**제조업 MES 공장자동화 시스템 현황 및 업무 특성**

**시스템 현황**

1. **한국의 온프레미스 인프라**
   * 기존 MES 시스템이 온프레미스 환경에서 운영 중.
   * 데이터 센터에 서버, 스토리지, 네트워크 장비 등이 구성되어 있음.
   * 데이터 보안 및 규제 준수 필요.
2. **사업 확장 계획**
   * 유럽과 베트남으로 공장을 확장할 예정.
   * 각 지역의 공장에서도 MES 시스템을 운영하고, 중앙에서 통합 관리 필요.

**업무 특성**

1. **실시간 데이터 처리**
   * 공장에서 실시간으로 생산 데이터와 품질 데이터를 수집 및 분석.
   * 데이터의 정확성과 신뢰성 확보가 중요.
2. **다국적 데이터 관리**
   * 한국, 유럽, 베트남에서 수집된 데이터를 통합 관리.
   * 각 지역의 법규 및 데이터 보호 규정을 준수해야 함.
3. **확장성 및 유연성**
   * 공장 확장에 따라 시스템의 유연한 확장이 필요.
   * 새로운 지역 공장에 신속하게 시스템을 배포할 수 있어야 함.
4. **보안**
   * 데이터 보호 및 네트워크 보안이 중요.
   * 각 지역에서 데이터 접근 및 전송 시 보안이 보장되어야 함.
5. **비용 효율성**
   * 효율적인 비용 관리 필요.
   * 클라우드 인프라를 사용하여 초기 투자 비용 절감.
6. **재해 복구**
   * 각 지역에서의 재해 발생 시 빠른 복구를 위한 계획 필요.
   * 데이터 백업 및 복구 전략이 중요.

**아키텍처 요구 사항**

1. **하이브리드 클라우드 인프라**
   * 온프레미스 인프라와 Azure 클라우드를 통합한 하이브리드 클라우드 아키텍처 구축.
   * Azure Arc를 사용하여 온프레미스 및 클라우드 리소스의 일관된 관리.
2. **네트워크 연결성**
   * 한국, 유럽, 베트남 간 안정적이고 고속의 네트워크 연결 구축 (Azure ExpressRoute 사용 고려).
   * VPN 게이트웨이를 통한 보안 네트워크 연결.
3. **데이터 통합 및 동기화**
   * 한국, 유럽, 베트남 간 데이터 동기화 및 통합 관리.
   * Azure Data Factory를 사용한 데이터 이동 및 변환.
4. **데이터 보호 및 보안**
   * Azure Security Center를 통한 통합 보안 관리.
   * 각 지역의 데이터 보호 규정 준수를 위한 Azure Policy 및 Azure Key Vault 사용.
5. **실시간 데이터 처리**
   * Azure IoT Hub를 통해 실시간으로 제조 데이터를 수집 및 처리.
   * Azure Stream Analytics를 사용하여 실시간 데이터 분석.
6. **확장성**
   * 필요에 따라 쉽게 확장 가능한 Azure Virtual Machines 및 Azure Kubernetes Service (AKS) 사용.
7. **재해 복구 및 고가용성**
   * Azure Site Recovery를 사용하여 재해 복구 계획 수립.
   * 여러 지역에 걸친 고가용성 아키텍처 구축 (Azure Availability Zones 사용).
8. **비용 관리**
   * Azure Cost Management를 통해 효율적인 비용 관리 및 모니터링.
   * 예약 인스턴스 및 Azure Hybrid Benefit을 통한 비용 절감.
9. **모니터링 및 관리**
   * Azure Monitor 및 Log Analytics를 사용하여 시스템 상태 모니터링.
   * Azure Automation을 통한 운영 효율성 향상.
10. **개발 및 배포 파이프라인**
    * Azure DevOps를 사용한 CI/CD 파이프라인 구축.
    * 코드 배포 자동화 및 인프라 구성 관리.
11. **AI 및 머신러닝**
    * Azure Machine Learning을 사용하여 생산 최적화 및 예측 분석 구현.
12. **API 및 서비스 통합**
    * Azure API Management를 사용하여 여러 서비스 및 API 통합 관리.
13. **사용자 접속/인증 관리**
    * Azure Active Directory B2C를 사용하여 사용자 인증 및 관리.
    * 각 지역의 사용자 접근 제어 및 보안 강화.
14. **대량 배치 작업 관리**
    * Azure Batch를 사용하여 대량 배치 작업을 효율적으로 관리.
    * 배치 작업이 온라인 서비스에 영향을 주지 않도록 구성.
15. **글로벌 트래픽 관리**
    * Azure Traffic Manager를 사용하여 글로벌 트래픽을 효율적으로 분산.
    * 지역별 최적화된 사용자 경험 제공.
16. **실시간 로그 수집 및 분석**
    * Azure Log Analytics를 사용하여 실시간 로그 수집 및 분석.
    * 문제 발생 시 빠른 대응 및 해결.
17. **네트워크 보안 및 제어**
    * Azure Firewall 및 Azure DDoS Protection을 사용하여 네트워크 보안 강화.
    * IP, Port, URL 기반으로 트래픽 제어.
18. **기존 인프라와의 연동**
    * 기존 온프레미스 시스템과의 원활한 연동을 위해 Azure Hybrid Connections 사용.
    * 데이터 및 애플리케이션의 이관을 위한 전략 수립.

각 항목별 상세 구현 방안

# 아키텍처 요구 사항: 1.하이브리드 클라우드 인프라

**구현 방안**

**1. 온프레미스 인프라와 Azure 클라우드 통합**

* **Azure Arc를 사용한 일관된 리소스 관리**
  + **설정**: Azure Arc를 사용하여 온프레미스와 멀티 클라우드 환경을 Azure와 통합하여 일관된 관리 및 거버넌스를 구현합니다.
  + **구성 방법**:
    1. **Azure Arc 설정**:
       - Azure Arc를 통해 온프레미스와 클라우드 자산을 연결합니다.
       - Azure Portal에서 Azure Arc 서비스를 활성화하고 리소스를 등록합니다.
       - Azure Policy 및 Azure Security Center를 사용하여 일관된 보안 및 정책 적용을 보장합니다.
    2. **하이브리드 리소스 관리**:
       - Azure Arc를 통해 Kubernetes 클러스터 및 서버를 관리합니다.
       - 온프레미스 및 멀티 클라우드 환경에서 Azure 리소스 관리자(ARM) 템플릿을 사용하여 인프라를 프로비저닝합니다.
    3. **모니터링 및 분석**:
       - Azure Monitor와 Azure Log Analytics를 사용하여 온프레미스와 클라우드 리소스의 상태를 모니터링합니다.
       - Azure Sentinel을 통해 통합된 보안 모니터링 및 위협 감지를 수행합니다.

**2. 네트워크 연결성 구성**

* **Azure ExpressRoute 사용**
  + **설정**: 온프레미스 데이터 센터와 Azure 간에 고속 전용 연결을 설정합니다.
  + **구성 방법**:
    1. **ExpressRoute 회로 설정**:
       - Azure Portal에서 ExpressRoute 회로를 생성합니다.
       - 로컬 통신사 또는 네트워크 제공자를 통해 ExpressRoute 연결을 설정합니다.
    2. **VPN 게이트웨이 설정**:
       - ExpressRoute와 함께 VPN 게이트웨이를 구성하여 추가적인 보안 레이어를 제공합니다.
       - Azure VNet을 온프레미스 네트워크와 연결하여 하이브리드 클라우드 네트워크를 구성합니다.
    3. **네트워크 성능 최적화**:
       - 네트워크 성능을 모니터링하고 최적화하기 위해 Azure Network Watcher를 사용합니다.
       - QoS(서비스 품질) 설정을 통해 중요한 트래픽의 우선 순위를 지정합니다.

**3. 데이터 통합 및 동기화**

* **Azure Data Factory 사용**
  + **설정**: Azure Data Factory를 사용하여 온프레미스와 클라우드 간의 데이터 이동 및 변환을 자동화합니다.
  + **구성 방법**:
    1. **데이터 파이프라인 설정**:
       - Azure Data Factory에서 데이터 이동 및 변환 파이프라인을 구성합니다.
       - 온프레미스 데이터베이스에서 Azure SQL Database 또는 Azure Data Lake Storage로 데이터를 이동합니다.
    2. **데이터 동기화**:
       - 데이터 동기화를 위해 주기적인 데이터 이동 작업을 설정합니다.
       - 데이터 변경 감지를 위해 Change Data Capture(CDC)를 구성합니다.
    3. **데이터 변환**:
       - 데이터 변환 로직을 구현하여 데이터 형식을 일관되게 유지합니다.
       - Azure Data Factory 내에서 데이터 흐름을 설계하여 데이터 정제 및 변환 작업을 수행합니다.

**기대 효과**

* **유연한 확장성 및 자원 관리**: 온프레미스와 클라우드 환경을 통합하여 자원을 효율적으로 관리하고 필요에 따라 확장할 수 있습니다.
* **통합된 운영 환경 제공**: Azure Arc를 사용하여 일관된 관리 및 거버넌스를 구현함으로써 운영 효율성을 높일 수 있습니다.
* **보안 및 규제 준수 강화**: 일관된 보안 정책과 모니터링을 통해 데이터 보호 및 규제 준수를 보장합니다.

**고려 사항 및 제약 사항**

* **네트워크 연결성 확보**: 온프레미스와 Azure 간의 안정적인 네트워크 연결을 확보해야 합니다. ExpressRoute 설치 및 유지 보수 비용이 발생할 수 있습니다.
* **데이터 전송 비용**: 데이터 전송 시 발생하는 비용을 고려해야 합니다. 특히 대량의 데이터를 주기적으로 전송할 경우 비용이 증가할 수 있습니다.
* **클라우드와 온프레미스 환경 간의 통합 문제**: 두 환경 간의 원활한 통합을 위해 호환성 및 성능 최적화를 지속적으로 관리해야 합니다.
* **보안 설정 복잡성**: 하이브리드 클라우드 환경에서 일관된 보안 정책을 적용하는 것이 복잡할 수 있습니다. Azure Policy와 Azure Security Center를 통해 관리합니다.
* **데이터 일관성 및 동기화 지연**: 데이터 통합 및 동기화 과정에서 발생할 수 있는 지연 및 일관성 문제를 고려해야 합니다. 이를 해결하기 위해 적절한 데이터 동기화 전략을 수립합니다.

**참조할 구성도**

* Azure 하이브리드 클라우드 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure 하이브리드 클라우드 아키텍처](https://docs.microsoft.com/en-us/azure/architecture/hybrid/)
  + [Azure Arc 개요](https://docs.microsoft.com/en-us/azure/azure-arc/overview)

이 구성을 통해 제조업 MES 공장자동화 시스템을 위한 하이브리드 클라우드 인프라를 구축하면 유연하고 효율적인 자원 관리와 안정적인 운영 환경을 제공할 수 있습니다.

