of these computers since they are running in the cloud.

Cloud computing provides the context of offering virtualized computing resources and services in a shared and scalable environment through the network. A big percentage of global IT firms and governmental entities have incorporated cloud services for a multitude of purposes such as those related to mission-oriented applications and thus sensitive data. In order to provide full-support for these applications and their sensitive data, it is vital to include ample provision of environments that incorporate dependable cloud computing.

The National Institute of Standards and Technology (NIST) in SP 800-145, NIST specifies that a cloud infrastructure should have the five essential characteristics listed below:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

NIST also specifies three primary cloud deployment models:

- Public
- Private
- Hybrid

As well as three primary cloud service models:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

2.2 Essential characteristics

2.2.1 On-demand self-service

In cloud computing, users have the ability to provision any IT resource that they require on demand from a cloud, whenever they want. Self-service means that the consumers themselves carry out all the activities required to provision the cloud resource.

To enable on-demand self-service, a cloud provider maintains a self-service portal, which allows consumers to view and order cloud services. The cloud provider publishes a service catalog on the self-service portal. The service catalog lists items, such as service offerings, service prices, service functions, request processes, and so on.

2.2.2 Broad network access

Consumers access cloud services on any client/endpoint device from anywhere over a network, such as the Internet or an organization's private network.

2.2.3 Resource pooling

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and re-assigned according to consumer demand. Usually, end-users have no knowledge about the exact location of the resources they may want to access, but they may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of such resources include storage, processing, memory, and network bandwidth.

2.2.4 Rapid elasticity

Rapid elasticity refers to the ability for consumers to quickly request, receive, and later release as many resources as needed. The characteristic of rapid elasticity gives consumers the impression that unlimited IT resources can be provisioned at any given time. It enables consumers (in few minutes) to adapt to the variations in workloads by quickly and dynamically expanding (scaling outward) or reducing (scaling inward) IT resources, and to proportionately maintain the required performance level.

2.2.5 Measured service

A cloud infrastructure has a metering system that generates bills for the consumers based on the services used by them. The metering system continuously monitors resource usage per consumer and provides reports on resource utilization. For example, the metering system monitors utilization of processor time, network bandwidth, and storage capacity.

2.3 Cloud deployment models

A cloud deployment model specifies how a cloud infrastructure is built, managed, and accessed. Each cloud deployment model may be used for any of the cloud service models: IaaS, PaaS, and SaaS. The different deployment models present a number of tradeoffs in terms of control, scale, cost, and availability of resources.

2.3.1 Public cloud

A public cloud is a cloud infrastructure deployed by a provider to offer cloud services to the general public and/or organizations over the Internet. In the public cloud model, there may be multiple tenants (consumers) who share common cloud resources. A provider typically has default service levels for all consumers of the public cloud. The provider may migrate a consumer’s workload at any time and to any location. Some providers may optionally provide features that enable a consumer to configure their account with specific location restrictions. Public cloud services may be free, subscription-based or provided on a pay-per-use model. **Figure 1** below illustrates a generic public cloud that is available to enterprises and to individuals.

2.3.2 Private cloud

A cloud infrastructure that is configured for a particular organization’s sole use is termed private cloud. Departments and business units within an organization rely on network services implemented on a private cloud for dedicated to consumers. Generally, organizations are likely to avoid the adoption of public clouds as they are used by the public to access their facilities over the Internet. A private cloud offers organizations a greater degree of privacy, and control over the cloud infrastructure, applications, and data. There are two variants of a private cloud:

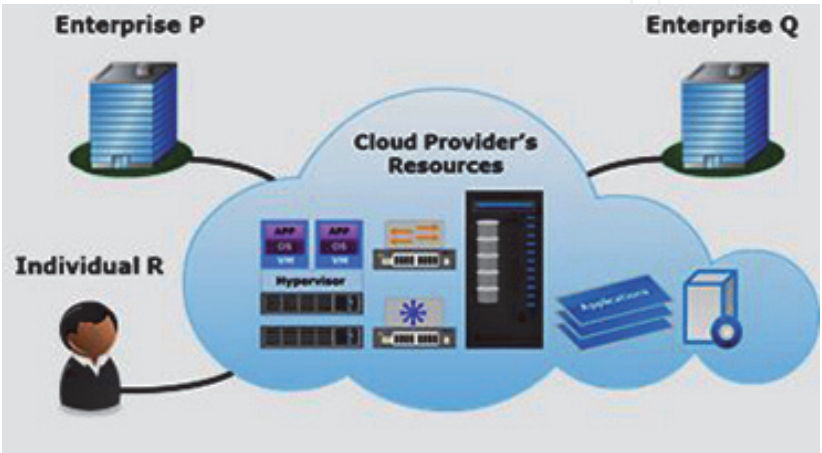


Figure 1.
A generic public cloud available to enterprises and individuals.

- On-premise: The on-premise private cloud, also known as an internal cloud, is hosted by an organization on its data centers within its own premises. The on-premise private cloud model enables an organization to have complete control over the infrastructure and data. In this model, the organization's IT department is typically the cloud service provider. In some cases, a private cloud may also span across multiple sites of an organization, with the sites interconnected via a secure network connection. **Figure 2** illustrates a private cloud of an enterprise that is available to itself.
- Externally hosted: In the externally hosted private cloud model, an organization outsources the implementation of the private cloud to an external cloud service provider. The cloud infrastructure is hosted on the premises of the external provider and not within the consumer organization's premises as shown in **Figure 3**. The provider manages the cloud infrastructure and facilitates an exclusive private cloud environment for the organization.

2.3.3 Hybrid cloud

The hybrid cloud infrastructure is a composition of two or more distinct cloud infrastructures (private or public).

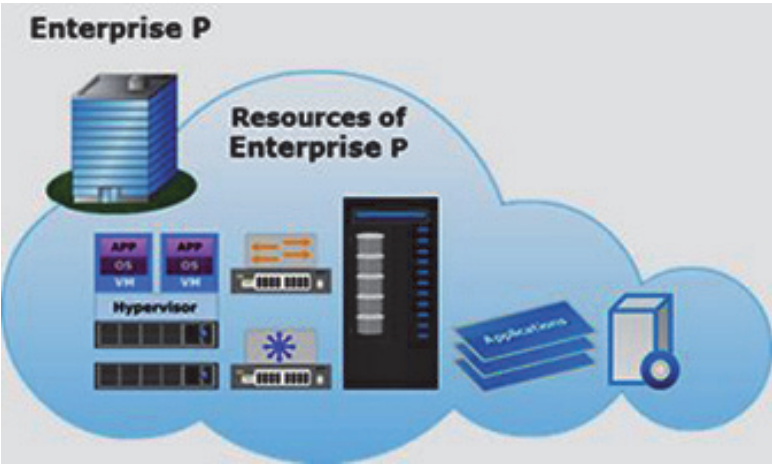


Figure 2.
An enterprise private cloud.

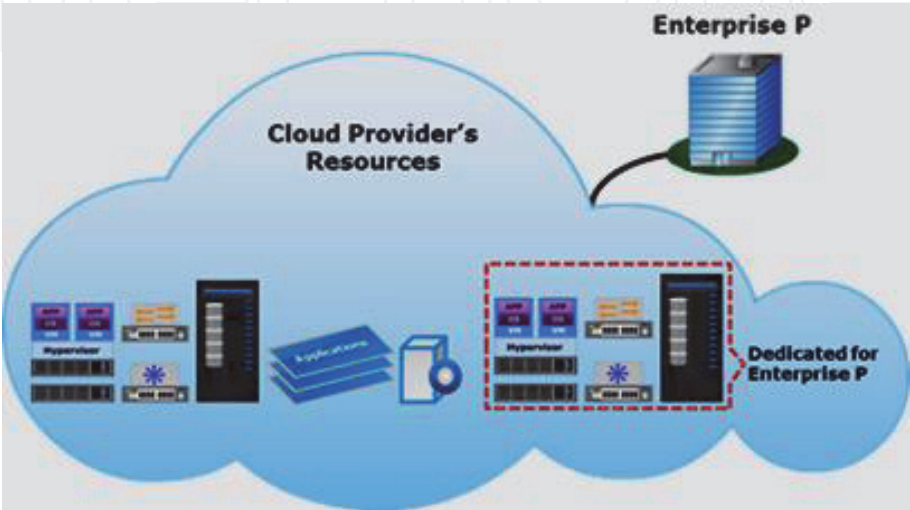


Figure 3.
An externally hosted private cloud.

In a hybrid cloud environment, the component clouds are combined with open or proprietary technology, interoperable standards, architectures, protocols, data formats; application programming interfaces (APIs), and so on. **Figure 4** illustrates a hybrid cloud that is composed of an on-premise private cloud deployed by an enterprise and a public cloud serving enterprise and individual consumers.

2.4 Cloud service models

2.4.1 Infrastructure-as-a-service (IaaS)

An end-user can access processing, storage, network resources hosted in a cloud infrastructure. The end-user can in particular run a multitude of software packages, which encompass a variety of applications as well as operating systems. The underlying infrastructure of the cloud is not managed or controlled by the consumer but he/she may have control over deployed applications, storage and operating systems, which may involve, for example in firewalls, a restricted use of specific components in the network.

IT resources such as storage capacity, network bandwidth and computing systems, may be hired by Consumers, for example in the IaaS model, from a Cloud Service Providers (CSP). The cloud service provider deploys and manages the underlying cloud infrastructure. While software, such as operating system (OS), database, and applications on the cloud resources, can be deployed and configured by Consumers.

In some organizations IaaS users are typically IT system administrators. In such cases, internal implementation of IaaS can even be carried out by the organization, with support given by the IaaS to its IT to manage the resources and services. In such examples the 2 options of Subscription-based or Resource-based (according to resource usage) can be implemented for IaaS pricing.

The IaaS provider pools the underlying IT resources and they are shared by multiple consumers through a multi-tenant model.

2.4.2 Platform-as-a-service (PaaS)

The capability provided to the consumer is the deployment of consumer-created or acquired applications created by means of programming languages, libraries, services, and tools supported by the PaaS provider. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage, but has control over the deployed applications and possibly the configuration settings for the application-hosting environment.

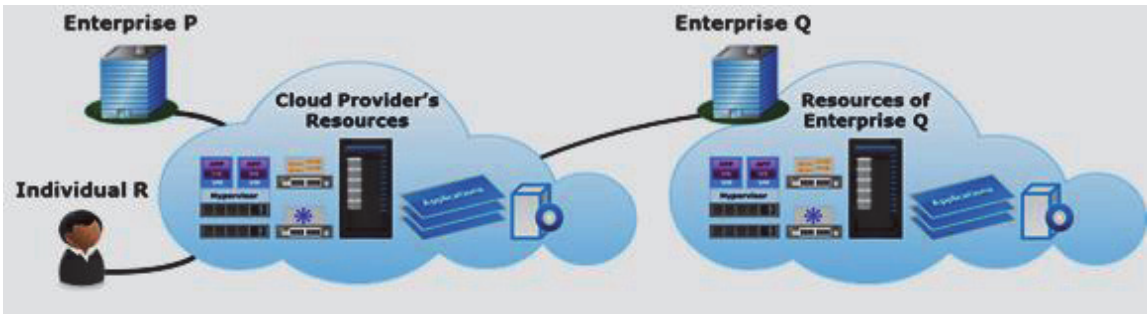


Figure 4.
A hybrid cloud composed of an on-premise private cloud and a public cloud.

2.4.3 Software-as-a-service (SaaS)

Within a SaaS context, the consumer can run SaaS provider’s applications in a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (for example, web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

In the SaaS model, a provider hosts an application centrally in the cloud and offers it to multiple consumers for use as a service. The consumers do not own or manage any aspect of the cloud infrastructure. In a SaaS context, a given version of an application, with a specific configuration (hardware and software) typically provides services to multiple consumers by partitioning their individual sessions and data. SaaS applications execute in the cloud and usually do not need installation on end-point devices. This enables a consumer to access the application on demand from any location and use it through a web browser on a variety of end-point devices. Some SaaS applications may require a client interface to be locally installed on an endpoint device. Customer Relationship Management (CRM), email, Enterprise Resource Planning (ERP), and office suites are examples of applications delivered through SaaS. **Figure 5** illustrates the three cloud service models.

2.4.4 Mobile “backend” as a service (MBaaS)

In the mobile “backend” as a service (m) model, also known as backend as a service (BaaS), web app and mobile app developers are provided with a way to link their applications to cloud storage and cloud computing services with application programming interfaces (APIs) exposed to their applications and custom software development kits (SDKs). Services include user management, push notifications, integration with social networking services [6] and more. This is a relatively recent model in cloud computing [7] with most BaaS startups dating from 2011 or later [8] but trends indicate that these services are gaining significant mainstream traction with enterprise consumers.

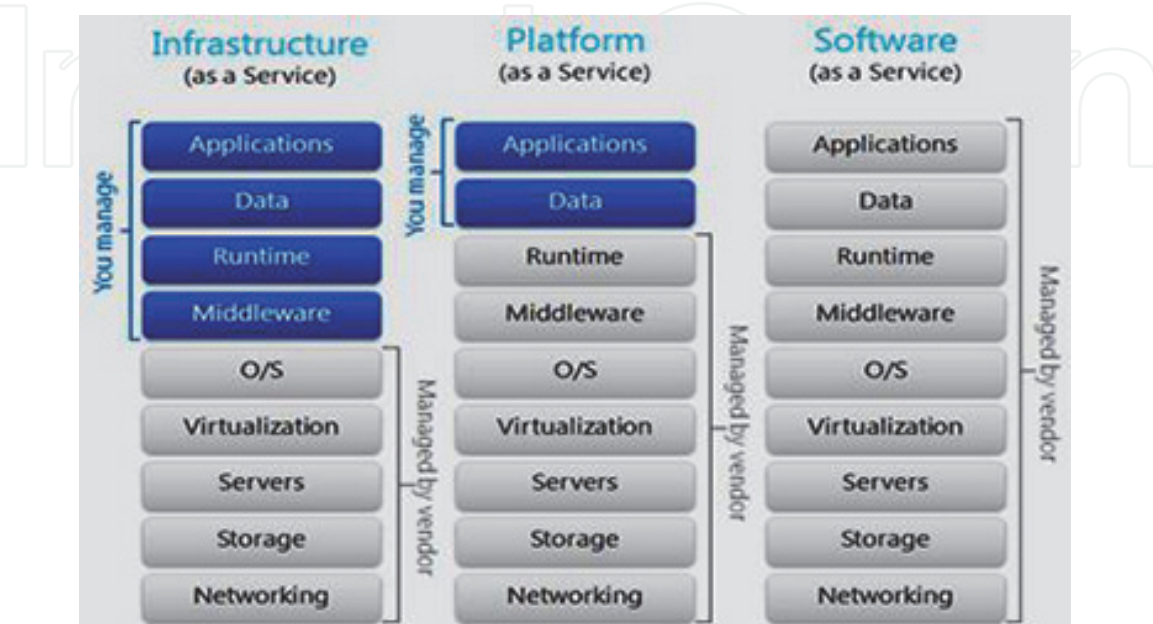


Figure 5.
Cloud service models.

