**What is identity and access management (IAM)?**

Identity and access management ensures that the right people, machines, and software components get access to the right resources at the right time. First, the person, machine, or software component proves they're who or what they claim to be. Then, the person, machine, or software component is allowed or denied access to or use of certain resources.

Here are some fundamental concepts to help you understand identity and access management:

**Identity**

A digital identity is a collection of unique identifiers or attributes that represent a human, software component, machine, asset, or resource in a computer system. An identifier can be:

* An email address
* Sign-in credentials (username/password)
* Bank account number
* Government issued ID
* MAC address or IP address

Identities are used to authenticate and authorize access to resources, communicate with other humans, conduct transactions, and other purposes.

At a high level, there are three types of identities:

* Human identities represent people such as employees (internal workers and front line workers) and external users (customers, consultants, vendors, and partners).
* Workload identities represent software workloads such as an application, service, script, or container.
* Device identities represent devices such as desktop computers, mobile phones, IoT sensors, and IoT managed devices. Device identities are distinct from human identities.

**Authentication**

Authentication is the process of challenging a person, software component, or hardware device for credentials in order to verify their identity, or prove they're who or what they claim to be. Authentication typically requires the use of credentials (like username and password, fingerprints, certificates, or one-time passcodes). Authentication is sometimes shortened to *AuthN*.

Multi-factor authentication (MFA) is a security measure that requires users to provide more than one piece of evidence to verify their identities, such as:

* Something they know, for example a password.
* Something they have, like a badge or [security token](https://learn.microsoft.com/en-us/azure/active-directory/develop/security-tokens).
* Something they are, like a biometric (fingerprint or face).

Single sign-on (SSO) allows users to authenticate their identity once and then later silently authenticate when accessing various resources that rely on the same identity. Once authenticated, the IAM system acts as the source of identity truth for the other resources available to the user. It removes the need for signing on to multiple, separate target systems.

**Authorization**

Authorization validates that the user, machine, or software component has been granted access to certain resources. Authorization is sometimes shortened to *AuthZ*.

**Authentication vs. authorization**

The terms authentication and authorization are sometimes used interchangeably, because they often seem like a single experience to users. They're actually two separate processes:

* Authentication proves the identity of a user, machine, or software component
* Authorization grants or denies the user, machine, or software component access to certain resources

Here's a quick overview of authentication and authorization:

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| Can be thought of as a gatekeeper, allowing access only to those who provide valid credentials. | Can be thought of as a guard, ensuring that only those with the proper clearance can enter certain areas. |
| Verifies whether a user, machine, or software is who or what they claim to be. | Determines if the user, machine, or software is allowed to access a particular resource. |
| Challenges the user, machine, or software for verifiable credentials (for example, passwords, biometric identifiers, or certificates). | Determines what level of access a user, machine, or software has. |
| Done before authorization. | Done after successful authentication. |
| Information is transferred in an ID token. | Information is transferred in an access token. |
| Often uses the OpenID Connect (OIDC) (which is built on the OAuth 2.0 protocol) or SAML protocols. | Often uses the OAuth 2.0 protocol. |

For more detailed information, read [Authentication vs. authorization](https://learn.microsoft.com/en-us/azure/active-directory/develop/authentication-vs-authorization).

**Example**

Suppose you want to spend the night in a hotel. You can think of authentication and authorization as the security system for the hotel building. Users are people who want to stay at the hotel, resources are the rooms or areas that people want to use. Hotel staff is another type of user.

If you're staying at the hotel, you first go to reception to start the "authentication process". You show an identification card and credit card and the receptionist matches your ID against the online reservation. After the receptionist has verified who you are, the receptionist grants you permission to access the room you've been assigned. You're given a keycard and can go now to your room.

The doors to the hotel rooms and other areas have keycard sensors. Swiping the keycard in front of a sensor is the "authorization process". The keycard only lets you open the doors to rooms you're permitted to access, such as your hotel room and the hotel exercise room. If you swipe your keycard to enter any other hotel guest room, your access is denied. Individual [permissions](https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/users-default-permissions?context=/azure/active-directory/roles/context/ugr-context), such as accessing the exercise room and a specific guest room, are collected into [roles](https://learn.microsoft.com/en-us/azure/active-directory/roles/concept-understand-roles) which can be granted to individual users. When you're staying at the hotel, you're granted the Hotel Patron role. Hotel room service staff would be granted the Hotel Room Service role. This role permits access to all hotel guest rooms (but only between 11am and 4pm), the laundry room, and the supply closets on each floor.

**Identity provider**

An identity provider creates, maintains, and manages identity information while offering authentication, authorization, and auditing services.

With modern authentication, all services, including all authentication services, are supplied by a central identity provider. Information that's used to authenticate the user with the server is stored and managed centrally by the identity provider.

With a central identity provider, organizations can establish authentication and authorization policies, monitor user behavior, identify suspicious activities, and reduce malicious attacks.

[Microsoft Azure Active Directory](https://learn.microsoft.com/en-us/azure/active-directory/) is an example of a cloud-based identity provider. Other examples include Twitter, Google, Amazon, LinkedIn, and GitHub.

