Azure 자격증 자료(AZ-300, AZ-301)

**1강. Azure Active Directory**

**1) AAD(Azure Active Directory)**

- 계정관리 솔루션 : 셀프 서비스 제공, 모든 플랫폼에 적용 가능

- 멀티 테넌트 클라우드 기반 디렉토리 및 ID 관리 : MS ,Google, 카카오, Facebook(Business Plan)

: 멀티 테넌트: 테넌트별로 각각의 서비스 분리 가능

: 사용자 식별 --> 권한 범위 내 접근 가능

- 단일 로그인 엑세스 제공

- 모든 ID 관리 기능을 제공

- MS의 모든 제품을 통합 관리: M365 등

**2) AD DS(Active Directory Domain Service)**

- Windows Server 기반 ID 관리 및 디렉토리 서비스

: 기존의 사내 구축 형 배포

: Azure AD를 직접 지원하지 않는 iaaS 워크로드의 클라우드 배포

- On-Premises AD --> AAD 단방향 :

- AAD 계정을 Emulation 해주는 서비스

- AAD와 AADS는 다르다

: 관리 서비스

: OAuth 인증

: Directory Lookup

: Federation : 인증을 위한 다른 곳으로의 연결

\* Claim : 필요한 정보(e-mail주소) -> Attribute를 가져오는 행위

\* Role 권한 부여 : 모든 권한, 특정 권한 부여

\* url 주소, Claim, 인증서 교환

- Soft Sync : On-Premise와 AAD 연결

- LDAP Query 가능 (사내 기기 및 장비에서 많이 사용)

- 그룹 정책 : 암호 변경 등

- 도메인 강비

- kerberos 및 NTLM

**3) Role-Based Access Control**

- Governance 수립이 중요

- Scope : 무한함

- built-in Roles 사용

: Control pannel, Management Pannel 구분

: Management Pannel 까지만 제어 가능

- Role Definition

- Azure PowerShell and CLI

- Configuration Self-Service Password reset

: Azure 포탈에서 직접 다운로드 가능

: 범위에서 사용자를 지정 : 암호 변경 권한 Role 적용

: Authentication methods

**4) Azure AD Connect**

- AD DS와 Azure AD를 통합

- AD DS의 사용자 암호를 Azure AD에 동기화 : Password Synchronization

- Sing-On Methods

: Password Synchronization

: Federation with AD FS

: Pass-through authentication : 인증 시스템을 통한 인증

첫번째 옵션 : 계정 + 암호(해시) 동기화 --> Seamless SSO 제공

두번째 옵션 : ID동기화 + ADFS -> 계정은 동기화

새로운 옵션 : ID 동기화 + SSO를 통한 인증 (Pass-through)

\* On-premises와 연동

**5) Azure AD Application Proxy**

- 서비스 계정과 사용자 계정 분리

- Application 계정 초기화 시 자동으로 기동 될 수 있도록 서비스 계정 사용 필요(최소의 권한 부여)

**6) Device Management**

- Azure AD 장치 통합을 지원

: 사용자가 사용하는 운영체제

: MDM(모바일 장치 관리) 사용

**7) Azure Joined Devices**

- 하이브리드 Azure AD에 가입한 장치

**[실습 자료]**

Implementing user-assigned managed identities for Azure resources

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T01_Lab_Mod05_Implementing%20user-assigned%20managed%20identities%20for%20Azure%20resources.md>

**2강. Application 에서 인증 구현**

**1) Certificate-based authentication**

- 디지털 인증서를 사용하여 ID를 설정

: Front-end 와 Back-end간 상호 작용

- 암호화된 통신

: IoT 시장에 꼭 필요 : 빅데이터

**2) Legacy authentication methods**

- Froms-based authentication

- Windows-based authentication

- Token-based authentication

: .NET 클레임 기반 인증

: 앱 서비스 인증 및 승인

**3) Multi-Factor authentication**

- 사용자를 인증 할 때 두가지 이상의 요인을 적용

: Knowledge, Possession, Inherence

**4) Claims**

- 아이디와 그 속성을 나타내는 name/value pair :

**5) Claim-based authentication**

**6) Role-based authorization**

**7) Role-based access control(RBAC)**

- 직무분리

**[실습 자료]**

Lab: Implementing custom Role Based Access Control (RBAC) roles

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T03_Lab_Mod04_Implementing%20custom%20Role%20Based%20Access%20Control%20roles.md>

**3강. Virtual Networks 구현**

**1) 가상 네트워크 생성**

- IP 주소 공간 정의

- 서브넷 생성

**2) Multiple NICs in Virtual Machines**

- 가상머신은 프로세스임

- 운영체제의 종속성

: SQL Cluster 서버인 경우 3개의 IP 필요

- Docker Container : Bridge, host, virtual network

**3) pubilc IP Address**

SKUs

- Bring Your Own IP : Public IP -> AWS에서 서비스 중임

**4) User Defined Routing**

- System Routing : Azure 내부적으로 모든 통신이 되도록 설정 함

- 특정한 Subnet에 특정한 Routing Table 설정

: 용도에 따른 Subnet 구분

\* NVA(Network Virtualization Appliance)

**5) Vnet-to-VNet Connections**

- VPN gateway 필요

**6) VNet Peering**

- SDN(Software Defined Network)

**7) Regional VNet Peering**

**8) Gateway Transit**

- 피어링 된 VNET에서 Routing 사용 허용

**9) Load Balancer**

- L4(TCP/UDP)

- Public Load Balancer

- Internal Load Balancer : Private IP로 통신

: LOB(Line of Business) -> Backend에서 제공

**10) Load Balancer SKUs**

- SKUs : Basic, Standard

**11) Backend Pool**

- 구성은 로드 밸런서 SU에 따라 다름

- Standard : Availability Set과 VM Scale Set의 VM을 포함, 동일한 가상 네트워크에서 최대 1000 개의 Azure VM