2.4.5 Serverless computing

Serverless computing is a cloud computing code execution model in which the cloud provider fully manages starting and stopping virtual machines as necessary to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request, rather than per virtual machine, per hour [9]. Despite the name, it does not actually involve running code without servers [9]. Serverless computing is so named because the business or person that owns the system does not have to purchase, rent or provision servers or virtual machines for the back-end code to run on.

2.4.6 Function as a service (FaaS)

Function as a service (FaaS) is a service-hosted remote procedure call that leverages serverless computing to enable the deployment of individual functions in the cloud that run in response to events. FaaS is included under the broader term serverless computing, but the terms may also be used interchangeably [10].

2.5 Actors in cloud computing

Five roles in cloud environments are described in the NIST Cloud Computing Standards Roadmap document. These five participating actors are the cloud provider, the cloud consumer, the cloud broker, the cloud carrier and the cloud auditor. **Table 1** presents the definitions of these actors [4].

2.6 Network functions virtualization telco cloud

2.6.1 History

A group of network service providers at the Software-Defined networking (SDN) and OpenFlow World Congress in October 2012, originally presented the Network Functions Virtualization (NFV) concept (11). These service providers aimed to simplify and speed up the process of adding new network functions or applications.

Cloud Role	Definition (see also ref. [11])
Consumer	Any individual person or organization that has a business relationship with cloud providers and consumes available services.
Provider	Any individual entity or organization which is responsible for making services available and providing computing resources to cloud consumers
Broker	An IT entity that provides an entry for managing performance and QoS of cloud computing services. In addition, it helps cloud providers and consumers with management of service negotiations.
Auditor	A party that can provide an independent evaluation of cloud services provided by cloud providers in terms of performance, security and privacy impact, information system operations and etc. in the cloud environments.
Carrier	An intermediary party that provides access and connectivity to consumers through any access devices such as networks. Cloud carrier transports services from a cloud provider to cloud consumers.

Table 1.
The five actors in cloud computing environment.

2.6.2 Definition

NFV is set of techniques that virtualize network functions that are traditionally supported on proprietary, dedicated hardware, such as traffic forwarding or Evolved Packet Core (EPC) capabilities. With NFV, functions like routing, load balancing and firewalls are hosted in virtual machines (VM) or in OS containers. Virtualized Network Functions (VNF), are an essential component of the NFV architecture, as shown in the **Figure 6** below from ETSI NFV [12]. Multiple VNFs can be added to a standard server and can then be monitored and controlled by a hypervisor. ETSI NFV [12].

2.6.3 Features

Because NFV architecture virtualizes network functions that are thus supported on commodity hardware, network managers can add, move, or change network functions at the server level during a provisioning process. If a VNF running on a virtual machine requires more bandwidth, for example, the decision to scale or move a VNF is to be taken by NFV management and orchestration functions that can move the virtual machine to another physical server or provision another virtual machine on the original server to handle part of the load. Having this flexibility allows an IT department to respond in a more agile manner to changing business goals and network service demands.

2.6.4 Benefits

While NFV can benefit enterprises, service providers have a more immediate use case for it. Many see NFV’s potential to improve scalability and better utilize network resources. If a customer requests a new function, for example, NFV enables the service provider to add the said function by configuring it in a virtual machine without upgrading or buying new hardware.

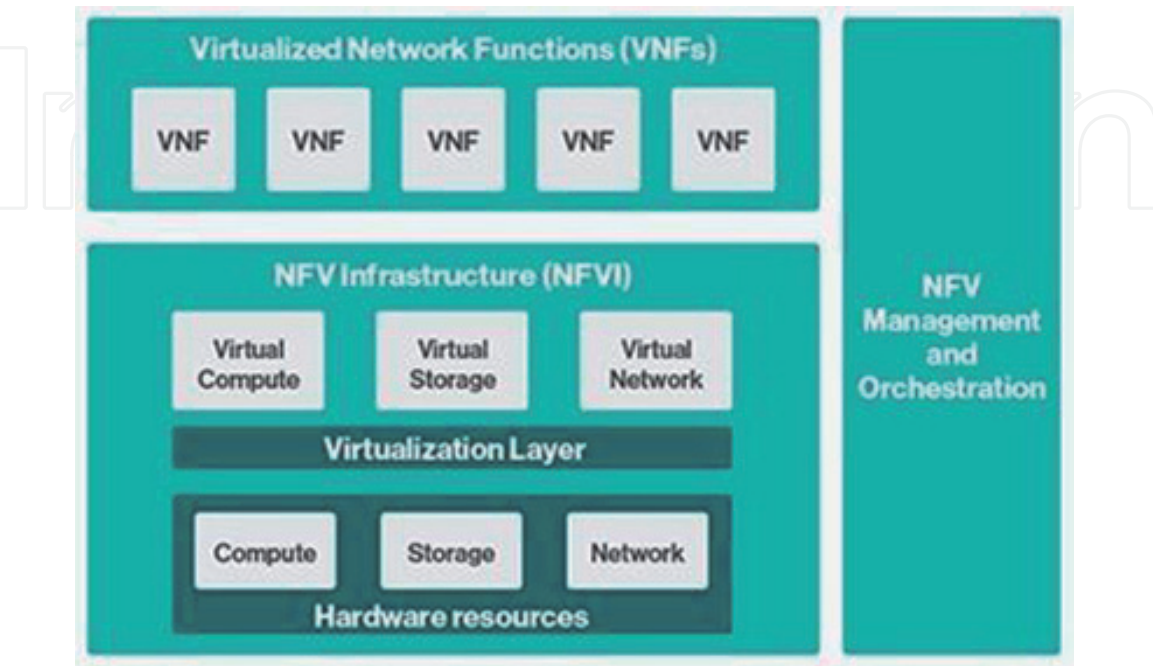


Figure 6.
Components of NFV architecture.

Basic NFV benefits also include reduced power consumption and increased the amiable physical space, since NFV eliminates the need for specific hardware appliances. NFV can then help reduce both operational and capital expenditures.

2.6.5 Cloud-native architecture is the foundation of 5G innovation

Through persistent effort and determination, Telecom operators are implementing a digital transformation to create a better digital world. To provide enterprises and individuals with a real time, on demand, all online experience requires an end-to-end (E2E) coordinated architecture featuring agile, automatic, and intelligent operation during each phase. The comprehensive cloud adaptation of networks, operation systems, and services is a prerequisite for this much-anticipated digital transformation.

In existing networks, operators have gradually used SDN and NFV to implement ICT network hardware virtualization, but retain a conventional operational model and software architecture. 5G networks require continuous innovation through cloud adoption to customize network functions and enable on-demand network definition and implementation and automatic O&M.

Physical networks are constructed based on DCs to pool hardware resources (including part of RAN and core network devices), which maximizes resource utilization.

The “All Cloud” strategy is an illuminated exploration into hardware resource pools, distributed software architecture, and automatic deployment. Operators transform networks using a network architecture based on data center (DC) in which all functions and service applications are running on the cloud DC, referred to as a Cloud-Native architecture.

In the 5G era, a single network infrastructure can meet diversified service requirements. Cloud-Native E2E network architecture has the following attributes:

- Provides logically independent network slicing on a single network infrastructure to meet diversified service requirements and provides DC-based cloud architecture to support various application scenarios.
- Uses CloudRAN to reconstruct radio access networks (RAN) to provide massive connections of multiple standards and implement on-demand deployment of RAN functions required by 5G.
- Simplifies core network architecture to implement on demand configuration of network functions through control and user plane separation, component-based functions, and unified database management.
- Implements automatic network slicing service generation, maintenance, and termination for various services to reduce operating expenses through agile network O&M.

3. Cloud computing for business continuity

Business continuity is a set of processes that includes all activities that a business must perform to mitigate the impact of service outage. BC entails preparing for, responding to, and recovering from a system outage that adversely affects business operations. It describes the processes and procedures a service provider establishes to ensure that essential functions can continue during and after a disaster. Business

continuity prevents interruption of mission-critical services, and reestablishes the impacted services as swiftly and smoothly as possible by using an automated process. BC involves proactive measures, such as business impact analysis, risk assessment, building resilient IT infrastructure, deploying data protection solutions (backup and replication). It also involves reactive countermeasures, such as disaster recovery, to be invoked in the event of a service failure. Disaster recovery (DR) is the coordinated process of restoring IT infrastructure, including data that is required to support ongoing cloud services, after a natural or human-induced disaster occurs. The basic underlying concept of DR is to have a secondary data center or site (DR site) and at a pre-planned level of operational readiness when an outage happens at the primary data center.

Cloud service availability refers to the ability of a cloud service to perform its agreed function according to business requirements and customer expectations during its operation. Cloud service providers need to design and build their infrastructure to maximize the availability of the service, while minimizing the impact of an outage on consumers. Cloud service availability depends primarily on the reliability of the cloud infrastructure (compute, storage, and network) components, business applications that are used to create cloud services, and the availability of data. The time between two outages, whether scheduled or unscheduled, is commonly referred as uptime, because the service is available during this time. Conversely, the time elapsed during an outage (from the moment a service becomes unavailable to the moment it is restored) is referred to as downtime. A simple mathematical expression of service availability is based on the agreed service time and the downtime.

$$\text{Service availability (\%)} = \frac{\text{Agreed service time} - \text{Downtime}}{\text{Agreed service time}} \quad (1)$$

Agreed service time is the period where the service is supposed to be available. For example, if a service is offered to a consumer from 9 am to 5 pm Monday to Friday, then the agreed service time would be $8 \times 5 = 40$ hours per week. If this service suffered 2 hours of downtime during that week, it would have an availability of 95%.

In a cloud environment, a service provider publishes the availability of a service in the SLA. For example, if the service is agreed to be available for 99.999 percent (also referred to as five 9 s availability) then the allowable service downtime per year is approximately 5 minutes. Therefore, it is important for the service provider to identify the causes of service failure, and analyze its impact to the business.

3.1 Causes of cloud service unavailability

This section listed some of the key causes of service unavailability. Data center failure is not the only cause of service failure. Poor application design or resource configuration errors can also lead to service outage. For example, if the web portal is down for some reason, then the services are inaccessible to the consumers, which leads to service unavailability. Even unavailability of data due to several factors (data corruption and human error) also leads to service unavailability. A cloud service might also cease to function due to an outage of the dependent services. Perhaps even more impactful on the availability are the outages that are required as a part of the normal course of doing business. The IT department is routinely required to take on activities such as refreshing the data center infrastructure, migration, running routine maintenance or even relocating to a new data center. Any of these activities can have its own significant and negative impact on service availability.

In general, the outages can be broadly categorized into planned and unplanned outages. Planned outages may include installation and maintenance of new hardware, software upgrades or patches, performing application and data restores, facility operations (renovation and construction), and migration. Unplanned outages include failure caused by human errors, database corruption, failure of physical and virtual components, and natural or human-made disasters.

3.2 Impact of cloud service unavailability

Cloud service unavailability or service outage results in loss of productivity, loss of revenue, poor financial performance, and damages to reputation. Loss of revenue includes direct loss, compensatory payments, future revenue loss, billing loss, and investment loss. Damages to reputations may result in a loss of confidence or credibility with customers, suppliers, financial markets, banks, and business partners. Other possible consequences of service outage include the cost of additional rented equipment, overtime, and extra shipping.

3.3 Methods to achieve required cloud service availability

With the aim of meeting the required service availability, the service provider should build a resilient cloud infrastructure. Building a resilient cloud infrastructure requires the following high availability solutions:

- Deploying redundancy at both the cloud infrastructure component level and the site (data center) level to avoid single point of failure
- Deploying data protection solutions such as backup and replication
- Implementing automated cloud service failover
- Architecting resilient cloud applications

For example when a disaster occurred at one of the service provider's data center, then BC triggers the DR process. This process typically involves both operational personnel and automated procedure in order to reactivate the service (application) at a functioning data center. This requires the transfer of application users, data, and services to the new data center. This involves the use of redundant infrastructure across different geographic locations, live migration, backup, and replication solutions.

4. Building fault tolerance cloud infrastructure

This section covers the key fault tolerance mechanisms at the cloud infrastructure component level and covers the concept of service availability zones. This section also covers automated service failover across zones along with zone configurations such as active/active and active/passive.

4.1 Single points of failure (SPOF)

Highly available infrastructures are typically configured without single points of failure as shown in **Figure 7** to ensure that individual component failures do not

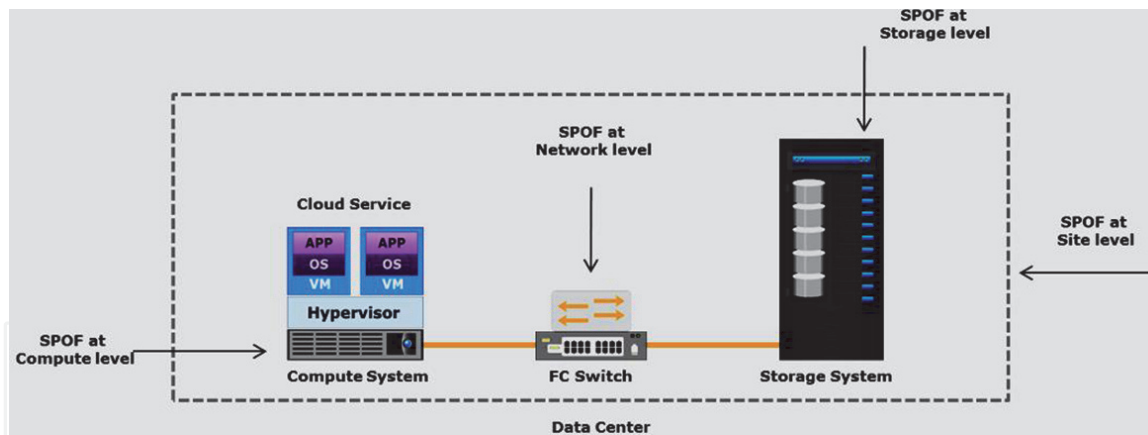


Figure 7.
 Single Point of Failure.

result in service outages. The general method to avoid single points of failure is to provide redundant components for each necessary resource, so that a service can continue with the available resource even if a component fails. Service provider may also create multiple service availability zones to avoid single points of failure at data center level.

