ICTCLD507 Build and Deploy Resources on Cloud Platforms

&

ICTCLD508 Manage Infrastructure in Cloud Environments

Version: v2.0  
  
Student Name: Warren Ikin

Student ID: 453470833

Submission Date:30/09/2022

Contents:

[Scope 1](#_Toc125620795)

[Client Details: 2](#_Toc125620796)

[1. Introduction to ABC Banking Corp 2](#_Toc125620797)

[2. Functionalities and resources that can be deployed in Cloud 3](#_Toc125620798)

[3. Cloud Computing Resource Options 7](#_Toc125620799)

[4. Cloud Resource Capacity Limits 1](#_Toc125620800)

[5. AWS Cloud Resource Capacity Limits 5](#_Toc125620801)

[5. Organisational Policies Implemented for the cloud 11](#_Toc125620802)

[6. Business Purpose Table. 14](#_Toc125620803)

[7. Repeatable Tasks and Automation Tools. 16](#_Toc125620804)

[8. Steps Required to Provision Resources. 20](#_Toc125620805)

[9. Provisioned Resources and Infrastructure Testing Model. 20](#_Toc125620806)

[10. Two policies for each of the following: 21](#_Toc125620807)

[Source Links: 24](#_Toc125620808)

Scope

To create a hybrid virtualised computer solution for ABC Bank and then move its critical infrastructure to the cloud to reduce its operational costs as well as provide 99.99% available services 24 hours per day, three-hundred, and sixty-five days of the year. These duties will be carried out in a professional manner as stipulated in the Uptown IT SLA, using services that are validated for financial services on the cloud. The initial overview as per the information and topology provided by the cloud architect is outlined below:

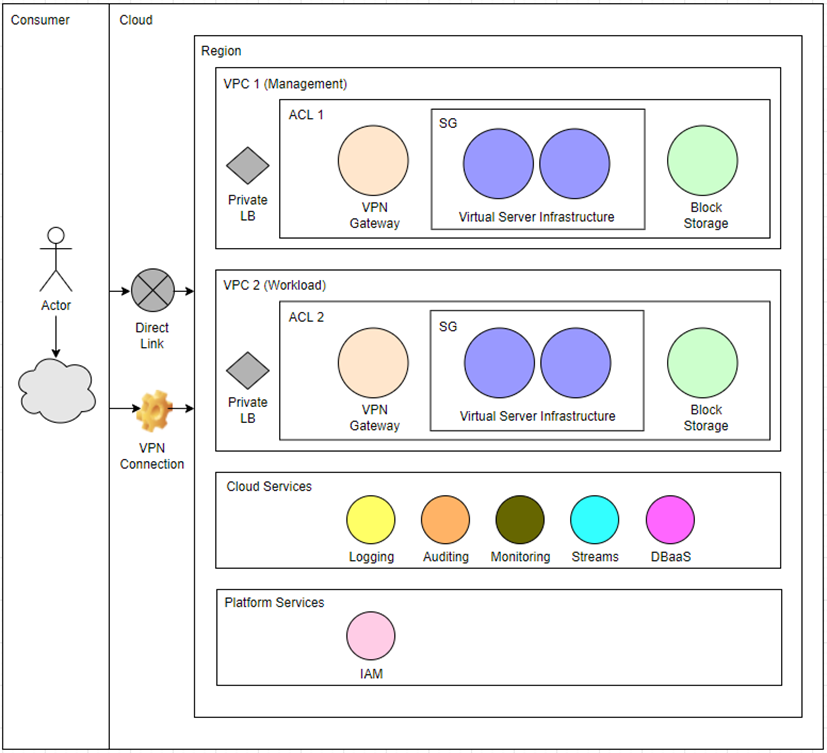


Figure 1

* Two Virtual Private Clouds (VPCs) containing:
  + 1 x management server configured with a domain controller and a file server, block storage, VPN connection.
  + 1 x web server for actual workloads, block storage, VPN connection
  + Individual Security Groups (SGs), Access Control Lists (ACLs), and Load Balancer (LB).
* Internet Gateway for public access connecting to the relevant AWS region.
* Each region in the availability zone consists of both VPCs, secure encrypted connections via IGW or VPN, AWS IAM platform services, and restricted access to essential cloud services such as: logging; auditing; monitoring; streams, and DBaaS.
* Establish site-to-site VPN connections to remote sites or to the branch office for communications and for synchronising with warm site backup.

Client Details:

|  |  |
| --- | --- |
| Business Name: | ABC Banking Corp |
| Business Location: | Branch Office, Level 27, 10 Market Street, Brisbane City |
| Date: | 30th September 2022 |
| Analyst Name: | Warren Ikin |
| Information Sources: | The consultant Cloud Architect |

1. Introduction to ABC Banking Corp

It is understood that Uptown IT has been tasked with implementing ABC Banking Corp’s new hybrid solution and to move its critical infrastructure to the cloud as recommended by the consultant cloud architect. The design for the prospective move is due to the significant costs in maintaining a data centre, its staff, and the increasing security risks. By this move, the bank is expecting to reduce its operational costs and to provide reliable services to its customers.

There are hundreds of various solutions and proprietors available and ready to offer the best solution for ABC Banking’s needs. Third party proprietors introducing new and better tools and plugins to augment cloud services every day. The mind boggles with the availability of choice. The challenge is to find the right one. ABC Banking already has considerable resources at hand. These resources already utilize many technologies that need to be migrated to cloud. ABC Banking’s infrastructure is already running on hypervisors and operating virtual systems on the on-premises data centre. VPNs already exist between the secret data centre and the central management control centre.

[Hybrid deployment](#hybridCloudServices) will allow ABC Banking to extend and scale their infrastructure into the cloud while still maintaining access to on-premises resources living on on-site servers. We can use the cloud deployment as backup in disaster recovery solution or vice versa. ABC Banking can maintain a working copy on premises, but make sure they have durable backup in the cloud. Because migration of existing IT systems takes a long time and is costly, hybrid deployment is a very effective in-between as resources are migrated to the cloud. AWS has many add-in services to facilitate a smooth transition to the cloud from the on-premises system.

2. Functionalities and resources that can be deployed in Cloud

Organizations of every type, size, and industry are using the cloud for a wide variety of use cases, such as data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics, and customer-facing web applications. Compute services provide virtual server hosting, container management, and serverless computing. You can set code to run to certain triggers using Lambda, run virtual machines using Elastic Compute Cloud or EC2, quickly set up and run small websites using LightSail, or create a unit of software to ship out to your users using Elastic Container Services or ECS. Compute services are backbones of cloud computing platforms as they provide the much-coveted computing resources that many companies are looking for. Instead of having to host their own servers in their own data centers, they can rent servers from AWS for pennies on the dollar. Storage services provide storage for both in-use and archival files. You can use Elastic File System or EFS to create shared folders in the cloud. You can upload flat files like images, videos, or text files to Simple Storage Service or S3 and link to it directly to use on your website. You can also archive files and store large amounts of data for cheap using Glacier or you can use Storage Gateway to take daily backups of your company's on-premises data and send them to the cloud for safekeeping. Storage solutions are cheaper than ever with cloud computing and AWS provides many options depending on the frequency of access and durability of data you require. AWS also offers fully managed relational and NoSQL databases. Their cost-efficient relational database is called Relational Database Service or RDS and a highly scalable NoSQL database is called DynamoDB. They also offer a fully managed, easily scalable petabyte-scale data warehouse service called Redshift and a highly scalable caching service called ElastiCache. ElastiCache allows you to run extremely intensive computations by caching necessary data in the cloud. All of the database services are highly scalable and cost efficient so you can crunch all the numbers and data you need for a fraction of the cost of an on-site database server. It's easy to get lost in the abundance of options but it's also exciting to consider the almost limitless potentials in what we can create using these resources.