- Basic : 최대 100개의 VM

**12) load Balancer Rules**

- 트래픽 분산 결정

\* 세션 지속성 : 5-tuple(source ip, sourceport, des ip, des port, protocol type)

**13) Multiple Frontends**

**14) Health Probes**

- HTTP '와 TCP 만 제공

**15) Application gateway**

- L7 Load Balancer

- SSL 인증서

- 리스너 : http / https

- Health Probes

\* HTTP (200~399)

- instance 기반

- Path-based routing

- host-based routing

- multiple site hosting

- redirection and session affinity

**16) Web Application Firewall**

- 백엔드 서버 풀 작업 부하에 대한 보호 기능을 제공

- OWASP 규칙 사용

**17) Site-to-Site VPN connection**

- IPSec IKEv1 또는 IKEv2

- 온디멘드 용량 절약 : Azure 서버를 빌려다 써서 온 프레미스 서버의 가용량 절약

**18) VPN Gateway**

- Gateway subnet

**19) Local Network Gateway**

- 하이브리드 시나리오에서 온프레미스 VPN 장치

**[실습 자료] Configuring and Managing Virtual Networks**

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T02_Lab_Mod03_Configuring%20VNet%20peering%20and%20service%20chaining.md>

**[실습 자료] Implementing Advanced Virtual Networking**

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T03_Lab_Mod03_Implementing%20Azure%20Load%20Balancer%20Standard.md>

**4강. ExpressRoute(전용선)**

**1) ExpressRoute**

- 온 프레미스 네트워크를 확장하는 사설 연결: Azure, Office 365, Dynamics 365 등

- 혜택

: 향상된 안정성

: 높은 대역폭

: 대기 시간 단축

: 보안 강화

- 일반적인 시나리오:

: 데이터 이동

: 비즈니스 연속성

**2) Connection Options**

- 클라우드 교환 코로케이션

- 지점 간 이더넷 연결

- Any-toany(IPVPN) 연결

**3) Site-to-Site and ExpressRoute Coexisting Connections**

- 구성: VPN 게이트웨이 및 ExpressRoute 게이트웨이가 동일한 Gateway Subnet에 배포

- 혜택

: 확장 된 연결 범위

: 비용 효과적인 Failover

**4) Site-to-Site VPN**

- 온프레미스와 Azure 가상 네트워크 간의 보안 강화 된 연결 제공

- 네트워크 전체에 직접 연결 제공

- 사내 구축 환경의 VPN 장치 필요

**5) Virtual network-to-virtual network Connections**

- Azure 가상 네트워크 연결에 Azure VPN 게이트웨이 활용

: 온프레미스에 대한 연결을 포함하여 다중 사이트 VPN 연결 용이

: IPSec 터널에 트래픽을 캡슐화 하여 처리

**6) Connecting across cloud providers**

- IPSec VPN을 활용하여 클라우드 공급자간 가상 네트워크 연결

**5강. VM 관리**

1. **VM 생성**
2. **VM 생성 예제**
3. **Template을 이용한 배포**

**4) Resource Manager Templates**

- ARM 전용 구문을 따르는 JSON 형식의 문서: 실습의 Json 문서를 참고하면 됨.

**5) Custom Images**

- 이미지는 일반화 된 운영 체제가 포함된 VHD파일임:

: Azure VM을 프로비저닝 하는데 사용

**6) Linux VM**

: 다양한 배포판을 지원

: 윈도vm과 동일한 배포 옵션을 지원

: DevOPS 툴을 지원함

**7) 백업 & 복구**

- 백업: VM의 애플리케이션 일관성 있는 백업 지원

: 지리적 중복 Recovery Vaults 에 저장된 복구 지점을 만든다.

: 전체 VM 또는 특정 파일만 복원할 수 있다.

- Azure Site Recovery: 전체 지역에 영향을 미치는 중단으로부터 VM보호

: 한번의 클릭으로 몇 분만에 복구 가능

- Managed Snapshots:

: 관리 디스크의 읽기 전용 전체 사본

: 원본 디스크와 독립적으로 존재

: VM 재구축을 위한 새 관리 디스크를 작성하는데 사용

: 디스크의 사용 된 부분에 따라 요금이 부과

**8) Backup Virtual Machines**

- 다음과 같은 3단계 적용

1. 지리적 중복 또는 로컬 중복 저장소로 Recovery Service vault를 만든다.

2. 복구 지점의 빈도와 보유 기간을 지정하여 백업 정책 정의

3. 백업 확장 기능을 설치하여 VM 백업 시작

**9) Restore Virtual Machine**

- Azure 포털에서 복원 작업을 트리거 할 수 있다.

: Azure Backup은 복원 작업을 생성

**10) 모니터링**

- Azure 포털의 Azure VM 개요 블레이드: CPU, 네트워크, 디스크, 디스크 오퍼레이션

- Azure 포털의 Azure VPM 모니터링 섹션: Metrics, Diagnostic settings, Advisor recommendations, Diagram.