Usually, each zone is isolated from others, so that the failure of one zone would not impact the other zones. It is also important to have high availability mechanisms that enable automated service failover within and across the zones in the event of component failure, data loss, or disaster.

$N + 1$ redundancy is a common form of fault tolerance mechanism that ensures service availability in the event of a component failure. A set of N components has at least one standby component. This is typically implemented as an active/passive arrangement, as the additional component does not actively participate in the service operations. The standby component is active only if any one of the active components fails. $N + 1$ redundancy with active/active component configuration is also available. In such cases, the standby component remains active in the service operation even if all other components are fully functional. For example, if active/active configuration is implemented at the site level, then a cloud service is fully deployed in both the sites. The load for this cloud service is balanced between the sites. If one of the site is down, the available site would manage the service operations and manage the workload.

4.2 Avoiding single points of failure

Single points of failure can be avoided by implementing fault tolerance mechanisms such as redundancy:

- Implement redundancy at component level: (i) Compute, (ii) Storage, (iii) Network.
- Implement multiple service availability zones: (i) Avoids single points of failure at data center (site) level, (ii) Enable service failover globally.

It is important to have high availability mechanisms that enable automated service failover.

4.3 Implementing redundancy at component level

The underlying cloud infrastructure components (compute, storage, and network) should be highly available and single points of failure at component level should be avoided. The example shown in **Figure 8** represents an infrastructure designed to mitigate the single points of failure at component level. Single points of failure at the compute level can be avoided by implementing redundant compute systems in a clustered configuration. Single points of failure at the network level can be avoided via path and node redundancy and various fault tolerance protocols. Multiple independent paths can be configured between nodes so that if a component along the main path fails, traffic is rerouted along another.

The key techniques for protecting storage from single points of failure are RAID, erasure coding techniques, dynamic disk sparing, and configuring redundant storage system components. Many storage systems also support redundant array independent nodes (RAIN) architecture to improve the fault tolerance. The following slides will discuss the various fault tolerance mechanisms as listed on the slide to avoid single points of failure at the component level.

4.4 Implementing multiple service availability zones

An important high availability design best practice in a cloud environment is to create service availability zones. A service availability zone is a location with its own set of resources and isolated from other zones to avoid that a failure in one zone will not impact other zones. A zone can be a part of a data center or may even be comprised of the whole data center. This provides redundant cloud computing facilities on which applications or services can be deployed. Service providers typically deploy multiple zones within a data center (to run multiple instances of a service), so that if one of the zone incurs outage due to some reasons, then the service can be failed over to the other zone. They also deploy multiple zones across geographically dispersed data centers (to run multiple instances of a service), so that the service can survive even if the failure is at the data center level. It is also important that there should be a mechanism that allows seamless (automated) failover of services running in one zone to another.

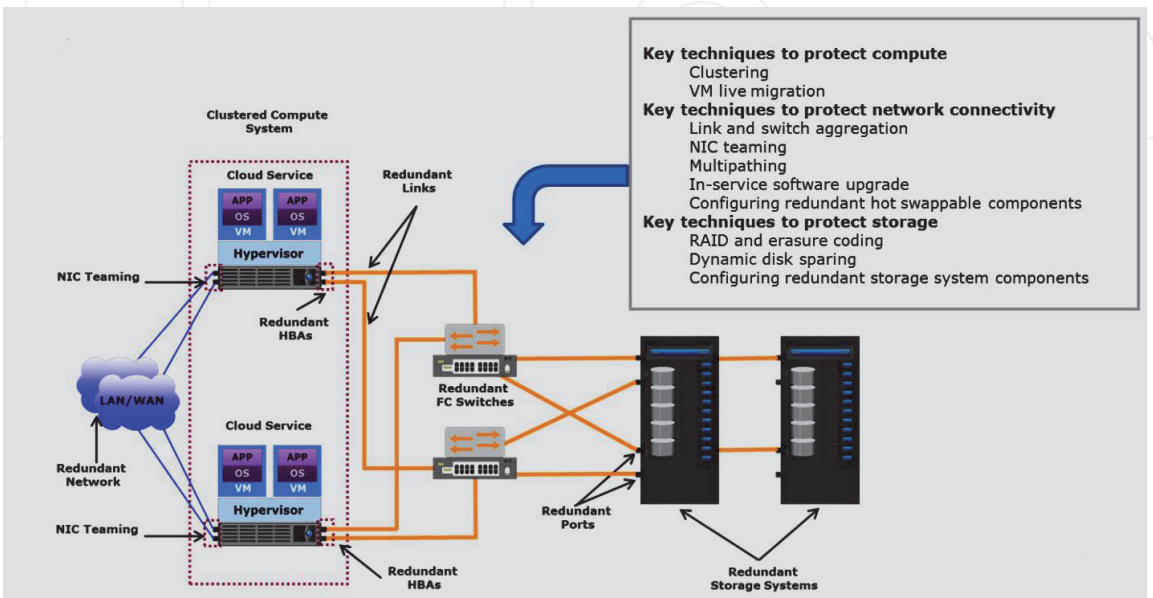


Figure 8.
Implementing Redundancy at Component Level.

To ensure robust and consistent failover in case of a failure, automated service failover capabilities are highly desirable to meet stringent service levels. This is because manual steps are often error prone and may take considerable time to implement. Automated failover also provides a reduced RTO when compared to the manual process. A failover process also depends upon other capabilities, including replication and live migration capabilities, and reliable network infrastructure between the zones. The following slides will demonstrate the active/passive and active/active zone configurations, where the zones are in different remote locations.

Figure 9 shows an example of active/passive zone configuration. In this scenario, all the traffic goes to the active zone (primary zone) only and the storage is replicated from the primary zone to the secondary zone. Typically in an active/passive deployment, only the primary zone has deployed cloud service applications. When a disaster occurs, the service is failed over to the secondary zone. The only requirement is to start the application instances in the secondary zone and the traffic is rerouted to this location.

In some active/passive implementation, both the primary and secondary zone have services running, however only the primary zone is actively handling requests from the consumers. If the primary zone goes down, the service is failed over to the secondary zone and all the requests are rerouted. This implementation provides faster restore of a service (very low RTO).

Figure 10 shows an example of implementing active/active configuration across data centers (zones), and the VMs running at both the zones collectively offer the same service. In this case, both the zones are active, running simultaneously, handling consumers requests and the storage is replicated between the zones. There should be a mechanism in place to synchronize the data between the two zones. If one of the zone fails, the service is failed over to the other active zone. The key point to be noted here is until the primary zone is restored, the secondary zone may have a sudden increase in workload.

So, it is important to initiate additional instances to handle the workload at secondary zone. The active/active design gives the fastest recovery time.

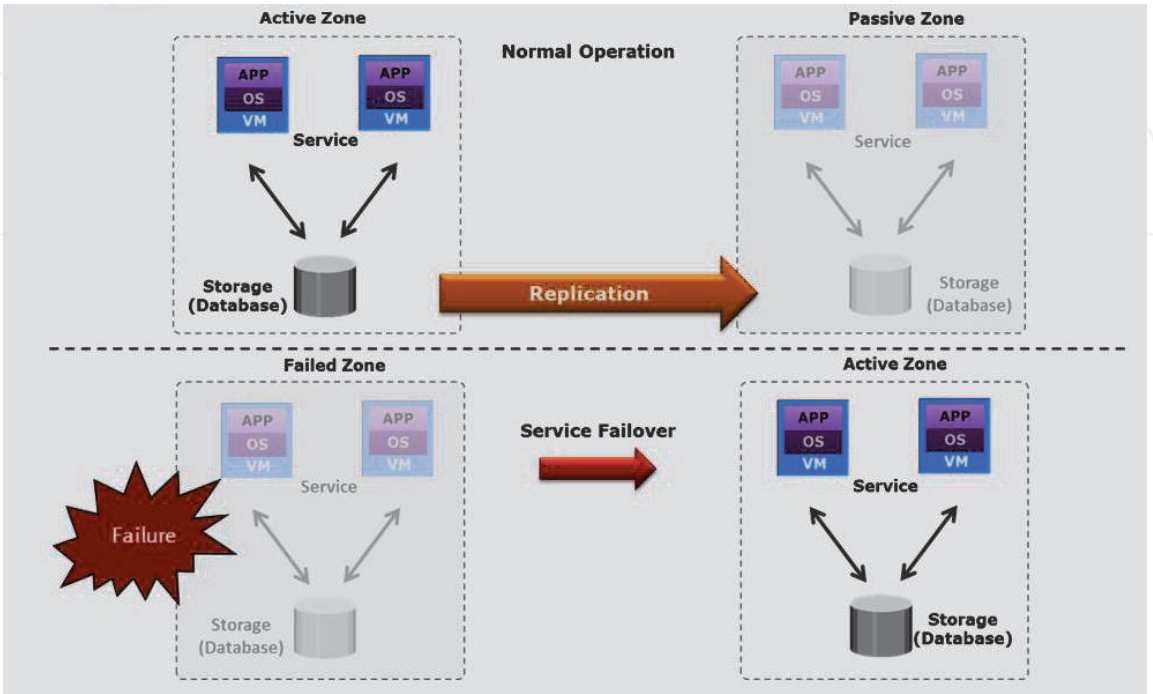


Figure 9.
Active/Passive Zone Configuration.

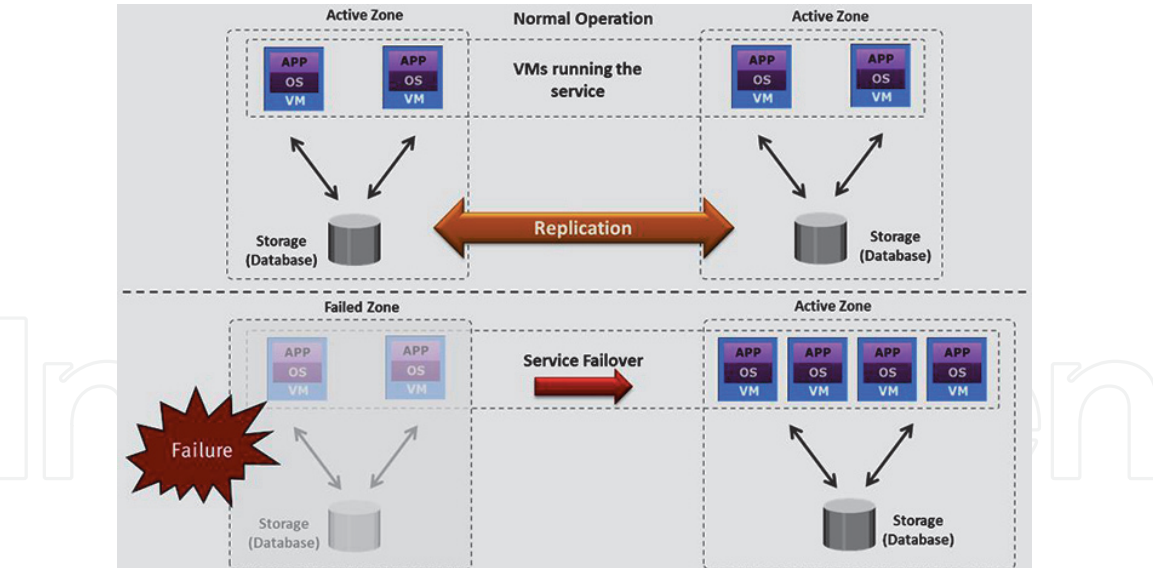


Figure 10.
Active/Active Zone Configuration.

The figure details the underlying techniques such as live migration of VMs using stretched cluster, which enables continues availability of service in the event of compute, storage, and zone (site) failure.

5. Data protection solution: backup

This section covers an introduction to backup and recovery as well as a review of the backup requirements in a cloud environment. This lesson also covers guest-level and image-level backup methods. Further, this section covers backup as a service, backup service deployment options, and deduplication for backup environment.

5.1 Data protection overview

Like protecting the infrastructure components (compute, storage, and network), it is also critical for organizations to protect the data by making copies of it so that it is available for restoring the service even if the original data is no longer available. Typically organizations implement data protection solution in order to protect the data from accidentally deleting files, application crashes, data corruption, and disaster. Data should be protected at local location and as well as to a remote location to ensure the availability of service. For example, when a service is failed over to other zone (data center), the data should be available at the destination in order to successfully failover the service to minimize the impact to the service.

One challenge to data protection that remains unchanged is determining the “right” amount of protection required for each data set. A “tiered approach” to data protection takes into account the importance of the data. Individual applications or services and associated data sets have different business values, require different data protection strategies. As a result, a well-executed data protection infrastructure should be implemented by a service provider to offer a choice of cost effective options to meet the various tiers of protection needed. In a tiered approach, data and applications (services) are allocated to categories (tiers) depending on their importance. For example, mission critical services are tier 1, important but less

time-critical services are tier 2, and non-critical services are tier 3. Using tiers, resources and data protection techniques can be applied more cost effectively to meet the more stringent requirements of critical services while less expensive approaches are used for the other tiers. The two key data protection solutions widely implemented are backup and replication.