# 아키텍처 요구 사항: 2.네트워크 연결성

**구현 방안**

**1. Azure ExpressRoute 사용**

* **설정**: 온프레미스 데이터 센터와 Azure 클라우드 간의 고속 전용 연결을 설정합니다. Azure ExpressRoute는 인터넷을 경유하지 않고 전용 네트워크 경로를 사용하여 안정적이고 빠른 연결을 제공합니다.
  + **구성 방법**:
    1. **ExpressRoute 회로 생성**:
       - Azure Portal에서 ExpressRoute 회로를 생성합니다.
       - 원하는 대역폭과 피어링 위치를 선택합니다.
    2. **ExpressRoute 서비스 제공자와의 연결 설정**:
       - 로컬 통신사 또는 네트워크 서비스 제공자를 통해 ExpressRoute 회로를 물리적으로 연결합니다.
       - 서비스 제공자가 제공한 라우터 또는 스위치 장비를 통해 연결을 설정합니다.
    3. **피어링 설정**:
       - Azure Portal에서 프라이빗 피어링 및 공용 피어링을 구성합니다.
       - VNet 피어링을 설정하여 Azure Virtual Network와 온프레미스 네트워크 간의 연결을 구성합니다.
    4. **라우팅 설정**:
       - BGP(Border Gateway Protocol)를 사용하여 라우팅 정보를 교환합니다.
       - 온프레미스 라우터와 Azure의 BGP 피어링을 구성하여 경로를 설정합니다.
    5. **네트워크 성능 최적화**:
       - QoS(서비스 품질) 설정을 통해 중요한 트래픽의 우선 순위를 지정합니다.
       - Azure Network Watcher를 사용하여 네트워크 성능을 모니터링하고 최적화합니다.

**2. VPN 게이트웨이 설정**

* **설정**: VPN 게이트웨이를 구성하여 온프레미스와 Azure 간의 보안 네트워크 연결을 설정합니다. ExpressRoute와 함께 VPN을 사용하여 추가적인 보안 레이어를 제공합니다.
  + **구성 방법**:
    1. **VPN 게이트웨이 생성**:
       - Azure Portal에서 VPN 게이트웨이를 생성합니다.
       - 가상 네트워크 게이트웨이를 선택하고 VPN 타입을 설정합니다.
    2. **VPN 연결 설정**:
       - 온프레미스 라우터 또는 방화벽에서 VPN 터널을 구성합니다.
       - IPsec/IKE 프로토콜을 사용하여 보안 연결을 설정합니다.
    3. **게이트웨이 서브넷 구성**:
       - Azure Virtual Network 내에 게이트웨이 서브넷을 설정하여 VPN 게이트웨이를 배치합니다.
    4. **로컬 네트워크 게이트웨이 생성**:
       - Azure Portal에서 로컬 네트워크 게이트웨이를 생성하여 온프레미스 네트워크 정보를 구성합니다.
    5. **VPN 연결 테스트**:
       - VPN 연결 상태를 모니터링하고, 필요 시 문제를 해결합니다.
       - Azure Network Watcher를 사용하여 VPN 연결 상태와 성능을 모니터링합니다.

**3. 네트워크 보안 및 최적화**

* **설정**: 네트워크 보안을 강화하고 성능을 최적화하기 위한 다양한 설정을 적용합니다.
  + **구성 방법**:
    1. **네트워크 보안 그룹(NSG) 설정**:
       - Azure Virtual Network 내의 서브넷 및 네트워크 인터페이스에 NSG를 적용하여 트래픽 필터링을 설정합니다.
       - 허용 및 차단 규칙을 정의하여 네트워크 보안을 강화합니다.
    2. **Azure Firewall 구성**:
       - Azure Firewall을 설정하여 중앙 집중식 보안 정책을 관리합니다.
       - 애플리케이션 및 네트워크 규칙을 정의하여 트래픽을 제어합니다.
    3. **DDoS 보호 설정**:
       - Azure DDoS Protection을 활성화하여 분산 서비스 거부(Distributed Denial of Service) 공격으로부터 네트워크를 보호합니다.
    4. **네트워크 성능 모니터링**:
       - Azure Network Watcher와 Traffic Analytics를 사용하여 네트워크 성능을 모니터링합니다.
       - 트래픽 패턴을 분석하고, 병목 현상을 해결합니다.

**기대 효과**

* **안정적이고 고속의 네트워크 연결**: Azure ExpressRoute와 VPN 게이트웨이를 통해 안정적이고 고속의 네트워크 연결을 확보할 수 있습니다.
* **보안 강화**: VPN과 네트워크 보안 그룹(NSG), Azure Firewall, DDoS 보호를 통해 네트워크 보안을 강화합니다.
* **글로벌 확장성 지원**: 다양한 지역에서의 네트워크 연결성을 확보하여 글로벌 비즈니스 확장을 지원합니다.

**고려 사항 및 제약 사항**

* **네트워크 연결 비용**: ExpressRoute와 VPN 설정 및 유지 보수 비용이 발생할 수 있습니다. 특히 고대역폭 회로의 경우 비용이 높을 수 있습니다.
* **복잡한 설정**: ExpressRoute와 VPN, NSG, Azure Firewall 등의 설정이 복잡할 수 있으며, 전문적인 네트워크 지식이 필요합니다.
* **라우팅 및 피어링 관리**: BGP 피어링 및 라우팅 설정을 올바르게 관리해야 하며, 잘못된 설정은 네트워크 성능 저하나 연결 문제를 야기할 수 있습니다.
* **네트워크 성능 모니터링 필요**: 네트워크 성능을 지속적으로 모니터링하고 최적화해야 합니다. 이를 위해 Azure Network Watcher와 같은 도구를 사용합니다.

**참조할 구성도**

* Azure 하이브리드 클라우드 네트워크 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure ExpressRoute 개요](https://docs.microsoft.com/en-us/azure/expressroute/expressroute-introduction)
  + [Azure VPN Gateway 개요](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways)
  + [Azure Network Watcher 개요](https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-monitoring-overview)

이 구현 방안을 통해 네트워크 연결성을 강화하고 안정적이고 고속의 네트워크 환경을 구축하여 제조업 MES 공장자동화 시스템의 글로벌 확장을 지원할 수 있습니다.

# 아키텍처 요구 사항: 3.데이터 통합 및 동기화

**구현 방안**

**1. Azure Data Factory를 사용한 데이터 통합**

* **설정**: Azure Data Factory (ADF)는 클라우드 기반 데이터 통합 서비스로, 다양한 데이터 소스 간의 데이터 이동, 변환 및 통합을 자동화합니다.
  + **구성 방법**:
    1. **Azure Data Factory 인스턴스 생성**:
       - Azure Portal에서 Data Factory를 생성합니다.
       - 리소스 그룹과 위치를 선택하고, Data Factory 이름을 지정합니다.
    2. **데이터 소스 및 싱크 설정**:
       - ADF 내에서 온프레미스 데이터 소스(예: SQL Server)와 클라우드 데이터 싱크(예: Azure SQL Database, Azure Data Lake Storage)를 설정합니다.
       - Integration Runtime(IR)을 사용하여 온프레미스 데이터에 안전하게 접근합니다.
    3. **파이프라인 생성**:
       - 데이터를 이동 및 변환할 파이프라인을 생성합니다.
       - 복사 활동(Copy Activity)을 사용하여 데이터를 소스에서 싱크로 이동합니다.
       - 데이터 변환이 필요한 경우 매핑 데이터 흐름(Data Flow)을 사용하여 데이터를 변환합니다.
    4. **스케줄링 및 트리거 설정**:
       - 파이프라인을 주기적으로 실행하기 위해 스케줄 트리거를 설정합니다.
       - 이벤트 기반 트리거를 설정하여 특정 이벤트 발생 시 파이프라인이 실행되도록 합니다.
    5. **모니터링 및 관리**:
       - ADF 모니터링 대시보드를 통해 파이프라인 실행 상태를 모니터링하고, 실패 시 경고를 설정합니다.

**2. Azure Data Factory를 사용한 데이터 동기화**

* **설정**: ADF를 사용하여 한국, 유럽, 베트남 간의 데이터 동기화 작업을 자동화합니다. Change Data Capture(CDC) 또는 Incremental Load를 사용하여 효율적으로 데이터를 동기화합니다.
  + **구성 방법**:
    1. **Change Data Capture(CDC) 설정**:
       - 소스 데이터베이스에서 CDC를 활성화하여 데이터 변경 사항을 캡처합니다.
       - ADF에서 CDC 데이터를 수집하여 변경된 데이터만 이동시킵니다.
    2. **Incremental Load 설정**:
       - ADF 파이프라인 내에서 Lookup Activity를 사용하여 마지막 동기화 이후 변경된 데이터를 식별합니다.
       - 데이터 이동 시 변경된 데이터만 이동하여 효율성을 높입니다.
    3. **Azure SQL Database와 Data Lake Storage 동기화**:
       - ADF 파이프라인을 사용하여 변경된 데이터를 Azure SQL Database 및 Data Lake Storage로 동기화합니다.
       - 데이터의 일관성을 유지하기 위해 데이터 무결성 검사를 포함합니다.
    4. **실시간 동기화 설정**:
       - Azure Event Grid 및 Azure Functions를 사용하여 실시간 데이터 동기화 트리거를 설정합니다.
       - 이벤트 기반 데이터 동기화를 통해 데이터 지연을 최소화합니다.

**기대 효과**

* **통합된 데이터 관리**: 다양한 지역에서 수집된 데이터를 중앙에서 통합 관리하여 데이터 일관성을 유지할 수 있습니다.
* **실시간 데이터 동기화**: 실시간 또는 주기적인 데이터 동기화를 통해 최신 데이터를 사용할 수 있습니다.
* **자동화된 데이터 처리**: 데이터 이동 및 변환 작업을 자동화하여 운영 효율성을 높입니다.

**고려 사항 및 제약 사항**

* **데이터 전송 비용**: 대량의 데이터를 클라우드로 이동하는 경우 데이터 전송 비용이 발생할 수 있습니다. 특히 빈번한 동기화 작업 시 비용이 증가할 수 있습니다.
* **데이터 보안**: 온프레미스에서 클라우드로 데이터를 이동할 때 데이터 보안이 중요합니다. 데이터 암호화 및 보안 설정을 철저히 관리해야 합니다.
* **네트워크 성능**: 데이터 전송 속도와 네트워크 지연을 최소화하기 위해 고속 네트워크 연결이 필요합니다. 네트워크 성능을 지속적으로 모니터링해야 합니다.
* **데이터 일관성 및 무결성**: 데이터 동기화 과정에서 데이터 일관성과 무결성을 유지해야 합니다. 이를 위해 데이터 무결성 검사와 오류 복구 메커니즘을 도입해야 합니다.
* **복잡한 설정**: ADF 설정 및 데이터 파이프라인 구축이 복잡할 수 있으며, 전문적인 데이터 엔지니어링 지식이 필요합니다.

**참조할 구성도**

* Azure Data Factory를 활용한 데이터 통합 및 동기화에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Data Factory 개요](https://docs.microsoft.com/en-us/azure/data-factory/introduction)
  + [Azure Data Factory를 사용한 데이터 이동](https://docs.microsoft.com/en-us/azure/data-factory/copy-data-tool)
  + [Change Data Capture (CDC) 구성](https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server)

이 구현 방안을 통해 데이터 통합 및 동기화를 효과적으로 수행하여 제조업 MES 공장자동화 시스템의 데이터 일관성 및 실시간 데이터를 보장할 수 있습니다.