Get Users Azure AD

# Connect to Azure AD

#Connect-MsolService

# Retrieve user information and select the required properties

$users = Get-MsolUser | Select-Object FirstName, LastName, UserPrincipalName, PhoneNumber, Title, Country, City, DisplayName, UsageLocation

# Export the data to a CSV file

$csvFilePath = "C:\Kamalakar\Training Meterials\ECP\August\Users-11Aug.csv"

$users | Export-Csv -Path $csvFilePath -NoTypeInformation

Write-Host "User information exported to $csvFilePath successfully."

Upload Blob data using Powershell

#Connect-AzAccount

#Get-AzSubscription -SubscriptionName "SUB-ECP-PROD-01" | Set-AzContext

$storageaccountname = "datatstaccnt"

$Storageaccountkey = "4xgyk/HGVk5P7LcJxTuzG/BZuo2F1qsbuhIirpFC93GPL5u0D0KYMQWq1ZNJ4bTUcdnw0v5TWY4Q+ASt29APpw=="

$containername = "documents"

$localpath = "C:\Kamalakar\Training Meterials\Enterprise Cloud Expert Course.pdf"

$blobname = "Enterprise Cloud Expert Course.pdf"

$context = New-AzStorageContext -StorageAccountName $storageaccountname -StorageAccountKey $Storageaccountkey

Set-AzStorageBlobContent -Container $containername -File $localpath -Blob $blobname -Context $context -Verbose

Get RSV Inventory

# Define your Azure subscription ID

#$subscriptionId = "Free Trial Account"

#Get-AzSubscription -SubscriptionName $subscriptionId | Set-AzContext

$subs = Get-AzSubscription

$outputData = @()

foreach ($sub in $subs)

{

Get-AzSubscription -SubscriptionName $sub.name | Set-AzContext

# Define the API version

$apiVersion = "2023-04-01"

# Authenticate and get the access token

$accessToken = Get-AzAccessToken

# Create the authorization header

$headers = @{

    "Authorization" = "Bearer $($accessToken.Token)"

}

# Get all Recovery Services Vaults in the subscription

$vaults = Get-AzRecoveryServicesVault

# Define the names to filter

$filterNames = @("GRSStorageUsage", "LRSStorageUsage", "RAGRSStorageUsage", "ZRSStorageUsage", "RAGZRSStorageUsage", "ArchiveGRSStorageUsage", "ArchiveLRSStorageUsage","AzureBackupServerProtectedItemCount","DPMProtectedItemCount", "IaaSVMProtectedItemCount", "MABContainersCount", "MABProtectedItemCount", "ManagedInstances", "ProtectedItem", "ProtectedItemCount", "RegisteredServersCount")

# Create an array to store the output data

$subscriptionData = @()

# Loop through each vault and fetch the data usage

foreach ($vault in $vaults) {

    # Define the REST API URL for the current vault

    $uri = "[https://management.azure.com](https://management.azure.com/)" + $vault.ID + "/usages?api-version=$apiVersion"

    # Make the REST API call

    $response = Invoke-RestMethod -Uri $uri -Headers $headers -Method Get

    # Loop through the response and filter the desired values

    foreach ($item in $response.value) {

      if ($filterNames -contains $item.name.value) {

            $data = [PSCustomObject]@{

                "RecoveryServicesVaultName" = $vault.Name

                "Type" = $item.name.value

               "CurrentValue" = $item.currentValue

               "Unit" =$item.unit

               "Location" = $vault.Location

               "SubscriptionName" = $sub.Name

                "ResourceGroupName" = $vault.ResourceGroupName

            }

            # Add the custom object to the output array

           $subscriptionData += $data

       }

    }

}

    $outputData += $subscriptionData

}

$outputData | Where-Object { $\_.CurrentValue -ne "0" } | Export-Csv -Path "RecoveryServicesVaults\_details.csv" -NoTypeInformation -NoClobber

Write-Host "CSV export completed. The report has been saved to RecoveryServicesVaults\_Report.csv"

Authenticate with Automation account

Please enable appropriate RBAC permissions to the system identity of this automation account. Otherwise, the runbook may fail..."

try

{

    "Logging in to Azure..."

    Connect-AzAccount -Identity

}

catch {

    Write-Error -Message $\_.Exception

    throw $\_.Exception

}

#Get all ARM resources from all resource groups

$ResourceGroups = Get-AzResourceGroup

foreach ($ResourceGroup in $ResourceGroups)

{

    Write-Output ("Showing resources in resource group " + $ResourceGroup.ResourceGroupName)

    $Resources = Get-AzResource -ResourceGroupName $ResourceGroup.ResourceGroupName

    foreach ($Resource in $Resources)

    {

        Write-Output ($Resource.Name + " of type " +  $Resource.ResourceType)

    }

    Write-Output ("")

}

Azure PowerShell Use cases - Azure Admin

Azure Operations team can utilize Azure PowerShell for various tasks and challenges in more detail. Here's an expanded explanation of the use cases:

1. Virtual Machine Management:
   * *Use Case*: Azure Operations teams can use Azure PowerShell to create, configure, and manage virtual machines (VMs) efficiently. This includes provisioning new VMs, starting and stopping existing ones, and resizing VMs to meet changing workload requirements. PowerShell allows automation of these tasks, saving time and reducing human errors.
2. Resource Monitoring and Alerting:
   * *Use Case*: With Azure PowerShell, Operations teams can set up custom monitoring and alerting rules for Azure resources. They can create and manage alerts that trigger notifications or actions when specific resource metrics or conditions exceed defined thresholds, helping to proactively address performance and availability issues.
3. Scaling Resources:
   * *Use Case*: Using Azure PowerShell, teams can automate the scaling of Azure resources. This includes scaling VMs, Azure App Service instances, and other services based on performance metrics or schedules. For example, during high traffic periods, PowerShell scripts can automatically add more VMs to distribute the load.
4. Backup and Disaster Recovery:
   * *Use Case*: Azure Operations teams can use Azure PowerShell to implement and manage backup and disaster recovery strategies. They can automate backup schedules for VMs, databases, and files, ensuring data resilience and simplified recovery in case of outages or data loss.
5. Azure Policy Enforcement:
   * *Use Case*: PowerShell scripts can be used to enforce organization-specific policies across the Azure environment. Teams can automate policy enforcement for security, compliance, and naming conventions to maintain a consistent and secure infrastructure.
6. Network Security:
   * *Use Case*: Azure PowerShell empowers Operations teams to configure and manage network security settings. This includes creating and managing Network Security Groups (NSGs), defining Virtual Network (VNet) peering relationships, and configuring Azure Firewall rules to secure network traffic.
7. Azure Active Directory Management:
   * *Use Case*: Azure AD management is crucial for Operations teams. They can automate user provisioning, deprovisioning, password resets, and role assignments using PowerShell scripts, ensuring that access control is efficient and secure.
8. Role-Based Access Control (RBAC):
   * *Use Case*: Operations teams can define and manage custom RBAC roles with PowerShell, tailoring permissions to the specific needs of the organization. This allows fine-grained control over who can access and modify Azure resources, enhancing security and compliance.
9. Resource Tagging and Management:
   * *Use Case*: PowerShell can automate resource tagging strategies to help Operations teams organize and track resources for billing, reporting, and compliance purposes. Consistent tagging simplifies cost allocation and resource management.
10. Container Orchestration:
    * *Use Case*: For organizations using Azure Kubernetes Service (AKS), Operations teams can manage containerized applications efficiently. They can automate tasks such as deployment, scaling, and updates using Azure PowerShell scripts.
11. Azure Key Vault Secrets Management:
    * *Use Case*: Azure Operations teams can use PowerShell to store, retrieve, and manage sensitive data securely in Azure Key Vault. This ensures the protection of secrets like API keys, certificates, and passwords.
12. Azure Policy Compliance Reporting:
    * *Use Case*: Teams can use PowerShell to generate compliance reports, demonstrating adherence to Azure policies and industry regulations. This is crucial for audits and maintaining a secure and compliant Azure environment.
13. Data Backup and Archiving:
    * *Use Case*: Operations teams can use Azure PowerShell to automate data backup and long-term archiving in Azure Blob Storage. This ensures data durability and availability for the long term.
14. Cost Management and Optimization:
    * *Use Case*: Using PowerShell, teams can analyze usage data, identify underutilized resources, and implement cost-saving measures. They can also automate the shutdown of non-essential resources during non-business hours to reduce costs.
15. Azure DevOps Integration:
    * *Use Case*: Operations teams can automate CI/CD processes with Azure DevOps and PowerShell. This streamlines application and infrastructure deployment, ensuring a continuous and efficient release pipeline.

Incorporating Azure PowerShell into daily operations enables Azure Operations teams to streamline tasks, reduce manual effort, enhance security, and optimize resource utilization, ultimately ensuring the smooth operation of Azure infrastructure.

Interview 1 ->

Understand the profile

* Skills -  Azure AD , Basics of VM
* Experience  - 13 years
* Tasks -
* Collaboration -

Technical

MOCK Interview - Azure L2+ & L3

Exp - 13 years

Azure 4+ years

   Cloud Infra & Hybrid, Identity Solutions

   DevOps Basics

    Power BI reports

10 members

 Cloud Support Team/operations Team

 Incidents/Requests

   On-Premise exp

           hybrid Setup

Daily tasks

    tickets-> Azure AD issues

                   Azure Services

Identity Solutions-> Sign In Issues, Enteprise Applications , Tshoot sign in issues, Security Solutions

Azure Resources

  Tshooting VMs

   Create VMs ->>> Terraform , PS , ARM Templates

   VM-> **Connectivity**, Performance issues, backup issues, jobs status, monitor, health checks of VM's

Monitoring -> Metrics (CPU, DISK, Memory), Alerts setup,  Few email alerts , critical ones are Auto tickets

AVSETs, AVZONES

Backup VM's ->

Site Recovery Replications ->

Azure VM Patching - Azure Update Manager (New Solution - POC) , SCCM is being used

NSG - ASG, AZ Firewall, App Gateways, Related config/tshootings

Disk Operations ->

Storage Operations ->

1. If user from external AAD, how can we assign roles ?

            Answer: External User has an option to assign AAD roles like Global admin etc.. But please note recommendation is to provide least privileges to external users. Review the permissions request and assign proper role or create customized role if needed.