<https://docs.aws.amazon.com/index.html>

The major advantages of the cloud over on-premises:

* Agility – IaaS, PaaS, and SaaS provide off-the-shelf customizable system environments. The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine. Pre-cooked infrastructure services, such as compute, storage, and databases, to machine learning, data lakes and analytics, and more.
* Elasticity - With cloud computing, you don’t have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the number of resources that you need. You can scale these resources up or down to instantly grow and shrink capacity as your business needs change. AWS EC2 instances augmented with ELB (Elastic Load Balancing) provide such flexibility.
* Cost savings - The cloud allows you to trade fixed expenses (such as data centres and physical servers) for variable expenses, and only pay for IT as you consume it. Plus, the variable expenses are much lower than what you would pay to do it yourself because of the economies of scale at AWS.
* Deploy globally in minutes - With the cloud, you can expand to new geographic regions and deploy globally in minutes. For example, AWS has infrastructure all over the world, so you can back up and deploy your services in multiple physical locations with just a few clicks.
* Plethora of choice in infrastructure as mentioned previously in Agility, but more detailed below: [Figure 2 Separation-of-Responsibilities schematic:](#Figure2)
  + Most of us use cloud file storage services such as Dropbox, Google Drive or OneDrive. We collaborate with each other using cloud-based applications like Zoom, Google-Docs and SharePoint. These applications are often referred to as Software as a Service, or SaaS. Basically, cloud providers offer software solutions to consumers through the internet. Another option is Platform as a Service, or PaaS. This solution provides complete software development and deployment environments for organizations. While SaaS provides a finished product, PaaS allows users to build their own internet cloud solutions by providing resources, including databases, business intelligence and data analytics tools, and programming interfaces to other existing applications and services. PaaS and SaaS will eventually be integrated into our IaaS framework as technologies and automated learning are properly integrated into the system. Infrastructure as a Service (IaaS) is the most logical hybrid link to existing on-premises system. AWS VPN, VPC, IAM, EC2, ELB, S3, EBS are PaaS and SaaS services that run concurrently within IaaS.

AWS IaaS services Pros:

* + - Resources are available as a service.
    - Cost depends on the consumption.
    - Highly scalable.
    - Allow multiple users to access a single piece of hardware.
    - The organization has complete control of the infrastructure.
    - Dynamic and flexible.
  + AWS IaaS services Cons:
    - Security threats which may arise from the host or other virtual machines.
    - Customers’ inability to access their data when vendor outages happen.
    - Required team training to learn how to manage new infrastructure.
  + Platform as a Service (PaaS) relates to services offered by a cloud platform that provides computing and software resources with minimal to no infrastructure management requirements. PaaS is the natural evolution of Infrastructure as Service (IaaS). It supports other compute services such as EC2, S3, and DynamoDB. PaaS leverages these instances with AWS Elastic Beanstalk, and Lambda.  
      
    AWS PaaS services Pros:
    - Resources can easily be scaled up or down as your business changes.
    - Provides a variety of services to facilitate the development, testing, and deployment of apps.
    - Multiple users can access via the same development application.
    - Able to integrate with web services and databases.
  + AWS PaaS Cons:
    - Dependent on the provider’s functional capabilities, speed, and reliability.
    - Compatibility problems may arise when existing infrastructure is incorporated into a new environment.
    - Security risks due to its availability in the public environment.
  + SaaS provides solutions that are ready to use and out-of-the-box to suit a specific business requirement. SaaS applications are hosted on software providers’ servers in the cloud. That frees users from licensing and hardware limitations, as well as patching and upgrade responsibilities.  
    The cloud is “Software Data-Storage & Processing-Power” that’s available via the web. Users therefore access cloud-based software as a service through web browsers and Apps.AWS SaaS Pros:
    - Managed from a central location.
    - Hosted on a remote server.
    - Accessible over the internet.
    - Users’ hardware or software updated by the provider.
  + AWS SaaS Cons:
    - Difficult to integrate with existing apps and services.
    - Security risks due to its availability in public environments.
    - Limited customization capabilities.
    - Loss of operational control which affects how PaaS solutions are managed, provisioned, and operated.

Source Link: <https://magenest.com/en/aws-iaas-paas-saas/>

Albeit we have chosen IaaS, this ‘Separation of Responsibilities’ diagram shows relationships and differences between IaaS, PaaS and SaaS.

**Separation of Responsibilities**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **You Manage** | | | | | **Other Manage** | | | | |
| **You manage** | On-Premises |  | Infrastructure (as a service) |  | | Platform (as a service) |  | SaaS (as a service) | **Other manage** |
| Applications | Applications | Applications | Applications |
| Data | Data | Data | Data |
| Runtime | Runtime | Runtime | Runtime |
| Middleware | Middleware | Middleware | Middleware |
| O/S | O/S | O/S | O/S |
| Virtualization | Virtualization | Virtualization | Virtualization |
| Servers | Servers | Servers | Servers |
| Storage | Storage | Storage | Storage |
| Networking | Networking | Networking | Networking |

Figure 2 Seperation of Resposibilities

3. Cloud Computing Resource Options

Given the ABC Bank’s significant current infrastructure, Uptown IT proposes we use AWS’s bare metal IaaS platform as most indicators identify this choice as the most logical for the bank’s hybrid design. We can migrate current virtual machine configurations or containers to the cloud account securely using AWS resource management tools or there are many third-party options including VMware. Hence our new cloud system will be built with AWS resources such as VPC, EC2, and S3 provisions over an IaaS framework. The functionalities of PaaS and SaaS can be employed as required once the initial migration is completed. These cloud computing resource options will be configured first as enumerated in the Gantt Chart in [Figure 3](#Figure3timeLine).

1. Administration/Security:
   1. [AWS Organizations](#awsOrganizations): AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet the budgetary, security, and compliance needs of your business. As an administrator of an organization, you can create accounts in your organization and invite existing accounts to join the organization.
      * [Click here to visit the Source Link for more features:](#awsOrganizationsfeat)
   2. [AWS Management Console](#awsConsole): With AWS IAM Identity Center, you can also obtain short-term credentials for use with the AWS SDK or CLI.

* + - [Click here to visit the Source Link for more features:](#awsConsolefeat)
  1. Performance and security with AWS Global Infrastructure. As per the [AWS SLA, 99.9%](#awsSLAnintyninepercent) resilience in AWS Availability Zones and design flexibility for choosing regions for offsite backups, and safe local zones for migrating data between hybrid cloud environments.
  2. AWS Governance & Security (Logging):
     + [CloudTrail](#awsCloudTrail): AWS CloudTrail monitors and records account activity across your AWS infrastructure, giving you control over storage, analysis, and remediation actions.  
       [Click here to visit the Source Link for more features:](#awsCloudTrailfeat)
     + [AWS Config](#awsConfig): AWS Config continually assesses, audits, and evaluates the configurations and relationships of your resources.  
       [Click here to visit the Source Link for more features:](#awsConfigfeat)
     + [CloudWatch](#awsCloudWatch): Amazon CloudWatch collects and visualizes real-time logs, metrics, and event data in automated dashboards to streamline your infrastructure and application maintenance.  
       [Click here to visit the Source Link for more features:](#awsCloudWatchfeat)

1. Network:

[AWS VPN](#awsVPN) (ITA secure direct link to AWS Cloud): AWS Client VPN is used by your remote workforce to securely access resources both on AWS and within your on-premises networks.  
[Click here to visit the Source Link for more features:](#awsVPNfeat)

* 1. [AWS VPC](#awsVPC) (Amazon Virtual Private Cloud): Gives full control over the virtual networking environment, including resource allocation, connectivity, and security.  
     [Click here to visit the Source Link for more features:](#awsVPCfeat)
  2. [AWS Route 53](#awsRoute53) (DNS, Firewall and Sub-Domains): Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Route 53 connects user requests to internet applications running on AWS or on-premises.  
     [Click here to visit the Source Link for more features:](#awsRoute53feat)

1. Compute:  
   1. [AWS EC2](#awsEC2) (Virtual Machines): Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 500 instances and choice of the latest processor, storage, networking, operating system, and purchase model to help you best match the needs of your workload. Windows Server, Web Server, File Server, Network Security, Storage (S3-Buckets), Scalability, Load Balancing, and High Availability.  
      [Click here to visit the Source Link for more features:](#awsEC2feat)
   2. [AWS Lambda](#awsLambda) (Serverless Design): AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and (SaaS) applications, and only pay for what you use. [Click here to visit the Source Link for more features:](#awsLambdafeat)
   3. [AWS EFS](#awsEFS) (Serverless Design, Elastic File System): Amazon Elastic File System (EFS) automatically grows and shrinks as you add and remove files with no need for management or provisioning.  
      Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.  
      [Click here to visit the Source Link for more features:](#awsEFSfeat)
   4. [AWS High Availability](#awsHAresilience) & [Scalability](#awsEC2autoScaling):
      * [ELB](#awsELB) (Elastic Load Balancing): automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more AZs (Availability Zones). It load balances applications, gateways, and networks.

