

TCT 시험 안내

(시스템&솔루션개발)

전사기술체계혁신 TF

Cloud와 On-Premise 환경에서 요구되는 솔루션의 최적화된 아키텍처를 설계하고 핵심 메커니즘을 유연하게 구현하기 위해 필요한 요소기술(통신, 데이터, 알고리즘, 프로세싱 등)을 분석/설계/개발할 수 있는 역량 측정

■ 기본사항

- ◆ 응시형태 : 집합응시
- ◆ 문항구성 : 필기형(객관식/단답식) 7문항 / 실기형(코딩) 1문항(소문항 5)
- ◆ 시험시간 : 필기+실기 250분 (단, 필기 시간은 20분으로 제한함)
- ◆ 인터넷 검색 : 가능
- ◆ 참고자료 지참 : 가능

■ 출제범위

소분류	대모듈	소모듈	문항수	배점
DX 기술 (이론)	Cloud	Cloud 기본	3	20
		Computing, Containers, Storage, Networking & Content Delivery, Security, Identity, & Compliance, Management Service		
		Cloud Architecture		
	AI	인공지능(AI)의 기초적인 개념	4	
		AI 활용 기술		
시스템&솔루션 개발 (실기)	언어	Java, C# (C는 주스킬 선택자가 소수일 경우, 변경 가능)	1 (소문항 5)	80
	아키텍처 설계	Software Architecture		
	기반 프레임워크	Base Framework		
	통신	Network programing		
	데이터	Basic Data Structure, File IO		
	알고리즘	Basic Algorithm, Encryption/Decryption		
	프로세싱	Thread, Async, Polling		
	DX 요소기술	API Gateway, Service Mesh, RealTime Streaming , AI 등 주요 기술		
합계				100

※ 모듈별 문항수와 배점은 차수에 따라 달라질 수 있습니다.

※ 실기시험에서 DX 요소기술을 솔루션화하거나 활용하는 시나리오로 출제가 됩니다.

- 구현 프로그램에서 활용할 수 있는 Mock 을 제공하며 개발 화면 환경은 '21 년과 동일
- 필요한 외부 라이브러리는 다운로드 후 활용 가능(C 언어 경우 사전 제공 라이브러리를 별도 공지함)

■ 주요 변경사항

- ♦ 시험 출제 및 응시 환경에 대해서는 21년과 동일한 방식을 유지하여 변화에 대한 스트레스를 최소화하는 방향으로 진행
- ♦ 실기형 문항 C 언어 선택은 '22년까지 유지하고, '23년부터는 폐지함

■ 학습자료

- ♦ Cloud 필기테스트 출제범위
 - TECH Wiki > Cloud > Cloud 서비스
 - TECH Wiki > Cloud > Cloud 서비스 > Private Cloud > Cloud 플랫폼 > Kubernetes 기반 솔루션 가이드
 - TECH Wiki > Cloud > Cloud 서비스 > Private Cloud > Cloud 플랫폼 > Serverless Computing 활용가이드
 - AWS Document(<https://docs.aws.amazon.com>)
 - GCP Document(<https://docs.microsoft.com/en-us/learn/>, <https://www.microsoft.com/handsonlabs>)
 - Azure Document(<https://cloud.google.com/docs/>)
 - * 베타 서비스 제외

- ♦ 필기테스트 상세 출제범위 (TECH Wiki > 아키텍처 > 하위 내용, CSP 3사 Document > 하위 내용)

[Cloud 서비스 상세 출제범위]

- > Cloud 서비스 > Amazon Web Service > 클라우드 개발 가이드
- > Cloud 서비스 > Amazon Web Service > 클라우드 아키텍처 가이드
- > Cloud 서비스 > Azure > 클라우드 개발 가이드(Azure)
- > Cloud 서비스 > Azure > 클라우드 아키텍처 가이드(Azure)
- > Cloud 서비스 > Google Cloud Platform > GCP 이해하기 - 1. GCP 개요

[Kubernetes 기반 솔루션 가이드 상세 출제범위]

- > Kubernetes 기반 솔루션 가이드 > AKS 가이드 (Azure) > AKS 아키텍처 가이드
- > Kubernetes 기반 솔루션 가이드 > EKS 가이드 (AWS) > EKS 아키텍처 가이드
- > Kubernetes 기반 솔루션 가이드 > GKE 가이드 (Google) > GKE 아키텍처 가이드

[Serverless Computing 활용가이드 상세 출제범위]

- > Serverless Computing 활용가이드 > 2. 서버리스 컴퓨팅이란 무엇인가?
- > Serverless Computing 활용가이드 > 3. 서버리스 컴퓨팅 특징
- > Serverless Computing 활용가이드 > 4. 서버리스 컴퓨팅 도입시 고려사항
- > Serverless Computing 활용가이드 > 6. 서버리스 컴퓨팅 아키텍처 설계

[AWS Document 상세 출제범위]

- › Compute > EC2, EC2 Image Builder, Lambda
- › Containers > ECR, ECS, EKS, App2Container
- › Networking & Content Delivery > API Gateway, VPC, Route3
- › Storage > S3, EBS, EFS, S3 Glacier
- › Security, Identity, & Compliance > IAM
- › Management & Governance > CloudFormation, CloudWatch, AppConfig, Auto Scaling,

[GCP Document 상세 출제범위]

- › 컴퓨팅 > Compute Engine, 선점형 VM, 보안 VM,
- › 컨테이너 > GKE, Container Registry,
- › 네트워킹 > VPC, Cloud Router, Cloud NAT, Cloud Load Balancing
- › 보안 및 ID > IAM, Resource Manager, 보안 VM, VPC 서비스 제어, Cloud Key Management Service
- › 서버리스 컴퓨팅 > Cloud Functions, Cloud Run
- › 관리도구 > Cloud Console, Cloud Shell, 비용 관리, Cloud APIs

[Azure Document 상세 출제범위]

- › Azure Fundamentals : Describe core Azure concepts
- › Azure Fundamentals : Describe core Azure services
- › Azure Fundamentals : Describe core solutions and management tools on Azure
- › Azure Fundamentals : Describe general security and network security features
- › Azure Fundamentals : Describe identity, governance, privacy, and compliance features
- › Azure Fundamentals : Describe Azure cost management and service level agreements

- ♦ AI 기초적인 개념 및 활용 기술
 - TECH Wiki > AI빅데이터 > AI > [AI Tech Letter] > 1 ~ 12편

■ 실습환경

- ♦ 응시 환경에서 HTTP 통신 시나리오가 활용되는 경우 아래 Library 를 포함하여 제공할 예정이며, 그 외 Library 를 선택하여 Download 하여 사용도 가능합니다.

[Java]

- Http Server : Jetty 9 Embedded
(<https://www.eclipse.org/jetty/documentation/jetty-9/index.html#jetty-helloworld>)
- Http Client : Jetty 9 HttpClient
(<https://www.eclipse.org/jetty/documentation/jetty-9/index.html#http-client>)
- Json : Google Gson 2.8.6
(<https://github.com/google/gson>)

[C#]

- Http Server
(<https://docs.microsoft.com/ko-kr/dotnet/api/system.net.http.listener>)
- Http Client
(<https://docs.microsoft.com/ko-kr/dotnet/api/system.net.http.httpclient>)
- Json : Newtonsoft.Json
(<https://www.newtonsoft.com/json>)

[C]

- Http Server : libmicrohttpd
(<https://www.gnu.org/software/libmicrohttpd/>)
- Http Client : libcurl
(<https://curl.se/libcurl/>)
- Json : json-c
(<https://github.com/json-c/json-c>)