

DP-203: Microsoft Azure Data Engineer Associate Exam Study Guide

April 20, 2022 by [manish](#)

★★★★★ 4.6/5 - (22 votes)

The DP 203 Certification is the next level after the DP-200 Designing an Azure Data Solution and Implementing an Azure Data Solution certifications. The article covers everything you should know before applying for DP-203 certification.

The topics covered in this blog are:

- [DP-203 Certification Overview](#)
- [Who is Azure Data Engineer?](#)
- [Why You Should Learn Data Engineer?](#)
- [Who This Certification is for?](#)
- [Benefits of DP-203 Certification](#)

- [DP-203 Exam Details](#)
- [DP-203 Exam Skills Measured](#)
- [How to Register for Azure DP 203 Exam](#)
- [Pre-requisites for DP-203 Certification](#)
- [DP 203 Study Guide](#)
- [DP-203 Exam Retake Policy](#)
- [DP 203 Exam Day Tips](#)
- [Conclusion](#)

DP-203 Certification Overview

The DP-203 is an advanced-level certification from Microsoft Azure for Data Engineer. After getting DP-203 certification, candidates get the credibility and validation for Azure Data Engineer skills such as Designing, implementing, processing, monitoring, optimizing data storage and security of Data.

With Azure Data Engineers, you can ensure that data pipelines and data stores are optimized for performance and reliability, based on your business needs and constraints. You will be able to design, implement, and monitor data platforms to meet the needs of data pipelines. Hence, once you understand and master this certification you can easily become a successful Azure Data Engineer.



Azure Data Engineer
Associate



Are you new to the Azure cloud? Do check out our blog post on the [Microsoft Azure Certification Path 2022](#) and choose the best certification for you.

Who is Azure Data Engineer?

Azure Data Engineers are responsible for clearly separating and differentiating raw data into structured data. They integrate, transform and consolidate data from unstructured data into structured data. And this structured data is used for building analytics solutions.

Moreover, they help the stakeholders to clearly understand the information through exploration. They construct efficiently and design particular instructions for processing pipelines with the help of specific tools and strategies. To produce better data for evaluation, they use [Azure](#) data offerings and languages.

Following are the responsibilities of an Azure Data Engineer:

- Developing and designing data processing and storage solutions for enterprises.
- Install, configure, and manage cloud-based data services, such as databases, blob services, and analytics.
- Securing the stored data and the platform, so that only necessary users have the access to the data.
- Monitoring the systems to make sure it is running properly and are cost-effective.

Why You Should Learn Data Engineer?

Data engineers are the people who understand and connect the raw data into structured data within a company. They accomplish this by doing,

- Accessing, collecting, auditing, and cleaning data from systems and converting it into useable data for enterprises.
- Maintaining the database
- Building pipelines
- Monitor and manage data systems
- Creating data scientists' output in a scalable manner.

So, they are the front lines of data strategy, the first people to tackle the unstructured data and convert it into structured one. They are people on whose shoulders data analysts and data scientists stand.

For this reason, there is a huge demand for Data engineers in the IT sector, as they daily produce raw data.

As for some resources, the average salary for a Data Engineer is \$116,591. It is also known that they make 261% more than the national average salary.

Who This Certification is For?

DP-203 certification is the ideal for the candidate,

- Who is interested in Data Engineering.
- For the professionals of Data architects, Data Administration and Business Intelligence.
- Candidates who know SQL, Python, Scala, or other data processing languages.
- Candidates who are good at parallel processing and data architecture patterns.
- Data Engineers who can transform and consolidate unstructured data into structured data.

Benefits of DP-203 Certification

- There is a huge demand for Data Engineers. In addition, Microsoft certification is globally recognized.
- After being DP-203 certified, 26% reported job promotions and 35% of technical professionals reported that certification led to salary or wage increments.
- DP-203 certification leads to rampant gain in jobs and earnings.
- The CV with Microsoft certification advances your job profile and increases the chances of getting chosen.