# 아키텍처 요구 사항: 4.데이터 보호 및 보안

**구현 방안**

**1. Azure Security Center를 통한 통합 보안 관리**

* **설정**: Azure Security Center는 통합 보안 관리 솔루션으로, 보안 상태를 평가하고 위협을 탐지하며 보안 규정을 준수하도록 돕습니다.
  + **구성 방법**:
    1. **Azure Security Center 활성화**:
       - Azure Portal에서 Security Center를 활성화합니다.
       - 구독 및 리소스를 Security Center에 등록하여 보호를 시작합니다.
    2. **보안 정책 설정**:
       - Azure Policy를 통해 조직의 보안 규정을 설정합니다.
       - 기본 제공되는 보안 정책을 사용하거나, 사용자 정의 정책을 작성하여 적용합니다.
    3. **보안 점수 및 권장 사항 모니터링**:
       - Security Center의 보안 점수를 확인하고, 권장 사항에 따라 보안 설정을 강화합니다.
       - 보안 경고 및 권장 사항을 모니터링하고, 즉시 대응합니다.
    4. **위협 탐지 및 경고 설정**:
       - Advanced Threat Protection (ATP)을 사용하여 위협을 탐지합니다.
       - 보안 경고를 설정하고, 경고 발생 시 알림을 받도록 구성합니다.
    5. **통합 보안 대시보드 사용**:
       - Security Center 대시보드를 통해 전체 보안 상태를 한눈에 파악합니다.
       - 다양한 보안 메트릭과 경고를 실시간으로 모니터링합니다.

**2. Azure Policy 및 Azure Blueprints를 통한 규정 준수 보장**

* **설정**: Azure Policy와 Azure Blueprints를 사용하여 규정 준수를 자동화하고, 조직의 보안 표준을 지속적으로 유지합니다.
  + **구성 방법**:
    1. **Azure Policy 정의 및 할당**:
       - Azure Policy를 사용하여 리소스 구성 및 배포 시 규정 준수를 보장합니다.
       - 기본 제공되는 정책을 사용하거나, 사용자 정의 정책을 작성하여 적용합니다.
    2. **Azure Blueprints 사용**:
       - Azure Blueprints를 사용하여 정책, 역할 기반 액세스 제어(RBAC), 리소스 그룹 등을 포함한 환경을 표준화합니다.
       - Blueprints를 통해 일관된 환경을 배포하고, 규정 준수를 보장합니다.
    3. **규정 준수 상태 모니터링**:
       - Azure Policy 대시보드를 통해 규정 준수 상태를 모니터링합니다.
       - 비준수 리소스를 식별하고, 즉시 수정 조치를 취합니다.

**3. Azure Key Vault를 통한 데이터 보호**

* **설정**: Azure Key Vault는 암호화 키, 비밀 및 인증서를 안전하게 저장하고 관리하는 서비스입니다.
  + **구성 방법**:
    1. **Azure Key Vault 생성**:
       - Azure Portal에서 Key Vault를 생성합니다.
       - 리소스 그룹과 위치를 선택하고, Key Vault 이름을 지정합니다.
    2. **암호화 키 및 비밀 관리**:
       - Key Vault에 암호화 키, 비밀 및 인증서를 저장합니다.
       - 암호화 키를 사용하여 데이터 암호화 및 복호화를 수행합니다.
    3. **액세스 제어 설정**:
       - RBAC 및 정책을 사용하여 Key Vault에 대한 액세스를 제어합니다.
       - 필요한 경우 Azure AD 인증을 통해 Key Vault에 대한 액세스를 관리합니다.
    4. **키 회전 및 수명 주기 관리**:
       - Key Vault의 자동 키 회전 기능을 설정하여 암호화 키의 수명 주기를 관리합니다.
       - 주기적으로 키를 교체하고, 보안성을 유지합니다.

**4. 데이터 암호화**

* **설정**: 데이터 전송 및 저장 시 암호화를 사용하여 데이터를 보호합니다.
  + **구성 방법**:
    1. **전송 중 데이터 암호화**:
       - Azure Storage, SQL Database 등에서 TLS(Transport Layer Security)를 사용하여 데이터를 암호화합니다.
       - HTTPS를 사용하여 웹 애플리케이션의 데이터 전송을 보호합니다.
    2. **저장된 데이터 암호화**:
       - Azure Storage 서비스에서 자동으로 제공되는 서버 측 암호화(Server-Side Encryption)를 사용합니다.
       - Azure SQL Database, Cosmos DB 등에서 TDE(Transparent Data Encryption)를 사용하여 데이터를 암호화합니다.
    3. **고객 관리 키(CMK) 사용**:
       - Key Vault를 통해 고객 관리 키(CMK)를 사용하여 암호화를 관리합니다.
       - BYOK(Bring Your Own Key) 기능을 사용하여 키를 직접 관리합니다.

**기대 효과**

* **강화된 데이터 보안**: 데이터 암호화, 키 관리 및 보안 정책을 통해 데이터를 보호하고, 보안 위협을 최소화합니다.
* **규정 준수 보장**: Azure Policy와 Blueprints를 사용하여 보안 규정을 준수하고, 감사 준비 상태를 유지합니다.
* **통합 보안 관리**: Azure Security Center를 통해 통합된 보안 관리와 위협 탐지를 수행하여 보안 상태를 지속적으로 개선합니다.

**고려 사항 및 제약 사항**

* **보안 설정 복잡성**: 다양한 보안 설정을 일관되게 관리하는 것이 복잡할 수 있습니다. 전문적인 보안 지식이 필요합니다.
* **비용**: 보안 서비스 사용에 따른 추가 비용이 발생할 수 있습니다. Key Vault, Security Center 등 고급 보안 기능을 사용하는 경우 비용이 증가할 수 있습니다.
* **성능 영향**: 데이터 암호화 및 복호화 작업은 시스템 성능에 영향을 미칠 수 있습니다. 특히 대규모 데이터 작업 시 성능 저하를 최소화하기 위한 최적화가 필요합니다.
* **키 관리 책임**: 고객 관리 키(CMK)를 사용하는 경우, 키 관리 및 회전에 대한 책임이 고객에게 있습니다. 키의 보안성과 가용성을 유지해야 합니다.

**참조할 구성도**

* Azure 데이터 보호 및 보안 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Security Center 개요](https://docs.microsoft.com/en-us/azure/security-center/security-center-intro)
  + [Azure Policy 개요](https://docs.microsoft.com/en-us/azure/governance/policy/overview)
  + [Azure Key Vault 개요](https://docs.microsoft.com/en-us/azure/key-vault/general/overview)
  + [데이터 암호화](https://docs.microsoft.com/en-us/azure/security/fundamentals/encryption-atrest)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 데이터를 보호하고, 보안을 강화하며, 규정 준수를 보장할 수 있습니다.

# 아키텍처 요구 사항: 5.실시간 데이터 처리

**구현 방안**

**1. Azure IoT Hub를 통한 데이터 수집**

* **설정**: Azure IoT Hub는 다양한 IoT 디바이스로부터 데이터를 수집하고, 이를 실시간으로 처리할 수 있도록 지원하는 서비스입니다.
  + **구성 방법**:
    1. **Azure IoT Hub 생성**:
       - Azure Portal에서 IoT Hub를 생성합니다.
       - IoT Hub의 이름, 위치, 가격 책정 계층을 선택합니다.
    2. **디바이스 등록 및 관리**:
       - IoT Hub 내에서 디바이스를 등록합니다.
       - 각 디바이스에 대해 고유한 인증서나 SAS 토큰을 생성하여 보안을 강화합니다.
    3. **메시지 라우팅 설정**:
       - IoT Hub에서 수신한 데이터를 다양한 엔드포인트(예: Azure Stream Analytics, Azure Event Hubs)로 라우팅합니다.
       - 조건 기반 라우팅 규칙을 설정하여 특정 조건을 만족하는 데이터만 라우팅합니다.
    4. **디바이스 트윈 및 디바이스 관리**:
       - 디바이스 트윈을 사용하여 디바이스의 상태를 관리하고, 원격으로 설정을 변경합니다.
       - IoT Hub의 디바이스 관리 기능을 사용하여 펌웨어 업데이트, 재부팅 등 디바이스 관리 작업을 수행합니다.

**2. Azure Stream Analytics를 통한 실시간 데이터 처리**

* **설정**: Azure Stream Analytics는 스트리밍 데이터를 실시간으로 처리하고 분석할 수 있는 완전 관리형 서비스입니다.
  + **구성 방법**:
    1. **Stream Analytics 작업 생성**:
       - Azure Portal에서 Stream Analytics 작업을 생성합니다.
       - 입력(Input), 출력(Output), 쿼리(Query)를 설정합니다.
    2. **입력 설정**:
       - IoT Hub에서 스트리밍 데이터를 입력으로 설정합니다.
       - Event Hubs 또는 Blob Storage 등 다른 데이터 소스도 입력으로 설정할 수 있습니다.
    3. **쿼리 작성**:
       - SQL 유사 언어를 사용하여 스트리밍 데이터를 실시간으로 처리하는 쿼리를 작성합니다.
       - 예를 들어, 특정 조건을 만족하는 데이터를 필터링하거나 집계 작업을 수행합니다.

sql

SELECT

DeviceId,

AVG(Temperature) AS AvgTemperature,

System.Timestamp AS EventTime

FROM

IoTHubInput

GROUP BY

TumblingWindow(minute, 1),

DeviceId

* + 1. **출력 설정**:
       - Stream Analytics 작업의 출력을 Azure SQL Database, Cosmos DB, Power BI, Blob Storage 등으로 설정합니다.
       - 실시간 대시보드를 통해 분석 결과를 시각화할 수 있습니다.
    2. **작업 모니터링 및 관리**:
       - Stream Analytics 작업의 상태를 모니터링하고, 성능을 최적화합니다.
       - 경고 및 알림을 설정하여 작업 실패 시 즉시 대응합니다.

**3. Azure Functions 및 Logic Apps를 통한 이벤트 기반 처리**

* **설정**: Azure Functions와 Logic Apps는 이벤트 기반 데이터 처리를 자동화할 수 있는 서비스입니다.
  + **구성 방법**:
    1. **Azure Functions 생성**:
       - Azure Portal에서 Function App을 생성합니다.
       - 특정 이벤트 발생 시 실행될 함수를 작성합니다. 예를 들어, IoT Hub에서 데이터를 수신할 때 트리거되는 함수를 작성합니다.

[FunctionName("ProcessIoTData")]

public static void Run(

[IoTHubTrigger("messages/events", Connection = "IoTHubConnectionString")] EventData message,

ILogger log)

{

// 데이터 처리 로직

log.LogInformation($"C# IoT Hub trigger function processed a message: {Encoding.UTF8.GetString(message.Body.Array)}");

}

* + 1. **Logic Apps 구성**:
       - Azure Portal에서 Logic App을 생성합니다.
       - Logic Apps 디자이너를 사용하여 워크플로를 구성합니다. 예를 들어, IoT Hub에서 데이터를 수신하고, 이를 데이터베이스에 저장하거나 알림을 보내는 작업을 구성합니다.
       - 다양한 커넥터를 사용하여 외부 시스템과의 통합을 쉽게 수행합니다.