* + - [Click here to visit the Source Link for more features:](#awsELBfeat)
    - [EFS](#awsEFS) (Elastic File System): automatically grows and shrinks as you add and remove files with no need for management or provisioning. With EFS, you only pay for what you use. EFS is another serverless file system that leverages AWS serverless services like, AWS Lambda, AWS Fargate, ECS (Elastic Container Service), and EKS (Elastic Kubernetes Service).

* + - [Click here to visit the Source Link for more features:](#awsEFSfeat)
    - [EBS](#awsEBS) (Elastic Block Storage): is an easy-to-use, scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2).  
      [Click here to visit the Source Link for more features:](#awsEBSfeat)

1. Storage:
   1. [AWS S3](#awsS3): Amazon Simple Storage Service (Amazon S3) is an object storage service offering scalability, data availability, security, and performance. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.  
      [Click here to visit the Source Link for more features:](#awsS3feat)
   2. [S3 Glacier](#awsS3Glacier) (Disaster Recovery): The Amazon S3 Glacier storage classes are purpose-built for data archiving, providing you with the highest performance, most retrieval flexibility, and the lowest cost archive storage in the cloud. All S3 Glacier storage classes provide virtually unlimited scalability and 99.999999999% resilience.  
      [Click here to visit the Source Link for more features:](#awsS3Glacierfeat)
   3. [Amazon RDS](#awsRDS): Relational Database Service (RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud. Choose which RDS engines suit best (Amazon Aurora/MySQL; Amazon Aurora/PostgreSQL; as well as MySQL; MariaDB; Oracle; PostgreSQL; SQL Server).  
      [Click here to visit the Source Link for more features:](#awsRDSfeat)
   4. [Amazon Aurora](#awsRDSAurora) (Serverless Design): Amazon Aurora is a relational database management system (RDBMS) built for the cloud with full MySQL and PostgreSQL compatibility. Aurora provides built-in security, continuous backups (up to 15 read replicas), automated multi-Region replication, and integrations with other AWS services.  
      [Click here to visit the Source Link for more features:](#awsRDSAurorafeat)

4. Cloud Resource Capacity Limits

[Time](#Figure3timeLine) ([Figure 3](#Figure3timeLine)) and [budget](#Figure5ec2instanceCost) ([Figure 5](#Figure5ec2instanceCost)), will dictate our AWS Cloud resource capacity limits. Capacity is only as broad as your imagination and affordable income. The AWS tools and resources for managing your system are immense and precise to your needs. The pricing schedule reflects the diverse array of options and use case scenarios. Eventually your system will become a fully configurable everlasting and evolving entity that is as resilient, cost effective, efficient, and secure as you wish to make it. However, budgetary and time constraints will no doubt have the final say in what fashion the project will roll out”. All things equal and according to plan, the process to begin the execution is described in the [flowchart](#Figure4rolloutProcess) ([Figure 4](#Figure4rolloutProcess)).



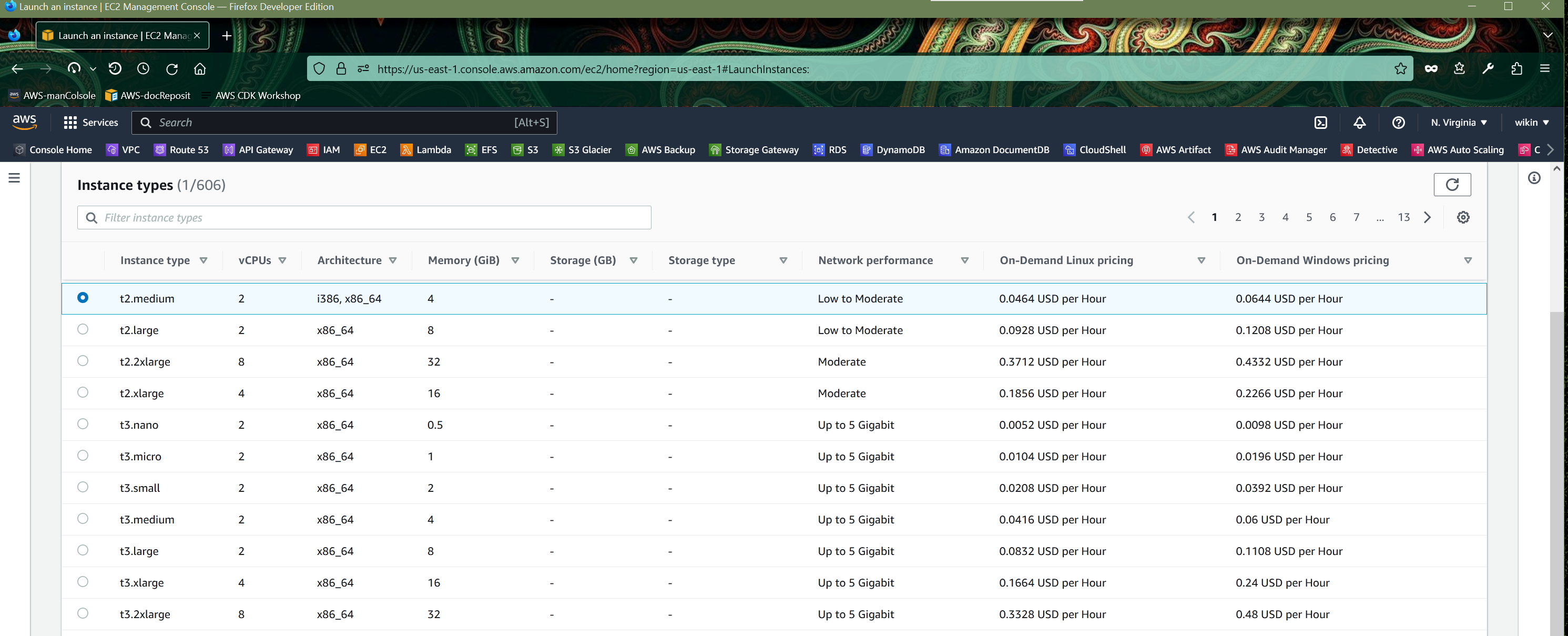
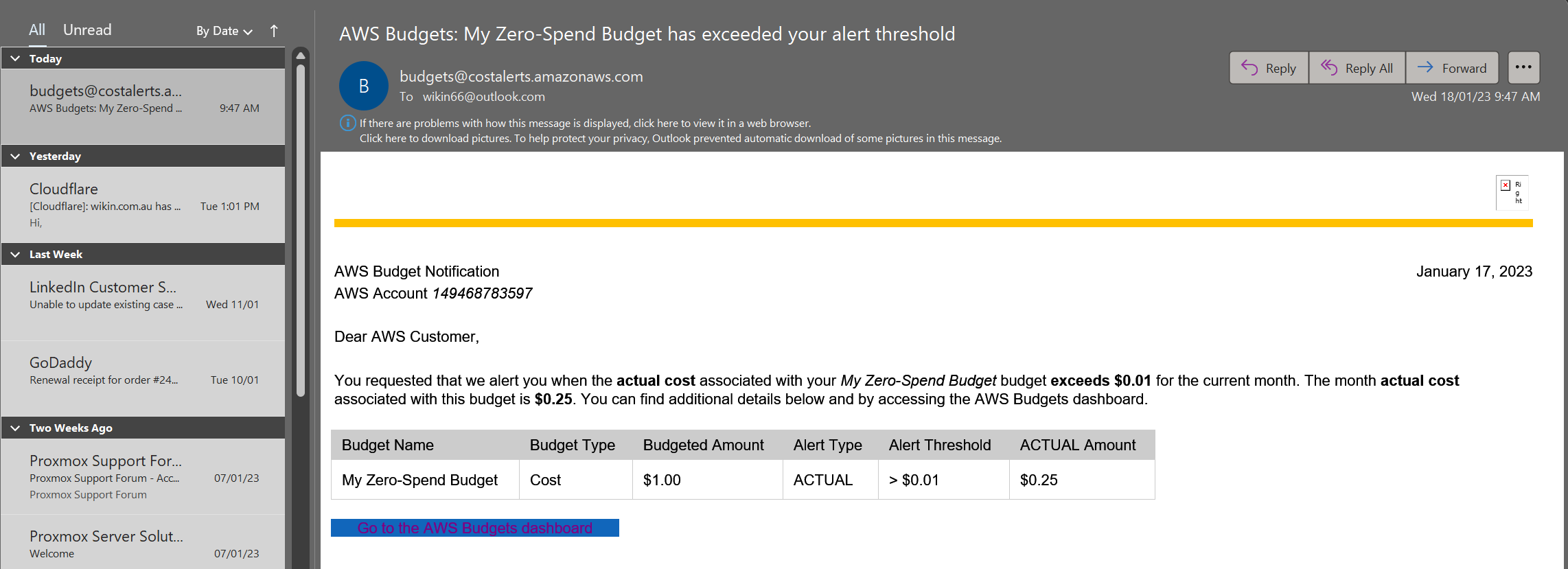
Timeline: Figure 3