Check Out: [ADF Interview Questions](#)

DP-203 Exam Details

Exam Name DP-203: Data Engineering on Microsoft Azure	Exam Duration 180 Minutes
Exam Type Multiple Choice Examination	Number of Questions 40 - 60 Questions
Exam Fee \$165	Eligibility/Pre-Requisite None
Exam validity 1 years	Exam Languages English, Japanese, Korean, and Simplified Chinese

DP-203 Exam Skills Measured

Design and implement data storage	40–45%
Design and develop data processing	25–30%
Optimize data storage and data processing	10-15%
Design and implement data security	10-15%

How to Register for Azure DP 203 Exam

You can register for the Microsoft Azure Data Engineer Exam (DP-203) by going to the [Official Microsoft Page](#).

Exam DP-203: Data Engineering on Microsoft Azure

Languages: English, Chinese (Simplified), Japanese, Korean, German, French, Spanish, Portuguese (Brazil), Arabic (Saudi Arabia), Russian, Chinese (Traditional), Italian, Indonesian (Indonesia)

Retirement date: none

This exam measures your ability to accomplish the following technical tasks: design and implement data storage; design and develop data processing; design and implement data security; and monitor and optimize data storage and data processing.

United States ▼

\$165 USD*

Price based on the country in which the exam is proctored.

[Schedule exam >](#)

Official practice test for Data Engineering on Microsoft Azure

All objectives of the exam are covered in depth so you'll be ready for any question on the exam.

Pre-requisites for DP-203 Certification

For this exam, candidates must have a decent knowledge of Data Processing Languages such as:

- SQL
- Python
- Scala

DP 203 Study Guide

Design and Implement Data Storage (40-45%)

Design a data storage structure

- Design an Azure Data Lake solution
 - [Introduction to Azure Data Lake Storage Gen2](#)

- [Building your Data Lake on Azure Data Lake Storage gen2](#)
- Recommend file types for storage
 - [Introduction to Azure Storage](#)
- Recommend file types for analytical queries
 - [Query data in Azure Data Lake using Azure Data Explorer](#)
 - [Query Azure Storage analytics logs in Azure Log Analytics](#)
- Design for efficient querying
 - [Design for querying](#)
 - [Guidelines for table design](#)
- Design for data pruning
 - [Dynamic file pruning](#)
- Design a folder structure that represents the levels of data transformation
 - [Copy and transform data in Data Lake Storage using Data Factory](#)
- Design a distribution strategy
 - [How to choose Right data distribution strategy for Azure Synapse?](#)
 - [Guidance for designing distributed tables in Azure Synapse](#)
- Design a data archiving solution
 - [Designing a Data Archiving Strategy on Microsoft Azure](#)
 - [Archive on-premises data to the cloud](#)

Design a partition strategy

- Design a partition strategy for files
 - [File Partition using Azure Data Factory](#)
 - [Incrementally copy new files by using the Copy Data tool](#)
- Design a partition strategy for analytical workloads
 - [Best practices: Delta Lake](#)
 - [Partitions in tabular models](#)
 - [Automated Partition Management with Azure Analysis Services](#)
- Design a partition strategy for efficiency/performance
 - [Designing partitions for query performance](#)
- Design a partition strategy for Azure Synapse Analytics

- [Partitioning tables in dedicated SQL pool](#)
- Identify when partitioning is needed in Azure Data Lake Storage Gen2
 - [ADLS Gen2](#)

Design the serving layer

- Design star schemas
 - [Star schema overview](#)
 - [Designing Star Schema](#)
- Design slowly changing dimensions
 - [Design a Slowly Changing Dimension \(SCD\) in Azure Data Factory](#)
- Design a dimensional hierarchy
 - [Hierarchies in tabular models](#)
- Design a solution for temporal data
 - [What is temporal data?](#)
 - [Getting started with temporal tables in Azure SQL Database](#)
- Design for incremental loading
 - [Incrementally load data from a source data store to a destination datastore](#)
 - [Incrementally load data from Azure SQL Database to Azure Blob storage](#)
- Design analytical stores
 - [Choose an analytical data store in Azure](#)
 - [What is Azure Cosmos DB analytical store?](#)
- Design meta stores in Azure Synapse Analytics and Azure Databricks
 - [Azure Synapse Analytics shared metadata tables](#)
 - [Apache Hive Metastore for Databricks](#)

Implement physical data storage structures

- Implement compression
 - [Data compression](#)
 - [Data compression on SQL Azure DB](#)