**기대 효과**

* **실시간 데이터 가시성**: IoT Hub와 Stream Analytics를 통해 실시간으로 데이터를 수집하고 분석하여, 공정의 상태를 실시간으로 모니터링할 수 있습니다.
* **빠른 문제 감지 및 대응**: 실시간 데이터 분석을 통해 이상 상태를 신속하게 감지하고, 문제 발생 시 즉시 대응할 수 있습니다.
* **자동화된 데이터 처리**: Azure Functions와 Logic Apps를 사용하여 이벤트 기반 데이터 처리를 자동화함으로써 운영 효율성을 높일 수 있습니다.

**고려 사항 및 제약 사항**

* **데이터 처리 비용**: 실시간 데이터 처리에는 많은 비용이 발생할 수 있습니다. 특히, 대량의 데이터를 지속적으로 처리할 경우 비용이 증가할 수 있습니다.
* **네트워크 대역폭**: 실시간 데이터 전송 시 네트워크 대역폭이 중요한 고려 사항입니다. 네트워크 지연이나 병목 현상을 최소화해야 합니다.
* **데이터 보안**: 실시간 데이터 전송 및 처리는 보안이 매우 중요합니다. 데이터 암호화 및 인증 메커니즘을 철저히 적용해야 합니다.
* **처리 지연**: 실시간 데이터 처리에서 지연 시간을 최소화하는 것이 중요합니다. 데이터 수집부터 분석까지의 전체 파이프라인을 최적화해야 합니다.
* **스케일링**: 처리량이 증가할 때 시스템이 자동으로 확장될 수 있도록 설정해야 합니다. Azure IoT Hub, Stream Analytics 및 Functions 모두 자동 확장 기능을 제공하지만, 적절한 설정이 필요합니다.

**참조할 구성도**

* Azure 실시간 데이터 처리 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure IoT Hub 개요](https://docs.microsoft.com/en-us/azure/iot-hub/about-iot-hub)
  + [Azure Stream Analytics 개요](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction)
  + [Azure Functions 개요](https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview)
  + [Azure Logic Apps 개요](https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템에서 실시간 데이터를 효과적으로 수집, 처리 및 분석할 수 있습니다.

# 아키텍처 요구 사항: 6.확장성

**구현 방안**

**1. Azure Virtual Machines (VMs) 사용**

* **설정**: Azure VMs는 필요에 따라 손쉽게 확장 및 축소가 가능한 가상 서버를 제공합니다.
  + **구성 방법**:
    1. **Azure VM 생성**:
       - Azure Portal에서 VM을 생성합니다.
       - VM 크기, 이미지, 리소스 그룹 등을 선택합니다.
    2. **가상 머신 스케일 세트(VMSS) 구성**:
       - VMSS를 사용하여 동일한 VM 인스턴스의 그룹을 자동으로 확장 및 축소합니다.
       - 자동 스케일링 규칙을 정의하여 CPU 사용률, 메모리 사용량 등의 기준에 따라 VM 인스턴스를 추가하거나 제거합니다.
    3. **관리형 디스크 사용**:
       - VM 인스턴스에 대해 관리형 디스크를 사용하여 성능을 최적화하고, 스케일링 시 디스크 관리를 단순화합니다.
    4. **로드 밸런서 구성**:
       - Azure Load Balancer를 사용하여 트래픽을 VM 인스턴스 간에 분산합니다.
       - 로드 밸런서 규칙을 설정하여 트래픽을 균등하게 분산하고, VM 인스턴스의 가용성을 보장합니다.

**2. Azure Kubernetes Service (AKS) 사용**

* **설정**: AKS는 컨테이너화된 애플리케이션을 손쉽게 배포, 관리 및 확장할 수 있는 완전 관리형 Kubernetes 클러스터를 제공합니다.
  + **구성 방법**:
    1. **AKS 클러스터 생성**:
       - Azure Portal에서 AKS 클러스터를 생성합니다.
       - 클러스터 이름, 노드 풀 크기, 가상 네트워크 설정 등을 지정합니다.
    2. **Horizontal Pod Autoscaler (HPA) 설정**:
       - HPA를 사용하여 CPU 사용률, 메모리 사용량 등의 메트릭에 따라 파드 수를 자동으로 확장 및 축소합니다.
       - kubectl 명령을 사용하여 HPA를 설정합니다.

yaml

코드 복사

apiVersion: autoscaling/v2beta2

kind: HorizontalPodAutoscaler

metadata:

name: myapp-hpa

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: myapp

minReplicas: 1

maxReplicas: 10

metrics:

- type: Resource

resource:

name: cpu

target:

type: Utilization

averageUtilization: 50

* + 1. **Cluster Autoscaler 설정**:
       - Cluster Autoscaler를 사용하여 노드 풀 크기를 자동으로 조정합니다.
       - az aks update 명령을 사용하여 Cluster Autoscaler를 설정합니다.

bash

코드 복사

az aks nodepool update \

--resource-group myResourceGroup \

--cluster-name myAKSCluster \

--name mynodepool \

--enable-cluster-autoscaler \

--min-count 1 \

--max-count 10

**3. Azure App Service 사용**

* **설정**: Azure App Service는 웹 애플리케이션을 자동으로 확장할 수 있는 완전 관리형 플랫폼입니다.
  + **구성 방법**:
    1. **App Service 생성**:
       - Azure Portal에서 App Service를 생성합니다.
       - 웹 애플리케이션을 배포하고, 필요한 설정을 구성합니다.
    2. **자동 스케일링 설정**:
       - Azure Portal에서 App Service의 스케일 아웃 규칙을 설정합니다.
       - CPU 사용률, 메모리 사용량, HTTP 요청 수 등의 기준에 따라 인스턴스를 자동으로 확장 및 축소합니다.
    3. **앱 인사이트 및 모니터링**:
       - Azure Application Insights를 사용하여 웹 애플리케이션의 성능을 모니터링하고, 자동 스케일링 트리거를 최적화합니다.

**기대 효과**

* **유연한 확장성**: 애플리케이션의 부하에 따라 자동으로 리소스를 확장 및 축소하여 필요에 맞게 자원을 효율적으로 사용합니다.
* **비용 절감**: 사용량에 따라 자원을 동적으로 조절함으로써 불필요한 리소스 사용을 줄이고, 비용을 절감할 수 있습니다.
* **향상된 성능 및 가용성**: 고가용성과 높은 성능을 유지하여 사용자 경험을 개선합니다.

**고려 사항 및 제약 사항**

* **비용 관리**: 자동 스케일링은 비용 절감에 도움이 되지만, 급격한 트래픽 증가로 인해 예상치 못한 비용이 발생할 수 있습니다. 비용 모니터링과 관리가 중요합니다.
* **성능 최적화**: 확장성 설정을 최적화하지 않으면 성능 저하가 발생할 수 있습니다. 스케일링 트리거를 적절히 설정하여 성능을 최적화해야 합니다.
* **네트워크 대역폭**: 스케일링 시 네트워크 대역폭 요구사항이 증가할 수 있습니다. 네트워크 인프라의 성능을 충분히 확보해야 합니다.
* **자동 스케일링 지연**: 스케일링 작업이 즉각적으로 수행되지 않을 수 있습니다. 스케일링 지연을 최소화하기 위해 적절한 초기 리소스를 설정하는 것이 중요합니다.
* **복잡한 설정**: 다양한 자동 스케일링 옵션을 설정하고 관리하는 것이 복잡할 수 있습니다. 전문적인 지식과 경험이 필요합니다.

**참조할 구성도**

* Azure 확장성 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Virtual Machines 개요](https://docs.microsoft.com/en-us/azure/virtual-machines/)
  + [Azure Kubernetes Service (AKS) 개요](https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes)
  + [Azure App Service 개요](https://docs.microsoft.com/en-us/azure/app-service/overview)
  + [Azure Auto Scaling](https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템에서 확장성을 효과적으로 관리하고, 고가용성과 성능을 보장할 수 있습니다.

# 아키텍처 요구 사항: 10.개발 및 배포 파이프라인

**구현 방안**

**1. Azure DevOps를 사용한 CI/CD 파이프라인 구축**

* **설정**: Azure DevOps는 지속적 통합(Continuous Integration, CI) 및 지속적 배포(Continuous Deployment, CD) 파이프라인을 구성할 수 있는 도구를 제공합니다. 이를 통해 코드 변경 사항을 자동으로 빌드, 테스트 및 배포할 수 있습니다.
  + **구성 방법**:
    1. **Azure DevOps 프로젝트 생성**:
       - Azure DevOps에서 새로운 프로젝트를 생성합니다.
       - 프로젝트 내에 필요한 리포지토리, 빌드 파이프라인, 릴리스 파이프라인 등을 설정합니다.
    2. **리포지토리 설정**:
       - Azure Repos를 사용하여 소스 코드를 저장하고 관리합니다.
       - Git 또는 Team Foundation Version Control (TFVC)을 사용하여 버전 관리를 수행합니다.
    3. **빌드 파이프라인 설정**:
       - Azure Pipelines를 사용하여 빌드 파이프라인을 설정합니다.
       - YAML 파일 또는 시각적 디자이너를 사용하여 빌드 정의를 작성합니다.
       - 코드 변경 사항이 푸시될 때마다 자동으로 빌드가 트리거되도록 설정합니다.

yaml

코드 복사

trigger:

- main

pool:

vmImage: 'ubuntu-latest'

steps:

- task: UseNode@1

inputs:

version: '10.x'

- script: npm install

displayName: 'Install dependencies'

- script: npm run build

displayName: 'Build the application'

- task: PublishBuildArtifacts@1

inputs:

PathtoPublish: '$(Build.ArtifactStagingDirectory)'

ArtifactName: 'drop'

* + 1. **테스트 자동화**:
       - 빌드 파이프라인 내에 단위 테스트 및 통합 테스트 단계를 포함합니다.
       - 테스트 도구(예: NUnit, JUnit, Selenium)를 사용하여 테스트를 자동으로 실행하고 결과를 확인합니다.
    2. **릴리스 파이프라인 설정**:
       - Azure Pipelines에서 릴리스 파이프라인을 설정합니다.
       - 빌드 아티팩트를 배포 환경(예: 개발, 테스트, 프로덕션)으로 배포합니다.
       - 각 단계에 대해 승인 워크플로를 설정하여 배포 전 검토 및 승인을 받습니다.
       - 배포 스크립트(예: ARM 템플릿, Terraform)를 사용하여 인프라를 프로비저닝하고 애플리케이션을 배포합니다.

yaml

코드 복사

stages:

- stage: Dev

jobs:

- deployment: DeployToDev

environment: Dev

strategy:

runOnce:

deploy:

steps:

- script: az group create --name myResourceGroup --location eastus

displayName: 'Create resource group'

- script: az deployment group create --resource-group myResourceGroup --template-file azuredeploy.json

displayName: 'Deploy ARM template'

**2. Infrastructure as Code (IaC) 도구 사용**

* **설정**: IaC 도구를 사용하여 인프라를 코드로 정의하고 관리합니다. 이를 통해 인프라 구성의 일관성을 유지하고, 자동화된 배포를 수행할 수 있습니다.
  + **구성 방법**:
    1. **ARM 템플릿 사용**:
       - Azure Resource Manager(ARM) 템플릿을 작성하여 Azure 리소스를 정의합니다.
       - JSON 형식의 ARM 템플릿을 사용하여 가상 네트워크, 가상 머신, 스토리지 계정 등 다양한 리소스를 정의합니다.
    2. **Terraform 사용**:
       - HashiCorp Terraform을 사용하여 클라우드 인프라를 코드로 정의합니다.
       - .tf 파일을 작성하여 인프라 리소스를 선언하고, terraform apply 명령을 사용하여 인프라를 프로비저닝합니다.