1. What if name location is not allowed, and Conditional Access policy we allowed location, what is priority?
   1. Name Location: We are just defining the name here , not applying any policy, this will be used in Conditional access policy, so nothing to worry about just Name Location
   2. We discussed these in Our call, just check the recording once
2. Azure Connect have below 3 accounts what are they?
   1. AD DS Connector -> This is used to connect to Active Directory to fetch the objects
   2. ADSYNC Service Account -> This is used for Managing the Sync engine in On-Premise AD
   3. Azure AD connector -> This is used to connect with AZURE AD and manage the aad objects.
3. What is SQL Server in AAD Connect?
   1. It is used to maintain the Objects and sync information
4. When we create any VM in which order the default resources will be provisioned?
5. When we have contributor role at the management level can we restrict the deletion of any resource by using resource lock?

They are in dependent resources connected

2. Resource Group
3. VNET
4. SUBNET   
   NIC   
   NSG   
   DISK   
   VM
6. This is the usually an order

Azure Interview Questions and Answers.

2) On-premise to Cloud Migration.

3) Diff between Load Balancer and Application Gateway?

Application Gateway is web traffic load balancer that enables you to manage traffic to your web applications,

4) How to enable Internet in Azure VM?

5) If I am unable to resize the VM, What is the reason?

Troubleshooting Azure VM remote access (RDP and SSH issue).

1) Reset Configuration Only

a) Open VM-> Under Support and Troubleshooting -> Reset Password ->

Configuration of Load balancer

Roles and Responsibilities:

Management of Azure Ad

Creating, Managing and deletion of User objects and management of groups.

Configuring Security settings like conditional access, Identity protection.

Troubleshooting user login issues.

Monitoring

Log Analytics work space

Troubleshooting performance issues.

Log Analytic work space

LogicalDisk

Memory

Network Interface

Processor Information

System

Process

Perf

|  where ObjectName == "LogicalDisk"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

|  where ObjectName == "Memory"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

|  where ObjectName == "Network Interface"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

|  where ObjectName == "Processor Information"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

|  where ObjectName == "System"

| where CounterName contains "up time"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Process information

Perf

|  where ObjectName == "Processor Information"

| where CounterName == "% Processor Time"

| extend CPU = CounterValue

| where CPU > 1

| project TimeGenerated, Computer, ObjectName, CPU, CounterName, CounterValue

Perf

| where ObjectName contains "Processor"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

| where ObjectName contains "System"

| where CounterName contains "System Up Time"

| extend uptime = CounterValue/60/60

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue, uptime

Perf

| where ObjectName contains "Memory"

| where CounterName contains "Available MBytes"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

Perf

| where ObjectName contains "LogicalDisk"

| where CounterName contains "% Free Space"

| project TimeGenerated, Computer, ObjectName, CounterName, CounterValue

**Task 11 Azure Migrations**

Please use below steps to get good understanding about the Azure Migrate projects and come with a good use case.

**Step 1: Initial Assessment and Inventory**

**Explanation:** This step involves conducting a thorough inventory of all 245 physical servers, ensuring that you have accurate and complete information about each server's configuration and dependencies.

**Examples:**

   - Server inventory includes details such as server names, IP addresses, hardware specifications, installed software, and roles.

   - Dependencies are documented, such as which servers rely on each other for applications or services.

**Known Issues:**

   - Incomplete inventory data can lead to missing servers or incorrect configurations.

   - Undocumented dependencies can result in application disruptions during migration.

**Step 2: Azure Environment Setup**

-**Explanation:** Here, you prepare your Azure environment, creating the necessary virtual networks, subnets, and resource groups to host the migrated servers.

-**Examples:**

   - Creation of VNets and subnets that mirror your on-premises network structure.

   - Configuration of resource groups for organization and management.

- **Known Issues:**

   - Resource limitations within the Azure subscription can lead to deployment errors or capacity constraints.

   - Incorrect network configurations can disrupt connectivity between on-premises and Azure.

**Step 3: Pre-migration Preparation**

-**Explanation:** In this step, you categorize servers based on their roles, application dependencies, and business priorities, and then plan how to group them into migration batches.

**- Examples:**

   - Grouping web servers and databases together to ensure application continuity.

   - Prioritizing migration batches based on business-critical applications.

**- Known Issues:**

   - Incorrect categorization or grouping can disrupt application dependencies.

   - Incomplete contingency plans for batch migrations can lead to data loss.

**Step 4: Networking and Connectivity**

**- Explanation:** Establishing connectivity between your on-premises environment and Azure is crucial for data replication and application access.

**- Examples:**

   - Setting up a site-to-site VPN to ensure secure communication.

   - Ensuring DNS and IP address assignments are consistent for seamless connectivity.

**- Known Issues:**

   - Inadequate network bandwidth can lead to slow data replication.

   - Misconfigured VPN or ExpressRoute connections can disrupt connectivity.

**Step 5: Batch Migration Plan**

**- Explanation:** Create a detailed migration plan for each batch, including lists of servers, dependencies, schedules, and contingency plans.

**- Examples:**

   - Staggering migrations to avoid resource contention during simultaneous migrations.

   - Defining rollback procedures in case of migration issues.

**- Known Issues:**

   - Overly aggressive schedules can overwhelm resources.

   - Lack of rollback procedures can result in data loss during migration problems.

**Step 6: Azure Migrate Setup**

**- Explanation:** Configure Azure Migrate in your Azure portal and install the Azure Migrate replication appliance on an on-premises server for data collection.

**- Examples:**

   - Creating an Azure Migrate project in the Azure portal.

   - Installing the replication appliance on a designated on-premises server.

**- Known Issues:**

   - Installation issues due to firewall or proxy restrictions.

   - Mismatched Azure Migrate versions can lead to compatibility problems.

**Step 7: Server Assessment with Azure Migrate**

**- Explanation**: Configure the Azure Migrate appliance to discover on-premises servers and run assessments to determine readiness for migration and estimate migration costs.

**- Examples:**

   - Assessing server performance and resource utilization.

   - Estimating monthly costs for running servers in Azure.

**- Known Issues:**

   - Assessments may report false positives/negatives due to configuration discrepancies.

   - Overestimation of costs can occur if assessment settings are not accurately configured.

**Step 8: Azure Migrate Replication Setup**

**- Explanation:** Create a replication group within your Azure Migrate project, add servers to it, and configure replication settings to prepare for migration.

**- Examples:**

   - Defining a replication group named "Batch1\_Servers."

   - Configuring replication settings such as target Azure region, storage account, and virtual network.

**- Known Issues:**

   - Replication setup may fail due to insufficient permissions or incorrect settings.

   - Mismatched replication frequencies can lead to outdated data in Azure.

**Step 9: Test Migration**

- **Explanation**: Initiate a test migration for the servers in the replication group to create test virtual machines in Azure based on replicated data.

**- Examples:**

   - Running a test migration for a subset of servers to validate the process.

   - Ensuring the test VMs function correctly and don't impact production.

**- Known Issues:**

   - Test migrations may uncover application compatibility issues not identified during assessments.

   - Resource constraints in the Azure test environment can affect test results.

**Step 10: Validation and Testing**

-**Explanation:** After a successful test migration, validate the test VMs in Azure for functionality, performance, and compatibility.

**- Examples:**

   - Conducting thorough testing of applications and services.

   - Identifying and addressing performance bottlenecks in the Azure environment.

**- Known Issues:**

   - Performance issues detected during testing may require application optimizations.

   - Complex dependencies between servers may lead to configuration challenges.

**Step 11: Final Migration**

**- Explanation:** Initiate the final migration for servers in the replication group to create production virtual machines in Azure based on replicated data.

**- Examples:**

   - Monitoring the final migration progress in the Azure portal.

   - Validating the functionality of production VMs and applications in Azure.

**- Known Issues:**

   - Data replication delays during final migration can lead to inconsistent data.

   - Application-specific configurations may require adjustment post-migration.

**Step 12: Post-Migration Cleanup**

**- Explanation:** Safely decommission physical servers in the batch, ensure data retention and compliance requirements are met, and remove unnecessary components.

**- Examples:**

   - Documenting and executing server decommissioning procedures.

   - Retaining critical data for regulatory compliance.

**- Known Issues:**

   - Inadequate decommissioning procedures may leave remnants of on-premises resources.

   - Compliance and data retention requirements may not be met adequately.

**Step 13: Repeat for Subsequent Batches**

**- Explanation:** Repeat Steps 4 to 12 for each batch of servers until all 245 physical servers are migrated to Azure.

**- Examples:**

   - Creating new replication groups and migration plans for each batch.

   - Ensuring that each batch undergoes thorough testing and validation.

**- Known Issues:**

   - Maintaining consistency and traceability across multiple batches can be challenging without proper documentation and tracking.

   - Resource limitations may become more pronounced as more batches are migrated.

**Step 14: Documentation and Knowledge Transfer**

-**Explanation:** Maintain detailed documentation of the entire migration process, including server configurations, migration plans, known issues, and solutions. Conduct knowledge transfer sessions to ensure team members understand the new Azure environment.

**- Examples:**

   - Documenting migration plans, test results, and issue resolutions.

   - Holding training sessions for IT staff to familiarize them with Azure management.

**- Known Issues:**

   - Inadequate documentation can lead to confusion and delays in issue resolution.

   - Insufficient knowledge transfer may hinder effective management of the Azure environment.

By following these detailed steps and considering potential examples and known issues, you can navigate the migration of 245 physical servers to Azure more effectively and mitigate challenges along the way. Thorough planning, testing, and documentation are key to a successful migration project.