- Implement partitioning
 - [Data partitioning strategies](#)
 - [How to partition your data in Azure Cosmos DB](#)
- Implement sharding
 - [Sharding pattern](#)
 - [Adding a shard using Elastic Database tools](#)
- Implement different table geometries with Azure Synapse Analytics pools
 - [Spatial Types – geometry \(Transact-SQL\)](#)
 - [Table data types for dedicated SQL pool](#)
- Implement data redundancy
 - [Azure Storage redundancy](#)
 - [Change how a storage account is replicated](#)
- Implement distributions
 - [Distributions in Azure Synapse Analytics](#)
 - [Examples for table distribution](#)
- Implement data archiving
 - [Archive on-premises data to the cloud](#)
 - [Blob rehydration from the Archive tier](#)

Implement logical data structures

- Build a temporal data solution
 - [Creating a system-versioned temporal table](#)
- Build a slowly changing dimension
 - [Azure Data Factory Data Flow: Building Slowly Changing Dimensions](#)
 - [How to implement Slow changing Dimension Type 1 in ADF?](#)
 - [Slowly Changing Dim Type 2 with ADF Mapping Data Flows](#)
- Build a logical folder structure
 - [Creating an Azure Blob Hierarchy](#)
 - [Modeling a Directory Structure on Azure Blob Storage](#)
- Build external tables

- [Use external tables with Synapse SQL](#)
- [Create and alter Azure Storage external tables](#)
- Implement file and folder structures for efficient querying and data pruning
 - [Query multiple files or folders](#)
 - [Query folders and multiple files](#)

Implement the serving layer

- Deliver data in a relational star schema
 - [Data models within Azure Analysis Services](#)
- Deliver data in Parquet files
 - [Parquet file](#)
 - [Parquet format in Azure Data Factory](#)
 - [Parquet format in Azure Data Lake Analytics](#)
- Maintain metadata
 - [Preserve metadata using copy activity in Azure ADF](#)
- Implement a dimensional hierarchy
 - [Create and manage hierarchies](#)

Design and Develop Data Processing (25-30%)

Ingest and transform data

- Transform data by using Apache Spark
 - [Transform data in the cloud by using a Spark activity in ADF](#)
 - [Transform data using Spark activity in Azure Data Factory](#)
- Transform data by using Transact-SQL
 - [Apply SQL Transformation](#)
- Transform data by using Data Factory
 - [Transform data in Azure ADF](#)
 - [Transform data using mapping data flows](#)
- Transform data by using Azure Synapse Pipelines

- [Use Azure Synapse Analytics to create a pipeline for data transformation](#)
- Transform data by using Stream Analytics
 - [Transform data by using Azure Stream Analytics](#)
- Cleanse data
 - [Data Cleansing](#)
 - [Clean Missing Data component](#)
- Split data
 - [Split Data](#)
 - [Split Data component](#)
- Shred JSON
 - [JSON in your Azure SQL Database?](#)
- Encode and decode data
 - [Azure Data Factory Copy Activity with Base64 encoded string](#)
 - [Handling data encoding issues while loading data to SQL Data Warehouse](#)
- Configure error handling for the transformation
 - [Handle SQL truncation error rows in Data Factory mapping data flows](#)
 - [Troubleshoot mapping data flows in Azure ADF](#)
 - [Error row handling](#)
- Normalize and denormalize values
 - [Normalize Data](#)
 - [Normalize Data component](#)
 - [How do I denormalize data in Azure Machine Learning Studio?](#)
- Transform data by using Scala
 - [Extract, transform and load data by using Azure Databricks](#)
- Perform data exploratory analysis
 - [Exploratory Data Analysis with Azure Synapse Analytics](#)
 - [Query data in Azure Data Explorer Web UI](#)