hcl

코드 복사

provider "azurerm" {

features {}

}

resource "azurerm\_resource\_group" "example" {

name = "example-resources"

location = "West Europe"

}

resource "azurerm\_storage\_account" "example" {

name = "examplestorageacc"

resource\_group\_name = azurerm\_resource\_group.example.name

location = azurerm\_resource\_group.example.location

account\_tier = "Standard"

account\_replication\_type = "LRS"

}

**3. 모니터링 및 알림 설정**

* **설정**: CI/CD 파이프라인의 상태와 성능을 모니터링하고, 문제 발생 시 알림을 설정합니다.
  + **구성 방법**:
    1. **Azure Monitor 사용**:
       - Azure Monitor를 사용하여 애플리케이션 및 인프라의 상태를 모니터링합니다.
       - 로그, 메트릭, 이벤트를 수집하고 분석하여 문제를 신속하게 파악합니다.
    2. **Azure Application Insights 설정**:
       - Application Insights를 사용하여 애플리케이션의 성능을 모니터링하고, 사용자의 행태를 분석합니다.
       - 애플리케이션 코드에 SDK를 통합하여 상세한 성능 데이터를 수집합니다.
    3. **알림 설정**:
       - Azure Alert를 사용하여 임계값을 초과하는 이벤트 발생 시 알림을 받습니다.
       - 이메일, SMS, 웹훅 등을 통해 알림을 설정하고, 즉시 대응할 수 있도록 합니다.

**기대 효과**

* **빠른 배포 주기**: CI/CD 파이프라인을 통해 코드 변경 사항을 신속하게 빌드, 테스트 및 배포할 수 있습니다.
* **개발 효율성 증가**: 자동화된 배포와 테스트를 통해 개발 팀의 효율성을 높이고, 품질을 유지합니다.
* **일관된 인프라 구성**: IaC 도구를 사용하여 인프라 구성을 코드로 관리함으로써 일관성과 재현성을 보장합니다.
* **문제 대응 시간 단축**: 모니터링 및 알림 설정을 통해 문제를 신속하게 파악하고 대응할 수 있습니다.

**고려 사항 및 제약 사항**

* **설정 복잡성**: CI/CD 파이프라인과 IaC 도구의 설정이 복잡할 수 있으며, 초기 설정에 많은 시간이 소요될 수 있습니다.
* **비용 관리**: Azure DevOps와 모니터링 도구의 사용에 따른 비용이 발생할 수 있습니다. 비용을 주기적으로 검토하고 최적화해야 합니다.
* **보안**: 코드 저장소, 빌드 아티팩트 및 배포 스크립트의 보안을 철저히 관리해야 합니다. 특히, 인증 정보와 같은 민감한 데이터를 안전하게 보호해야 합니다.
* **성능 최적화**: 대규모 애플리케이션의 경우 빌드 및 배포 시간이 길어질 수 있습니다. 이를 최적화하기 위한 전략이 필요합니다.

**참조할 구성도**

* Azure DevOps CI/CD 파이프라인 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure DevOps 개요](https://docs.microsoft.com/en-us/azure/devops/?view=azure-devops)
  + [Azure Pipelines 개요](https://docs.microsoft.com/en-us/azure/devops/pipelines/?view=azure-devops)
  + [Infrastructure as Code (IaC) 개요](https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 개발 및 배포 프로세스를 효율적으로 관리하고, 자동화된 CI/CD 파이프라인을 통해 신속하고 일관된 배포를 실현할 수 있습니다.

# 아키텍처 요구 사항: 9.모니터링 및 관리

**구현 방안**

**1. Azure Monitor를 통한 시스템 상태 모니터링**

* **설정**: Azure Monitor는 Azure 리소스 및 애플리케이션의 상태를 모니터링하고, 로그 및 메트릭을 수집하여 분석할 수 있는 서비스입니다.
  + **구성 방법**:
    1. **Azure Monitor 활성화**:
       - Azure Portal에서 Azure Monitor를 활성화합니다.
       - 모니터링할 리소스를 선택하고, 메트릭과 로그 수집을 설정합니다.
    2. **메트릭 수집**:
       - VM, 애플리케이션, 데이터베이스, 네트워크 등의 메트릭을 수집합니다.
       - 주요 메트릭에는 CPU 사용률, 메모리 사용량, 네트워크 트래픽, 디스크 I/O 등이 포함됩니다.
    3. **로그 수집**:
       - Azure Log Analytics 워크스페이스를 생성하여 로그 데이터를 수집합니다.
       - 애플리케이션 로그, 시스템 로그, 보안 로그 등을 수집합니다.
    4. **대시보드 생성**:
       - Azure Portal에서 사용자 정의 대시보드를 생성하여 주요 메트릭과 로그를 시각화합니다.
       - Grafana와 같은 도구를 사용하여 대시보드를 확장하고, 시각화 기능을 강화할 수 있습니다.

**2. Azure Log Analytics를 통한 로그 분석**

* **설정**: Azure Log Analytics는 로그 데이터를 수집, 분석 및 시각화하는 데 사용됩니다. 이를 통해 다양한 소스에서 수집된 로그를 중앙에서 관리할 수 있습니다.
  + **구성 방법**:
    1. **Log Analytics 워크스페이스 생성**:
       - Azure Portal에서 Log Analytics 워크스페이스를 생성합니다.
       - 워크스페이스에 대한 리소스 그룹과 위치를 선택합니다.
    2. **데이터 소스 연결**:
       - Azure VM, Azure App Service, Azure Functions 등 다양한 소스에서 로그 데이터를 수집하도록 설정합니다.
       - Log Analytics 에이전트를 설치하여 온프레미스 및 클라우드 리소스에서 로그를 수집합니다.
    3. **쿼리 작성 및 분석**:
       - Kusto Query Language (KQL)를 사용하여 로그 데이터를 쿼리하고 분석합니다.
       - 예를 들어, 특정 시간 범위 동안의 에러 로그를 조회하는 쿼리를 작성합니다.

kql

코드 복사

AppLogs

| where Timestamp > ago(1h)

| where Level == "Error"

| summarize count() by bin(Timestamp, 5m)

* + 1. **경고 및 알림 설정**:
       - 특정 조건이 충족될 때 경고를 트리거하도록 설정합니다.
       - 경고가 발생하면 이메일, SMS, 웹훅 등을 통해 알림을 받도록 설정합니다.

**3. Azure Automation을 통한 운영 효율성 향상**

* **설정**: Azure Automation은 반복적인 작업을 자동화하여 운영 효율성을 향상시키는 서비스입니다. 이를 통해 스크립트 실행, 업데이트 관리, 상태 구성 등을 자동화할 수 있습니다.
  + **구성 방법**:
    1. **Automation 계정 생성**:
       - Azure Portal에서 Automation 계정을 생성합니다.
       - 계정에 대한 리소스 그룹과 위치를 선택합니다.
    2. **Runbook 작성 및 실행**:
       - PowerShell 또는 Python을 사용하여 Runbook을 작성합니다.
       - 예를 들어, 주기적으로 VM 상태를 확인하고 필요한 작업을 수행하는 Runbook을 작성합니다.

powershell

코드 복사

$vmList = Get-AzVM

foreach ($vm in $vmList) {

if ($vm.Statuses[1].Code -eq 'PowerState/running') {

Write-Output "$($vm.Name) is running"

} else {

Write-Output "$($vm.Name) is not running"

}

}

* + 1. **업데이트 관리**:
       - Azure Automation Update Management를 사용하여 VM의 업데이트를 관리합니다.
       - 업데이트 배포 주기를 설정하고, 패치 적용 상태를 모니터링합니다.
    2. **Desired State Configuration (DSC)**:
       - Azure Automation DSC를 사용하여 VM의 구성 상태를 원하는 상태로 유지합니다.
       - 구성 드리프트를 감지하고, 자동으로 복구 작업을 수행합니다.

**4. Azure Application Insights를 통한 애플리케이션 성능 모니터링**

* **설정**: Application Insights는 애플리케이션 성능을 모니터링하고, 사용자의 행태를 분석하는 데 사용됩니다. 이를 통해 애플리케이션의 성능 문제를 신속하게 파악할 수 있습니다.
  + **구성 방법**:
    1. **Application Insights 리소스 생성**:
       - Azure Portal에서 Application Insights 리소스를 생성합니다.
       - 리소스 그룹과 위치를 선택하고, 애플리케이션 유형을 지정합니다.
    2. **SDK 통합**:
       - 애플리케이션 코드에 Application Insights SDK를 통합하여 성능 데이터를 수집합니다.
       - 예를 들어, ASP.NET 애플리케이션의 경우 Application Insights NuGet 패키지를 설치하고 초기화 코드를 추가합니다.

csharp

코드 복사

using Microsoft.ApplicationInsights;

var telemetryClient = new TelemetryClient();

telemetryClient.TrackEvent("PageView");

* + 1. **성능 메트릭 모니터링**:
       - 응답 시간, 처리량, 에러율 등의 성능 메트릭을 모니터링합니다.
       - 애플리케이션 맵을 사용하여 호출 간의 관계를 시각화하고, 성능 병목을 파악합니다.
    2. **분석 및 진단**:
       - 애플리케이션 성능 문제를 진단하고, 원인을 분석합니다.
       - 사용자 세션을 추적하여 사용자의 행태를 분석하고, 개선할 부분을 식별합니다.

**기대 효과**

* **종합적인 시스템 모니터링**: Azure Monitor, Log Analytics, Application Insights를 통해 시스템 및 애플리케이션의 종합적인 상태를 모니터링할 수 있습니다.
* **빠른 문제 감지 및 대응**: 실시간 경고와 알림 설정을 통해 문제 발생 시 신속하게 대응할 수 있습니다.
* **운영 효율성 향상**: Azure Automation을 통해 반복적인 작업을 자동화하여 운영 효율성을 높일 수 있습니다.
* **애플리케이션 성능 최적화**: Application Insights를 통해 애플리케이션 성능을 모니터링하고, 성능 문제를 신속하게 해결할 수 있습니다.

**고려 사항 및 제약 사항**

* **비용 관리**: 모니터링 서비스 사용에 따른 비용이 발생할 수 있습니다. 수집되는 데이터의 양과 저장 기간을 관리하여 비용을 최적화해야 합니다.
* **데이터 보안 및 프라이버시**: 수집된 로그와 메트릭 데이터의 보안을 철저히 관리해야 합니다. 민감한 데이터가 포함되지 않도록 주의해야 합니다.
* **성능 오버헤드**: 모니터링 에이전트와 SDK 통합으로 인해 약간의 성능 오버헤드가 발생할 수 있습니다. 성능 최적화를 위해 적절히 설정해야 합니다.
* **설정 복잡성**: 다양한 모니터링 및 자동화 도구의 설정이 복잡할 수 있으며, 초기 설정에 많은 시간이 소요될 수 있습니다.