5.2 Backup and recovery

A backup is an additional copy of production data, created and retained for the sole purpose of recovering the lost or corrupted data. With the growing business and the regulatory demands for data storage, retention, and availability, cloud service providers face the task of backing up an ever-increasing amount of data. This task becomes more challenging with the growth of data, reduced IT budgets, and less time available for taking backups. Moreover, service providers need fast backup and recovery of data to meet their service level agreements. The amount of data loss and downtime that a business can endure in terms of RPO and RTO are the primary considerations in selecting and implementing a specific backup strategy. RPO specifies the time interval between two backups. For example, if a service requires an RPO of 24 hours, the data need to be backed up every 24 hours. RTO relates to the time taken by the recovery process. To meet the defined RTO, the service provider should choose the appropriate backup media or backup target to minimize the recovery time. For example, a restore from tapes takes longer to complete than a restore from disks. Service providers need to evaluate the various backup methods along with their recovery considerations and retention requirements to implement a successful backup and recovery solution in a cloud environment.

5.3 Backup requirements in a cloud environment

In a cloud environment, applications typically run on VMs. Multiple VMs are hosted on single or clustered physical compute systems. The virtualized compute system environment is typically managed from a management server, which provides a centralized management console for managing the environment. The integration of backup application with the management server of virtualized environment is required. Advanced backup methods require the backup application to obtain a view of the virtualized environment and send configuration commands related to backup to the management server. The backup may be performed either file-by-file or as an image. Similarly, recovery requires either/both file level recovery and/or full VM recovery from the image. Cloud services have different availability requirement and that would affect the backup strategy. For example, if the consumer chose a higher backup service level (e.g. platinum) for their VM instances, then the backup would happen more frequently, have a lower RTO, and also have longer term retention when compared to lower-level service tiers. Typically, cloud environment has large volume of redundant data. Backing up of redundant data would significantly affect the backup window and increase the operating expenditure. Service provider needs to consider deduplication techniques to overcome these challenges. It is also important to ensure that most of the backup and recovery operations need to be automated.

5.4 Backup as a service

The unprecedented growth in data volume confronting today's IT organizations challenges not only IT management and budgets, but also all aspects of data

protection. The complexity of the data environment, exemplified by the proliferation and dynamism of virtual machines, constantly outpaces existing data backup plans. Deployment of a new backup solution takes weeks of planning, justification, procurement, and setup. Some organizations find the traditional on-premise backup approach inadequate to the challenge.

Service providers offer backup as a service that enables an organization to reduce its backup management overhead. It also enables the individual consumer to perform backup and recovery anytime, from anywhere, using a network connection. Backup as a service enables enterprises to procure backup services on-demand. Consumers do not need to invest in capital equipment in order to implement and manage their backup infrastructure. Many organizations' remote and branch offices have limited or no backup in place. Mobile workers represent a particular risk because of the increased possibility of lost or stolen machines. Backing up to cloud ensures regular and automated backups for these sites and workers who lack local IT staff, or who lack the time to perform and maintain regular backups.

5.5 Backup service deployment options

There are three common backup service deployment options that a cloud service providers offer to their consumers. These deployment options are:

- **Local backup service (Managed backup service):** This option is suitable when a cloud service provider is already providing some form of cloud services (example: compute services) to the consumers. The service provider may choose to offer backup services to the consumers, helping protect consumer's data that is being hosted in the cloud.
- **Replicated backup service:** This is an option where a consumer performs backup at their local site but does not want to either own or manage or incur the expense of a remote site for disaster recovery purposes. For such consumers, a cloud service provider offers replicated backup service that replicates backup data to a remote disaster recovery site.
- **Remote backup service:** In this option, consumers do not perform any backup at their local site. Instead, their data is transferred over a network to a backup infrastructure managed by the cloud service provider.

6. Data protection solution: replication

This section covers the replication and its types. This section also covers local replication methods such as snapshot and mirroring. This section further covers remote replication methods such as synchronous and asynchronous remote replications along with continuous data protection (CDP). Finally, this lesson covers a replication use case, Disaster Recovery as a Service (DRaaS).

6.1 Introduction to replication

It is necessary for cloud service providers to protect mission-critical data and minimize the risk of service disruption. If a local outage or disaster occurs, faster data and VM restore, and restart is essential to ensure business continuity. One of the ways to ensure BC is replication, which is the process of creating an exact copy (replica) of the data. These replica copies are used for restore and restart services if data loss occurs. Based on the availability requirements for the service being offered

to the consumer, the data can be replicated to one or more locations. Service provider should provide the option to consumers for choosing the location to which the data is to be replicated in order to comply with regulatory requirements. Replication can be classified into two major categories: local replication and remote replication. Local replication refers to replicating data within the same location. Local replicas help to restore the data in the event of data loss or enables to restart the application immediately to ensure BC. Snapshot and mirroring are the widely deployed local replication techniques. Remote replication refers to replicating data across multiple locations (locations can be geographically dispersed). Remote replication helps organizations to mitigate the risks associated with regional outages resulting from natural or human-made disasters. During disasters, the services can be moved to a remote location to ensure continuous business operation. In a remote replication, data can be synchronously or asynchronously replicated.

Replicas are immediately accessible by the application, but a backup copy must be restored by backup software to make it accessible to applications. Backup is always a point-in-time copy, but a replica can be a point-in-time copy or continuous. Backup is typically used for operational or disaster recovery but replicas can be used for recovery and restart. Replicas typically provide faster RTO compared to recovery from backup.

6.2 Local replication: snapshot

A snapshot is a virtual copy of a set of files, or volume as they appeared at a specific PIT. A snapshot can be created by using compute operating environment (hypervisor), or storage system operating environment. Typically the storage system operating environment takes snapshot at volume level, that may contain multiple VMs data and configuration files. This option does not provide an option to restore a VM in the volume. The most common snapshot technique implemented in a cloud environment is virtual machine snapshot. A virtual machine snapshot preserves the state and data of a virtual machine at a specific point-in-time. The VM state includes VM files, such as BIOS, VM configurations, and its power state (powered-on, powered-off, or suspended). This VM snapshot is useful for quick restore of a VM. For example, a cloud administrator can snapshot a VM, then make changes such as applying patches, and software upgrades. If anything goes wrong, administrator can simply restore the VM to its previous state using the previously created VM snapshot.

The hypervisor provides an option to create and manage multiple snapshots. When a VM snapshot is created, a child virtual disk (delta disk file) is created from the base image or parent virtual disk. The snapshot mechanism prevents the guest operating system from writing to the base image or parent virtual disk and instead directs all writes to the delta disk file. Successive snapshots generate a new child virtual disk from the previous child virtual disk in the chain. Snapshots hold only changed blocks. This VM snapshot can be used for creating image-based backup (discussed earlier) to offload the backup load from a hypervisor.

6.3 Local replication: mirroring

Mirroring as shown in **Figure 11** can be implemented within a storage system between volumes and between the storage systems. The example shown on the slide illustrates mirroring between volumes within a storage system. The replica is attached to the source and established as a mirror of the source. The data on the source is copied to the replica. New updates to the source are also updated on the replica. After all the data is copied and both the source and the replica contain

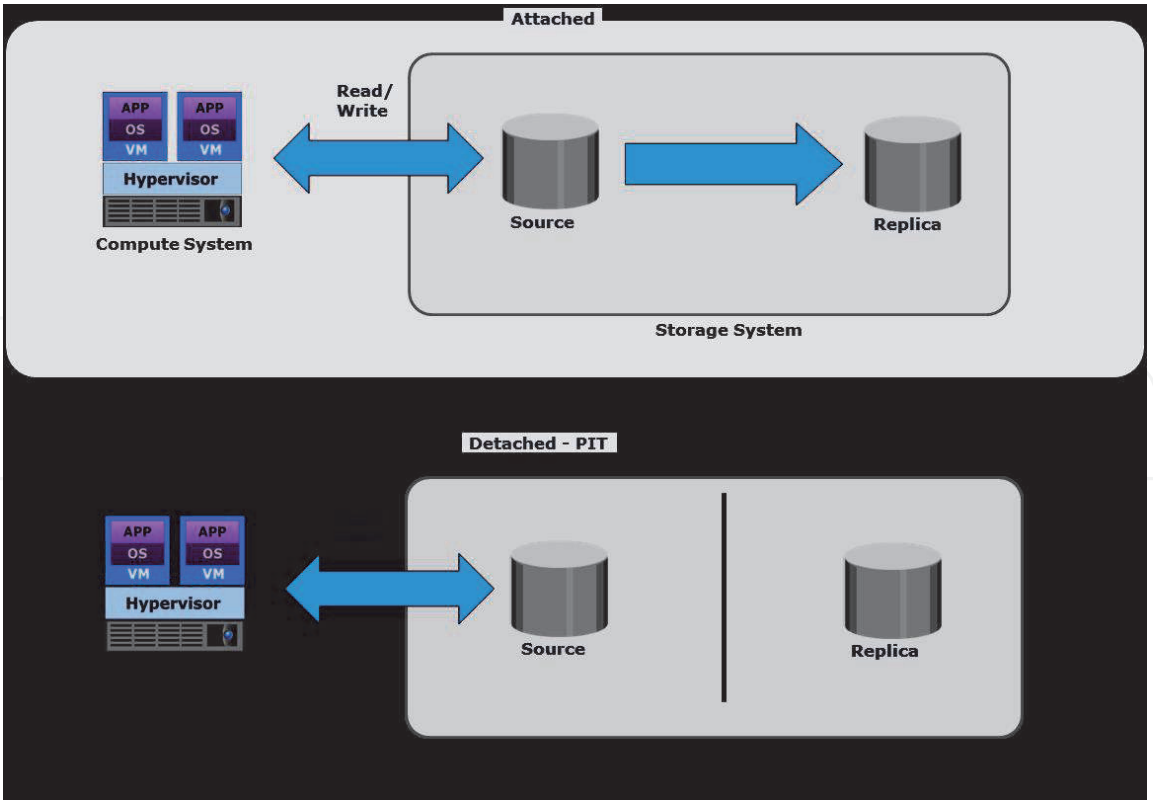


Figure 11.
Local Replication: Mirroring.

identical data, the replica can be considered as a mirror of the source. While the replica is attached to the source, it remains unavailable to any other compute system. However, the compute system continues to access the source. After the synchronization is complete, the replica can be detached from the source and made available for other business operations such as backup and testing. If the source volume is not available due to some reason, the replica enables to restart the service instance on it or restores the data to the source volume to make it available for operations.

6.4 Remote replication: synchronous

Synchronous replication as shown in **Figure 12** provides near zero RPO where the replica is identical to the source at all times.

In synchronous replication, writes must be committed to the source and the remote replica (or target) prior to acknowledging “write complete” to the compute

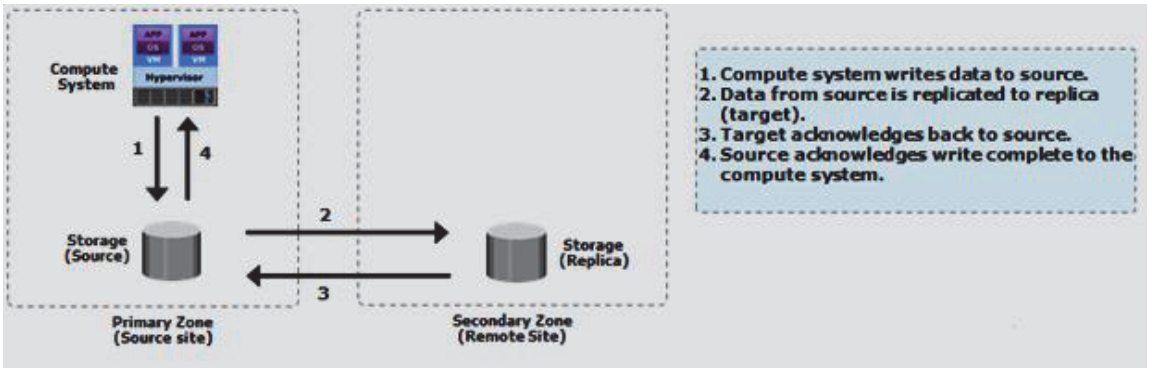


Figure 12.
Remote Replication: Synchronous.

system. Additional writes on the source cannot occur until each preceding write has been completed and acknowledged. This ensures that data is identical on the source and the replica at all times. Further, writes are transmitted to the remote site exactly in the order in which they are received at the source. Therefore, write ordering is maintained. The figure on the slide illustrates an example of synchronous remote replication. Data can be replicated synchronously across multiple sites. If the primary zone is unavailable due to disaster, then the service can be restarted immediately in other zone to meet the required SLA.

Application response time is increased with synchronous remote replication because writes must be committed on both the source and the target before sending the “write complete” acknowledgment to the compute system. The degree of impact on response time depends primarily on the distance and the network bandwidth between sites. If the bandwidth provided for synchronous remote replication is less than the maximum write workload, there will be times during the day when the response time might be excessively elongated, causing applications to time out. The distances over which synchronous replication can be deployed depend on the application’s capability to tolerate the extensions in response time. Typically synchronous remote replication is deployed for distances less than 200 KM (125 miles) between the two sites.

6.5 Remote replication: asynchronous

It is important for a service provider to replicate data across geographical locations in order to mitigate the risk involved during disaster. If the data is replicated (synchronously) between zones and the disaster strikes, then there would be a chance that both the zones may be impacted. This leads to data loss and service outage. Replicating data across zones which are 1000s of KM apart would help service provider to face any disaster. If a disaster strikes at one of the regions then the data would still be available in another region and the service could move to the location.

In asynchronous remote replication as shown in **Figure 13**, a write from a computer system is committed to the source and immediately acknowledged to the compute system.