Design and develop a batch processing solution

- Develop batch processing solutions by using Data Factory, Data Lake, Spark, Azure Synapse Pipelines, PolyBase, and Azure Databricks
 - [Batch processing](#)
 - [Choose a batch processing technology in Azure](#)
 - [Building Batch Data Processing Solutions in Microsoft Azure](#)
 - [Process large-scale datasets by using Data Factory and Batch](#)
 - [Run Spark Jobs on Azure Batch using Azure Container Registry and Blob storage](#)
 - [Batch Processing with Databricks and Data Factory in Azure](#)
- Create data pipelines
 - [Create a pipeline](#)
 - [Build a data pipeline by using Azure Data Factory, DevOps, and ML](#)
- Design and implement incremental data loads
 - [load data from Azure SQL Database to Azure Blob storage](#)
 - [Implement incremental data loading with ADF](#)
- Design and develop slowly changing dimensions
 - [Processing Slowly Changing Dimensions with ADF Data](#)
- Handle security and compliance requirements
 - [Azure security baseline for Batch](#)
 - [Azure Policy Regulatory Compliance controls for Azure Batch](#)
- Scale resources
 - [Automatically scale compute nodes in an Azure Batch pool](#)
- Configure the batch size
 - [Choose a VM size and image for compute nodes](#)
- Design and create tests for data pipelines
 - [Unit testing Azure Data Factory pipelines](#)
- Integrate Jupyter/Python notebooks into a data pipeline
 - [Set up a Python development environment for AML](#)
 - [Explore Azure ML with Jupyter Notebooks](#)
- Handle duplicate data
 - [Handle duplicate data in Azure Data Explorer](#)
 - [Dedupe rows and find nulls by using data flow snippets](#)

- [Remove Duplicate Rows component](#)
- Handle missing data
 - [Clean Missing Data component](#)
 - [Methods for handling missing values](#)
- Handle late-arriving data
 - [Late arriving events](#)
 - [Late Arrival Tolerance](#)
- Upsert data
 - [Optimize Azure SQL Upsert scenarios](#)
 - [Implement UpSert using Dataflow Alter Row Transformation](#)
- Regress to a previous state
 - [Monitor Batch solutions by counting tasks and nodes by state](#)
- Design and configure exception handling
 - [Error handling and detection in Azure Batch](#)
- Configure batch retention
 - [Manage task lifetime](#)
- Design a batch processing solution
 - [Batch processing](#)
- Debug Spark jobs by using the Spark UI
 - [Track an application in the Spark UI](#)

Design and develop a stream processing solution

- Develop a stream processing solution by using Stream Analytics, Azure Databricks, and Azure Event Hubs
 - [Implement a Data Streaming Solution with Azure Streaming Analytics](#)
 - [Stream processing with Azure Databricks](#)
 - [Stream data into Azure Databricks using Event Hubs](#)
- Process data by using Spark structured streaming
 - [Structured Streaming](#)
 - [Overview of Apache Spark Structured Streaming](#)
 - [Structured Streaming tutorial](#)

- Monitor for performance and functional regressions
 - [Understand Stream Analytics job monitoring](#)
- Design and create windowed aggregates
 - [Introduction to Stream Analytics windowing functions](#)
 - [Windowing functions \(Azure Stream Analytics\)](#)
- Handle schema drift
 - [Schema drift in mapping data flow](#)
- Process time-series data
 - [Time series data](#)
 - [Understand time handling in Azure Stream Analytics](#)
- Process across partitions
 - [Stream processing with Azure Stream Analytics](#)
 - [Use repartitioning to optimize processing with Azure Stream Analytics](#)
- Process within one partition
 - [Maximize throughput with repartitioning](#)
- Configure checkpoints/watermarking during processing
 - [Checkpoint in Azure Stream Analytics](#)
 - Watermarks
 - [Illustrated example of watermarks](#)
 - [how to calculate watermark for Azure Streaming Analytics](#)
- Scale resources
 - [Understand and adjust Streaming Units](#)
 - [Scale an Azure Stream Analytics job to increase throughput](#)
- Design and create tests for data pipelines
 - [Test live data locally using Azure Stream Analytics tools](#)
 - [Test an Azure Stream Analytics job in the portal](#)
- Optimize pipelines for analytical or transactional purposes
 - [Use repartitioning to optimize processing](#)
 - [Leverage query parallelization](#)
- Handle interruptions
 - [Avoid service interruptions in Azure Stream Analytics jobs](#)
- Design and configure exception handling

- [Azure Stream Analytics output error policy](#)
- [Exception handling](#)
- Upsert data
 - [Upserts from Stream Analytics](#)
 - [Azure Stream Processing upsert to DocumentDB](#)
- Replay archived stream data
 - [Estimate replay catch-up time](#)
- Design a stream processing solution
 - [Stream processing with Azure Stream Analytics](#)