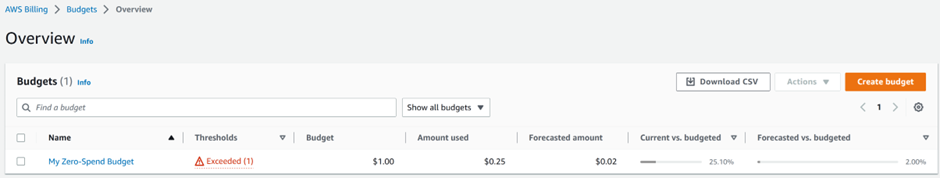
From Phase-1, IaaS is the chosen framework, but it will need many different AWS resources and instances. To start, we need to open our AWS account and set user profiles and security policies. VPC, EC2 and S3 will be employed first. Phase-5 will use AWS resources like Lambda, ELB, EBS, and EFS to build into and augment the final design. Once the cloud infrastructure is in place, we then continue to furnish the design with scripts and automated commands using the AWS Management Console with CloudTrail-API Reference to protect and monitor the entire hybrid design.  
As the new system design migrates and matures, it is feasible and likely that the entire system will establish itself to the cloud as a Platform as a Service (PaaS).  
 

Execution Stage: Figure 4

* Phase-1: Launch barebones cloud system AWS infrastructure. Go Live!  
  Requirements: security, time to deploy, performance, reliability, scalability, communication model best suited to ABC Bank’s hybrid cloud.
  + Connect to AWS Cloud:
    - Launch hybrid network: Selected VPN Connection.
      * Encryption enabled and connection established.
    - Establish SD-WAN Connectivity: ACLs, SSH, ICMP enabled.
      * Ping and TraceRoute connection available.
    - Establish Internet Connectivity: DNS lookup working.
      * AWS Route53 has communication with ITA office.
    - Test Line Capacity: Sufficient internetwork bandwidth.
      * Internetwork speed test performed to confirm bandwidth adequacy across the WAN and SD-WAN.
* Phase-2: Administer user privileges and enable security protocol scripts.  
  Requirements: AWS CLI or SDK, IDE (VS Studio), Remote Access/PuTTy, AWS Organizations, AWS Management Console, AWS IAM, AWS Backup.
* Phase-3: Migrate software services and technologies from on premises system the AWS cloud environment.
  + Assimilate new and old system environments.
    - Combine OUs from Active Directories
      * User profiles established. DNS configured.
* Phase-4: Test and debug unforeseen transitional challenges and conflicts.
* Phase-5: Enhance and augment updated cloud environment with native AWS technologies.
* Phase-6: Create topology map and security plan for new hybrid system and implement new policies and procedures to ABC Banking’s SLA and other policy and regulatory requirements to ABC Bank’s existing Policies and Procedures. Monitoring and reporting will now be performed as per new ABC Bank’s Hybrid AWS Cloud – Security, Monitoring & Logging policy.

Source Link: [Hybrid Connectivity. AWS Whitepaper](#hybridWhitepaper)

Below are some costing considerations considered as part of our AWS Cloud environment. The AWS resources are immense, and the pricing options are to.  
As aforementioned, time and cost are major contributing factors toward our ‘Well-Architected Design’. Until proper stress testing, debugging, and patching have been completed on the capacity and performance limitations of the initial launch system, we will be constantly manipulating and augmenting. The new Hybrid/Cloud design will be furnished over time as job resources are required.  
IaC (Infrastructure as Code) will no doubt be a major implementation that will automate and facilitate many tasks within the cloud environment. Automation reduces human error and makes time for other tasks previous administered by humans. Automation is also used for security as an event monitor and reporter.  
  
  
  




Graphical user interface, application

Description automatically generated

This is the monthly cost based on T2.large x 2 EC2 instances for the highly available Domain Controller/File server. T2.medium x 2 EC2 instances for the highly available Web Server. Gp2 x 2 EBS instances for the highly available Database Storage with RESTful APIs to communicate snapshots and backups.

Graphical user interface, text, application

Description automatically generated

<https://calculator.s3.amazonaws.com/index.html>

EC2 and S3 Instance Costings: Figure 5

**AWS Network/Security Capacity Limits.**  
  
We are bound be AWS governance when it comes to networking our cloud resources, we are limited by the AWS Terms of Agreement and requirements which stipulates quotas and capacity limits. These resources are also subject to typical internet protocols (layer 7/OSI) like HTTPS, TLS, IP/TCP, FTP, SSH etc. AWS has mitigated many security and networking issues by introducing its own network framework with VPCs (Virtual Private Networks). Within the AWS VPC and Route 53 framework there are AWS Security Groups (SGs). SGs are reusable groups of network security rules, and they define incoming and outgoing rules. They can be attached to almost any resource in AWS, such as EC instances, load balancers and RDS databases, and they act like local firewalls for that resource. AWS Security Groups deny all incoming traffic by default, and they allow all outgoing traffic by default. Any deny is treated as primary to overtake the other rules that may be more specific below it. Security Group permissions allow HTTPS traffic from the world wide web, and SSH traffic from inside private IPs. When you deploy your IT resources into AWS Cloud, you benefit from the global network of data centers and architecture built with security in mind. AWS helps you keep your data safe in their highly secure data centers, and there are safeguards in place to help protect customer privacy. There are dozens of compliance programs embedded into AWS to help you meet your compliance requirements for data security. Securing your data on AWS Cloud allows you to maintain the highest standard of security without having to manage your own data centers, which saves you time and money. It also allows you to scale the size of your business quickly, as AWS is designed to keep data safe no matter how big or small your cloud usage is.  
Some of the major security related AWS services, are AWS Identity Access Management or IAM, AWS Web Application Firewall, or WAF, and AWS Trusted Advisor. The following tables list the quotas, formerly referred to as limits, for Amazon VPC resources per Region for your AWS account. Unless indicated otherwise, you can request an increase for these quotas. For some of these quotas, you can view your current quota using the Limits page of the Amazon EC2 console. If you request a quota increase that applies per resource, we increase the quota for all resources in the Region.

|  |  |  |  |
| --- | --- | --- | --- |
| **Amazon VPC Quotas.** | | | |
| *Resource* | *Default Capacity* | *Limit Adjustable?* | *Comments* |
| VPCs per Region | 5 | Yes | Increasing this quota increases the quota on internet gateways per region by the same amount.  You can increase this limit so that you can have 100s of VPCs per Region. |
| Subnets per VPC | 200 | Yes |  |
| IPv4 CIDR blocks per VPC | 5 | Yes (up to 50) | This primary CIDR block and all secondary CIDR blocks count toward this quota. |
| IPv6 CIDR blocks per VPC | 5 | No |  |
| *Link:* [*https://docs.aws.amazon.com/vpc/latest/userguide/amazon-vpc-limits.html*](https://docs.aws.amazon.com/vpc/latest/userguide/amazon-vpc-limits.html) | | | |