It enables replication of data over distances of up to several thousand kilometers between the primary zone and the secondary zones (remote locations). Asynchronous replication also mitigates the impact to the application’s response time because the writes are acknowledged immediately to the compute system. In this case, the required bandwidth can be provisioned equal to or greater than the average write workload. Data can be buffered during times when the bandwidth is insufficient and moved later to the remote zones. Therefore, adequate buffer capacity should be provisioned. RPO depends on the size of the buffer, the available network

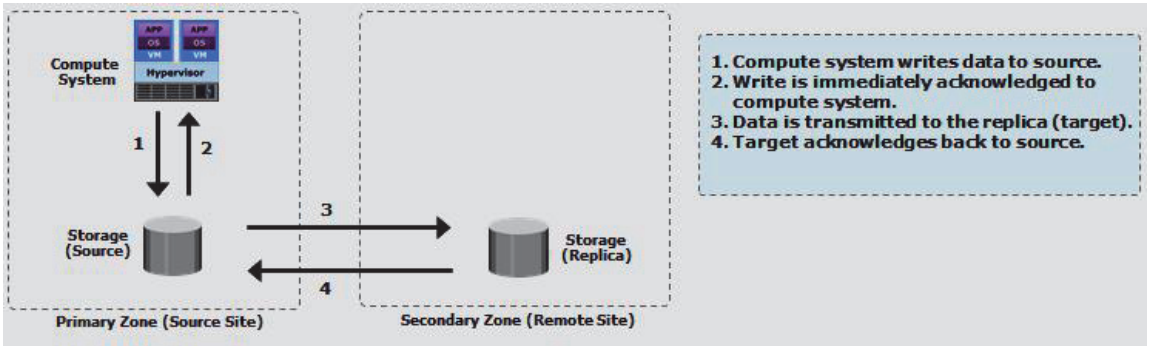


Figure 13.
Remote replication: asynchronous.

bandwidth, and the write workload to the source. Asynchronous replication implementations can take advantage of locality of reference (repeated writes to the same location). If the same location is written multiple times in the buffer prior to transmission to the remote zones, only the final version of the data is transmitted. This feature conserves link bandwidth.

6.6 Advanced replication solution: continuous data protection (CDP)

Mission-critical applications running on computer systems often require instant and unlimited data recovery points. Traditional data protection technologies offer a limited number of recovery points. If data loss occurs, the system can be rolled back only to the last available recovery point. Mirroring offers continuous replication; however, if logical corruption occurs to the production data, the error might propagate to the mirror, which makes the replica unusable. Ideally, CDP provides the ability to restore data to any previous point-in-time images (PIT). It enables this capability by tracking all the changes to the production devices and maintaining consistent point-in-time images. CDP enables one to perform operational recovery (protection against human errors, data corruption, and virus attacks) through local replication and disaster recovery through remote replication. CDP minimizes both RPO and RTO. In CDP, data changes are continuously captured and stored in a separate location from the primary storage. With CDP, recovery from data corruption poses no problem because it allows going back to any PIT image prior to the data corruption incident.

6.7 Replication use case: disaster recovery-as-a-service (DRaaS)

Facing an increased reliance on IT and the ever-present threat of natural or man-made disasters, organizations need to rely on business continuity processes to mitigate the impact of service disruptions. Traditional disaster recovery methods often require buying and maintaining a complete set of IT resources at secondary data centers that matches the business-critical systems at the primary data center. This includes sufficient storage to house a complete copy of all of the enterprise's business data by regularly replicating production data on the mirror systems at secondary site. This may be a complex process and expensive solution for a significant number of organizations.

Disaster Recovery-as-a-Service (DRaaS) has emerged as a solution to strengthen the portfolio of a cloud service provider, while offering a viable DR solution to consumer organizations. The cloud service provider assumes the responsibility for providing resources to enable organizations to continue running their IT services in the event of a disaster. From a consumer's perspective, having a DR site in the cloud reduces the need for data center space and IT infrastructure, which leads to significant cost reductions, and eliminates the need for upfront capital expenditure. Resources at the service provider can be dedicated to the consumer or they can be shared. The service provider should design, implement, and document a DRaaS solution specific to the customer's infrastructure. They must conduct an initial recovery test with the consumer to validate complete understanding of the requirements and documentation of the correct, expected recovery procedures.

For enterprises, the main goal of DR is business continuity, which implies the ability to resume services after a disruption. The Recovery Time Objective (RTO) and Recovery Point Objective (RPO) are two important parameters that all recovery mechanisms to improve. RTO is the time duration between disruption until the service is restored, and RPO denotes the maximum amount of tolerable data loss

that can be afforded after a disaster. By minimizing RTO and RPO, business continuity can be achieved.

Failover delays consist of five steps depending on the level of backup [13]:

- S1: Hardware setup
- S2: OS initiation time
- S3: Application initiation time
- S4: Data/process state restoration time
- S5: IP forwarding time

Therefore, RPO and RTO can be defined as:

The Recovery Time Objective (RTO)

The Recovery Point Objective (RPO)

$$PRO \propto \frac{1}{Fb} \tag{2}$$

Where Fb is Frequency of backup

$$RTO = fraction\ of\ PRO + \sum_{S1}^{S5} T_j \tag{3}$$

During normal production operations as shown in **Figure 14**, IT services run at the consumer’s production data center. Replication of data occurs from the consumer production environment to the service provider’s location over the network. Typically when replication occurs, the data is encrypted and compressed at the production environment to improve the security of data and reduce the network bandwidth requirements. Typically during normal operating conditions, a DRaaS implementation may only need a small share of resources to synchronize the application data and VM configurations from the consumer’s site to the cloud. The full set of resources required to run the application in the cloud is consumed only if a disaster occurs.

In the event of a business disruption or disaster as shown in **Figure 15**, the business operations will failover to the provider’s infrastructure as shown in the figure on the slide. In such a case, users at the consumer organization are redirected to the cloud.

For applications or groups of applications that require restart in a specific order, a sequence is worked out during the initial cloud setup for the consumer and recorded in the disaster recovery plan. Typically VMs are allocated from a pool of compute resources located in the provider’s location.

Returning business operations back to the consumer’s production environment is referred to as failback. This requires replicating the updated data from the cloud repository back to in-house production systems before resuming the normal

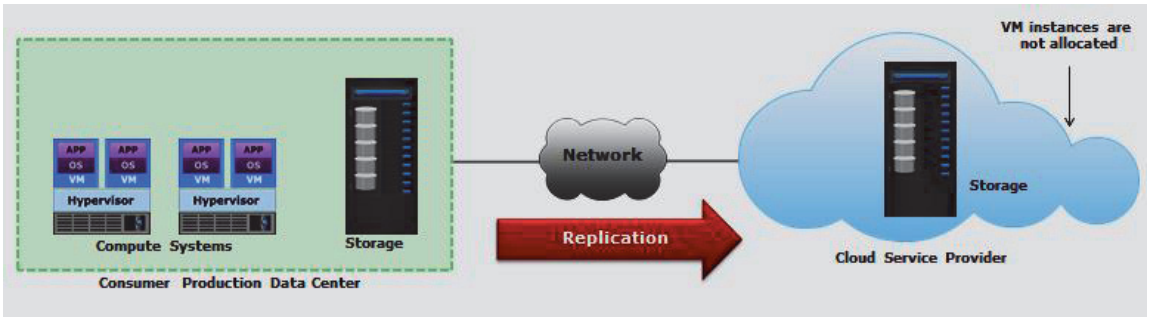


Figure 14.
DRaaS – Normal Production Operation.

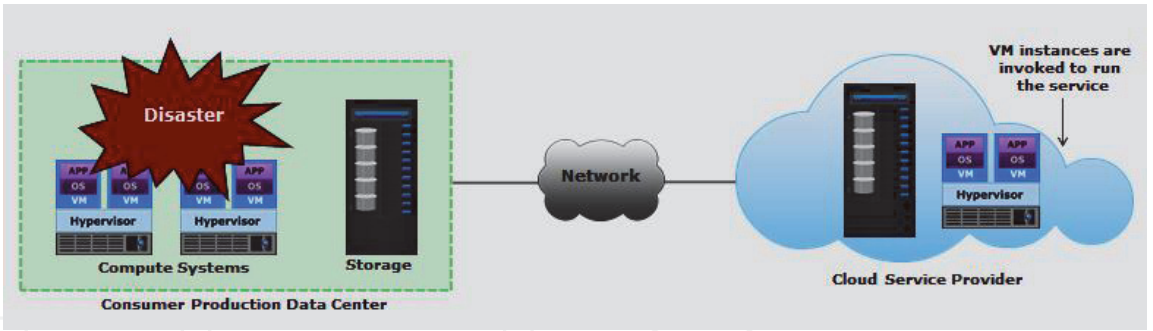


Figure 15.
DRaaS – Business Disruption.

business operations at consumer’s location. After starting the business operations at the consumer’s infrastructure, replication to the cloud is re-established. To offer DRaaS, the service provider should have all the necessary resources and technologies to meet the required service level.

Cloud-based DR solutions are attractive because of their ability to tolerate disasters and to achieve reliability and availability goals. Such solutions can be even more useful in small and medium enterprises (SMEs) environments, because the latter does not have many resources (unlike large companies). As shown in **Table 2**, data level, system level, and application level are three DR levels, which are defined in terms of system requirements [14].

The performance requirements that have to be met by DR are to minimize RPO and RTO. There are different DR approaches to develop a recovery plan. All these approaches are based on redundancy and backup strategies if resources are systematically reserved in the backup. The redundancy strategy relies upon distinct sites that have the ability to start up the applications after a disaster; whereas backup strategy relies upon replication technology [15]. The speed and protection degree of these approaches depend on the level of DR services that is shown in **Table 3** [16].

The objective of disaster recovery planning is to minimize RTO, RPO, cost, and application latency by considering system constraints such as CPU, network, and

DR Level	Description
Data level	Security of application data
System level	Reducing recovery time as short as possible
Application level	Application continuity

Table 2.
DR levels.

Model	Synchronize Time	Recovery Time	Backup Characteristics	Tolerance support
Hot	Seconds	Minutes	Physical Mirroring	Very High
Modified Hot	Minutes	1 Hour	Virtual Mirroring	High
Warm	Hours	1 to 24 Hours	Limited Physical Mirroring	Moderate
Cold	Days	More than 24 Hours	Off-site backup	Limited

Table 3.
Cloud-based DR models.

storage requirements. DR recovery planning can be considered as an optimization problem. DR plans include at least two phases:

- Matching Phase: In this phase, several candidate DR solutions are matched against the requirements to minimize RPO and RTO) of any data container, (a data container means a data set with identical DR requirements).
- Plan composition phase: Selecting an optimal DR solution which can minimize cost with respect to required latency for each data container.

6.8 Disaster recovery-as-a-service (DRaaS) challenges

In this section, we investigate some common DRaaS challenges in cloud environments.

6.8.1 Cost

One of the main factors to choose cloud as a DRaaS is its price. CSPs always seek cheaper ways to provide recovery mechanisms by minimizing different costs.

6.8.2 Replication latency

DRaaS depends on the replication process to create backups. Current replication strategies are divided into two categories: synchronous and asynchronous. However, they both have certain advantages and disadvantages. Synchronized replication, guarantees excellent RPO and RTO, but is dear and might affect system performance because of over-optimization. This problem is incredibly bad for multi-tier web applications because it can greatly increase the trip Time (RRR) between the first and backup sites. On the choice hand, the backup model adopted with asynchronous replication is cheaper and also the system has fewer problems, but the standard of the DRaaS is reduced. Thus, the transaction between costs, the performance of the system and replication latency is an undeniable challenge in cloud disaster solutions.

6.8.3 Data storage

Business database storage is one of the problems of enterprises that can be solved by cloud services. By increasing cloud usage in business and market, enterprises need to store huge amounts of data on cloud-based storage. Instead of conventional data storage devices, cloud storage service can save money and is more flexible. To satisfy applications and to guarantee the security of data, computing has to be distributed but storage has to be centralized. Therefore, storage a single point of failure and data loss are critical challenges to store data in cloud service providers [17].

6.8.4 Lack of redundancy

When a disaster happens, the primary site becomes unavailable and the backup site has to be activated to replace the primary site. It is a serious threat to the system. This issue is temporary and will be removed once the primary site is recovered.

6.8.5 Failure

The early detection failure time significantly affects the recovery time of the system, so it is important to detect and report rapid and accurate DRaaS failure. On the other hand, in many backup sites there is a big question: How to separate network failures and service interruptions.

6.8.6 Security

As mentioned earlier, disaster can be created by nature or can be man-made. The cyber-terrorist attack is one of the most man-made disasters that can occur for a variety of reasons. In this case, protecting and restoring important data will be the focus of the DRaaS programs other than system restoration [18].

6.9 Disaster recovery-as-a-service (DRaaS) solutions

In this section we discuss some DRaaS solutions that can address the problems and challenges raised by cloud-based DR.

6.9.1 Local backup

A solution to the dependency problem has been proposed in [19]. A Local Backup can be deployed on the side of customers to control data and to get backup of both data and even complete application on local storage. Local storage are often updated through a secured channel. By this technique, migration between CSPs and migration from public to private clouds, and from private to public clouds is possible. In the event of a disaster, local backups can provide the services that used to be provided by the CSP.

6.9.2 Geographical redundancy and backup (GRB)

Geographical Redundancy can be used in a traditional context. Two cloud zones mirror each other [17]. If one zone gets down, then the zone will be on and provide the services. A module monitors the zones to detect disaster.

Another research [20] has been proposed proposes a method to select optimal locations for multiple backups. The number of places is decided based on the nature of the application and service priority. Distance and bandwidth are two factors to settle on the simplest sites. However, this work neglects some critical factors such as the capacity of mirror sites and the number of node sources that can be hosted in each location.

6.9.3 Inter-private cloud storage (IPCS)

According to the Storage Networking Industry Association (SNIA), at least three backup locations are necessary for business data storage. Users' data should be stored in three different geographical locations: Servers, Local Backup Servers (LBS) and Remote Backup Server (RBS) [21].

6.9.4 Resource management

Hybrid clouds consist of many different hardware and software. In cloud-based enterprises, all business data are stored in storage resources of the cloud. Therefore, data protection, safety and recovery are critical in these environments. Data

protection is challenged when the data that has been processed at the primary host has not been stored in the backup host yet. There are three solutions for data recovery purposes [22]:

- Using fastest disk technology in the event of a disaster for data replication
- Prediction and replacement of risky devices: Some important factors such as power consumption, heat dissipation, and carbon footprint and data criticality (stored on each disk) can be calculated for a specific period

6.9.5 Pipelined replication

This replication technique [23] aims to gain both the performance of asynchronous replication and the consistency of sync replication. In synchronous replication, processing cannot continue until replication is completed at the backup site. Whereas, in asynchronous replication, after storing data in the local storage the process can be started.