Manage batches and pipelines

- Trigger batches
 - [Tutorial: Trigger a Batch job using Azure Functions](#)
- Handle failed batch loads
 - [Check for pool and node errors](#)
- Validate batch loads
 - [Job and task error checking](#)
- Manage data pipelines in Data Factory/Synapse Pipelines
 - [Monitor and manage Azure Data Factory pipelines](#)
 - [Managing the mapping data flow graph](#)
- Schedule data pipelines in Data Factory/Synapse Pipelines
 - [Create a trigger that runs a pipeline on a schedule](#)
- Implement version control for pipeline artifacts
 - [Source control in Azure Data Factory](#)
- Manage Spark jobs in a pipeline
 - [Monitor a pipeline](#)

Design and Implement Data Security (10-15%)

Design security for data policies and standards

- Design data encryption for data at rest and in transit

- [Azure Data Encryption at rest](#)
- [Azure Storage Encryption for data at rest](#)
- [Protect data in transit](#)
- Design a data auditing strategy
 - [Auditing for Azure SQL Database and Azure Synapse Analytics](#)
- Design a data masking strategy
 - [Dynamic data masking](#)
 - [Static Data Masking for Azure SQL Database and SQL Server](#)
- Design for data privacy
 - [Privacy in Azure](#)
- Design a data retention policy
 - [Understand data retention in Azure Time Series Insights Gen1](#)
- Design to purge data based on business requirements
 - [Data purge](#)
 - [Enable data purge on your Azure Data Explorer cluster](#)
- Design Azure role-based access control (Azure RBAC) and POSIX-like Access Control List (ACL) for Data Lake Storage Gen2
 - [Role-based access control \(Azure RBAC\)](#)
 - [Access control lists \(ACLs\) in Azure Data Lake Storage Gen2](#)
- Design row-level and column-level security
 - [Column-level security](#)

Implement data security

- Implement data masking
 - [Get started with SQL Database dynamic data masking](#)
- Encrypt data at rest and in motion
 - [Transparent data encryption for SQL Database](#)
- Implement row-level and column-level security
 - [Column-Level Security](#)
- Implement Azure RBAC
 - [Assign an Azure role for access to blob data](#)
- Implement POSIX-like ACLs for Data Lake Storage Gen2

- [Use PowerShell to manage ACLs in Azure Data Lake Storage Gen2](#)
- Implement a data retention policy
 - [Configuring retention in Azure Time Series Insights Gen1](#)
- Implement a data auditing strategy
 - [Set up auditing for your server](#)
- Manage identities, keys, and secrets across different data platform technologies
- Implement secure endpoints (private and public)
 - [Use private endpoints for Azure Storage](#)
 - [Use Azure SQL MI securely with public endpoints](#)
 - [Configure public endpoint in Azure SQL Managed Instance](#)
- Implement resource tokens in Azure Databricks
 - [Authentication using Azure Databricks personal access tokens](#)
- Load a DataFrame with sensitive information
 - [DataFrames tutorial](#)
- Write encrypted data to tables or Parquet files
 - [use Parquet with Azure Data Lake Analytics](#)
- Manage sensitive information
 - [Security Control: Data Protection](#)

Monitor and Optimize Data Storage and Data Processing (10-15%)

Monitor data storage and data processing

- Implement logging used by Azure Monitor
 - [Azure Monitor Logs overview](#)
 - [Collect custom logs with Log Analytics agent in Azure Monitor](#)
- Configure monitoring services
 - [Tutorial: Monitor Azure resources with Azure Monitor](#)
 - [Enable VM insights overview](#)
- Measure performance of data movement
 - [Copy activity performance and scalability guide](#)

- Monitor and update statistics about data across a system
 - [Update statistics](#)
 - [UPDATE STATISTICS \(Transact-SQL\)](#)
- Monitor data pipeline performance
 - [Monitor and Alert Data Factory by using Azure Monitor](#)
- Measure query performance
 - [Query Performance Insight for Azure SQL Database](#)
 - [How to measure the performance of the Azure SQL DB?](#)
- Monitor cluster performance
 - [Monitor cluster performance in Azure HDInsight](#)
- Understand custom logging options
 - [Collect text logs with Log Analytics agent in Azure Monitor](#)
- Schedule and monitor pipeline tests
 - [Monitor and manage Azure Data Factory pipelines](#)
- Interpret Azure Monitor metrics and logs
 - [Azure Monitor Metrics overview](#)
 - [Overview of Azure platform logs](#)
- Interpret a Spark directed acyclic graph (DAG)
 - [Directed Acyclic Graph DAG in Apache Spark](#)