|  |  |  |
| --- | --- | --- |
| **Amazon DNS** | | |
| *Resource* | *Capacity* | *Limit* |
| Amazon Route 53 | DNS messaging is subject to factors that affect how you create and use hosted zones and records.   * *Maximum response size.* To comply with DNS standards, responses sent over UDP are no more than 512 bytes in size. Responses exceeding 512 bytes are truncated and the resolver must re-issue the request over TCP. If the resolver supports EDNS0 (as defined in RFC 2671), and advertises the EDNS0 option to Amazon Route 53, Route 53 permits responses up to 4096 bytes over UDP, without truncation. * *Authoritative section processing.* For successful queries, Route 53 appends name server (NS) records for the relevant hosted zone to the Authority section of the DNS response. For names that are not found (NXDOMAIN responses), Route 53 appends the start of authority (SOA) record (as defined in RFC 1035) for the relevant hosted zone to the Authority section of the DNS response. * *Additional section processing.*  Route 53 appends records to the Additional section. If the records are known and appropriate, the service appends A or AAAA records for any target of an MX, CNAME, NS, or SRV record cited in the Answer section. More information on these DNS record types, found on Google: Supported DNS record types. | You can currently use Service Quotas to view and manage only Route 53 and Route 53 Resolver quotas. Domain registration quotas aren't available.  For information on getting current quotas (formerly referred to as "limits"), see the following Route 53 actions:   * GetAccountLimit – Gets quotas on health checks, hosted zones, reusable delegation sets, traffic flow policies, and traffic flow policy records. * GetHostedZoneLimit – Gets quotas on records in a hosted zone and on Amazon VPCs that you can associate with a private hosted zone. * GetReusableDelegationSetLimit – Gets the quota on the number of hosted zones that you can associate with a reusable delegation set. * Each EC2 instance can send 1024 packets per second per network interface to Route 53 Resolver (specifically the .2 address, such as 10.0.0.2 and 169.254.169.253), this quota cannot be increased. * Number of DNS queries per second supported by Route 53 Resolver varies by the type of query, size of the response, and the protocol in use. Information and recommendations for a scalable DNS architecture: [AWS Hybrid DNS with Active Directory White Paper below:](#awsHybridWhitePater) |
| Link:  <https://d1.awsstatic.com/whitepapers/aws-hybrid-dns-with-active-directory.pdf> | | |

|  |  |  |
| --- | --- | --- |
| **AWS Client VPN quotas** | | |
| *Resource* | *Capacity* | *Limit Adjustable?* |
| Authorization rules per Client VPN endpoint | 50 | Yes |
| Client VPN endpoints per region. | 5 | Yes |
| Concurrent client connections per Client VPN endpoint | This value depends on the number of subnet associations per endpoint.   * 1 – 7,000. * 2 – 36,500. * 3 – 66,500. * 4 – 96,500. * 5 – 126,000. | Yes |
| Concurrent operations per Client VPN endpoint. | 10 | No |
| Entries in a client certificate revocation list for Client VPN endpoints. | 20,000 | No |
| Routes per Client VPN endpoint. | 10 | Yes |
| Link: <https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/limits.html> | | |

|  |  |  |
| --- | --- | --- |
| **Default quotas for IAM entities** | | |
| *Resource* | *Capacity* | *Limit* |
| Customer managed policies in an AWS account | 1500 | 5000 |
| Groups in an AWS account | 300 | 500 |
| Instance profiles in an AWS account | 1000 | 5000 |
| Managed policies attached to an IAM role | 10 | 20 |
| Managed policies attached to an IAM user | 10 | 20 |
| Role trust policy length | 2048 characters | 4096 characters |
| Roles in an AWS account | 1000 | 5000 |
| Server certificates stored in an AWS account | 20 | 1000 |
| Virtual MFA devices (assigned or unassigned) in an AWS account | Equal to the user quota for the account | Not applicable |
| Link:  <https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-quotas.html#autoapproved> | | |

**AWS Instances/Virtual Machines Capacity Limits.**

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

Amazon EC2 provides you with a large number of options across ten different instance types, each with one or more size options, organized into distinct instance families optimized for different types of applications.

Amazon EC2 allows you to choose between multiple storage options based on your requirements. Amazon EBS is a durable, block-level storage volume that you can attach to a single, running Amazon EC2 instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates.

|  |  |  |
| --- | --- | --- |
| **Compute Capacity and Limitations** | | |
| Resource | Capacity | Limitation |
| AWS EC2 | * There are multiple ways to pay for Amazon EC2 instances: On-Demand, Savings Plans, Reserved Instances, and Spot Instances. You can also pay for Dedicated Hosts, which provide EC2 instance capacity on physical servers dedicated for your use. For more information on how to optimize your Amazon EC2 spend, visit the Amazon EC2 Cost and Capacity page. * Capacity Reservation requests are granted or denied based on your EC2 instance limits and our available capacity. When you create a Capacity Reservation, AWS reserves the specified capacity for your use. The reserved capacity is charged at the selected instance type’s On-Demand rate whether an instance is running in it or not. You can also use your regional reserved instances with your Capacity Reservations to benefit from billing discounts. | * The maximum number of launch configurations for your account = 200 * The maximum number of Auto Scaling groups for your account. = 500 * On-Demand instances are recommended for:   + Users that prefer the low cost and flexibility of Amazon EC2 without any upfront payment or long-term commitment.   + Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted.   + Applications being developed or tested on Amazon EC2 for the first time. |

**AWS Storage/Back-up Capacity Limits.**

Amazon EC2 allows you to choose between multiple storage options based on your requirements. Amazon EBS is a durable (SSD), block-level storage volume that you can attach to a single, running Amazon EC2 instance. AWS S3 is also called AWS Simple Storage Service. S3 is a storage service. Amazon EBS should be used as a primary storage device for data that requires frequent and granular updates. It is important to back up the data with AWS EBS snapshots.  
Magnetic volumes provide the lowest cost per gigabyte of all EBS volume types. Magnetic volumes are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important.  
RDS runs relational databases in AWS Cloud. It supports: AWS Aurora; PostgreSQL; MySQL; MariaDB; Oracle Database; Microsoft SQL Server.

|  |  |  |
| --- | --- | --- |
| **Storage Capacity and Limitations** | | |
| Resource | Capacity | Limitation |
| AWS S3 | * An Amazon S3 bucket is owned by the AWS account that created it. * When you create a bucket, you choose its name and the AWS Region to create it in. * There is no difference in performance whether you use many buckets or just a few. * Account bucket limit to a maximum of 1,000 buckets. * Multi-Region Access Points in Amazon S3 have the following restrictions and limitations.   + Multi-Region Access Point names:   + Must be unique within a single AWS account.   + Must begin with a number or lowercase letter.   + Must be between 3 - 50 characters long.   + Can't begin or end with a hyphen (-)   + Can't contain underscores (\_), uppercase letters, or periods (.)   + Can't be edited after they are created. * Multi-Region Access Point minimum requirements:   + Transport Layer Security (TLS) v1.2   + Signature Version 4 (SigV4A) | * Bucket ownership is not transferable to another account. * After you create a bucket, you can't change its name or region. * By default, you can create up to 100 buckets in each of your AWS accounts. * Multi-Region Access Point aliases are generated by Amazon S3 and can't be edited or reused. * Multi-Region Access Point limitations:   + IPv6 is not supported.   + Amazon S3 on Outposts buckets are not supported.   + CopyObject is not supported, as a source or destination.   + The S3 Batch Operations feature is not supported.   + Certain AWS SDKs are not supported. To confirm which AWS SDKs are supported for Multi-Region Access Points, see Compatibility with AWS SDKs. * The service quotas for Multi-Region Access Points are as follows:   + There is a maximum of 100 Multi-Region Access Points per account.   + There is a limit of 17 Regions for a single Multi-Region Access Point. |
| Link:  <https://docs.aws.amazon.com/AmazonS3/latest/userguide/BucketRestrictions.html>  <https://docs.aws.amazon.com/AmazonS3/latest/userguide/MultiRegionAccessPointRestrictions.html> | | |

|  |  |  |
| --- | --- | --- |
| Resource | Capacity | Limitation |
| AWS RDS | * By default, you can have up to a total of 40 DB instances. RDS DB, Aurora DB, Amazon Neptune, and Amazon DocumentDB instances apply to this quota. * If your application requires more DB instances, you can request additional DB instances by opening the Service Quotas console. * Relational Database Service (Amazon RDS), choose a quota and follow directions to request a quota increase. For more information, see Requesting a quota increase in the Service Quotas [User Guide](#awsRDS). * For RDS for Oracle and RDS for SQL Server, the read replica limit is 5 per source database for each Region. * Data transfer rates and resilience have significant resources cost vs. speed capacity and limits. The amount vs. speed matrix below outlines hypothetical data rate flow. | * 10 for each SQL Server edition (Enterprise, Standard, Web, and Express) under the "license-included" model. * 10 for Oracle under the "license-included" model. * 40 for MySQL, MariaDB, or PostgreSQL. * 40 for Oracle under the "bring-your-own-license" (BYOL) licensing model. * Nomenclature must contain 1–63 alpha numeric characters or hyphens. First character must be a letter. Can't end with a hyphen or contain two consecutive hyphens. Must be unique for all DB instances per AWS account, per AWSr. |
| Link:  <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Limits.html> | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Amount vs. Speed** | | | |
|  | **100 Mbps** | **1 Gbps** | **10 Gbps** |
| **1 TB** | **30 hours** | **3 hours** | **18 minutes** |
| **10 TB 100 TB** | **12 days** | **30 hours** | **3 hours** |
| **100 TB** | **124 days** | **12 days** | **30 hours** |
| **1 PB** | **3 years** | **124 days** | **12 days** |
| **10 PB** | **34 years** | **3 years** | **124 days** |

Figure 6 Data Amount vs Speed.

Graphical user interface, text, application

Description automatically generated

Figure 7 Data Transfer Pricing Examples

5. Organisational Policies Implemented for the cloud

When managing resources on the AWS cloud, the tagging of resources is a mandatory best practice. All tagging policies are available through AWS Systems Manager and AWS IAM. Most policies exist as code in the documentation page on the AWS website and can be scribed into configuration templates that run AWS instances and containers or virtual machines in the cloud. A policy is an object in AWS that, when associated with an identity or resource, defines their status and permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents that are attached to an IAM identity (user, group of users, or role). Identity-based policies include AWS managed policies, customer managed policies, and inline policies. All preconfigured virtual machines will have most policies delivered via IaC however it is good practice to review a new instance before launching it to the cloud. AWS Code Commit is the default repository for IaC templates.

All application systems must maintain strict controls over their communication network for these purposes:

1. Safeguarding data
   1. Tightly controlling access to network devices through management approval and subsequent audits with AWS IAM, System Manager and TrustedAdvisor. MFA to be configured as per NIST.SP\_800-53\_r5-Security and Privacy Controls for Information Systems.  
      Link: [NIST.SP\_800-53\_r5.prn](https://d.docs.live.net/4a3f1e8ed8318687/Desktop/Diploma%20Of%20IT%20Networking/TERM-3/ICTCLD507_508_ProjectProposal/AT2/NIST.SP_800-53_r5-Security%20and%20Privacy%20Controls%20for%20Information%20Systems.prn)
   2. All users must terminate all unused sessions and remote communications when no business need exists.
   3. Applied ephemeral instances and lifecycle policies with AWS IAM and IaC automated templates.
   4. Logging and monitoring remote access. ACLs applied to allow SSH only. All other traffic has an explicit deny all policy attached by default. Secure Site-to-Site VPN’s have been configured for authorized administrators only.
   5. Delete access key and secret key after they have been downloaded and archived in a secure location. Once the private and public keys have been established, there is no need to keep a record on AWS.
   6. Securing remote access devices. SSH has been configured to allow static IP address from port 22. All other ingress traffic has been explicitly denied. AWS also has Session Manager that can be used as an alternative secure option.
   7. Using AWS IAM for strong authentication and encryption to secure communications. Strong passwords and multifactor identification functions have been provided through IAM with lifecycle IaCs attached. MFA to be configured as per NIST.SP\_800-53\_r5-Security and Privacy Controls for Information Systems.  
      Link: [NIST.SP\_800-53\_r5.prn](https://d.docs.live.net/4a3f1e8ed8318687/Desktop/Diploma%20Of%20IT%20Networking/TERM-3/ICTCLD507_508_ProjectProposal/AT2/NIST.SP_800-53_r5-Security%20and%20Privacy%20Controls%20for%20Information%20Systems.prn)
   8. All confidential bank data will be encrypted while in transit on AWS Site-to-Site VPNs within AWS VPC networks and beyond, or stored on any device on premises, or on a private or public cloud. Confidential bank information, including authentication credentials, will be encrypted while in transit over any public network or wireless network. Key management procedures will be employed that assure the confidentiality, integrity, and availability of cryptographic key material. The use of encryption products complies with both local restrictions and regulations in the relevant region as per NIST.SP\_800-37\_r2-Risk Management Framework for Information Systems and Organizations.  
      Link: [NIST.SP\_800-37\_r2.prn](https://d.docs.live.net/4a3f1e8ed8318687/Desktop/Diploma%20Of%20IT%20Networking/TERM-3/ICTCLD507_508_ProjectProposal/AT2/NIST.SP_800-37_r2-Risk%20Management%20Framework%20for%20Information%20Systems%20and%20Organizations.prn)
2. Scalability
   1. The architecture frameworks and templates must be able to support more than hundred thousand customers, over a million transactions per day, and manage assets of more than USD 4 billion. The architecture must not only meet the current requirements but also anticipate the future needs of a dynamic, growth-focused institution. Amazon EC2 Auto Scaling has an IaC template to automatically launch and terminate EC2 instances based on user-defined scaling policies, scheduled actions, and health checks.
3. Availability
   1. ABC Banking need to operate in a 24/7 environment. The infrastructure must be available to the customers whenever they want, wherever they are, and on whatever channel they prefer to interact. Mission-critical applications such as payment processing and other core banking transaction processing require 99.99% availability. AWS Load Balancing and Autoscaling tools allow for 99.99% availability across the AWS ecosystem. AWS System Manager in conjunction with CloudFormation and CloudWatch as well as snapshots on demand in AWS Elastic Block Storage (EBS).
4. Performance and response times.
   1. Some of the application components need a high throughput, low-latency, real-time performance. Examples include card authorizations and transactions, payment transactions, fraud detection, and securities processing.
   2. Business continuity and disaster recovery
   3. Application systems must have formal documented recovery plans and tools to identify the resources and specify actions that are required to minimize losses if a disruption occurs to the business unit, its supporting group units, applications, or infrastructure components. Solution is same as Question 3 – Availability.
5. Backup and offsite storage.
   1. All application systems must adhere to the backup policy and associated procedures for backing up data in a scheduled and timely manner. AWS System Manager IaC templates available in AWS Code Commit.
   2. Effective controls established to safeguard backed-up data onsite and offsite.
      * Bank data is securely transferred or transported to and from backup locations via AWS Site-to-Site VPN.
      * Conduct periodic tests to ensure that data can be safely recovered from backup devices.
   3. The cloud infrastructure will hold all the critical application and database servers. The existing data center will be used as a warm site with daily data synchronization with the cloud systems.
6. Business Purpose Table.

**AWS Network/Security.**  
AWS cloud security and Compliance is a shared responsibility between AWS and the customer. AWS does the heavy lifting with stringent hardware protection and security. The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall.   
AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet the budgetary, security, and compliance needs of your business. You can group your accounts into organizational units (OUs) and attach different access policies to each OU. You then can attach a policy to that OU that blocks access to services that do not meet those regulatory requirements. You can nest OUs within other OUs to a depth of five levels, providing flexibility in how you structure your account groups. You can use tag policies to maintain consistent tags, including the preferred case treatment of tag keys and tag values. AWS Organizations is a global service with a single endpoint that works from all AWS Regions. You don't need to explicitly select a region to operate in. As a managed service, AWS Organizations is protected by the AWS global network security procedures that are described in [Best Practices for Security, Identity, and Compliance](#awsBestPractice).  
Amazon Virtual Private Cloud (Amazon VPC) lets you provision a private, isolated section of the AWS Cloud where you can launch AWS resources in a virtual network. Amazon Route 53 Resolver responds recursively to DNS queries from AWS resources for public records, Amazon VPC-specific DNS names, and Amazon Route 53 private hosted zones, and is available by default in all VPCs.

|  |  |  |
| --- | --- | --- |
| **ABC Resource** | **AWS Service** | **Requirements** |
| Rack mounted server, router, and switches.  CIA Multi factor authentication  DNS, DMZ | VPN  VPC  Cognito  Route 53 | Secure encrypted connection  Administration control centre.  Need to determine cross account authentication. MFA. CIA  Network host, Router, DNS, Monitor/Log, Security, DMZ, Software Firewall, NAT/PAT |
| Active Directory | IAM | Role and permission allocation and enforce predefined condition. Policy repository and control. |
| Firewall | EC2 - Bastion Host or AWS WAF | ACLs, Policies, Honey Pot, DMZ |
| NFS (Network File System) | EFS | Restrict access to a file system and Encryption. Allow/Deny ACLs. |

**AWS Instances/Virtual Machines.**

Instances in AWS are basically virtual environments. These virtual environments are isolated from the underlying base OS. It’s an On-demand service, i.e., a user can rent the virtual server (instances) on an hourly base and deploy their applications on it. EC2 Instances are highly scalable, meaning, you can scale up or scale down based on your requirement dynamically. Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones (AZs). Using EC2 Instances as your cloud computing environment eliminates the need to invest in hardware and software dependencies.

|  |  |  |
| --- | --- | --- |
| **ABC Resource** | **AWS Service** | **Requirements** |
| Servers x 5 Domain Controller. Web Server. File server. Database Server Bastion Host. | EC2 x 5 Windows Sever AD DC.  AWS Linux/Apache WS. AWS Linux Ubuntu FS.  AWS Linux EC2/S3 Dbs.  AWS Linux | Equivalent: t2.lge, x86-64, 8GB, 2 vCPUs. Equivalent: T2.med, x86-64, 4GB, 2 vCPUs. Equivalent: T2.med, x86-64, 4GB, 2 vCPUs. Equivalent: MySQL Relational database. Equivalent: T2.nano, x86-64, 0.5GB, 1 vCPUs. |

**AWS Storage/Back-up.**

The main reason to implement a hybrid storage environment is cost savings. Purchasing network storage can be increasingly expensive to acquire and maintain. By using cloud storage, you would forgo the large upfront costs. The hybrid model is also commonly used as a backup solution using Glacier or Glacier Deep Archive for cheap long-term storage.

|  |  |  |
| --- | --- | --- |
| **ABC Resource** | **AWS Service** | **Requirements** |
| Rack mounted servers and database storage.  Active Directory, NFS, Policies, Windows Event Monitor, Share policies | S3  AWS CloudTrail, CloudWatch, or EC2 + S3 (DataSync FSx). | Storage and backup of all data and resilience- 99.9999999%.  Monitoring, logging, and reporting. Budgeting, alerting, and automated resiliency. |
| MySQL Relational Database | RDS | Database equivalent to current on-premises system. |
| Emergency back-up (VMs) | EBS | On-hand disk image backups |
| Automated filing system | EFS | Data encryption, restrict access to a file system, implement access control, Monitor, and manage disk size, space, and scalability. |

7. Repeatable Tasks and Automation Tools.

(PLEASE NOTE: AWS Tools are highlighted in grey)

A computer with internet and a plain text editor for manipulating scripts and the CLI (Command Line Interface) or SDK from which we can launch common network testing tools like ping and Trace Route, NMAP and NetStat. Then there is a plethora of AWS and third-party testing tools that you can use. The AWS Management Console to create an IaC stack from an example template from the AWS CloudFormation sample template library to create basic templates. A template is a text file that describes a stack, or a collection of AWS resources you want to deploy together as a group. You use these templates to define all the AWS resources you want in your stack. This can include Amazon Elastic Compute Cloud instances (EC2), Amazon Relational Database Service (RDS) DB Instances, and other resources. For a list of resource types, see the AWS resource and property types reference. You can author AWS CloudFormation templates in JSON or YAML formats. You can store all your templates, scripts and secrets in AWS Code Commit as the repository to support all AWS CloudFormation features and functions for both formats, including in AWS CloudFormation Designer. Also consider that YAML inherently provides some features, such as commenting, that aren't available in JSON.

1. Format Version (optional)  
   The AWS CloudFormation template version that the template conforms to. The template format version isn't the same as the API or WSDL version. The template format version can change independently of the API and WSDL versions.
2. Description (optional)  
   A text string that describes the template. This section must always follow the template format version section.
3. Metadata (optional)  
   Objects that provide additional information about the template.
4. Parameters (optional)  
   You use the Parameters section to declare values that can be passed to the template when you create the stack. A parameter is an effective way to specify sensitive information, such as users’ names and passwords, that you don't want to store in the template itself. It's also a way to specify information that might be unique to the specific application or configuration you are deploying, for example, a domain name or an instance type. You can refer to parameters from the Resources and Outputs sections of the template.
5. Rules (optional)  
   Validates a parameter or a combination of parameters passed to a template during a stack creation or stack update.
6. Mappings (optional)  
   A mapping of keys and associated values that you can use to specify conditional parameter values, like a lookup table. You can match a key to a corresponding value by using the Fn::FindInMap intrinsic function in the Resources and Outputs sections.
7. Conditions (optional)  
   Conditions that control whether certain resources are created or whether certain resource properties are assigned a value during stack creation or update. For example, you could conditionally create a resource that depends on whether the stack is for a production or test environment.
8. Transform (optional)  
   For serverless applications like, AWS Lambda based applications, it specifies the version of the AWS Serverless Application Model (AWS SAM) to use. When you specify a transform, you can use AWS SAM syntax to declare resources in your template. The model defines the syntax that you can use and how it's processed. You can also use AWS::Include transforms to work with template snippets that are stored separately from the main AWS CloudFormation template. You can store your snippet files in an Amazon S3 bucket and then reuse the functions across multiple templates.
9. Resources (required)  
   Specifies the stack resources and their properties, such as an Amazon Elastic Compute Cloud instance or an Amazon Simple Storage Service bucket. You can refer to resources in the Resources and Outputs sections of the template.
10. Outputs (optional)  
    Describes the values that are returned whenever you view your stack's properties. For example, you can declare an output for an S3 bucket name and then call the aws cloudformation describe-stacks AWS CLI command to view the name.

The following examples show an AWS CloudFormation template structure and its sections. The following example shows a YAML template using inline comments.

---

AWSTemplateFormatVersion: "*version date*"

Description:

*String*

Metadata:

*template metadata*

Parameters:

*set of parameters*

Rules:

*set of rules*

Mappings:

*set of mappings*

Conditions:

*set of conditions*

Transform:

*set of transforms*

Resources:

*set of resources*

Outputs:

*set of outputs*

**YAML**

Figure 8 YAML

The following example shows a JSON-formatted template fragment using quotes- “”, colons- : and curly braces- {}.

{

"AWSTemplateFormatVersion" : "*version date*",

"Description" : "*JSON string*",

"Metadata" : {

*template metadata*

},

"Parameters" : {

*set of parameters*

},

"Rules" : {

*set of rules*

},

"Mappings" : {

*set of mappings*

},

"Conditions" : {

*set of conditions*

},

"Transform" : {

*set of transforms*

},

"Resources" : {

*set of resources*

},

"Outputs" : {

*set of outputs*

}

}

**JSON**

Figure 9 JSON

8. Steps Required to Provision Resources.

1. Choose a home Region  
   The home region at which you set a root account can’t be changed once chosen. Instances and services can be hosted in different regions, and you can have many root accounts however you cannot move an account to another region unless you first delete it and reopen in the region of choice.
2. Choose a configuration type  
   AWS provides a library of configuration types that automate common setup tasks and deploy configurations for services based on AWS best practices. We will be implementing a hybrid system utilizing AWS services.

Our IaaS environment will begin as a framework in the cloud and then augmented and enhanced over time. Eventually our environment will be PaaS as many of the current on premises resources are Refactored, Rehosted, Replatformed, Repurchased, or Retired.

1. Specify configuration options  
   Specify your preferred values for the options of the configuration type you chose, and which resources to target with your configuration. You can deploy configurations to a single AWS Region for your current account or integrate with AWS Organizations so you can target multiple organization units (OUs) and Regions.
   1. Determine cross account authentication.
   2. Evaluate the current authentication infrastructure.
   3. Analyze the AWS resources at an account level.
   4. Decide on an auditing policy for both authentication and access.
   5. Determine a way to design a network. This includes things like:
      1. Outlining IP strategy for VPCs.
      2. Determining DNS strategy.
      3. Classifying network traffic and security.
      4. Determine connectivity needs for hybrid environments.
      5. Determine a way to audit network traffic.
2. Deploy and review  
   Deploy our configuration and review the summarized results. We can use the summary page to troubleshoot failed deployments, and to view more details about your configuration. The summary page also notifies us when new versions of relevant software become available, as well as newly supported features and Regions. Please see: [Figure 3 (Gantt Chart)](#Figure3timeLine) and [Figure 4 (Flow Chart)](#Figure4rolloutProcess).

9. Provisioned Resources and Infrastructure Testing Model.

Straight up before any major configurations can be made and debugged, you must test the network connectivity. For this the most common tools are listed below or else please read on to learn of more intricate processes:

* Ping: tracert; NMAP; nslookup
* Bash/Text editor/IDE | Test Scripts
* Command Line Interface (CLI or SDK)
* AWS DataSync, MariaDB, MySQL, DynamoDB
* Kali Linux | Burpsuite | sqlmap | Web Browser

There are many software tools and frameworks available for automating the process of running tests, but proper infrastructure must be in place. This involves provisioning infrastructure resources, initializing the resources with a sample dataset, deploying the software to be tested, orchestrating the test runs, and collecting results. The challenge is not only to have enough resources to deploy the complete application with all the different servers or services it might require, but to be able to initialize the test environment with the right software and the right data over and over.  
Test environments should be identical between test runs, otherwise, it is more difficult to compare results. You can create and manage test environments programmatically using the AWS abundant tool sets.

Provisioning complex environments or platforms composed of multiple resources can be done using AWS CloudFormation stacks, or some of the other automation tools and resources. Some tools for testing infrastructure process listed below:

* API Gateway
* AWS CodePipeline
* AWS CloudFormation
* AWS CodeCommit
* AWS Systems Manager

Defined runbooks through Systems Manager automation documents to automate critical response and provided detailed steps to first responders. Used AWS CloudFormation and Incidents Manager to automate the creation of response plans. IaC templates are available in AWS Code Commit.

Scheduled script testing policy. Run scripts in virtualized environment to ensure integrity on various platforms and versions. All anomalies to be reported to the administrator on discovery.

AWS offers you the ability to experiment and iterate with a rapidly changeable infrastructure. Your project teams are free to use inexpensive capacity to perform any kind of tests or to experiment with new ideas, with no upfront expenses or long-term commitments, making AWS a platform of choice for development and test.

10. Two policies for each of the following:

1. Tagging policy and categorise cloud resources:  
   1. Mandatory tagging policies must be always adhered to:
      * All resources deployed on AWS cloud infrastructure must have a name, department and description tag applied in the correct category.
      * All tagging categories and policies are available through AWS Systems Manager and AWS IAM. Most policies exist as YAML or JSON statements in CodeCommit and the documentation pages on the AWS website. Please find scripts:   
        [Figure 8 YAML](#Figure8YAML) and [Figure 9 JSON](#Figure9JSON).
      * Automate configuration templates that run AWS instances and containers or virtual machines in the cloud.
      * Tags applied automatically must be manually checked by an administrator after launching a new instance.
      * Attribute-based access control to administrator-managed attributes such as tags attached to both AWS resources and AWS identities.
      * AWS Organizations taggable resources include AWS accounts, the organization's root, organizational units (OUs), or policies.
      * Organizations resources, must use tagging policy to control who can access those resources.
      * Add ‘Condition’ elements to AWS Identity and Access Management (IAM) permissions policy statements that check whether certain tag keys and values are present before allowing the action.
      * The condition of a statement must be manually checked for accuracy before launch of an instance
   2. Educate account administrators as follows:
      * Communicate your tagging strategy.
      * Emphasize that administrators need to use tags on specific resource types. This is important, as untagged resources don't show as noncompliant in compliance results.
      * Provide guidance on checking compliance with tag policies.
      * Instruct administrators to find and correct noncompliant tags on resources in their account.
      * Encourage all staff to become familiar with the security principals outlined in NIST.SP\_800-181\_r1-Workforce Framework for Cyber Security (NICE Framework)  
        Link: [NIST.SP\_800-181\_r1-Workforce(NICE Framework)](https://d.docs.live.net/4a3f1e8ed8318687/Desktop/Diploma%20Of%20IT%20Networking/TERM-3/ICTCLD507_508_ProjectProposal/AT2/NIST.SP_800-181_r1-Workforce%20Framework%20for%20Cyber%20Security%20(NICE%20Framework).prn)
2. Configuration policy for cloud resources:
   1. Architecture and design must not only meet the current system requirements but also anticipate the future needs of a dynamic, growth-focused institution.
      * The infrastructure must boost the business, not limit it.
      * Configurations must provide a demonstrated ability to scale and align to the business objectives as the financial institution grows.
      * Policies dedicated to Load balancing and Auto Scaling must be implemented as default settings throughout architecture and concept design of all instances. (Within budget).
   2. All users must be certified within ABC Banking’s client management software.
      * While using AWS services, IAM must be configured to accommodate all system permissions for users and groups.
      * All users must have AWS IAM roles applied and meet security requirements by adhering to NIST.SP\_800-53\_r5-Security and Privacy Controls for Information Systems.
      * All clients must use MFA and secure passwords at login. New passwords must be generated at the end of each quarter.
      * Password manager must be employed at all times.  
        Link: [NIST.SP\_800-53\_r5.prn](https://d.docs.live.net/4a3f1e8ed8318687/Desktop/Diploma%20Of%20IT%20Networking/TERM-3/ICTCLD507_508_ProjectProposal/AT2/NIST.SP_800-53_r5-Security%20and%20Privacy%20Controls%20for%20Information%20Systems.prn)
3. Data retention policy in cloud environment:
   1. Configure all EC2 and S3 instances to synchronize with ABC Banking’s secret warm site at set intervals via encrypted Site-to-Site connection on the ABC Banking Hybrid Cloud.
      * Amazon EBS to be used for data that must be quickly accessible and requires long-term persistence.
      * Back up the data on Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots within budget constraints.
      * IaC templates must be recreated and maintained regularly and made available on AWS Code Commit. (Schedule TBA)
   2. Use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs.
      * Configure AWS services for monitoring CloudWatch Events and use AWS CloudTrail and Amazon Data Lifecycle Manager.
      * Each EC2 instance must be a minimum of 20GB and configured via policies applied through AWS Organizations.
      * Provide a complete backup solution for Amazon EC2 including S3 instances with EFS and individual EBS volumes.

Source Links:

1. <https://docs.aws.amazon.com/index.html>
2. <https://aws.amazon.com/blogs/architecture/extending-your-saas-platform-with-aws-lambda/> | <https://aws.amazon.com/solutions/resilience/>

1. <https://docs.aws.amazon.com/vpn/> | <https://aws.amazon.com/vpn/features/>

1. <https://docs.aws.amazon.com/vpc/> | <https://aws.amazon.com/vpc/features/>

1. <https://aws.amazon.com/about-aws/whats-new/2019/03/aws-systems-manager-announces-service-level-agreement/> | <https://aws.amazon.com/console/>
2. <https://aws.amazon.com/about-aws/global-infrastructure/> | <https://aws.amazon.com/console/features/>

1. <https://aws.amazon.com/route53/> | <https://aws.amazon.com/route53/features/>
2. <https://aws.amazon.com/ec2/> | <https://aws.amazon.com/ec2/autoscaling/> | <https://aws.amazon.com/ec2/autoscaling/features/>

1. <https://aws.amazon.com/efs/> | <https://aws.amazon.com/efs/features/>

1. <https://aws.amazon.com/elasticloadbalancing/> | <https://aws.amazon.com/elasticloadbalancing/features/>

1. <https://aws.amazon.com/ebs/> | <https://aws.amazon.com/ebs/features/>

1. <https://aws.amazon.com/hybrid/> | <https://aws.amazon.com/hybrid/services/>

1. <https://aws.amazon.com/cloudtrail/> | <https://aws.amazon.com/cloudtrail/features/>

1. <https://aws.amazon.com/config/> | <https://aws.amazon.com/config/features/>

1. <https://aws.amazon.com/cloudwatch/> | <https://aws.amazon.com/cloudwatch/features/>

1. <https://aws.amazon.com/ec2/> | <https://aws.amazon.com/ec2/features/>

1. <https://aws.amazon.com/lambda/> | <https://aws.amazon.com/lambda/features/>

1. <https://aws.amazon.com/s3/> | <https://aws.amazon.com/s3/features/>

1. <https://aws.amazon.com/s3/storage-classes/glacier/> | <https://aws.amazon.com/s3/storage-classes/glacier/features/>

1. <https://aws.amazon.com/rds/> | <https://aws.amazon.com/rds/features/>

1. <https://aws.amazon.com/rds/aurora/> | <https://aws.amazon.com/rds/aurora/features/>

1. <https://docs.aws.amazon.com/awscloudtrail/latest/APIReference/Welcome.html>

1. <https://docs.aws.amazon.com/whitepapers/latest/hybrid-connectivity/introduction.html>

1. <https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html> | [https://aws.amazon.com/architecture/security-identity-compliance/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=\*all&awsf.methodology=\*all](https://aws.amazon.com/architecture/security-identity-compliance/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all) <https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html#features>

1. <https://docs.aws.amazon.com/organizations/latest/userguide/orgs_integrate_services_list.html>
2. <https://www.pcisecuritystandards.org/about_us/> | <https://www.pcisecuritystandards.org/about_us/>