**11) Diagnostic 세팅**

- Windows VM: Performance counter, logs, crash dumps, agent, sinks

- Linux: Metrics, Syslog, Agent

**12) Advisor Recommendations**

- Azure 배포 최적화를 위한 맞춤형 클라우드 서비스

: 리소스 구성 및 사용 원격 측정법 분석

: 네 가지 범주로 그룹화된 권장 사항 제공

1. High availability

2. Security

3. Performance

4. Cost

**[실습 자료] Online Lab: Implementing Custom Azure VM Images:**

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T01_Lab_Mod03_Implementing%20Custom%20Azure%20VM%20Images.md>

**6강. Migrate to Azure**

**1) Migration 목표**

- 기술 중심 및 비즈니스 중심

: 하드웨어 노후화 사이클 해결

: '구매 전 용량' 모델에서 탈피

: IT민첩성 부족

: 핵심 역량에 다시 집중하려는 열망

: 글로벌 존재감 유지 비용

: 재해 복구 시나리오 활성화

**2) Migration 단계**

- Azure로 마이그레이션을 계획할 때 고려 사항:

1. Discover : 사내 구축 환경의 작업 부하를 보다 잘 파악하고 Azure에서 실행하기에 최적의 리소스 수준을 평가

2. Migrate: 선택한 작업을 Azure로 이동

3. Optimize: Azure 기반 작업 부하를 미세 조정하고 ROI를 극대화

백업, 모니터링, 보안 평가 및 비용 관리를 도와주는 많은 파트너가 있다.

**3) Azure Migration Service**

- Azure로의 마이그레이션을 위해 온 프레미스 워크로드 평가 제공:

: 구내 시스템의 마이그레이션 적합성

: 성능 기반 크기 조정

**4) 아키텍처**

: VMware vCenter 환경의 구성 요소 통합

**5) 프로세스**

1. 프로젝트 만들기

2. 기계 발견

3. 정보 수집

4. 프로젝트 평가

**6) Creating a Collector**

- 인프라 Discovery를 처리하는 가상 어플라이언스

**7) Accessing VM Sizing**

- Azure Migration VM 평가는 두 가지 유형의 sIZING 제공: 성능 기반 크기 조정, On-prem 기준

**8) 비용 견적: Azure VM의 총 비용 제공(월 비용)**

**9) Migration 트러블슈팅**

- 자주 발생하는 문제:

: 실적 데이터 없음

: 마이그레이션 프로젝트를 만들지 못함.

: 수집기가 인터넷 연결 불가

: 서버 접속 불가

: 날짜/시간 동기화 오류

**[실습 자료] 300T02\_Lab\_Mod01\_Implementing Azure to Azure migration.md**

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T02_Lab_Mod01_Implementing%20Azure%20to%20Azure%20migration.md>

**7강. Azure Site Recovery(ASR)**

**1) ASR 시나리오**

- Hyper-V VM 복제:

: VMM을 사용하여 Azure 또는 보조 사이트로 복제

: VMM없이 Azure로 복제

- VMWare VM 복제

: VMware 또는 Azure를 실행하는 보조 사이트로 복제

- 물리 서버 복제

: VMware 또는 Azure를 실행하는 보조 사이트로 복제

**2) ASR 기능**

- 재해 복구 사이트의 필요성 제거

- 인프라 비용 절감 (디스크 복제만 이루어지므로 DR센터에 서버가 기동되어 있지 않아도 됨)

- Azure에 자동으로 복제

- 복잡한 작업 부하의 중단 방지

- 용량 확장 또는 증대

- 지속적인 상태 모니터링

**3) Using ASR: A Look Ahead**

- 아키텍처

1. 1차 및 2차 사이트

2. 온프레미스 또는 Azure

- 프로세스

1. 인프라 구성

2. 복제 구성

3. 재해 복구 구성

4. 장애 복구 구성

**4) Configuring Azure**

- 대상 Azure 구독 내 권한 위임

- Azure 저장소 계정 만들기

**5) ASR Deployment Planner**

**6) 인프라 구축 프로세스**

- Azure Portal 구동 순서

1. 보호 목표 (예: Azure에 Hyper-V 시스템 복제)

2. 소스 환경 (예: Hyper-V 사이트)

3. 대상 화경 (예: Azure 구독의 리소스 그룹)

4. 복제 설정

5. 배치 계획

**7) Recovery Services Vault**

- ASR 메타 데이터 복제된 VM디스크 호스트

**8) Customize the Recovery Plan**

- 복제 목표 선택

- 소스 및 대상 혼경 설정

- 복제 정책 설정

- 복제 사용

- 테스트 마이그를 실행하여 모든 것이 제대로 작동하는지 확인

- Azure로 한 번 장애 조치 실행

**9) Planed and Unplaned Failovers**

- ASR은 두가지 유형의 장애 조치

: Planed

: Unplaned

- ASR Failovers:

: 자동이 아님

: Azure Portal에서 작업 가능

**10) Test Failover**

- 복제 및 재해 복구 전략의 영향을 받지 않는 검증:

- Best practices:

: 자동이 아님

: Azure Portal에서 작업 가능

**11) Failback**

- 기본 사이트로 다시 계획된 장애 조치

- 두 가지 데이터 동기화 방법 제공:

1. Full downtime

2. Minimize Downtime

**12) Virtual Machine Replication**

- 설정:

1. 원본 컴퓨터 선택

2. 각각에 대해 복제 할 OS유형

3. 디스크 선택

**13) 복제 정책**

- 복사 빈도

- 복사 지점

**14) 고가용성과 재해 복구**

- 고가용성(HA): 장애에도 불구하고 응용 프로그램이 계속 실행 되는 기능

- 재해복구(DR): 전체 지역에 영향을 미치는 서비스 중단과 같은 주요 사고를 복구 하는 기능

**15) 데이터 백업**

- 백업은 데이터 복제와 다름.

- 데이터 복제는 거의 실시간으로 데이터를 복제하여, 빠르게 복구 하는게 목적임.

**16) 요구사항 파악**

- RTO: 복구 시간 목표, 사고 후 응용 프로그램을 사용할 수 없는 최대 허용 시간

- RPO: 복구 지점 목표, 재해 중에 허용되는 최대 데이터 손실 기간.

**8강. 가용성 개선**

**1) FMA(Failure Mod Analysis)**

- FMA는 설계 단계 초기에 애플리케이션에 탄력성을 구축하는 프로세스임.