**참조할 구성도**

* Azure 모니터링 및 관리 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Monitor 개요](https://docs.microsoft.com/en-us/azure/azure-monitor/overview)
  + [Azure Log Analytics 개요](https://docs.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview)
  + [Azure Automation 개요](https://docs.microsoft.com/en-us/azure/automation/automation-intro)
  + [Azure Application Insights 개요](https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 모니터링 및 관리를 효과적으로 수행하여 시스템 및 애플리케이션의 안정성을 보장할 수 있습니다.

# 아키텍처 요구 사항: 13.사용자 접속/인증 관리

**구현 방안**

**1. Azure Active Directory (Azure AD) B2C를 통한 사용자 인증 관리**

* **설정**: Azure Active Directory B2C (Business to Consumer)는 사용자 인증을 위한 확장 가능한 클라우드 솔루션으로, 다양한 인증 시나리오를 지원합니다.
  + **구성 방법**:
    1. **Azure AD B2C 테넌트 생성**:
       - Azure Portal에서 새로운 Azure AD B2C 테넌트를 생성합니다.
       - 테넌트 이름, 도메인 이름, 리소스 그룹을 지정합니다.
    2. **애플리케이션 등록**:
       - Azure AD B2C 테넌트 내에서 애플리케이션을 등록합니다.
       - Redirect URI 및 기타 애플리케이션 설정을 구성합니다.
    3. **사용자 흐름(User Flows) 생성**:
       - 사용자 등록, 로그인, 비밀번호 재설정 등의 사용자 흐름을 생성합니다.
       - 각 흐름에 대해 사용자 경험을 커스터마이즈하고, 인증 제공자(예: 이메일, 소셜 계정)를 설정합니다.
    4. **정책 설정**:
       - 사용자 인증 정책을 설정하여 다중 인증(MFA), 계정 잠금 등 보안 기능을 적용합니다.
    5. **클라이언트 애플리케이션 통합**:
       - 애플리케이션 코드에 Azure AD B2C를 통합하여 사용자를 인증합니다.
       - 예를 들어, ASP.NET Core 애플리케이션의 경우 Microsoft.Identity.Web 패키지를 사용하여 Azure AD B2C와 통합합니다.

csharp

코드 복사

services.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)

.AddMicrosoftIdentityWebApp(Configuration.GetSection("AzureAdB2C"));

**2. Azure AD를 통한 권한 관리 및 SSO**

* **설정**: Azure AD를 사용하여 권한 관리를 수행하고, Single Sign-On (SSO)을 통해 사용자 경험을 개선합니다.
  + **구성 방법**:
    1. **Azure AD 애플리케이션 등록**:
       - Azure Portal에서 Azure AD 애플리케이션을 등록합니다.
       - 애플리케이션 ID 및 클라이언트 시크릿을 생성합니다.
    2. **역할 기반 액세스 제어(RBAC) 설정**:
       - Azure AD에서 역할(Role)을 정의하고, 사용자 또는 그룹에 할당합니다.
       - 예를 들어, 관리자, 사용자, 읽기 전용 등의 역할을 정의합니다.
    3. **SSO 구성**:
       - Azure AD SSO를 구성하여 사용자가 하나의 로그인으로 여러 애플리케이션에 접근할 수 있도록 설정합니다.
       - SAML, OAuth, OpenID Connect 등의 프로토콜을 사용하여 SSO를 구성합니다.
    4. **애플리케이션 통합**:
       - 클라이언트 애플리케이션에 Azure AD 인증을 통합합니다.
       - 예를 들어, JavaScript 애플리케이션의 경우 MSAL.js 라이브러리를 사용하여 Azure AD와 통합합니다.

javascript

코드 복사

const msalConfig = {

auth: {

clientId: "your-client-id",

authority: "https://login.microsoftonline.com/your-tenant-id",

redirectUri: "http://localhost:3000"

}

};

const msalInstance = new msal.PublicClientApplication(msalConfig);

**3. 보안 강화 및 사용자 관리**

* **설정**: 보안 강화를 위해 다중 인증(MFA)을 적용하고, 사용자 및 그룹 관리를 효율적으로 수행합니다.
  + **구성 방법**:
    1. **다중 인증(MFA) 설정**:
       - Azure AD에서 MFA 정책을 설정합니다.
       - 사용자가 로그인할 때 추가 인증 단계를 요구하여 보안을 강화합니다.
    2. **조건부 액세스 정책**:
       - 조건부 액세스 정책을 설정하여 특정 조건(예: 위치, 디바이스 상태)에 따라 액세스를 제어합니다.
       - 예를 들어, 회사 네트워크 외부에서 로그인 시 MFA를 요구하도록 설정합니다.
    3. **사용자 및 그룹 관리**:
       - Azure AD에서 사용자 및 그룹을 생성하고 관리합니다.
       - 그룹 기반 접근 제어를 통해 관리 효율성을 높입니다.

**기대 효과**

* **강화된 보안**: Azure AD B2C와 Azure AD의 통합을 통해 강력한 인증 및 권한 관리 기능을 제공하며, MFA 및 조건부 액세스 정책을 통해 보안을 강화할 수 있습니다.
* **사용자 경험 개선**: SSO를 통해 사용자가 한 번의 로그인으로 여러 애플리케이션에 접근할 수 있어 사용자 경험이 향상됩니다.
* **효율적인 사용자 관리**: 사용자 및 그룹 관리를 중앙에서 효율적으로 수행하여 관리 복잡성을 줄일 수 있습니다.
* **확장성**: Azure AD B2C는 수백만 명의 사용자를 지원할 수 있는 확장성을 제공하여 비즈니스 성장에 따라 유연하게 대응할 수 있습니다.

**고려 사항 및 제약 사항**

* **초기 설정 복잡성**: Azure AD B2C와 Azure AD의 초기 설정이 복잡할 수 있으며, 전문가의 도움이 필요할 수 있습니다.
* **비용**: Azure AD B2C 및 Azure AD 서비스 사용에 따른 비용이 발생할 수 있습니다. 특히, MFA 및 조건부 액세스 정책 사용 시 추가 비용이 발생할 수 있습니다.
* **사용자 데이터 보호**: 사용자 인증 및 프로필 데이터의 보안을 철저히 관리해야 합니다. 데이터 암호화 및 접근 제어를 강화해야 합니다.
* **통합 문제**: 기존 시스템과의 통합 과정에서 호환성 문제나 추가 개발 작업이 필요할 수 있습니다.

**참조할 구성도**

* Azure 사용자 접속 및 인증 관리 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure AD B2C 개요](https://docs.microsoft.com/en-us/azure/active-directory-b2c/overview)
  + [Azure AD 애플리케이션 등록](https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app)
  + [Azure AD SSO 구성](https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/what-is-single-sign-on)
  + [Azure AD 조건부 액세스 개요](https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 사용자 접속 및 인증 관리를 효과적으로 수행하고, 보안을 강화할 수 있습니다.

# 아키텍처 요구 사항: 14.대량 배치 작업 관리

**구현 방안**

**1. Azure Batch를 통한 대량 배치 작업 관리**

* **설정**: Azure Batch는 대규모 병렬 및 고성능 컴퓨팅 애플리케이션을 관리하고 실행할 수 있는 완전 관리형 서비스입니다. 이를 통해 대량 배치 작업을 효율적으로 처리할 수 있습니다.
  + **구성 방법**:
    1. **Azure Batch 계정 생성**:
       - Azure Portal에서 Azure Batch 계정을 생성합니다.
       - 계정 이름, 리소스 그룹, 위치를 지정합니다.
    2. **풀(Pool) 생성**:
       - Azure Batch 계정 내에서 컴퓨팅 리소스 풀을 생성합니다.
       - 풀의 VM 크기, 노드 수, 이미지 유형을 설정합니다.
       - 자동 스케일링을 설정하여 작업 부하에 따라 풀의 크기를 동적으로 조절합니다.

json

코드 복사

{

"formula": "$TargetDedicatedNodes=2;",

"maxNodeCount": 10,

"minNodeCount": 1,

"nodeDeallocationOption": "TaskCompletion"

}

* + 1. **작업(Job) 및 작업(Task) 정의**:
       - 배치 작업을 정의하고, 이를 실행할 개별 작업을 설정합니다.
       - 작업 스크립트 및 애플리케이션을 컨테이너로 패키징하여 배포할 수 있습니다.
       - 작업이 실행될 스토리지 계정과 입력 파일을 연결합니다.

json

코드 복사

{

"id": "job1",

"poolInfo": {

"poolId": "mypool"

},

"task": [

{

"id": "task1",

"commandLine": "cmd /c myscript.bat"

}

]

}

* + 1. **작업 모니터링 및 관리**:
       - Azure Portal 또는 Azure Batch Explorer를 사용하여 작업의 진행 상태를 모니터링합니다.
       - 작업 로그 및 결과를 저장하고 분석합니다.

**2. 데이터 전처리 및 후처리**

* **설정**: Azure Data Factory를 사용하여 배치 작업의 데이터 전처리 및 후처리를 자동화합니다.
  + **구성 방법**:
    1. **Azure Data Factory 생성**:
       - Azure Portal에서 Data Factory 인스턴스를 생성합니다.
       - 리소스 그룹과 위치를 지정합니다.
    2. **파이프라인 생성**:
       - 데이터 이동 및 변환 작업을 자동화하기 위해 파이프라인을 생성합니다.
       - 배치 작업 전후에 데이터 전처리 및 후처리를 수행합니다.
    3. **트리거 설정**:
       - 스케줄 기반 또는 이벤트 기반 트리거를 설정하여 파이프라인을 자동으로 실행합니다.
    4. **연결된 서비스 설정**:
       - Azure Storage, Azure SQL Database 등 데이터 소스와 싱크를 설정하여 데이터 이동을 관리합니다.

**3. 작업 스케줄링 및 조율**

* **설정**: Azure Logic Apps 또는 Azure Functions를 사용하여 배치 작업의 스케줄링 및 조율을 자동화합니다.
  + **구성 방법**:
    1. **Azure Logic Apps 생성**:
       - Azure Portal에서 Logic App을 생성합니다.
       - Logic Apps 디자이너를 사용하여 워크플로를 작성합니다.
    2. **워크플로 정의**:
       - 배치 작업을 시작, 모니터링 및 종료하는 워크플로를 정의합니다.
       - 예를 들어, 배치 작업 완료 후 후속 작업을 수행하도록 설정합니다.
    3. **Azure Functions 통합**:
       - 필요한 경우 Azure Functions를 사용하여 맞춤형 작업을 수행합니다.
       - Logic Apps에서 Azure Functions를 호출하여 배치 작업의 조율을 강화합니다.

**기대 효과**

* **효율적인 대량 데이터 처리**: Azure Batch를 사용하여 대규모 배치 작업을 병렬로 처리하여 효율성을 높일 수 있습니다.
* **자동 스케일링**: 작업 부하에 따라 자동으로 리소스를 확장 및 축소하여 비용을 절감하고, 성능을 최적화할 수 있습니다.
* **데이터 이동 자동화**: Azure Data Factory를 통해 데이터 전처리 및 후처리를 자동화하여 전체 워크플로를 간소화할 수 있습니다.
* **작업 스케줄링 및 조율**: Azure Logic Apps와 Azure Functions를 사용하여 복잡한 배치 작업을 스케줄링하고 조율할 수 있습니다.