6.9.6 Dual-role operation

To maximize resource usage, [24] introduces a technique which enables a host to operate as the primary host for some applications and to be the backup host for some other applications. In this architecture, clients send their requests to the backup host first, then the backup host transmits those requests to the primary host. After processing, the primary host sends a log to the backup and eventually replies to the clients. When a failure happens, the primary host becomes unavailable, and the backup host has to handle the requests sent to the failed host. However, this technique cannot guarantee a good service restoration by itself, because the backup site must share the resources between the requests that are directly sent to it and the redirected requests.

6.10 Disaster recovery-as-a-service (DRaaS) platforms

In this section some DRaaS systems are briefly introduced. In addition, benefits and weaknesses of each system are discussed.

6.10.1 Second site

The Second Site [25] is a disaster tolerance as a service system cloud. This platform is intended to cope three challenges: Reducing RPO, failure detection, and service restoration. Using a backup site: There is a geographically separated backup site that allows replicating groups of virtual machines through Internet links.

6.10.2 Distributed

Cloud System Architecture: In [26] the authors present a cloud architecture that provides high dependability of the system based on severe redundancy. The architecture is composed of multiple datacenters that are geographically separated from each other. Each datacenter includes VMs are active in physical. To perform DR, there is a backup server that stores a copy of each VM. In the case of a disaster, which makes a datacenter unavailable, the backup site transmits VM copies to another datacenter. Although this system architecture is expensive, it highly increases the dependability which can be adequate for (IaaS clouds. In addition, the

aforementioned paper describes a hierarchical approach to model cloud systems based on dependability metrics as well as disaster occurrence using the Statistic Petri Net approach.

6.11 Disaster recovery-as-a-service (DRaaS) open issues and future directions

In the previous sections, we described the main properties and challenges of DRaaS systems. However, some issues still require more effort to reach a level of DRaaS mechanisms in cloud computing.

6.11.1 Maximizing resource utilization

Cloud customers pay for DRaaS resources only after a disaster happens. However, these resources must always be available whenever they are needed. The revenue of the DRaaS servers is less. Therefore, CSPs need solutions to increase both the usage and the revenue of DRaaS servers while guaranteeing the availability of DRaaS services.

6.11.2 Correlated failures

Disasters that affect a specific area can lead to vast service interruption, and consequently, many customers have to be recovered by CSPs. In this case, it is possible that related servers cannot handle all the customers' requests. Therefore, it can be critical to multiplex customers' data of the same area in different servers. One major challenge in this case is how to distribute customers' traffic and data between cloud servers to minimize correlated failure risks with respect to required QoS for each server and also cloud SLAs [14].

6.11.3 Failover and failback procedures

In the event of a disaster, the failover procedure excludes failed resources and redirects workloads to a secondary site. Client-transparent procedures and fast IP failover requirements are two main challenges raised by this context. On the other hand, the cloud environment recovers from the disaster, application control has to be reverted to the original site. For this purpose, bidirectional state replication must be supported by the DRaaS mechanism. A portion of data may be lost because of the disaster in the primary site and new data will be created in the backup site. Therefore, one major challenge is how to determine new and old data which must be resynchronized to the primary site [24].

6.11.4 Disaster monitoring

In the case of a disaster, the sooner the failure is detected in either the primary site or the backup site, the better RTO. So, the challenge is how should the status of cloud be monitored and how a disaster can be detected as soon as possible [27].

6.11.5 Resource scheduling

The number of cloud-based services is increasing day by day and so has increased the complexity of cloud infrastructures. Hence, resource scheduling is a critical issue in modern cloud environments. This issue is more crucial for cloud-base platforms since they face unpredictable incoming traffic and have to consider a

variety of catastrophic situations. Building on this, more efficient resource scheduling techniques are needed in order for current DRaaS platforms to be optimized.

7. Application resiliency for cloud

This section covers the overview of resilient cloud application. This section also covers the key design strategies for application resiliency and monitoring applications for availability.

7.1 Resilient cloud applications overview

The cloud infrastructures are typically built on a large number of commodity systems to achieve scalability and keep hardware costs down. In this environment, it is assumed that some components will fail. Therefore, in the design of a cloud application the failure of individual resources often has to be anticipated to ensure an acceptable availability of the application. For existing applications, the code has to be rewritten to make them “cloud-ready” i.e., the application should have the required scalability and resiliency. A reliable application is able to properly manage the failure of one or more modules and continue operating properly. If a failed operation is retried a few milliseconds later, the operation may succeed. These types of error conditions are called as transient faults. Fault resilient applications have logic to detect and handle transient fault conditions to avoid application downtime. Key application design strategies for improving availability:

- Graceful degradation of application functionality
- Retry logic in application code
- Persistent application state model
- Event-driven processing

7.2 Graceful degradation of application functionality

Graceful degradation of application functionality refers to the ability of an application to maintain limited functionality even when some of the components, modules, or supporting services are not available. A well designed application or service typically uses a collection of loosely coupled modules that communicate with each other. The purpose of graceful degradation of application functionality is to prevent the complete failure of a business application or service. For example, consider an e-commerce application that consists of modules such as product catalog, shopping cart, order status, order submission, and order processing. Assume that the payment gateway is unavailable due to some problem. It is impossible for the order processing module of the application to continue. If the application or service is not designed to handle this scenario, the entire application might go offline. However, in this same scenario, it is possible that the product catalog module can still be available to consumers to view the product catalog. Also, the application could allow to place the order and move it into shopping cart. This provides the ability to process the orders when the payment gateway is available or after failing over to a secondary payment gateway.

7.3 Retry logic in application code

A key mechanism in a highly available application design is to implement retry logic within a code to handle service that is temporarily down. When applications use other cloud-based services, errors can occur because of temporary conditions such as intermittent service, infrastructure-level faults, or network issues. Very often, this form of problem can be solved by retrying the operation a few milliseconds later, and the operation may succeed. The simplest form of transient fault handling is to implement this retry logic in the application itself. To implement this retry logic in an application, it is important to detect and identify that particular exception which is likely to be caused by a transient fault condition. Also, a retry strategy must be defined to state how many retries can be attempted before deciding that the fault is not transient and define what the intervals should be between the retries. The logic will typically attempt to execute the action(s) a certain number of times, registering an error, and utilizing a secondary service if the fault continues.

7.4 Persistent application state model and event-driven processing

In a stateful application model, the session state information of an application (for example user ID, selected products in a shopping cart, and so on) is usually stored in compute system memory. However, the information stored in the memory can be lost if there is an outage with the compute system where the application runs. In a persistent application state model, the state information are stored out of the memory and usually stored in a repository (database). If a VM running the application instance fails, the state information is still available in the repository. A new application instance is created on another VM which can access the state information from the database and resume the processing.

In a tightly integrated application environment, user requests are processed by a particular application instance running on a server through synchronous calls. If that particular application instance is down, the user request will not be processed. For cloud applications, an important strategy for high availability design is to insert user requests into a queue and code applications to read requests from the queue (asynchronously) instead of synchronous calls. This allows multiple applications instances to process requests from the queue. This also enables adding multiple application instances to process the workload much faster to improve performance. Further, if an application instance is lost, the impact is minimal, which could be a single request or transaction. The remaining requests in the queue continue to be distributed to other available instances. For example, in an e-commerce application, simultaneous requests from multiple users, for placing orders, are loaded into a queue and the application instances running on multiple servers process the orders (asynchronously).

7.5 Monitoring application availability

A specialized monitoring tool can be implemented to monitor the availability of application instances that runs on VMs. This tool adds a layer of application awareness to the core high availability functionality offered by compute virtualization technology. The monitoring tool communicates directly with VM management software and conveys the application health status in the form of an application heartbeat. This allows the high availability functionality of a VM management software to automatically restart a VM instance if the application heartbeat is not received within a specified interval. Under normal circumstance, the resources that

comprise an application are continuously monitored at a given interval to ensure proper operation. If the monitoring of a resource detects a failure, the tool attempts to restart the application within the VM. The number of attempts that will be made to restart an application is configurable by the administrator. If the application does not restart successfully, the tool communicates to high availability functionality of a VM management software through API in order to trigger a reboot of the VM. The application is restarted as part of this reboot process. This integration between the application monitoring tool and the VM high availability solutions protects VMs, as well as the applications that run inside them.

8. Solutions and recommendations

To create a disaster recovery solution, an alternative location must be prepared to be able to recover a datacenter at failure occurring and the business can continue to run.. As a proof of concept of this solution, Microsoft Azure is used. Microsoft Azure is the public cloud to offer Disaster Recovery solution for applications running on Infrastructure as a Service (IaaS) by replicating VMs into another region even failure occurs on region level. The second proposed solution is a way of implementing highly available virtualized network element using Microsoft Windows Server and Microsoft System Center tools called High Availability Solution over Hybrid Cloud Using Failover Clustering Feature.

8.1 The first solution: network function virtualization over cloud-disaster recovery solution

Cloud Computing is making big inroads into companies today. Smaller businesses are taking advantage of Microsoft cloud services like Windows Azure to migrate their line-of business applications and services to the cloud instead of hosting them on-premises. The reasons for doing this include greater scalability, improved agility, and cost savings. Large enterprises tend to be more conservative with regards to new technologies mainly because of the high costs involved in widespread rollout of new service models and integrating them with existing organization's datacenter infrastructure.

Disaster recovery (DR) is an area of security planning that aims to protect an organization from the effects of significant disastrous events. It allows an organization to maintain or quickly resume mission-critical functions following a disaster.

Network & Mobile organizations require features that enable data backup or automate the restoring of an environment, while incurring minimal downtime. This allows organizations to maintain the necessary levels of productivity.

Network & Mobile organizations require features that enable data backup or automate the restoring of an environment, while incurring minimal downtime. This allows organizations to maintain the necessary levels of productivity.

Microsoft System Center has components that enable Network & Mobile organizations to back-up their data and automate the recovery process. The components used in this solution are Data Protection Manager (DPM) and Orchestrator.

Cloud computing is making big inroads into companies today. Smaller businesses are taking advantage of Microsoft cloud services like Windows Azure to migrate their line-of business applications and services to the cloud instead of hosting them on-premises.

The reasons for doing this include greater scalability, improved agility, and cost savings. Large enterprises tend to be more conservative with regards to new technologies mainly because of the high costs involved in widespread rollout of new

service models and integrating them with existing organization's datacenter infrastructure.

Windows Azure Pack is designed to help large enterprises overcome these obstacles by providing a straightforward path for implementing hybrid solutions that embraces both the modern datacenter and cloud-hosting providers.

Microsoft Windows Azure Pack brings Windows Azure technologies to private cloud integrating Windows Server, System Center to offer a self-service portal and cloud services. Microsoft Windows Azure Pack is now the preferred interface for private cloud environments. A hybrid cloud solution helps enterprises transform their current infrastructure to public cloud to achieve cost effectiveness. With advanced offloads, acceleration, virtualization, and advanced scale-out storage features, this reference architecture provides the most efficient, multi-tenant cloud that is built on top of Windows Azure Pack. Building on a familiar foundation of Windows Server and System Center, the hybrid cloud platform offers a flexible and familiar solution for enterprises to deliver business agility through self-provisioning and automation of resources.

The objective of this project is creating a hybrid cloud with Microsoft System Center 2016 and Windows Azure Pack (WAP) for storage management service.

A storage cloud can help the business units become more agile and dynamic to meet the fluctuating resource demands of their clients. Storage cloud also helps the larger organization to implement a pay-per-use model to accurately track and recover infrastructure costs across the business units.

In this solution, the software components required for deploying Windows Azure Pack (WAP) are downloaded and installed. Microsoft System Center 2016 components are downloaded and installed. The System Center Virtual Machine Manager (VMM), Operation Manager (OM), Service Manager (SM) and Orchestrator (ORCH) are configured and integrated for delivering self-service and automation. The problem of limited and shared storage owned by most of enterprises can be addressed by renting large storage from public cloud provider.

The scenario for deploying a hybrid cloud solution for enterprises to overcome this problem is introduced in this project. The tenant portal of WAP is configured to be a web portal interface for the users to request the services by himself. The services are pre-configured by admin portal WAP and accessed by the users through WAP tenant portal. Admin portal of WAP allows administrator managing clouds over a web browser. This tool also allows to enabling self-service and automation for end users in order to create virtual machines inside clouds. This solution introduces for enterprises to ensure the continuity of the service that is provided to the users by using a hybrid cloud solution of this project for managing the allocated storage [28].

The solution is based on 2 sites and one Public Cloud, Site 1 is the main datacenter which has the services such as Database Servers (VMs), Site 2 is the hot disaster recovery site which it has all equipment needed to receive the recovered data, The Public Cloud has the resources essential for the recovery process itself which are DPM and VMM. The user can be a mobile phone, computer or any other device that can connect to the network.

Backups are typically performed on a daily basis to ensure necessary data retention at a single location, for the single purpose of copying data. Disaster recovery requires the determination of the RTO (recovery time objective) in order to designate the maximum amount of time the business can be without IT systems post-disaster. Traditionally, the ability to meet a given RTO requires at least one duplicate of the IT infrastructure in a secondary location to allow for replication between the production and DR site.

Disaster recovery is the process of failing over your primary environment to an alternate environment that is capable of sustaining your business continuity.

Backups are useful for immediate access in the event of the need to restore a document but does not facilitate the failover of your total environment should your infrastructure become compromised. They also do not include the physical resources required to bring them online.

A backup is simply a copy of data intended to be restored to the original source. DR requires a separate production environment where the data can live. All aspects of the current environment should be considered, including physical resources, software, connectivity and security.

Planning a backup routine is relatively simple, since typically the only goals are to meet the RPO (recovery point objective) and data retention requirements. A complete disaster recovery strategy requires additional planning, including determining which systems are considered mission critical, creating a recovery order and communication process, and most importantly, a way to perform a valid test the overall benefits and importance of a DR plan are to mitigate risk and downtime, maintain compliance and avoid outages. Backups serve a simpler purpose. Make sure you know which solution makes sense for your business needs.

In normal situation, Customer can access their service through Microsoft Windows Azure Pack (WAP) portal from site1. System Center Data Protection Manager (DPM) makes a backup for site1 Virtual Machines (VMs). System Center Orchestrator (ORCH) operates a runbook for DPM to take recovery points of(VMs) of site1 to reduce any failure down-time Then, ORCH operates a runbook for System Center Virtual Machine Manager (VMM) to live migrate virtual machines of site 1. In a scheduled loop, ORCH operates a runbook for DPM to take recovery points of VMs of site1 as shown in **Figure 16**.