- 목표

: 응용 프로그램에서 어떤 유형의 장애가 발생하는지 식별

: 각 유형의 장애가 응용 프로그램에 미치는 잠재적 영향 파악

: 복구 전략 식별

**2) Avoiding a Single Point of Failure**

**3) Auto Scaling and Load Balancing**

- 더 많은 인스턴스를 추가하여 클라우드 애플리케이션을 확장 할 수 있어야 함.

**4) Workload Service-Level Objectives**

- 서비스 수준 목표(SLO)에 따라 워크로드를 구성하시오.

: 다른 작업과 분리 될 수 있는 개별 기능 또는 컴퓨팅 작업

- 웹에서 웹앱을 분리(APIs Azure App Service)

: 솔루션에 웹 프러ㄴ트 엔드 및 웹API가 모두 있는 경우 별도의 App Service앱으로 분해하는 것 고려

- 서비스 종속석 최소화

: 서비스 수를 최소화하고 시스템 내 기능 및 서비스 종속성을 식별하시오.

**5) Enhancing Security**

- 분산 서비스 거부(DDoS) 공격에 대한 보호를 보장하시오.

- 응용 프로그램의 액세스 최소 권한 원칙 준수.

**6) 추가 Resiliency Tips**

- 중요한 트랜잭션에 고 가용성을 구현하는 메시지 브로커 사용

- 정상적으로 저하되는 응용 프로그램 설계

- 대량 사용자 스로틀

- 로드 밸런싱을 사용하여 트래픽 급증 완화

- 타사 서비스 모니터링

- 가능할 때마다 비동기 작업 구현

**7) 배포 & 유지 보수 업무**

- 배포 및 유지 관리 작업을 자동화하고 테스트.

- App Service 배포의 경우 구성을 앱 설정으로 저장.

- 리소스에 의미 있는 이름을 부여

- 기능 및 수명주기별로 리소스 그룹을 구성.

**8) Infrastructure as Code and Immutable Infrastructure**

**9)Maximize Application Availability**

- 응용 프로그램 가용성을 최대화하기 위한 릴리즈 프로세스 설계

: 블루/그린 배포

: 카탈리나 배포

**9강. 데이터 관리**

**1) Replicating Data**

- 지역 복제 데이터 베이스

: Azure SQL DB와 Cosmos DB는 모두 지역 복제를 지원하므로 다른 지역에서 보조 DB복제본 구성 가능.

- Azure Storage의 지역 복제 데이터

: Azure Storage의 데이터는 데이터 센터 내에서 자동으로 복제 됨.

- VM의 경우 VM디스크(VHD파일)를 복원하기 위해 RA-GRS복제에 의존하지 마시오.

**2) Additional Data Management Considerations**

- 데이터 관리 추가 고려 사항

: Shading

: 낙관적 동시성과 최종 일관성

: 문서 데이터 소스 장애 조치 및 장애 복구 프로세스를 수행한 후 테스트

: 정기적인 백업 및 특정 시점 복원.

: 단일 사용자 계정이 프로덕션 및 백업 데이터 모두에 액세스 할 수 없도록 함.

: 데이터 백업을 확인.

**10강. 저장소 구현 및 관리**

**1) Azure Storage**

- Azure 가상 컴퓨터 저장소

- 비정형 데이터 저장(Blobs, Queues, Data Lake)

- 정형 데이터 저장(Tables, Cosmos DB, Azure SQL)

**2) Azure Storage Accounts**

- Storage (General Purpose v1)

: contain blobs, tables, queues, files

: 성능 계층: standard, premium

- Blob Storage

: Contain blobs only

- Storage V2

: contain blobs, tables, queues, files

: 성능 계층: standard, premium

**3) Standard, Premium Storage Accounts**

- 성능 계층:

: Standard: 높은 성능의 I/O가 필요하지 않은 경우 적합, GB당 저렴하며, HDD 지원

: Premium: IO성능이 높은 경우, 낮은 지연과 높은 throughput/IOPS, SSD지원

**4) Custom Domain Names설정**

- CNAME 연결: CNAME record를 Azure storage account Endpoint로 생성

- asverify를 사용한 중개 매핑: 자신의 CNAME레코드에 asverify를 앞에 추가한다. 도메인의 DNS레코드를 수정한다.

**5) 가격 및 청구**

- 저장소 비용: 데이터 양, 성능 계층, 접근 계층

- 저장소 포함 비용:

: 데이터 접근 비용

: 트랜잭션 비용

: 지리적 복제 데이터 비용

: 아웃바운드 데이터 전송 비용

: 저장소 계청 변경

**6) Locally Redundant Storage**

**7) Geo-Redundant Storage**

- 각 Azure 지역간 3copy씩 총 6copy

- 전략: 동일한 지역에서 3개의 복제본을 통해 동기로 복제 된 데이터,

데이터는 각 영역 내에서 동기식으로 복제되고 지역간 비동기식으로 복제

- 보호: 로컬 하드웨어 오류와 지역간 재해 보호

**8) Zone Redundant Storage**

- 동일한 지역에 3개의 다른 가용 영역에 각각 저장

- 전략: 데이터는 동일한 지역에 있는 별도의 가용 영역에 3개의 복제본을 통해 동기식으로 복제

- 보호: 로컬 하드웨어 오류와 가용 영역 보호

**9) Azure Storage Explorer**

- 저장소 관리를 위한 오픈소스 응용 프로그램

- 지원 플랫폼: Windows, MAC, Linux

- Azure 구독에 연결

- 로컬 개발 저장소 작업

- 저장소 계정의 이름과 키를 사용하여 저장소 계정에 연결

- 공유 액세스 서명을 사용하여 저장소 리소스에 접근

**11강. 데이터 저장 및 접근**

**1) Import / Export Service**

- 온프레미스와 Azure 간 대량의 데이터 전송: Import데이터를 Azure Blob 저장소 및 Azure Files로 전송.