Optimize and troubleshoot data storage and data processing

- Compact small files
 - [Auto Optimize](#)
- Rewrite user-defined functions (UDFs)
 - [Modify User-defined Functions](#)
- Handle skew in data
 - [Resolve data-skew problems](#)
- Handle data spill
 - [Data security Q&A](#)
- Tune shuffle partitions
 - [Use Unravel to tune Spark data partitioning](#)
- Find shuffling in a pipeline

- [Lightning-fast query performance with Azure SQL Data Warehouse](#)
- Optimize resource management
 - [How to optimize your Azure environment](#)
 - [Azure resource management tips to optimize a cloud deployment](#)
- Tune queries by using indexers
 - [Automatic tuning for SQL Database](#)
- Tune queries by using cache
 - [Performance tuning with result set caching](#)
- Optimize pipelines for analytical or transactional purposes
 - [Hyperspace: An indexing subsystem for Apache Spark](#)
- Optimize pipeline for descriptive versus analytical workloads
 - [Optimize Apache Spark jobs in Azure Synapse Analytics](#)
- Troubleshoot a failed spark job
 - [Troubleshoot Apache Spark by using Azure HDInsight](#)
 - [Troubleshoot a slow or failing job on an HDInsight cluster](#)
- Troubleshoot a failed pipeline run
 - [Troubleshoot pipeline orchestration in Azure Data Factory](#)

DP-203 Exam Retake Policy

The DP-203 exam retake policy is as follows:

1. If a candidate fails on the first attempt, they must wait for 24 hours before retaking the exam.
2. If a candidate again fails on the second attempt, then the candidate will have to wait for 14 days.
3. A candidate will be given a maximum of five attempts to retake an exam in a year.

DP-203 Exam Day Tips

- With practice tests, you can become familiar with the test format while broadening your knowledge at the same time.

- Those questions are very similar to those that you will find on test day, and more importantly, each answer is explained with a reference to documentation.
- Let your common sense and previous knowledge take center stage in the first learning phase, and try to answer every question.
- At least a week before the exam, make sure you schedule it.
- A quiet space should be provided for test-taking.
- Do not read the question loudly otherwise, you may get disqualified
- Please remove all paper, pencils, external keyboards, etc. from sight before taking photos.
- During the test, be mindful of your eyes. Pearson VUE will monitor you throughout your test by using the front-facing camera on your device. You may be accused of cheating if you fail to pay attention to your computer screen during the test.
- Avoid staring into the distance while you are thinking during your exam.

Conclusion

The certification is for those candidates who want to demonstrate expertise in designing and implementing data solutions with the use of [Microsoft Azure](#) data services.

Through this certification, you will learn to integrate, transform and consolidate unstructured data into structured data, that are suitable for building analytical solutions for the company.

Make sure to understand the concept behind the answers and eventually you will be able to use this knowledge to pass every practice and actual DP-203 certification test.

I hope this article is helpful to you and wish you good luck!

If you have any questions, please feel free to ask in below comment section.

Related/References

- [DP-100 Exam Study Guide](#)
- [PL-300 Exam Study Guide](#)
- [AZ-500 Exam Study Guide](#)
- [AZ-204 Exam Study Guide](#)
- [AZ-900 Exam Study Guide](#)
- [AI-900 Exam Study Guide](#)



1 thought on “DP-203: Microsoft Azure Data Engineer Associate Exam Study Guide”



Sneha

August 8, 2022 at 4:31 pm

thanks

[Reply](#)

Leave a Comment

Name *

Email *

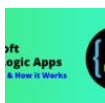
☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

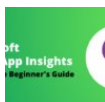
Recent Posts



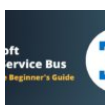
Azure Storage Explorer: Download, Install, and Setup Overview



What are Azure Logic Apps: Components, Advantages and How it Works



Microsoft Azure Application Insights: A Complete Beginners Guide



Microsoft Azure Service Bus: A Complete Beginners Guide



Azure App Service: Types, Benefits and Limitations

[Privacy Policy](#) [About](#)

Copyrights © 2021-22, cloudkeeda. All Rights Reserved