9 Azure Backup

Here's a comprehensive plan for implementing Azure Backup Solution for Azure VMs and Azure SQL Database backups running on Azure VMs using Azure Recovery Services Vault, including explanations, known issues, and examples:

Step 1: Azure Backup Policy Configuration

Explanation: Start by defining backup policies that specify backup retention, frequency, and application-consistent snapshot settings for Azure VMs and Azure SQL Databases.

Examples:

   Create a daily backup policy for VMs with a retention period of 30 days.

   Configure an Azure SQL Database policy to take weekly full backups and daily transaction log backups with a 7-day retention period.

Known Issues:

   Misconfigured backup policies can result in excessive storage costs or inadequate data retention.

   Incompatible settings may lead to backup failures.

Step 2: Azure Recovery Services Vault Creation

Explanation: Set up an Azure Recovery Services Vault to centrally manage backup and recovery operations for VMs and databases.

Examples:

   Create a vault named "MyBackupVault" in your Azure portal.

   Configure the vault to store backup data in a specific Azure region for compliance.

Known Issues:

   Vault creation errors may occur due to subscription limitations or naming conflicts.

   Choosing an inappropriate storage region may result in performance issues.

Step 3: VM Backup Configuration

Explanation: Configure Azure VMs to use Azure Backup by enabling the Backup feature and associating them with the previously defined backup policies.

Examples:

   Enable Azure Backup for a VM named "MyVM."

   Associate the VM with the daily backup policy.

Known Issues:

   VMs with misconfigured agent extensions or permissions may not register with Azure Backup.

   Mismatched policies can lead to unexpected backup schedules.

Step 4: Azure SQL Database Backup Configuration

Explanation: Configure Azure SQL Databases running on VMs for backup using Azure Backup. This involves installing the Azure Backup extension and associating the databases with the appropriate policies.

Examples:

   Install the Azure Backup extension on the VM hosting the SQL Server.

   Associate the databases with the Azure SQL Database backup policy.

Known Issues:

   Incomplete extension installation or lack of permissions can prevent backup setup.

   Policy misalignment may cause data retention issues.

Step 5: Triggering Manual Backups

Explanation: Trigger manual backups for both VMs and Azure SQL Databases when needed, such as before significant updates or maintenance activities.

Examples:

   Initiate an immediate backup of VM "MyVM" from the Azure portal.

   Trigger an on-demand backup for an Azure SQL Database.

Known Issues:

   Backup initiation may fail due to resource constraints or transient Azure issues.

   Failure to confirm backup completion can lead to data loss assumptions.

Step 6: Monitoring and Alerts

Explanation: Implement monitoring and alerting to keep track of backup jobs, status, and potential issues. Configure alerts to notify you of backup failures or performance anomalies.

Examples:

   Set up Azure Monitor to track backup job success rates.

   Configure Azure Monitor alerts to send notifications in case of backup failures.

Known Issues:

   Misconfigured alerts or lack of monitoring can lead to unnoticed backup failures.

   Alert fatigue from excessive notifications can hinder issue resolution.

Step 7: Restoration and Recovery

Explanation: Test and document the restoration process to ensure you can recover VMs and Azure SQL Databases when needed. Regularly perform recovery drills.

Examples:

   Test VM recovery by restoring "MyVM" to a specified point in time.

   Practice Azure SQL Database recovery by restoring a database to a different Azure SQL Server.

Known Issues:

   Inadequate testing may result in data loss during recovery attempts.

   Lack of documentation can lead to confusion during critical recovery scenarios.

By following this detailed plan, you can effectively implement Azure Backup for your Azure VMs and Azure SQL Databases running on VMs using Azure Recovery Services Vault. Proactive monitoring, testing, and adherence to backup policies are essential to maintaining data availability and recovery readiness.

Azure Automation

Hello Team,

We have to conduct a POC for VVH IT Solutions Group for automation solution, how Microsoft Azure can provide automated solutions for their regular cloud operations & better utilization of the Azure Native services.

Please leverage Azure Automation account Service to fulfil this POC requirement.

Use managed Identity solution for automated authentication solution

Please automate below tasks using azure runbooks

* Cost Saving Project
  + Customer wanted to save the cost by
    - identifying the servers that can be powered Off during non-business hours & power On them before the business hours starts
      * Create a PowerShell Script to
        + fetch list of VM's belongs to specific application/product that customer mentioned
        + Send an email before Power Off those Machines
        + Power Off the Machines after business hours
        + Power On them an hour before the business hours
        + Ensure you perform the validation such as app services/components
        + Send an email to CloudOps team if you see any issues
        + If no issues, please send an email to application team & CloudOps team.
    - Identify the Zombie disks & Un-used Snapshots
      * Fetch the report of each zombie disk & un used snapshots
      * Identify the owner of those resources by looking at the Owner Tag value on these resources
      * Send an email to them by asking why they are kept un-used or zombie & keep CloudOps in CC
      * Once CloudOps receive an update , please have another script which can clean up such resources
* Bulk operations on Azure VM (OS LEVEL)
  + Install applications remotely on multiple Azure VM's
    - Create an Azure VM with Win 2016 OS minimum
    - Create an automation account
    - Register above server as Hybrid worker group agent
    - Keep all the Software/applications on Azure Storage
    - Build a script that would go onto the Server to download applications from Azure Storage & Install them on Azure VMs
    - The Script should run as Hybrid worker group

 Patching Solution

Hi Team,  

We need to provide a POC for one of our customer on Patching solution using Azure Native Service. The solution should minimize the infra setup & reduce the cost, administrative efforts for customer.

Please utilize Azure Update Manager to provide the POC, please document each and every step with detailed analysis and understanding of the product.

Please ready with the benefits, pros & Cons of the solution

Your document should be presented to the customer and it should be highly professional with the detailed , easy to understand content.

Create VM's in below regions

* East US
* West Europe

Create 2 VM's minimum in each region

Create Log Analytics workspace (Centralized)

Create automation account and enable Update Manager

The solution should contains below

* Enable the solution for Existing & Future VMs
* Monitor & track how many VM's are part of Update Manager & how many of them are not ( this gives you the compliance of your environment)
* Monitor Azure Hybrid worker group agent (Agent Readiness) and trigger alert if the agent is not responding for more than an hour
* Monitor Microsoft Monitoring Agent heartbeat and alert if it is not responding for 30 minutes
* Create a Workbook that contains below
  + Total VM's on Azure
  + Total VM's enabled Azure Update Manager
  + Total VM's compliant Vs Non-Compliant , Not Ready
  + Total Updates , Installed updates, Pending Updates
  + Patching Compliance
* Create Schedules as described below
  + Define the Schedules based on the Environment, Customer, Application , Operating System
  + Schedule Should runs every month after 2nd Tuesday
  + Machines Discovery should be based on the Tags so define the Patching tags & Value based on schedule window
  + Identify pre & post patching activities
  + Create Azure Automation runbooks for pre & post patching activities and incorporate them on Patching Schedule
  + VM's should be always rebooted
  + Please send out an automated email notification after the patching with the status and compliance
* Define what is Azure Hybrid worker Group Agent
* How do you monitor the patching Job
* What do you do if Patching fail? Which logs you refer?
* Can I cancel the Patching job in between?
* How many VM's you can add in one Schedule? What is the recommendation?

Infra Monitoring

Hi Team,

Please setup Monitoring solution for the Cloud Infrastructure.

Please deploy VM's in below regions

EAST US

West Europe

Each region should have the at-least 2 Virtual Machines with Standard HDD as disk tier

Please create centralized Log Analytics Workspace

Configure Monitoring for all Azure VM's

Ensure you enable Monitoring for CPU, DISK, Memory Performance counters

Please configure Data collection Rule to collect performance counters for every   5 Minutes

Please create Alerts if CPU, Memory, Disk throughput going beyond 90%

The alert subject should be easy to understand & alert Body should contain the useful information

Please create Dashboard using the Azure Monitor workbook feature, the Dashboard should have the

  Filter to select

* Subscription
* Time Range
* VM Name
* Log Analytics Workspace

It should contains the Tabs such as Disk , CPU, Memory, Network, Uptime

When you click on each Tab it should contains the information related to that component

Users should be able to export the data to an excel sheet

Each table should have an option to Pin to Dashboard Option

Please use Azure Logic Apps & Azure Alerts for automated alerts and define which solution is best and why?

Please advise can we Monitor Azure VM performance without Log Analytics Workspace?

* If Yes, Why and how
* If No, Why we cannot

What is the difference between Azure VM Metrics & Insights?

What is heartbeat of Microsoft Monitoring Agent and how it is useful?

What is the query to fetch heartbeat for each and every Machine

What is Data Collection Rule Set

Please provide the query for VM Uptime

Please provide query for CPU, Memory, Disk utilization

How do you know the inventory of total VM's in subscription & out of which how many are part of Monitoring & how many are not part of Monitoring

What are the steps you follow if you do not see heatbeat  from an azure vm

**Infra Setup**

Hello Team,

VVHIT Tech group has signed up for new contract & as part of our contract we need to setup the application infrastructure as below.  As per of the contract we have to provision the infra following below requirement.

Application Name:  Payment Gateway System (PGS)

Number of environments: 3

* Development
* Testing
* Production

Primary Location: East US2

Secondary Location (DR):Central US

Number of Servers to host the applications: 5 Servers per environment - OS - Windows

* Application Servers (2)
* Web Servers (2)
* Database Server (1) - SQL Server Latest version

**VNET->** VNET-SOL-EUS2-01

**Subnet -** Create one per environment by customer by application

**Network Security Group->** Create one per environment and associate to respective subnet

**Application Security Group -> C**reate one per Server type (application 1, Web Server 1, DB 1)

**Availability Set** - Per Environment and Application

* Example: Create one Availability Set for application Servers in each environment , such as AVSET-APP-DEV, AVSET-APP-PROD, AVSET-APP-TEST, AVSET-WEB-TEST, AVSET-WEB-PROD, these are just examples please follow the naming standards.