- 시나리오: 데이터를 클라우드로 마이그레이션, 콘텐츠 배포, 백업 및 데이터 복구

**2) Import / Export Jobs**

- 식별: 데이터, 드라이브 수, 대상 Blob 또는 Files

- Job 생성: Azure Portal, import/export REST API

- 배송 정보 제공

**3) AzCopy**

- Azure Storage 작업을 자동호하는 CLI유틸리티

: Azure Blob, Files, Table 저장소를 원본 또는 대상으로 복사

: 파일 시스템과 저장소 계정 간 또는 저장소 계정간 복사

: AWS S3에서 저장소 계정으로 복사.

**4) CDN**

- AZure 콘텐츠 캐싱 서비스

: Azure에서 제공하는 오디오, 비디오, 이미지 등을 신속하게 전달할 수 있도록 설계

: 데이터 흐름:

1. Alice가 CDN위치를 가리키는 URL을 사용하여 파일을 요청, DNS 가 요청을 최적의 POP로 라우팅.

2. Edge 서버에 파일이 캐싱되어 있지 않은 경우, Origin에서 콘텐츠를 가져옴.

3. Origin은 TTL값을 포함하여 콘텐츠 반환.

4. Edge서버는 파일을 캐시하여 Alice에게 반환.

5. 후속 요청은 Edge 서버의 캐시에서 제공

6. Edge 서버에 캐시된 파일의 TTL값이 만료되면 Origin에 콘텐츠를 재요청

7. Origin에서 파일이 변경되었다면, 변경된 파일의 TTL값을 포함하여 콘텐츠 반환.

8. Edge서버에서 TTL값 갱신 또는 콘텐츠 캐시

**5) CDN 프로파일**

- CDN endpoints:

: 가격 책정 계층 및 제공업체: Verizon, Akamai, Microsoft

- 과금 기준

: 아웃 바운드 에서 전송(per GB)

**6) CDN Endpoints**

- 콘텐츠 Origin 설정:

: CDN endpoint name/value

: Origin Type: storage, cloud service, web app, custom origin

: Origin hostname

: 콘텐츠 제공에 영향을 미치는 CDN 기능: 압축, 쿼리 문자열, 지역 필터링, 맞춤 도메인

**7) CDN Time-to-Live**

- CDN TTL

: 콘텐츠 오리진의 HTTP응답에서 cache-directive 헤더에 포함

**8) CDN 압축**

- 파일 전송 속도 향상 및 페이지로드 성능 향상

- 지원: 원본에서 압축 사용

CDN에지 서버에서 직접 압축 사용.

MIME 형식 목록에서 압축 사용.

**12강. Database Encryption**

**1) 정규화 된 단위**

- 하이퍼 스케일 클라우드

**2) DTUs - Azure SQL DB**

- The Database Transaction Unit(DTU) 모델

: 조합 항목 - CPU, MEM, Read-Write 퍼포먼스

: 구성 항목

1. DTUs - 단일 데이터 베이스

2. eDTUs - 신축성 있는 데이터 베이스 풀

: Capacity 설명 - 3가지 서비스 등급 (Basic, Standard, Premium)

**3) 구조화 및 비정형 데이터**

- 다각형 지속성:

: 다양한 데이터 저장소 기술을 사용하는 솔루션을 나타낸다.

: 요구 사항에 따라 데이터 저장소를 선택해야 하는 중요성을 강조한다.

**4) 구조화 된 데이터 저장소 사용**

- 관계형 데이터베이스:

: 데이터 조직 제공

1. 잘 정의 된 스키마에 기반

2. 행과 열이 있는 일련의 2차원 테이블

3. SQL지원

4. ACID 모델과의 트랜잭션 일관성 및 준수

: 강력한 일관성이 필요한 시나리오에 적합

: 스케일 아웃이 필요한 시나리오에는 적합하지 않음

: Azure에서 관리하는 구조화 된 데이터 저장소 포함

1. Azure SQL DB

2. Azure DB for MySQL

3. Azure DB for PostgreSQL

**5) Encryption**

- 평문을 암호문으로 변역하는 과정.

- 암호화 알고리즘과 하나 또는 두 개의 키를 사용:

: 알고리즘의 목적은 키를 사용하지 않고 암호문을 해독하는 것을 가능한 한 어렵게 만드는 것.

: 대칭 암호화: 동일한 키가 암호화 및 암호 해독에 사용됨, 많은 양의 데이터를 암호화

: 비대칭 암호화: 암호화 및 암호 해독을 위한 다른 키, 소량의 데이터 또는 대칭 키 암호화 용

**6) 암호화된 상태에서의 암호화**

- 데이터가 지속될 때 암호화:

: 데이터가 저장된 하드웨어에 대한 물리적 액세스를 얻고 포함 된 데이터를 손상시키려는 시도로부터 보호

: 준수 및 보안 요구 사항으로 인해 많은 시나리오에서 필수 항목

- Azure의 암호화 된 상태에서의 암호화:

: 쓰기/읽기 중 동적으로 암호화/복호화

: 대칭 암호화 키 사용

: 파티션에서 다른 키 사용

: 안전 위치에 키 저장

: 지원

1. Azure Storage Encryption

2. Azure SQL Database encryption

3. Azure Cosmos DB Encryption

**7) 하이브리드 백업 암호화**

- 3중 암호화

**13강. End-to-End Encryption**

**1) TDE(Transparent Data Encryption)로 데이터 암호화**

- 주요 TDE특성:

: 데이터 및 로그 파일에 대한 암호화:

1. SQL Server

2. AZure SQL DB

3. Azure SQL Data Warehouse

: 페이지 수준에서 실시간 I/O 암호화 및 암호 해독

: 데이터베이스 암호화 키(DEK) 사용

1. 키는 복구 중 가용성을 위해 데이터베이스 부트 레코드에 저장.