**고려 사항 및 제약 사항**

* **비용 관리**: 대규모 배치 작업 실행 시 발생하는 비용을 관리해야 합니다. 자동 스케일링 설정을 통해 비용 효율성을 극대화할 수 있습니다.
* **데이터 전송 속도**: 배치 작업 전후의 데이터 이동 속도가 중요한 고려 사항입니다. Azure Storage와 같은 고속 스토리지 솔루션을 사용하여 데이터 전송 속도를 최적화해야 합니다.
* **작업 실패 및 재시도**: 작업 실패 시 자동 재시도 메커니즘을 설정하여 작업의 신뢰성을 높여야 합니다.
* **보안 및 규정 준수**: 배치 작업 중 처리되는 데이터의 보안을 보장하고, 관련 규정을 준수해야 합니다. 데이터 암호화 및 접근 제어를 철저히 관리해야 합니다.

**참조할 구성도**

* Azure Batch를 사용한 대량 배치 작업 관리 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Batch 개요](https://docs.microsoft.com/en-us/azure/batch/batch-technical-overview)
  + [Azure Data Factory 개요](https://docs.microsoft.com/en-us/azure/data-factory/introduction)
  + [Azure Logic Apps 개요](https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview)
  + [Azure Functions 개요](https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 대량 배치 작업을 효율적으로 관리하고, 자동화된 데이터 처리와 조율을 통해 운영 효율성을 높일 수 있습니다.

# 아키텍처 요구 사항: 18.기존 인프라와의 연동

**구현 방안**

**1. Azure Hybrid Connections**

* **설정**: Azure Hybrid Connections는 온프레미스 네트워크와 Azure 앱 간의 연결을 쉽게 설정할 수 있는 서비스입니다. 이를 통해 클라우드 애플리케이션이 온프레미스 리소스에 접근할 수 있습니다.
  + **구성 방법**:
    1. **Hybrid Connection 생성**:
       - Azure Portal에서 App Service에 대한 Hybrid Connection을 생성합니다.
       - 온프레미스 서버와 연결할 엔드포인트 정보를 입력합니다.
    2. **Hybrid Connection Manager 설치**:
       - 온프레미스 서버에 Hybrid Connection Manager (HCM)를 설치합니다.
       - HCM을 설정하여 Azure와 연결을 구성합니다.
       - HCM에서 생성한 연결 문자열을 사용하여 Azure App Service와 온프레미스 리소스를 연결합니다.
    3. **네트워크 설정**:
       - 필요한 포트와 프로토콜이 온프레미스 방화벽에서 허용되도록 설정합니다.
       - 연결 테스트를 수행하여 온프레미스 리소스에 접근할 수 있는지 확인합니다.

**2. Azure Virtual Network (VNet) 및 VPN Gateway**

* **설정**: Azure VNet과 VPN Gateway를 사용하여 온프레미스 네트워크와 Azure 네트워크 간의 보안 연결을 설정합니다. 이를 통해 온프레미스와 클라우드 리소스 간의 통신을 안전하게 수행할 수 있습니다.
  + **구성 방법**:
    1. **Virtual Network 생성**:
       - Azure Portal에서 VNet을 생성하고, 서브넷을 설정합니다.
       - 필요한 주소 범위를 지정합니다.
    2. **VPN Gateway 구성**:
       - VNet 내에 VPN Gateway를 생성합니다.
       - VPN Gateway에 대한 공용 IP 주소를 할당합니다.
    3. **로컬 네트워크 게이트웨이 설정**:
       - 온프레미스 네트워크에 대한 정보를 가진 로컬 네트워크 게이트웨이를 생성합니다.
       - 온프레미스 VPN 디바이스와의 연결 정보를 입력합니다.
    4. **VPN 연결 설정**:
       - VNet의 VPN Gateway와 온프레미스 VPN 디바이스 간의 VPN 연결을 설정합니다.
       - IPsec/IKE 프로토콜을 사용하여 보안 연결을 구성합니다.
       - 연결 상태를 확인하고, 트래픽이 정상적으로 흐르는지 확인합니다.

**3. Azure ExpressRoute**

* **설정**: Azure ExpressRoute는 전용 네트워크 연결을 통해 온프레미스 데이터 센터와 Azure 간의 고속, 저지연 연결을 제공합니다.
  + **구성 방법**:
    1. **ExpressRoute 회로 생성**:
       - Azure Portal에서 ExpressRoute 회로를 생성합니다.
       - 필요한 대역폭과 피어링 위치를 선택합니다.
    2. **서비스 제공자와 연결 설정**:
       - 로컬 네트워크 서비스 제공자와 협력하여 물리적 연결을 설정합니다.
       - ExpressRoute 경로를 설정하여 온프레미스 네트워크와 Azure 간의 연결을 구성합니다.
    3. **피어링 설정**:
       - 프라이빗 피어링과 공용 피어링을 설정하여 다양한 Azure 서비스에 접근할 수 있도록 합니다.
       - BGP(Border Gateway Protocol)를 통해 경로를 교환하여 네트워크 트래픽을 최적화합니다.

**4. Azure Arc**

* **설정**: Azure Arc를 사용하여 온프레미스 및 멀티 클라우드 환경의 서버와 Kubernetes 클러스터를 Azure에서 관리할 수 있습니다.
  + **구성 방법**:
    1. **Azure Arc 활성화**:
       - Azure Portal에서 Azure Arc를 활성화합니다.
       - 온프레미스 서버 및 Kubernetes 클러스터를 Azure Arc에 등록합니다.
    2. **서버 및 클러스터 관리**:
       - Azure Arc를 통해 온프레미스 리소스의 모니터링, 정책 적용, 업데이트 관리를 수행합니다.
       - Azure Policy와 Azure Monitor를 사용하여 일관된 관리 및 보안 정책을 적용합니다.

**기대 효과**

* **통합된 관리**: Azure Hybrid Connections, VNet, VPN Gateway, ExpressRoute 및 Azure Arc를 통해 온프레미스와 클라우드 리소스를 통합 관리할 수 있습니다.
* **보안 강화**: VPN과 ExpressRoute를 통해 안전하고 신뢰할 수 있는 네트워크 연결을 제공하여 데이터 전송 보안을 강화합니다.
* **고성능 연결**: ExpressRoute를 통해 고속, 저지연의 네트워크 연결을 제공하여 대용량 데이터 전송 시 성능을 최적화할 수 있습니다.
* **유연한 확장성**: Azure Arc를 통해 온프레미스와 클라우드 리소스를 유연하게 확장하고, 일관된 관리 정책을 적용할 수 있습니다.

**고려 사항 및 제약 사항**

* **초기 설정 복잡성**: 다양한 연결 옵션의 초기 설정이 복잡할 수 있으며, 네트워크 및 보안 전문가의 도움이 필요할 수 있습니다.
* **비용 관리**: ExpressRoute 및 VPN Gateway 사용에 따른 비용이 발생할 수 있습니다. 비용 효율성을 고려하여 적절한 연결 옵션을 선택해야 합니다.
* **네트워크 대역폭 및 성능**: 온프레미스와 클라우드 간의 데이터 전송 속도와 대역폭 요구사항을 충분히 고려해야 합니다.
* **보안 설정**: 온프레미스와 클라우드 간의 보안 설정을 철저히 관리하여 데이터 무결성과 기밀성을 유지해야 합니다.

**참조할 구성도**

* Azure 인프라 연동 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Hybrid Connections 개요](https://docs.microsoft.com/en-us/azure/app-service/app-service-hybrid-connections)
  + [Azure Virtual Network (VNet) 개요](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview)
  + [Azure VPN Gateway 개요](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways)
  + [Azure ExpressRoute 개요](https://docs.microsoft.com/en-us/azure/expressroute/expressroute-introduction)
  + [Azure Arc 개요](https://docs.microsoft.com/en-us/azure/azure-arc/overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 기존 온프레미스 인프라와 Azure 클라우드 간의 원활한 연동을 실현할 수 있습니다.

# 아키텍처 요구 사항: 17.네트워크 보안 및 제어

**구현 방안**

**1. Azure Firewall을 통한 중앙 집중식 네트워크 보안**

* **설정**: Azure Firewall은 중앙에서 관리되는 네트워크 보안 서비스로, 애플리케이션 및 네트워크 규칙을 통해 트래픽을 제어하고 보호합니다.
  + **구성 방법**:
    1. **Azure Firewall 배포**:
       - Azure Portal에서 Azure Firewall을 생성합니다.
       - 가상 네트워크와 서브넷을 지정하여 배포합니다.
    2. **애플리케이션 규칙 구성**:
       - Azure Firewall에서 애플리케이션 규칙을 정의하여 특정 애플리케이션 트래픽을 제어합니다.
       - 예를 들어, 특정 도메인이나 URL에 대한 액세스를 허용 또는 차단합니다.
    3. **네트워크 규칙 구성**:
       - 네트워크 규칙을 설정하여 IP 주소, 포트, 프로토콜 기반으로 트래픽을 제어합니다.
       - 인바운드 및 아웃바운드 트래픽에 대한 세부적인 제어를 설정합니다.
    4. **DNS 프록시 및 필터링**:
       - DNS 프록시 기능을 활성화하여 DNS 요청을 필터링하고 보호합니다.
       - 악성 도메인에 대한 접근을 차단하여 보안을 강화합니다.
    5. **로깅 및 모니터링**:
       - Azure Monitor와 통합하여 방화벽 로그를 수집하고 분석합니다.
       - 트래픽 및 보안 이벤트를 모니터링하여 이상 징후를 빠르게 감지합니다.

**2. Azure DDoS Protection을 통한 분산 서비스 거부 공격 방어**

* **설정**: Azure DDoS Protection은 Azure Virtual Network 리소스를 분산 서비스 거부(Distributed Denial of Service, DDoS) 공격으로부터 보호하는 서비스입니다.
  + **구성 방법**:
    1. **DDoS Protection 계획 생성**:
       - Azure Portal에서 DDoS Protection 계획을 생성합니다.
       - 보호할 VNet을 선택하고 계획을 적용합니다.
    2. **자동 방어 설정**:
       - DDoS Protection Standard를 활성화하여 자동으로 DDoS 공격을 방어합니다.
       - DDoS Protection Metrics를 모니터링하여 공격 탐지 및 방어 상태를 확인합니다.
    3. **알림 구성**:
       - DDoS 공격이 감지될 경우 이메일, SMS, 웹훅 등을 통해 알림을 받도록 설정합니다.
       - Azure Monitor와 통합하여 경고 및 알림을 설정합니다.

**3. 네트워크 보안 그룹(NSG)을 통한 서브넷 및 NIC 보안 제어**

* **설정**: 네트워크 보안 그룹(NSG)는 Azure Virtual Network의 서브넷 및 네트워크 인터페이스(NIC)에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 규칙을 설정합니다.
  + **구성 방법**:
    1. **NSG 생성**:
       - Azure Portal에서 NSG를 생성합니다.
       - NSG를 서브넷 또는 NIC에 연결합니다.
    2. **인바운드 및 아웃바운드 규칙 설정**:
       - 인바운드 규칙을 설정하여 허용 또는 차단할 트래픽을 정의합니다.
       - 아웃바운드 규칙을 설정하여 허용 또는 차단할 트래픽을 정의합니다.
    3. **우선순위 설정**:
       - 규칙의 우선순위를 설정하여 적용 순서를 정의합니다.
       - 기본적으로 낮은 숫자가 높은 우선순위를 가집니다.
    4. **로깅 및 모니터링**:
       - NSG 흐름 로그를 활성화하여 트래픽을 모니터링합니다.
       - Azure Network Watcher와 통합하여 네트워크 흐름을 분석합니다.