Proximity Placement Groups-> Each AVSET should be part of one Proximity placement group (  DEV APP AVSET should be part of DEV APP PPG)

Internal load Balancer -> Create one ILB by Environment and application , keep application servers on Backend members

**Backup ->** Create one recovery services vault for customer in primary region and configure backup for all environment belongs to that customer . For app & Web servers configure Azure VM backup , for SQL DB Servers, please configure SQL DB backup on Recovery services vault

**Site Recovery ->** Enable Replication for All Production Servers of customer to Secondary Location (DR) , create Recovery services vault in DR region. Make sure you use DR VNET as  and create DR subnet to enable replication. Create respective NSG, ASG, AVSET, PPG in advance in DR region before you enable replication

NOTE: If you see there is no quota available for VM Size , please raise a quota request.

Naming Standards:

Environment - Prod -> PR, Dev - DV , Test - TS

Application - PGS

Location - East US - EUS2, Central US -> CUS, West Europe - EUW, North Europe - EUN

Virtual Network - VNET

Subnet - SN

Network Security Group -> NSG

Proximity Placement Group - PPG

Application Security Group -> ASG

Recovery Services Vault - RSV

Availability Set - AV

Virtual Machine - VM

**Examples:**

Subnets->

* SN-**PR**-**TMS**-**VVH**-EUS2-01
* SN-PR-ESM-VVH-EUS2-01
* SN-PR-TMS-VVH-DR-CUS-01
* SN-PR-ESM-VVH-DR-CUS-01

Network Security Group->

* NSG-VVH-PR-ESM-EUS2-01
* NSG-VVH-PR-TMS-EUS2-01

Virtual machines->

* PR-TMS-VVH-EUS2-01
* PR-TMS-VVH-EUS2-02

Proximity Placement Group:

* PPG-VVH-PR-TMS-EUS2-01

Availability Set:

* AV-TMS-VVH-EUS2-01

**\*NOTE\*: \*Please add your first 2 letters of your name on resource group & resources you are deploying\***

 Reference links:

Azure VM Backup -> [Day 19 - Mastering Microsoft Azure Backup: Secure Your Azure Virtual Machines](https://www.youtube.com/watch?v=Gc-u6qE0sVw)

Azure Site Recovery -> [Disaster Recovery for Azure VMs  For Regional Level Outages](https://www.youtube.com/watch?v=-BNFFuwGyCU)

Azure Network Security Groups:  [Azure Network Security Groups](https://www.youtube.com/watch?v=pGEnoEuKh2Y)

[Day 23 - Azure Site Recovery Workflow & Use cases Part-1](https://www.youtube.com/watch?v=nrujqFIW0ns)

[Day 24 - Azure Site Recovery Deep Dive Part 2](https://www.youtube.com/watch?v=H5vhu_NYqbU)

[Day 22 - Azure Backup Interview Questions & Answers - MARS AGENT](https://www.youtube.com/watch?v=6tMi34e7fHc)

[Day 21 - Master Azure SQL DB Backups & Centralized Monitoring](https://www.youtube.com/watch?v=xMMVahalpO8)

[Day 20 -Azure Backup: Admin Day to Day tasks](https://www.youtube.com/watch?v=4qh0SFlu16A)

[Day 18 - Mastering Azure Backup: A Comprehensive Guide for Azure Admins](https://www.youtube.com/watch?v=GGoUOfe17Ac)

|  |  |  |
| --- | --- | --- |
| [Day 13 Azure VM Tags - Azure Admin Tas](https://www.youtube.com/watch?v=hDpd1HRsIHs) |  | ks- Challenges- Human Erros |

[Day 11 - Azure VM - High Availability - Options](https://www.youtube.com/watch?v=x3xhBdFhbaw)

[1) How can you ensure high availability for a critical VM - Azure Admin Interview Q&A](https://www.youtube.com/watch?v=UuZOYFSWmoQ)

 Documents to refer:

Bulk User Deletion:-> <https://learn.microsoft.com/en-us/azure/active-directory/enterprise-users/users-bulk-delete>

Azure AD B2B Users: <https://learn.microsoft.com/en-us/azure/active-directory/external-identities/what-is-b2b>

<https://learn.microsoft.com/en-us/azure/active-directory/external-identities/invitation-email-elements>

Azure RBAC : <https://learn.microsoft.com/en-us/azure/active-directory/roles/custom-overview>

Assign Roles on Azure Portal -> <https://learn.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal>

Azure AD Custom Roles: <https://learn.microsoft.com/en-us/azure/role-based-access-control/custom-roles>

Azure Tags-> <https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources>

<https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources-portal>

Azure Naming Standards: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming>

Abbreviation examples for Azure resources: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-abbreviations>

Create Azure VNET: <https://learn.microsoft.com/en-us/azure/virtual-network/quick-create-portal>

Azure VWAN: <https://learn.microsoft.com/en-us/azure/virtual-wan/virtual-wan-about>

[Day 17 - Azure VWAN- Network Architecture- Enterprise Cloud Environment](https://www.youtube.com/watch?v=NjbFlgype14&t=3308s)

Create Templates for VM infra on Azure

* Create VM with
  + Backup
  + VNET
  + Subnet
  + Disks
  + NSG
  + ASG
* Create Multiple VM's with ARM templates