**2) 항상 암호화 된 데이터 암호화**

- Azure SQL 데이터베이스 및 SQL Server의 암호화 기술

: 중요한 데잍가 데이터베잇 시스템 내에 일반 텍스트로 나타나지 않도록 보장.

: 클라이언트가 클라이언트 응용 프로그램 내부의 중요한 데이터를 암호화하고 암호화 키를

데이터베이스 엔진에 공개하지 못하게 한다.

: 중요한 데이터를 보호하는데 도움이 됨.

: 구분

1. 데이터를 소유한 사람들.

2. 데이터를 관리하는 사람들.

: 클라이언트 컴퓨터에 설치된 드라이버가 응용 프로그램의 중요한 데이터를 자동으로 암호화하고 복호화.

**3) Azure confidential computing**

- 사용중인 데이터를 암호화하는 많은 Azure 서비스의 기능 참조:

**4) SSL and TLS overview**

- TLS 및 SSL은 암호화 프로토콜

- TLS는 SSL의 후속 제품

: 컴퓨터 네트워크를 통한 통신 버전 제공

**5) Azure Key Vault**

- 보안이 강화 된 비밀 저장소로 작동하는 클라우드 서비스

: Vault라고 하는 보안이 강화 된 여러 개의 컨테이너를 만들 수 있습니다.

: 특성

1. 암호, 키 및 인증서와 같은 비밀 정보를 지원

2. 키 저장 및 암호화 작업에 하드웨어 보안 모듈(HSM) 사용

3. TLS인증서 요청 및 갱신 기능.

4. 모든 작업의 로깅

**14강. 구독과 리소스 관리**

**1) Azure 모니터링 서비스 소개**

- 모니터링: 데이터를 수집 및 분석하여 작업 부하의 성능, 상태, 가용성을 결정

- 지원 사항:

: Core Monitoring

: Deep Infrastructure Monitoring

: Depp App Monitoring

: Shared monitoring Capabilities

**2) Azure Monitor: Key Capabilities**

- Azure Core Monitoring은 다음과 같은 구성을 가진다.

- Metrics:

: Azure 리소스의 성능 통계

: Azure VM의 운영 체제 메트릭

: 1분 간격 데이터(중요한 알림 및 알림에 적합)

- Activity Logs : Azure에서 수행한 Activity에 대한 로그

- Diagnostic logs : Appl. 레벨의 로그 (액세스 로그 등)

**3) Azure Advisor**

- Azure 배포 최적화를 위한 맞춤형 클라우드 서비스:

: 자원 구성 및 사용 원격 측정법 분석

: 네 가지 범주로 그룹화 된 권장 사항 제공:

1. High availiability

2. Security

3. Performance

4. Cost

**4) Azure Monitor Alerts**

- Azure Monitor는 다양한 혜택을 포함한 alert을 지원:

: Action Groups을 통한 nitification 시스템 개선

: Alert과 Alert Rule을 분리

: 통합된 제작 환경

: 향상된 Workflow

: Azure 퐅ㄹ에서 직접 로그 분석 알림에 액세스

**5)Alert Rules**

- Alert 생성 3단계

1. Alert 조건 정의

: 대상 선택

: Alert 기준(사용량 등)

2. Alert 세부 정의

: Alert Rule 이름, 설명, 심각도

3. Action Group 정의

: 전자 메일 및 문자 메시지를 통해 알림을 보내거나 Webhook 사용

**6) Action Groups**

- Action Group은 Alert에 대한 작업 구성에 용이

: Email/SMS/Push/Voice

: Logic App

: Webhook

: Automation Runbook

: IT Service Mangement

**7) Signal Types 과 Metrics**

- Signal은 대상 자원에 의해 출력

: Metrics

: Activity Log

: Application Insights

: Logs

- 최신 통계 알림 제공:

: 향상된 Latency

: 다차원 Metrics 지원

: Metrics 조건에 대한 추가 제어

: 다중 Metrics이 결합된 모니터링

: Metrics from Logs

**8) Activity Log 소개**

- 구독 레벨의 로그 캡처:

: Azure 리소스 매니저 운영 데이터

: Service Health Events

- 모든 쓰기/변경 작업의 감사 추적 기능 포함:

: 무엇을 누가 언제

: 각 작업의 상태를 제공

- 기록되는 내용:

: Event Hub로 스트리밍 / Power BI 분석 됨.

: Azure Portal에서 조회, Log Analytics 전달

: PowerShell, CLI, and REST API 처리

: Azure Storage Account에 저장.

**9) Activity Log 쿼리**

- Azure 포털에서 직접 사용할 수 있으며 다음 항목을 기반으로 필터링:

: Subscription

: Resource Group

: Resource(name)

: Resource type

: Operation

: Timespan

: Event category

: Event Severity

: Event Initiated by

- 필터 선택사항 저장 가능.

**10) Activity Log와 Log Analytics**

- Active Log를 Azure Log Analytics 작업 공간에 추가:

: 동일하거나 다른 구독(동일한 Azure AD tenant일시)

: 지원하는 기능:

1. 사용자 활동

2. 활동 로그를 다른 애저 서비스 및 App에 연결

3. 여러 Azure 구독에서 활동 로그 분석

4. 상태 별로 집계된 운영 활동

5. 사전 정의 된 보기로 활동 로그 분석

6. 정전 또는 서비스 상태 식별

7. 승인 변경 사항에 대한 리포트

8. 활동 동향 분석

**11) Collet Across Subscriptions**

- Logic Apps 용 Azure Log Analytics Data Collector connector를 사용하여 Azure 활동 로그를 Log Analytics

작업 공간으로 수집 가능.

: 다른 Azure AD Tenant가 있는 구독에서 작업 영역을 쉽게 사용 가능.