**4. Azure Application Gateway 및 Web Application Firewall(WAF)**

* **설정**: Azure Application Gateway는 웹 애플리케이션의 트래픽을 관리하고 보호하는 서비스로, WAF 기능을 통해 웹 애플리케이션을 보호합니다.
  + **구성 방법**:
    1. **Application Gateway 생성**:
       - Azure Portal에서 Application Gateway를 생성합니다.
       - 프론트엔드 IP, 백엔드 풀, 리스너 등을 설정합니다.
    2. **WAF 정책 설정**:
       - WAF 정책을 생성하여 OWASP 규칙 집합을 적용합니다.
       - 사용자 정의 규칙을 추가하여 특정 공격 벡터를 차단합니다.
    3. **SSL/TLS 종단 설정**:
       - SSL/TLS 인증서를 구성하여 웹 애플리케이션 트래픽을 암호화합니다.
       - SSL/TLS 종단을 설정하여 암호화된 트래픽을 처리합니다.
    4. **로깅 및 모니터링**:
       - Application Gateway 로그를 활성화하여 트래픽 및 보안 이벤트를 모니터링합니다.
       - Azure Monitor와 통합하여 로그 데이터를 분석합니다.

**기대 효과**

* **강화된 보안**: Azure Firewall, DDoS Protection, NSG, Application Gateway, WAF를 통해 네트워크 보안을 강화하고, 다양한 공격 벡터로부터 리소스를 보호할 수 있습니다.
* **중앙 집중식 관리**: 중앙에서 네트워크 보안 정책을 관리하여 일관된 보안 상태를 유지할 수 있습니다.
* **실시간 모니터링**: 보안 이벤트와 트래픽을 실시간으로 모니터링하여 빠르게 대응할 수 있습니다.
* **규정 준수**: 보안 정책과 로그를 통해 규정 준수 요구 사항을 충족할 수 있습니다.

**고려 사항 및 제약 사항**

* **설정 복잡성**: 다양한 보안 서비스와 기능을 설정하고 관리하는 것이 복잡할 수 있으며, 네트워크 및 보안 전문가의 도움이 필요할 수 있습니다.
* **비용**: Azure Firewall, DDoS Protection, Application Gateway 등의 사용에 따른 비용이 발생할 수 있습니다. 비용을 주기적으로 검토하고 최적화해야 합니다.
* **성능 오버헤드**: 보안 서비스 적용으로 인한 성능 오버헤드가 발생할 수 있습니다. 성능을 최적화하기 위해 적절한 설정과 모니터링이 필요합니다.
* **정책 관리**: 여러 보안 정책을 일관되게 관리하고 업데이트하는 것이 중요합니다. 정책 변경 시 전체 네트워크에 미치는 영향을 고려해야 합니다.

**참조할 구성도**

* Azure 네트워크 보안 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Firewall 개요](https://docs.microsoft.com/en-us/azure/firewall/overview)
  + [Azure DDoS Protection 개요](https://docs.microsoft.com/en-us/azure/ddos-protection/ddos-protection-overview)
  + [네트워크 보안 그룹(NSG) 개요](https://docs.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview)
  + [Azure Application Gateway 개요](https://docs.microsoft.com/en-us/azure/application-gateway/overview)
  + [Azure Web Application Firewall(WAF) 개요](https://docs.microsoft.com/en-us/azure/web-application-firewall/overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 네트워크 보안과 제어를 효과적으로 관리하고 강화할 수 있습니다.

# 아키텍처 요구 사항: 15.글로벌 트래픽 관리

**구현 방안**

**1. Azure Traffic Manager를 통한 글로벌 트래픽 관리**

* **설정**: Azure Traffic Manager는 DNS 기반 트래픽 부하 분산 서비스로, 사용자 요청을 최적의 엔드포인트로 라우팅하여 전 세계적으로 고가용성과 빠른 응답 시간을 제공합니다.
  + **구성 방법**:
    1. **Traffic Manager 프로필 생성**:
       - Azure Portal에서 Traffic Manager 프로필을 생성합니다.
       - 프로필 유형을 선택합니다. (예: 성능, 가중치, 우선 순위)
    2. **엔드포인트 구성**:
       - 각 엔드포인트를 정의하고, Azure 리소스(예: App Service, VM, Public IP) 또는 외부 리소스를 엔드포인트로 추가합니다.
       - 엔드포인트 상태 모니터링을 설정하여 가용성을 확인합니다.
    3. **라우팅 방법 설정**:
       - 성능 라우팅: 사용자 위치에 따라 최적의 엔드포인트로 라우팅합니다.
       - 가중치 라우팅: 각 엔드포인트에 가중치를 설정하여 트래픽을 분산합니다.
       - 우선 순위 라우팅: 기본 엔드포인트를 설정하고, 장애 시 백업 엔드포인트로 라우팅합니다.
    4. **DNS 구성**:
       - Traffic Manager 프로필의 DNS 이름을 설정하고, 도메인 이름 시스템(DNS) 설정을 통해 사용자의 도메인 이름을 Traffic Manager 프로필로 지정합니다.
    5. **모니터링 및 알림 설정**:
       - Azure Monitor와 통합하여 Traffic Manager 프로필의 상태를 모니터링합니다.
       - 엔드포인트 상태 변화 시 알림을 설정하여 신속하게 대응합니다.

**2. Azure Front Door를 통한 글로벌 애플리케이션 가속 및 보안**

* **설정**: Azure Front Door는 글로벌 웹 애플리케이션에 대한 빠르고 안전한 액세스를 제공하는 서비스로, 애플리케이션 가속, 부하 분산, SSL 오프로드, 웹 애플리케이션 방화벽(WAF) 기능을 제공합니다.
  + **구성 방법**:
    1. **Front Door 프로필 생성**:
       - Azure Portal에서 Front Door 프로필을 생성합니다.
       - Front Door에 대한 프론트엔드 호스트, 백엔드 풀, 라우팅 규칙 등을 설정합니다.
    2. **프론트엔드 호스트 설정**:
       - 사용자 도메인 이름을 Front Door 프로필의 프론트엔드 호스트로 추가합니다.
       - SSL 인증서를 설정하여 HTTPS 트래픽을 처리합니다.
    3. **백엔드 풀 구성**:
       - 애플리케이션 백엔드를 정의하고, 각 백엔드에 대한 상태 프로브를 설정합니다.
       - 백엔드 간 부하 분산을 위한 라우팅 규칙을 구성합니다.
    4. **라우팅 규칙 설정**:
       - 트래픽 라우팅 규칙을 정의하여 요청을 적절한 백엔드로 라우팅합니다.
       - 경로 기반 라우팅, 대기 시간 기반 라우팅 등을 설정합니다.
    5. **WAF 설정**:
       - Front Door의 WAF 기능을 활성화하여 애플리케이션을 웹 공격으로부터 보호합니다.
       - OWASP 규칙 집합을 적용하고, 사용자 정의 규칙을 추가합니다.
    6. **모니터링 및 로깅**:
       - Azure Monitor와 통합하여 Front Door의 성능과 보안을 모니터링합니다.
       - 요청 로그와 진단 로그를 분석하여 성능 최적화를 수행합니다.

**3. Content Delivery Network (CDN) 통합**

* **설정**: Azure CDN은 전 세계적으로 분산된 서버를 통해 콘텐츠를 캐싱하고, 사용자에게 빠른 응답을 제공하는 서비스입니다.
  + **구성 방법**:
    1. **CDN 프로필 및 엔드포인트 생성**:
       - Azure Portal에서 CDN 프로필을 생성합니다.
       - CDN 엔드포인트를 생성하고, 캐싱할 콘텐츠의 원본을 설정합니다.
    2. **캐싱 규칙 설정**:
       - 콘텐츠의 캐싱 동작을 제어하기 위해 캐싱 규칙을 설정합니다.
       - TTL(Time-To-Live) 설정, 캐시 만료 규칙 등을 정의합니다.
    3. **HTTPS 설정**:
       - CDN 엔드포인트에 HTTPS를 설정하여 안전한 콘텐츠 전송을 보장합니다.
       - SSL/TLS 인증서를 설정합니다.
    4. **CDN 도메인 설정**:
       - 사용자 도메인 이름을 CDN 엔드포인트에 매핑합니다.
       - DNS 설정을 통해 CDN 엔드포인트로 트래픽을 라우팅합니다.
    5. **모니터링 및 분석**:
       - CDN 성능과 사용량을 모니터링하여 최적화합니다.
       - CDN 로그를 분석하여 캐싱 효율성을 평가합니다.

**기대 효과**

* **글로벌 사용자 경험 향상**: Azure Traffic Manager와 Front Door를 사용하여 사용자 위치에 따라 최적의 엔드포인트로 라우팅함으로써 응답 시간을 단축하고 사용자 경험을 향상시킬 수 있습니다.
* **고가용성 보장**: 장애 발생 시 Traffic Manager와 Front Door의 자동 페일오버 기능을 통해 서비스 가용성을 보장합니다.
* **보안 강화**: Azure Front Door의 WAF 기능과 Azure CDN의 HTTPS 설정을 통해 웹 애플리케이션 보안을 강화합니다.
* **콘텐츠 전송 최적화**: Azure CDN을 통해 전 세계 사용자에게 빠른 콘텐츠 전송을 제공하여 성능을 최적화합니다.

**고려 사항 및 제약 사항**

* **비용 관리**: Traffic Manager, Front Door, CDN 사용에 따른 비용이 발생할 수 있습니다. 비용 효율성을 주기적으로 검토하고 최적화해야 합니다.
* **설정 복잡성**: 다양한 서비스와 설정을 통합하고 관리하는 것이 복잡할 수 있으며, 전문가의 도움이 필요할 수 있습니다.
* **성능 모니터링**: 글로벌 트래픽 관리의 성능을 지속적으로 모니터링하고 최적화해야 합니다. 특히, 트래픽 패턴 변화에 따른 성능 영향을 평가해야 합니다.
* **규정 준수**: 글로벌 데이터 전송과 관련된 규제 및 데이터 보호 법규를 준수해야 합니다. 각 지역별 데이터 보호 규정을 확인하고 준수해야 합니다.

**참조할 구성도**

* Azure 글로벌 트래픽 관리 아키텍처에 대한 Microsoft의 공식 문서와 구성도를 참조할 수 있습니다:
  + [Azure Traffic Manager 개요](https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview)
  + [Azure Front Door 개요](https://docs.microsoft.com/en-us/azure/frontdoor/front-door-overview)
  + [Azure CDN 개요](https://docs.microsoft.com/en-us/azure/cdn/cdn-overview)

이 구현 방안을 통해 제조업 MES 공장자동화 시스템의 글로벌 트래픽을 효과적으로 관리하고, 성능과 보안을 최적화할 수 있습니다.