In a Disaster situation of site1, System Center Operation Manager (OM) detects failure of site1 and sends failure alert to ORCH. ORCH senses failure alerts, then runbooks is running automatically and operates DPM to perform a recovery for the backed up site1 and add the last recovery points taken. So that, our customer can access site 2 and find their service up, with decreased down-time as shown in **Figure 17**.

Setups (Installation), there are common Prerequisite for all of System Center Products which is SQL Database if we want a database for each product to store configuration files on them.

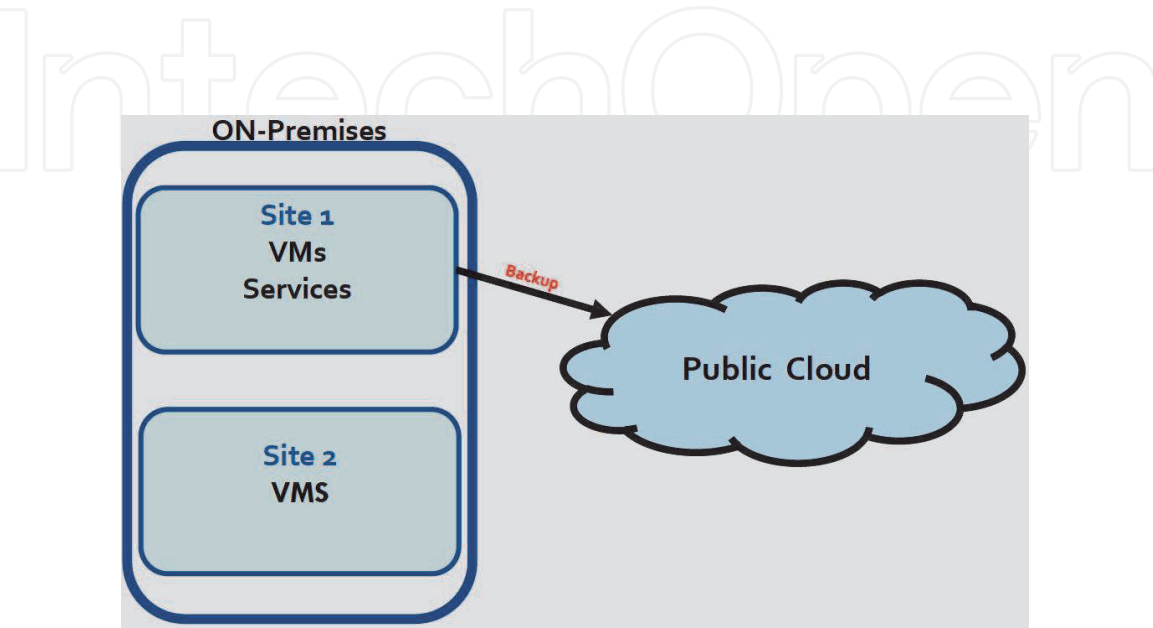


Figure 16.
Normal Case.

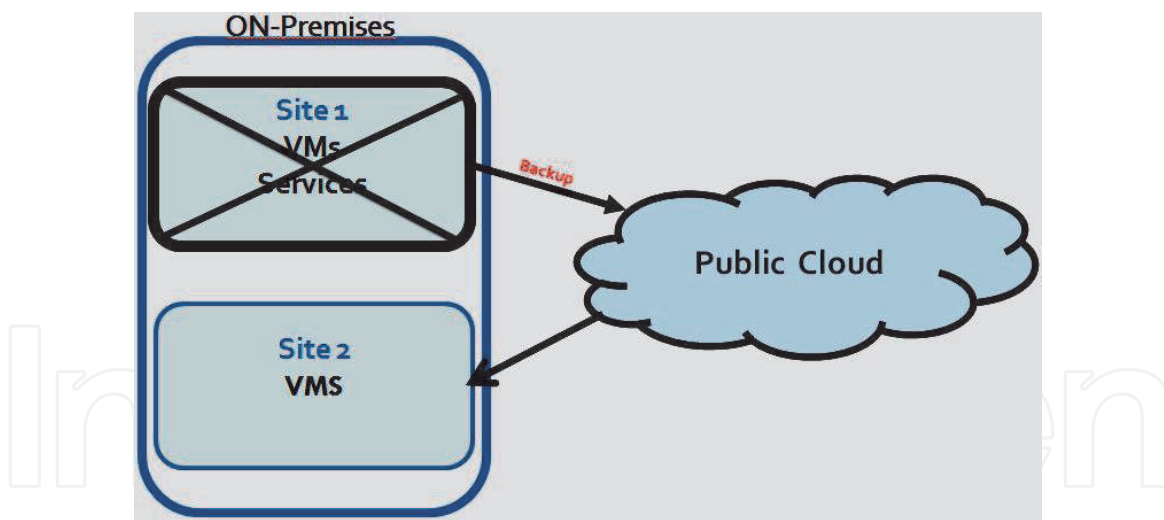


Figure 17.
Disaster Response.

8.2 The second solution: high availability solution over hybrid cloud using failover clustering feature

This solution is mainly concerned with two complementary aspects, Network Function Virtualization (NFV) followed by Hybrid cloud computing. Initially, virtualization of administrative components on premise of a company took place, such as Active Directory Domain Services (ADDS), System Center tools and Service Provider Foundation (SPF). Then we virtualized the Private Branch Exchange (PBX), the network function we are concerned with (NFV), by installing Elastix software on a virtual machine. This network function is one of the most important services adopted by various entities such as mobile operators. Thus, one of its main concerns is high availability. The availability of a system at time 't' is referred to as the probability that the system is up and functional correctly at that instance in time.

In our solution, we planned to achieve highly available voice service using Failover Cluster feature on Windows Server 2012. This is achieved by having any two identical nodes, a primary and a secondary one. Rather than having both nodes on premises of the company which may be not as cost efficient and may also be less safe since it is subjected to the same kind of failures on premise, we decided to have the secondary node on Microsoft Azure public cloud. This is called hybrid cloud computing.

In conclusion, the voice service is always running on the primary node on premises as long as there are no failures. In case of a critical failure directly affecting the voice service, its virtual machine would be migrated automatically to the secondary node on Azure's public cloud with minimum delay and not affecting the call. And thus, the voice service is proved to be highly available maintaining customer's confidence and preventing revenue losses (**Figure 18**).

The solution provides high availability for a VNF of a mobile operator, which is a Virtual Machine with cloud PBX-Elastix-installed on it as a proof of concept. **Figure 19** shows the topology of the project, which is a hybrid cloud. The on-premises part (private) represents the mobile operator and Microsoft Azure part (public) is the cloud service provider that provides a secondary failover cluster node as a part of a tenant plan. So when the Elastix server fails on premise, the Elastix service (virtual machine) is transferred to the cluster node in Microsoft Azure, making Elastix highly available [29].

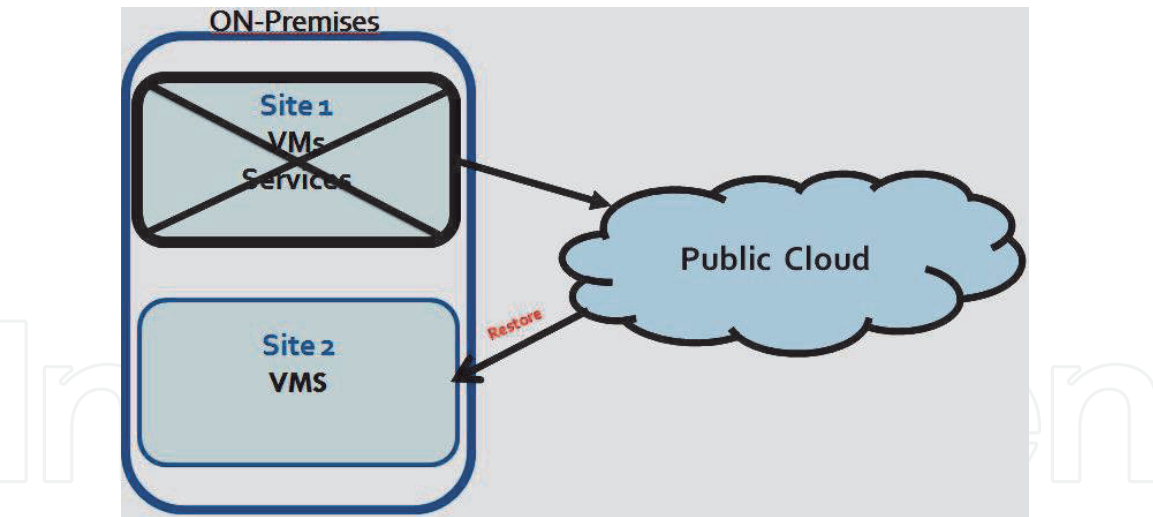


Figure 18.
Disaster Response.

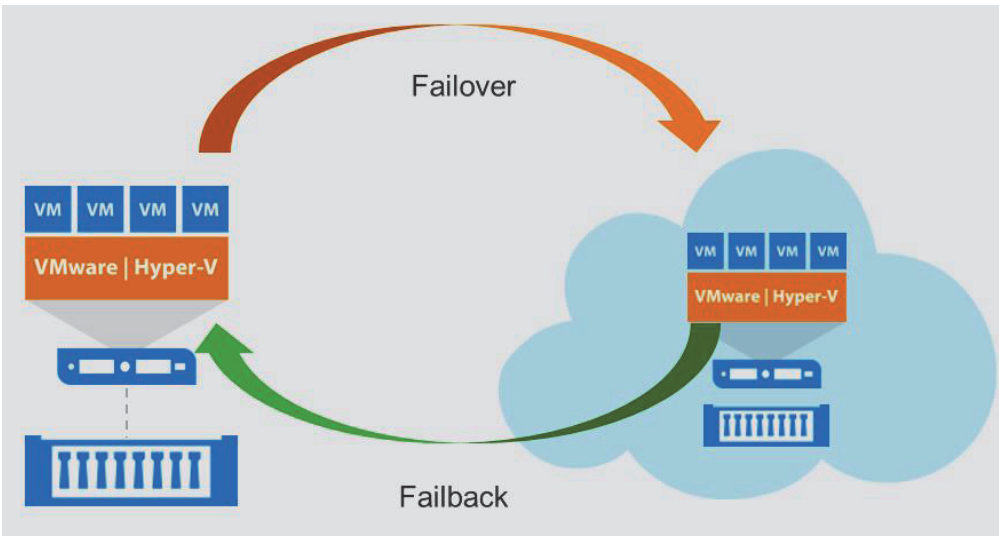


Figure 19.
Cloud Failover Scenario.

The topology components on-premises are:

- AD DS: It is the domain on premises, used for creating and authorizing Service Accounts of Microsoft System Center products. And logged on by all on-premises users and computers.
- VMM: The primary virtualization and management product of Microsoft System Center. It is used for adding hosts and clusters to the VMM library and for creating clouds, and so on. An administrator uses VMM for all management tasks including remote ones.
- Orchestrator: Its console is installed so that SPF can be installed because SPF is a sub-component of ORCH.
- SPF: SPF facilitates communication between Azure Pack and System Center products (VMM in this case). It reflects all changes that the participants (Azure Portals and VMM) made.

- **Azure Admin Portal:** This portal is managed by the on-premises IT administrators and is where user accounts and plans are created, and where resources are assigned to on-premises tenants through linking the users (tenants) with the plans.
- **Azure Tenant Portal:** This portal is used by tenants, the on-premises employees in this case. Tenants can view the services available to them and configure their own environment, such as deploying and managing VMs using the tenant portal.
- **Failover Cluster Node 1:** This is the on-premises node (primary node) of the Failover Cluster which has the virtual machine of Elastix on it along with other VNFs.

The additional components at Microsoft Azure are:

- **Microsoft System Center:** It is a suite of systems management products. The core products are: VMM, ORCH, SM, OM and DPM. They help IT build reliable and manageable systems and better automate processes.
- **Azure Tenant Portal:** This portal is used by azure subscribers; azure subscribers can be individual users or enterprises. In this case the mobile operator is a tenant assigned to a plan created by Microsoft Azure's Admin portal.
- **Failover Cluster Node 2:** The secondary public node of the Failover Cluster which Elastix virtual machine is migrated to and which takes over if node 1 of the on-premise fails.
- **SMB:** Provides cluster nodes with a shared access to shared storage files.

Zoiper application is installed on the subscribers' mobile phones to make VoIP calls. Using Elastix web portal, a SIP extension is created for each subscriber to be able to create their own accounts on Zoiper (**Figure 20**).

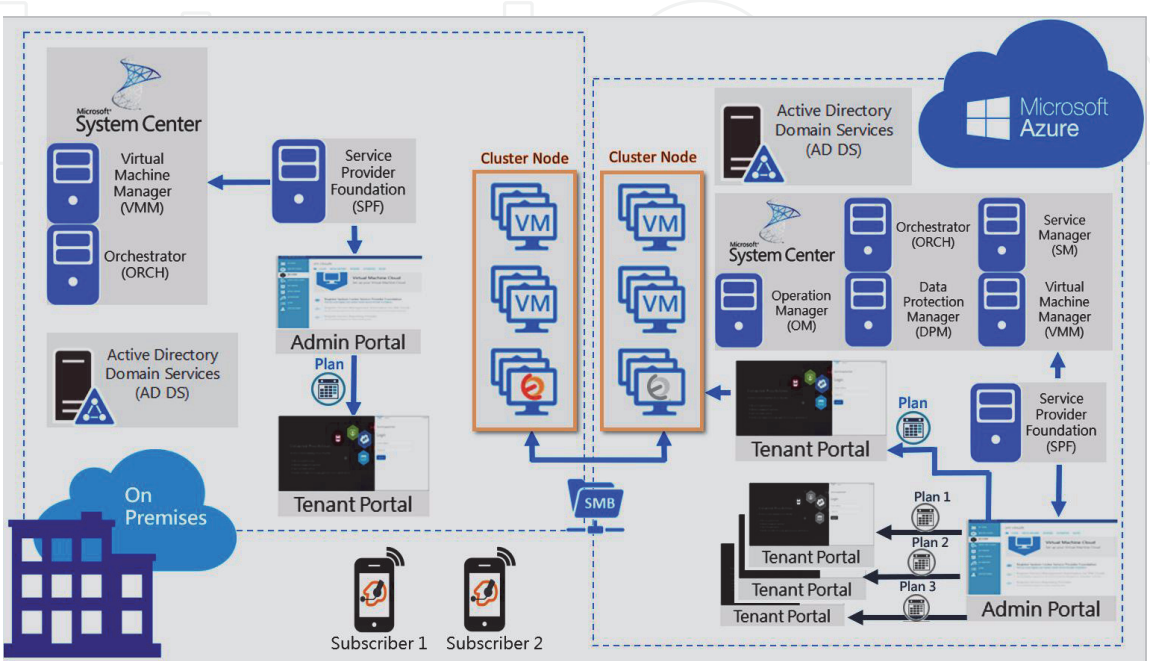


Figure 20.
Network Topology of the Project.

Initially, a call is initiated between the two subscribers through Zoiper. The two subscribers will access the Elastix server deployed on the on-premises node as demonstrated in **Figure 21**.

In the case of failure of the on-premise node (primary node) while the call is ongoing, the Elastix virtual machine will be migrated to the Microsoft Azure node (secondary node), which now becomes the primary node. The Elastix virtual machine will continue running on the Microsoft Azure node by accessing SMB storage as demonstrated in **Figure 22**.

During the migration process, the downtime ranges from 2 to 3 seconds, which is barely recognizable by the user, and then the call proceeds normally. And thus, the voice service using Elastix is proved to be highly available. When the on-premises node is up again, the Elastix virtual machine will be manually migrated using the Failover Cluster Manager.

The overall goal of this solution is providing high availability solution for a virtualized network element using hybrid cloud computing. This will improve the performance of cloud services. Moreover, reduce the downtime of a service, which

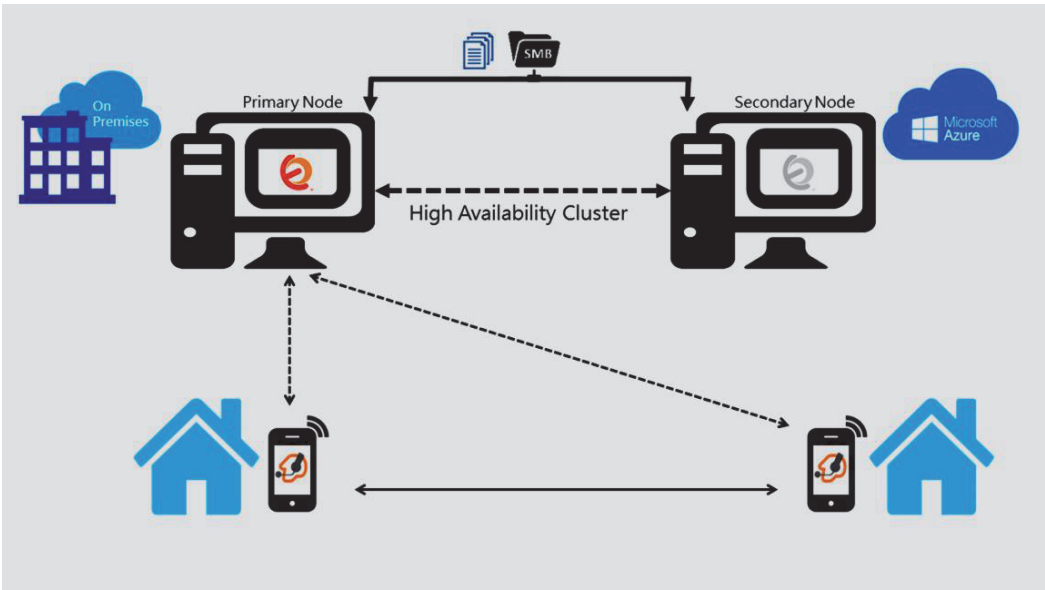


Figure 21.
The Call before Primary node failure.

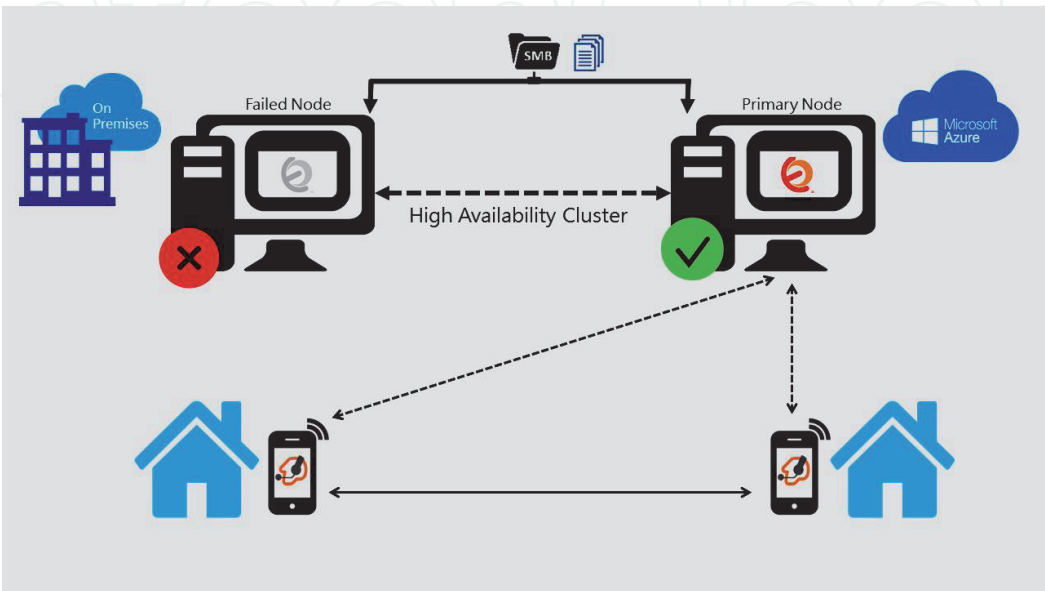


Figure 22.
The Call after Primary node failure.

would otherwise degrade the performance. This paper focused on conducting the solution using Failover cluster feature along with Microsoft System Center tools. Failures that might occur include but are not restricted to the following: network vulnerability, human mistakes, server, storage or power failures and need to be avoided.

As a conclusion, the cloud will remain subject to failure and failures can occur in the cloud as well as the IT traditional environment. Thus, high availability cannot be ensured, but it can be increased and improved, by avoiding common system failures through the implementation of different solutions and techniques.

9. Conclusion

This chapter covered the importance of business continuity in a cloud environment and how business continuity enables to achieve the required service availability. First, we briefly introduced cloud computing, including background, properties, advantages and challenges. This chapter also covered various fault tolerance mechanisms for cloud infrastructure to eliminate single points of failure. This chapter further covered data protection solutions such as backup and replication. Then, we discussed the details of DRaaS approaches. In addition, we also derived the main challenges of DRaaS mechanisms and proposed solutions to overcome them. Furthermore, the main DRaaS platforms are discussed, followed by open issues and future directions in the field of DRaaS mechanisms. Finally, this chapter also covered the key design strategies for cloud application resiliency.

Finally and as a proof of concept of cloud DR solutions, Microsoft Azure is used. Microsoft Azure is the public cloud to offer Disaster Recovery solution for applications running on Infrastructure as a Service (IaaS) by replicating VMs into another region even failure occurs on region level. The second proposed solution is a way of implementing highly available virtualized network element using Microsoft Windows Server and Microsoft System Center tools called High Availability Solution over Hybrid Cloud Using Failover Clustering Feature. The two solutions were successfully implemented and provided high performance and excellent results.

Author details

Wagdy Anis Aziz¹, Eduard Babulak^{2*} and David Al-Dabass³


¹ Ain Shams University Faculty of Engineering/Orange, Egypt

² National Science Foundation, USA

³ Nottingham Trent University, England

*Address all correspondence to: babulak@yahoo.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Reid, S., Kicker, H., Matzke, P., Bartels, A., & Lisserman, M. (2011). Sizing the cloud. Technical report. Retrieved from <http://www.forrester.com/-/E-/Sizing+The+Cloud/fulltext/RES58161objectid=RES58161>
- [2] White paper. (2013). *UK Cloud Adaption and Trends*. Retrieved from <http://cloudindustryforum.org/white-papers/uk-cloud-adoption-and-trends-for-2013>
- [3] Kashiwazaki, H. (2012). Practical uses of cloud computing services in a Japanese university of the arts against aftermath of the 2011 Tohoku earthquake. *Proceedings of the ACM SIGUCCS 40th annual conference on Special interest group on university and college computing services* (pp. 49-52).
- [4] Mesbahi, Rahmani and Hosseinzadeh (2018). Reliability and high availability in cloud computing environments: a reference roadmap.
- [5] EMC Corporation (2014). Module: Business Continuity.
- [6] Carney, Michael (2013-06-24). "AnyPresence partners with Heroku to beef up its enterprise mBaaS offering". Pando Daily. Retrieved 24 June 2013.
- [7] Alex Williams (11 October 2012). "Kii Cloud Opens Doors For Mobile Developer Platform With 25 Million End Users". TechCrunch. Retrieved 16 October 2012.
- [8] Dan Rowinski (9 November 2011). "Mobile Backend As A Service Parse Raises \$5.5 Million in Series A Funding". ReadWrite. Retrieved 23 October 2012.
- [9] Miller, Ron (24 Nov 2015). "AWS Lambda Makes Serverless Applications A Reality". TechCrunch. Retrieved 10 July 2016.
- [10] Sbarski, Peter (2017-05-04). *Serverless Architectures on AWS: With examples using AWS Lambda* (1st ed.). Manning Publications. ISBN 9781617293825.
- [11] Arean, O. (2013). Disaster recovery in the cloud. *Network Security*, 9, 5-7. [http://dx.doi.org/10.1016/S1353-4858\(13\)70101-6](http://dx.doi.org/10.1016/S1353-4858(13)70101-6)
- [12] ETSI NFV (<https://www.etsi.org/technologies/nfv>)
- [13] Alhazmi, O. H., & Malaiya, Y. K. (2013). Evaluating disaster recovery plans using the cloud. *Reliability and Maintainability Symposium (RAMS), IEEE Proceedings-Annual* (pp. 1-6). <http://dx.doi.org/10.1109/RAMS.2013.6517700>
- [14] Wood, T., Cecchet, E., & Ramakrishnan, K. K. (2010). Disaster recovery as a cloud service: Economic benefits & deployment challenges. *2nd USENIX Workshop on Hot Topics in Cloud Computing* (pp. 1-7).
- [15] Lwin, T. T., & Thein, T. (2009). High Availability Cluster System for Local Disaster Recovery with Markov Modeling Approach. *International Journal of Computer Science Issues*, 6(2), 25-32. Error! Hyperlink reference not valid.
- [16] Guster, D., & Lee, O. F. (2011). Enhancing the Disaster Recovery Plan Through Virtualization. *Journal of Information Technology Research*, 4 (4), 18-40. <http://dx.doi.org/10.4018/jitr.2011100102> IBM white paper. (2012). Virtualizing disaster recovery using cloud computing, IBM global technology services.
- [17] Pokharel, M., Lee, S., & Park, J. S. (2010). Disaster Recovery for System Architecture using Cloud Computing. *10th IEEE/IPSJ International Symposium*

on Applications and the Internet (SAINT) (pp. 304-307).

[18] C. Bucur and E. Babulak, "Security validation testing environment in the cloud," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 4240-4247, doi: 10.1109/BigData47090.2019.9006202.

[19] Javaraiah, V. (2011). Backup for cloud and disaster recovery for consumers and SMBs. IEEE 5th International Conference on Advanced Networks and Telecommunication Systems (ANTS) (pp. 1-3). <http://dx.doi.org/10.1109/ANTS.2011.6163671>

[20] Khan, J. I., & Tahboub, O. Y. (2011). Peer-to-Peer Enterprise Data Backup over a Ren Cloud. *IEEE 8th International Conference on Information Technology: New Generations (ITNG)* (pp. 959-964). <http://dx.doi.org/10.1109/ITNG.2011.164>

[21] Jian-hua, Z., & Nan, Z. (2011). Cloud Computing-based Data Storage and Disaster Recovery. *IEEE International Conference on Future Computer Science and Education (ICFCSE)*, (pp. 629-632). <http://dx.doi.org/10.1109/ICFCSE.2011.157>

[22] Patil, S. R., Shiraguppi, R. M., Jain, B. P., & Eda, S. (2012). Methodology for Usage of Emerging Disk to Ameliorate Hybrid Storage Clouds. *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* (pp.1-5). <http://dx.doi.org/10.1109/CCEM.2012.6354615>

[23] Wood, T., Lagar-Cavilla, H. A., Ramakrishnan, K. K., Shenoy, P., & Van der Merwe, J. (2011). PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery. *2nd ACM Symposium on Cloud Computing*. <http://dx.doi.org/10.1145/2038916.2038933>

[24] Aghdaie, N., & Tamir, Y. (2003). Fast transparent failover for reliable web service. 15th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS) (pp. 757- 762).

[25] Rajagopalan, S., Cully, B., Connor, R. O., & Warfield, A. (2012). SecondSite: disaster tolerance as a service. *ACM SIGPLAN Notices*, 47(7), 97-107. <http://dx.doi.org/10.1145/2365864.2151039>.

[26] Silva, B., Maciel, P., Tavares, E., & Zimmermann, A. (2013). Dependability models for designing disaster tolerant cloud computing systems. *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp.1-6). <http://dx.doi.org/10.1109/DSN.2013.6575323>

[27] Aceto, G., Botta, A., Donato, W., & Pescapé, A. (2013). Cloud monitoring: A survey. *Computer Networks*, 57(9), 2093-2915. <http://dx.doi.org/10.1016/j.comnet.2013.04.001>

[28] Wagdy A. Aziz et al: Network Function Virtualization Over Cloud - Disaster Recovery Solution Over Hybrid Cloud, ijssst.info/Vol-21/No-4/paper15

[29] Wagdy A. Aziz et al: High Availability Solution Over Hybrid Cloud Using Failover Clustering Feature, ijssst.info/Vol-21/No-4/paper16