: 이와 같은 구성 시 여러 장점이 있다.

1. Azure Activity Log가 이벤트 허브로 스트리밍 되기 때문에 대기 시간이 짧다.

2. 배포 할 인프라 구성 요소가 없는 Serverless 서비스

3. 최소한의 사용자 정의 코드

**12) Log Analytics**

- 시나리오

: 시나리오1: 업데이트 평가 (시스템 업데이트 요구 사항 화인, 업데이트 배포 계획)

: 시나리오2: 변경 내역 추적

**13) Analyzing Log Analytics Data**

- OMS Portal

: 레거시 인터페이스.

: 사용 중지 예정.

- Azure Portal

**14) Log Analytics Querying**

- Log Analytics Queries:

: 데이터 검색, 집계 미 분석 용이.

: 저장 후 조회.

**[실습 자료] Lab: Exploring Monitoring Capabilities in Azure**

<https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies/blob/master/Instructions/AZ-300T01_Lab_Mod01_Exploring%20Monitoring%20Capabilities%20in%20Azure.md>

**15강. 컴퓨팅 및 스토리지 솔루션 선택**

**1) Azure Architecture Center**

- 가이드, 디자인 패턴, 모범 사례:

: Azure 애플리케이션 아키텍쳐 가이드

: Azure 데이터 아키텍처 가이드

: Azure 가상 데이터 센터

: 참조 아키텍처

: Azure Building 블록

: 클라우드 모범 사례

: 설계 검토 체크 리스트

: AWS 전문가를 위한 Azure

: Azure에서 마이크로서비스 제작

- 링크: <https://docs.microsoft.com/en-us/azure/architecture/>

**2) Retry Pattern**

- 문제점: 클라우드 서비스의 일시적인 오류

- 솔루션: 일시적으로 실패한 요청을 다시 시도하는 응용 프로그램 논리

**3) Competing Consumers Pattern**

- 문제점: 다양한 양 또는 많은 양의 요청 처리

- 솔루션: 다양한 메시지 생성자 및 소비자와의 비동기 메시징

**4) Cache-aside Pattern**

- 문제점: 캐시 된 데이터 일관성

- 솔루션: 연속 읽기 및 연속 기입 캐싱

:

**5) Sharding Pattern**

- 문제점: 전통적인 단일 인스턴스 저장소에서 많은 양의 데이터 호스팅

- 솔루션: 많은 노드에서 데이터를 가로로 파티션하기.

**6) Partitioning Workloads**

- 예제: 사진 공유 응용 프로그램 (아래와 같이 각 구성 요소는 특정 요구 사항을 충족시키지 위해 격리되어 확장)

: 가상 컴퓨터(Signal-R허브)

: 저장소 서비스(썸네일 저장)

: 웹앱(웹 프론트 엔드)

: Background Worker

**7) The Valet Key Pattern**

- 응용 프로그램이 클라이언트를 대신하여 리소스에 액세스하면 서버에 추가 로드가 발생.

- 리소스를 공개적으로 설정하고 응용 프로그램에서 클라이언트의 유효성 검사하고 싶지 않다.

- Valet Key 패턴은 응용 프로그램이 클라이언트의 유효성을 검사한 다음, 클라이언트에 토큰을 반환하는 것.

그러면 클라이언트는 자체 하드웨어 및 대역폭을 사용하여 직접 리소스를 검색할 수 있다.

- 클라이언트가 응용 프로그램에 자원을 요청.

- 응용 프로그램이 클라이언트의 유효성을 검사한 다음 액세스 토큰을 반환.

- 클라이언트는 반환된 토큰을 사용하여 리소스에 직접 접근.

**8) CQRS 패턴**

- 읽기와 쓰기간 잠재적인 불일치

- 데이터 접근 계층에서 가장 많은 양의 요청 발생(데이터베이스, ID와 보안)

- 읽기 및 쓰기 모델을 분리하여 문제 해결

: 각 모델은 별도로 관리 할 수 있다.

: 모델을 배치하여 계산 인스턴스를 분리 할 수 있다.

-> 쓰기 작업량이 많은 작업에는 쓰기 인스턴스가 더 필요.

**9) Throttling 패턴**

- 사용상 “상승 한계”를 만들어서 문제를 해결

:

**10) Static Content Hosting 패턴**

- 정적 콘텐츠는 종종 CDN에서 호스팅되어 가장 효율적인 방식으로 클라이언트에 직접 콘텐츠를 제공

- 웹 서버는 정적 컨텐츠를 제공 할 수 있지만 올바른 선택이 아닐 수 있다.

- 컴퓨팅 비용을 최소화 하는 웹 호스팅: 특히 정적 콘텐츠로만 구성된 웹 사이트의 경우

- 최종 사용자 컨텐츠 성능 향상.

**16강. Azure 워크로드 요구 사항 결정**

1. **Workload 요구 사항 결정을 위한 샘플 예제**
2. **Checklist of Assessment Goals**

- After Migration

- Understanding Compatibility

- Minimize Impact of Potential Compatibility Issues

- Costs Associated with the Infrastructure

**[실습 자료] Managing Security and Identity for Azure Solutions. Lab Answer Key: Securing Secrets in Azure**

<https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T01_Lab_Mod01_Securing%20Secrets%20in%20Azure.md>

**17강. 응용 프로그램 서비스 구현 및 관리**

**1) Web app**

- App Service 운영이 포함된 서비스

: 여러 언어 및 프레임 워크

: 개발 최적화

: 고가용성을 갖춘 글로벌 규모

: SaaS 플랫폼 및 사내 구축 형 데이터에 대한 연결

: 보안 및 규정 준수

: 응용 프로그램 템플릿

: Visual Studio 통합

: API 및 모바일 기능

: 서버리스 코드

**2) Continuous Development**

- 소스 제어 코드에 대한 업데이트 자동 게시

: Visual Studio Team Services

: BitBucket

: GitHub

: Git

**3) App Service 스테이징 환경**

- A separate deployment slot of App Service services:

: 웹 앱, 모바일 앱 및 API 앱에서 지원

: 높은 가격 책정 등급 필요:

1. Standard
2. Premium
3. Isolated

: 여러 가지 이점 제공:

1. 앱 변경 사항에 대한 추가 확인
2. 스왑 전에 슬롯 워밍업
3. 자동 스왑 지원
4. 즉각적인 롤백

**4) swap Deployment Slots**

- 슬롯 스와핑 설정의 세 가지 유형

: 스왑 되지 않은 설정

: 스왑 된 설정

: 구성 가능한 설정(스왑 또는 스왑되지 않음)

**5) Enabling Continuous Deployment**

- Azure 포털에서 구성 가능

: VSTS

: OneDrive

: Local Git

: GitHub

: BitBucket

: Dropbox

: External Repository

**6) Backup Your App**

- Azure 웹앱 백업을 구성하는데 필요한 요구사항

1. 백업 저장소를 지정하는 백업 저장소(Azure 저장소 계정 및 컨테이너)

2. 백업 데이터 베이스(웹 응용 프로그램에서 백엔드 데이터베이스를 사용하는 경우)

3. 백업 일정 (백업을 자동화 하기 위해)

**7) Restore**

- Azure Portal에서 다운로드 가능

- 사용 가능한 가격 계층

: Standard

: Premium

**8) Snapshot**

- 웹 앱의 자동, 증분 섀도 복사본

: 프리미엄 가격 책정 단계 필요

: 제약 조건: 동일한 앱 또는 해당 앱에 속한 슬롯에만 복원 가능.

**9) Cloning an app**

- 동일하거나 다른 지역에 웹 앱의 복제본 생성

: 프리미엄 가격 책정 단계 필요.

**10) 보안**

- 인프라 및 플랫폼 보안

: Azure 플랫폼에 의해 제어

: 컴퓨팅, 스토리지, 네트워크, 웹 프레임 워크, 통합 기능 등을 포함.

: 지속적인 보안 및 적합성 검사 포함.

- 응용 프로그램 보안:

: 고객이 관리

: Azure AD통합, 인증서 관리, 보안 통신 등을 포함.

**11) App Service Authentication**

- 아이디 공급자(IdP)와의 트러스트 관계에 의존

: 앱 서비스 앱의 ID제공 업체를 선택

: 아이디 공급자가 인증을 처리하고 인증 토큰을 생성

- 인증 요청을 인증 endpoint로 보냄.

- 가장 많이 사용되는 ID 제공 업체 Built-in

**12) Authentication Providers**

- Azure 포털에서 직접 구성 가능

: 인증/권한 부여 기능에 의존

: federated ID 사용

**18강. Serverless Computing**

**1) 컨셉**

- 서버, 인프라 및 운영 체제 추상화

- 거의 실시간으로 이벤트 및 트리거에 반응

- 다양한 혜택을 제공:

: 관리 오버 헤드 제거

: 개발자는 비즈니스 로직에 집중할 수 있습니다.

: 유연한 스케일링 구현.

: 비용 절감.

**2) Managing Azure Functions**

- Serverless compute service

: 인프라를 명시적으로 규정하거나 관리 할 필요가 없다.

: 주문형 또는 트리거에 대한 응답으로 코드 실행 가능.

: 활성화 된 경우에만 요금 청구

**3) Function Templates**

- 특정 트리거 유형에 대한 함수 작성 단순화

**4) implementing Functions**

- In-portal authoring

-In-portal monitoring

**5) Bindings**

- 다른 서비스와 기능 연결:

: Inputs

: Outputs

**6) Function Scaling**

- 플랫폼 관리 형 스케일 컨트롤러에 의존

- 스케일 컨트롤러

: 이벤트 비율 모니터링

: 수평 스케일링을 제어합니다.

: 휴리스틱에 의존

**7) Overview of Event Grid**

- 다양한 혜택을 제공하는 관리형 이벤트 라우팅 서비스:

: 이벤트 전달 간소화

: 안정적인 전달

**8) Event Grid Concepts**

- Event Sources

- Topics

- Event Subscriptions

- Event Handlers

**[실습 자료] Authoring Serverless Applications in Azure. Lab Answer Key: Deploying Serverless Workloads to Azure**

<https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T03_Lab_Mod03_Deploying%20Serverless%20Workloads%20to%20Azure.md>

**19강. Service Bus**

**1) Queues**

- Azure 서비스 버스 대기열의 제공 기능

: 비동기식 중개인 메시징

**2) Topics and Subscriptions**

- 게시/구독 패턴의 1:N 통신

- 많은 수의 수신자를 확장하는데 유용

**3) Service Bus Features**

- Load Leveling

- Loose coupling

- Load Balancing

20강. Azure Kubernetes

1) AKS 개념

2) AKS 서비스 연동

3) Azure Container Registry

- 주요 개념

: Registry

: Repository

: Image

: Container

- ACR은 Docker Registry 2.0에 기반한 관리 컨테이너 레지스트리이다.

4) Azur Container Instances overview

- 다양한 혜택을 제공하는 관리 컨테이너 호스팅 옵션

: 빠른 시작 시간

: 공용IP연결 및 DNS 이동

5) Implement an application using Virtual Kubelet

- Virtual Kubelet provider는 실험적인 오픈 소스 프로젝트

- Virtual Kubelet provider를 설치하려면 전제 조건 충족 확인 필요

: 기존 AKS클러스터

: Azure CLI 2.0.33 및 Helm 설치

: Tiller와 함께 사용할 서비스 계정 및 역할 바인딩

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

:

3)

-

: