Cloud Data Engineer

212 שעות לימוד אקדמיות

תיאור התפקיד:

היום כדי להצליח ולהשתלב בעבודה בסביבת Big Data וענן, נדרשים אנשי דאטה ומפתחים לבצע קפיצת מדרגה ולשדרג את הכישורים והמיומנויות הטכנולוגיות שלהם בעולמות הדאטה, גם בצד האנליטי וגם בצד האפליקטיבי והתשתיתי.

עליהם להיות מסוגלים לקבל החלטות נכונות בעבודה עם טכניקות עיבוד וניתוח של כמות אינסופית של נתונים ב- scale משתנה, לבחור נכון את הכלים הרלוונטיים למשימה ספציפית, ולדעת לגשת לכמויות גדולות של נתונים גולמיים ובלתי מובנים ולמצוא בהם תובנה עסקית או כל מידע שימושי אחר.

בשנים האחרונות מחפשים ארגונים מומחי דאטה חדשים.

כאלה שיודעים לכרות את הנתונים הגולמיים ולהפוך אותם לתובנות עסקיות בהן משתמשים מנהלים כדי לבנות אסטרטגיה עסקית. כל זה תוך מיקוד וטיפול במשאבי מידע עצומים ובפיתוח פתרונות לשינוע היעיל והרווחי ביותר של דאטה בין מחלקות/בן מוצרים/בין כלים בארגוני הייטק וחברות בתחומים נוספים.

אז מה תפקידם של ה-Cloud Data Engineer?

מהנדסי נתונים בסיסית הענן הם אלה המעצבים, מפתחים ומנהלים פתרונות לשינוע הנתונים (בניית Pipelines), באמצעות עבודה עם טכנולוגיות Big Data וענן מגוונות ומתקדמות.

על מהנדסי נתונים בסיסית הענן לשלוט בטכנולוגיות המידע השונות, כולל בסביבות הענן ושירותי דאטה שהן מספקות, בכלים המובילים לשינוע ועיבוד מידע בנפחים גדולים ובשפת התכנות המובילה היום שהיא Python.

תיאור הכשרת Cloud Data Engineer של נאיה קולג'

הכשרת Cloud Data Engineer פותחה בהתאם לדרישות הקיימות כיום בתעשיית ההייטק לעבודה עם מסדי נתונים מבוזרים שאינם מובנים, עם מערכות מבוזרות ועם שירותי דאטה בענן.

ההכשרה מותאמת לדרישות התפקיד בארגונים ולכן הינה מגוונת וכוללת היכרות עם טכנולוגיות מידע וכלים שונים.

מודול הליבה הראשון של ההכשרה הינו אפליקטיבי ומטרתו להקנות ידע ויכולות בעיבוד נתונים (Data Processing) ובכתיבת Data Pipeline.

במודול זה נתמקד בלימוד מעמיק של שפת Python, השפה המובילה לכתיבת תהליכי ETL בארגונים מבוססי דאטה. נלמד כיצד Python והחבילות העוטפות (Python APIs) מאפשרות לאסוף נתונים ממקורות רבים ומגוונים (בסיסי נתונים, JSON, TEXT,XML, LOG, אינטרנט וכלי Big Data) לעבד אותם, ובסופו של דבר לבנות תשתית לתעבורת הנתונים כאשר התוצר שלה הינו דאטה מוכן לגורמים הרלוונטיים בארגון (אנליסטים, מדעני נתונים, מפתחים ועוד.)

מודול הבא של ההכשרה מתמקד בטכנולוגיות Big Data וגם בנושא DataOps.

בעולמות הביג דאטה לא ניתן לעבוד היום ללא היכרות בסיסית לפחות במערכת הפעלה Linux, לכן ניגע במערכת הפעלה זו לרמת היכרות מספקת להרצת פקודות רלוונטיות בסביבת ביג דאטה.

בצד של הדאטה – ידע בכלים כמו Kafka הוא חובה וכלן גם כאן נלמד ונתרגל את הפעולות הקריטיות של המערכת הזו.

בצד של הקוד – נלמד לנהל את הגרסאות של הקוד ב-GIT, נלמד לעבוד עם Docker שכיום זה כבר לא רק כלי עבודה של אנשי DevOps אלא גם של מפתחים ומהנדסי נתונים בפרט. ולאחר מכן נתמקד בתחום CI/CD - זוהי גישה המיישמת המשכיות תקינה של תהליכי פיתוח, תוך כדי אינטרקציה עם כלל הגורמים המעורבים בפיתוח במקביל ומאפשרת ניהול סיכונים חכם, פיקוח ובקרה לאורך כל תהליכי האוטמוציה בבנייה, בדיקה ופריסה של יישומים ואפליקציות.

כיום כשהדרישות ל- Delivery מהיר ואיכותי הן גבוהות מאי פעם, לא ייתכנו תהליכי פיתוח מודרניים ללא אינטגרציה על פי הגישות האלו. ביניהם גם תהליכי פיתוח דאטה. כשבאים לבחון את היתרונות בעבודה על פי גישה זו ניתן למצוא נושאים כגון זמני פיתוח קצרים מאי פעם המאפשרים לטפל בבאגים במהירות, שילוב פונקציונליות רבה יותר כבר בשלב הפיתוח הראשוני ותהליכי בדיקות קצרים ויעילים יותר. לכן הבנה בכלים רלוונטיים בתחום הזה הינה קריטית גם למהנדסי נתונים.

המודול הבא יעסוק בסביבת Spark, אחת מהטכנולוגיות הנדרשות ביותר כיום בשוק לצורך עיבוד נתונים בסביבת Big Data. במודול זה נבין כיצד עבודה עם PySpark מאפשרת לנהל את ה-Pipeline בסביבה מבוזרת, תוך כדי מינוף האופטימיזציות ש-Spark מביאה איתה. שליטה ב-Spark תקנה לך יתרון משמעותי ביחס למועמדים אחרים!

המודול הבא הוא עוד שלב מהותי, שיעסוק בכלים וטכנולוגיות דאטה בסביבת הענן. נכיר את הארכיטקטורה של הכלים, את אופן קליטת ושמירת הנתונים, נרכוש יכולות שליפה, עיבוד ואינטגרציה של הנתונים וכן ניצור תהליכי אוטומציה בזרימת הדאטה. במודול זה נעסוק באופן מעשי בכל סוגי הענן הקיימים, נקיים תרגולים פרקטיים שנלקחו מהשטח ונתמקד בענן של AWS ובטכנולוגיות רלוונטיות לעבודה בו, וביניהן: Athena, Glue, S3, RDS , Redshift Lambda & API Gateway, Message Queues.
בנוסף, במהלך הכשרה זו יילמדו גם הטכנולוגיות החדשניות והמדוברות ביותר בתחום כמו: Apache Airflow, Apache NiFi ועוד.
המודול האחרון בהכשרה זו מתמקד בטכנולוגיות NoSQL.
האתגרים בתחום הדאטה בשנים אחרונות מציבים אותנו בפני מצב שבסיסי הנתונים הרלציוניים כבר אינם מסוגלים להתמודד עם נפחים של דאטה ועם תהליכים כבדים כגון AI ו- Data Science. כלי NoSQL פותרים את הבעיה הזו. הם מתבססים על ארכיטקטורות מבוזרות ובסיסי נתונים לא רלציונים ומאפשרים יכולת אחסון לדאטה ב- Scale גבוה ושליפה יעילה ומהירה של הנתונים מתוכם. הם מסוגלים להתמודד עם Use Cases מגוונים ומורכבים של כל ארגון .

בנוסף ללימוד התיאורטי ולתרגולים השוטפים במהלך ההכשרה, כולל הקורס גם פרויקט גמר מקיף המאפשר לסטודנטים להתנסות במכלול הטכנולוגיות הנלמדות, לבצע אינטגרציה לבישורים ולידע שאספו במודולים השונים ולתפור פתרון Big Data אמיתי מקצה לקצה.

למי מתאימה ההכשרה?

קורס Cloud Data Engineer לא מתאים לכל אחד.

הוא פונה לאנשים בעלי נסיון משמעותי בעבודה עם בסיסי נתונים רלציוניים, בעלי נסיון ו/או הכרות טובה עם תהליכי ETL המעוניינים לקחת את הקריירה שלהם צעד משמעותי קדימה לתוך העולם Big Data & Cloud ביניהם: אנשי BI, אנליסטים, Product Data Managers, מפתחים, DBA וכל אחד שעבד עם נתונים בצורה משמעותית.

דרישות קדם:

- ידע וניסיון בעבודה מול RDBMS
- ידע וניסיון בשפת SQL ברמה בסיסית ומתקדמת
- נסיון ו/או ידע בנושא ETL
- אנגלית ברמה גבוהה
- מעבר בהצלחה מבחן כניסה וראיון אישי

עם מה יוצאים מההכשרה?

בוגרי הקורס יצאו עם ידע פרקטי רב ותיק עבודות רחב המציג שימוש בטכנולוגיות השכיחות והחמות ביותר בתעשיית ההייטק בארץ . הידע שנרכש במהלך ההכשרה יאפשר להם להשתלב בתפקידי מפתח ולהוביל תהליכי בניית תשתיות לתעבורת הנתונים בסביבות ענן ו-Big Data.

וזה עוד לא הכל - בתום הלימודים מקבלים הבוגרים סדנת פיתוח קריירה, הכוללת כתיבת קורות חיים, סימולציית ראיון, וקישור למשרות ולארגונים מעסיקים!
לקראת ראיונות העבודה הם מקבלים ליווי והכנה לראיונות מקצועיים על ידי מומחי הדאטה הבכירים שלנו.

ההכשרה על כל שלביה ובעיקר פרויקט הגמר בסופה, מבינה את הסטודנטים באופן מיטבי להתמודדות עם האתגרים בהם יתקלו בהמשך דרכם המקצועית ומוציאה אותם מובנים ומנוסים לשלב חיפוש העבודה.

תכני הקורס:

Big Data Technologies Introduction for Data Engineers

- Hadoop Eco-System
- NoSQL
  - Introduction to NoSQL
  - Main characteristics of NoSQL solutions
  - The four types of NoSQL solutions, leading technologies and real world use cases
- Technologies and trends in the world of Big Data
  - Search engines (Elastic search, SOLR)
  - Cloud Computing
  - NewSQL databases (Vertica, VoltDB, MemSQL)

Python and PyData

- **Basic Python**
  - Working environments (Anaconda, Jupiter, PyCharm, etc.)
  - Data types (numbers, strings, Booleans, etc.)
  - Data collections (lists, dictionaries, etc.)
  - Flow control (if, for, while etc.)
  - Textual interface (input and formatting)
- **Intermediate Python**
  - Functions
    - User-defined functions
    - *args and **kwargs
    - Built-in functions
    - Lambda expressions
  - Debugging and Error Handling
  - Text files
  - The standard library
    - import
    - datetime
- **Pandas and Data Resources**
  - The NumPy library
    - Array
    - Broadcasting
  - The matplotlib library
    - matplotlib object
    - Plotting

        - Seaborn
    - The pandas library
        - Series and Index
        - DataFrame
        - GroupBy
        - Visualizations
    - General tools and DB
        - Intro to regular expressions (re)
        - JSON
        - DBs with SQLAlchemy package and Pandas

Big Data and DataOps

- **Hadoop Introduction**
    - The rise of Big Data recap
    - Introduction to Apache Hadoop
    - Apache Hadoop core components
    - Apache Hadoop ecosystem
- **Basic Linux**
    - Connecting Using Putty
    - Working With Directories , Files, File Contents
    - Control Operators
    - Shell Variables
    - I/O Redirection
    - Filters
    - File Permissions
- **Apache kafka**
    - ETL Vs Data streaming
    - Kafka and data streaming
    - Kafka use cases
    - Architecture
    - Fundamental concepts
    - Core APIs
    - Main distributions
    - Basics steps (Linux/Mac)
- **Git**
    - Introduction to Source control
    - Git Branches, tags and release management
    - gitignore
    - pull requests (PR)
- **Docker**
    - What is Docker & containers
    - Docker main components and terminology

- Docker architecture, hub, desktop and commands

- **Airflow**
  - Introduction to Apache Airflow
  - Airflow Components
  - DAGs
  - Infrastructure
  - Advanced
  - Integrations
- **CI/CD**
  - CI/CD pipeline
  - Jenkins
  - Build machine
  - Artifact management
  - Tagging and releases

Spark

- Introduction to Spark
- RDD – Low Level API
- Broadcast & Accumulators
- Spark Partitioning
- Spark SQL API
- Working with Data Sources
- DataFrame Operations
- Spark UI

AWS

- **Cloud Deployment Models**
  - Service models:
    - Infrastructure as a service (Iaas)
    - Platform as a service (Paas)
    - Software as a service (Saas)
- **Fundamentals Overview**
  - Storage
  - Networking and security
  - Compute
  - Virtualization and Containers
  - Managed services
  - Databases – Relational and NoSQL
- **Amazon and AWS History & Region**
- **AWS Management Console**
- **AWS Command Line Interface**

- **Amazon S3**
  - What is a Big Data Data-Lake S3
  - Amazon S3 Integration
  - Data Lake on S3 Use-Cases
- **Amazon RDS**
  - What is Amazon RDS
  - Supported Database Engines
  - Features and Advantages
  - Creation Method
  - Database Engine Type & Instance Sizes
  - Amazon RDS Storage Types
  - Setting the Availability
  - RDS Parameter Groups
  - Amazon Aurora
- **Amazon Redshift**
  - What is Amazon Redshift
  - Amazon Redshift and S3 Integration
  - Architecture
  - Performance Optimization
  - Managing and Query Methods
  - Amazon Redshift SQL Editor
  - Settings and Configuration
  - Amazon Redshift Lab
- **Amazon Athena**
  - Features and Advantages
  - Athena SQL Engine
  - Data model for Athena & Partitioning
  - Athena Table Definitions and Schema
  - Cost Considerations
  - Amazon Athena AWS Services Integrations
  - Amazon Athena Real World Projects
- **Amazon Glue**
  - What is AWS Glue
  - Features and Advantages
  - Main Concepts:
    - supported data Sources
    - supported data Targets
  - Main Components
    - Data Catalog
    - Crawlers and Classifiers
    - ETL Operations

- Streaming ETL
- Tables
- Connections
- Glue Use-Cases

- **Lambda & API Gateway**
  - Lambda Features and Advantages
  - Lambda Code Editor
  - Function Creation Options
  - Lambda Considerations
  - Exposure Rest API and handle events
  - Integration with sources and targets Lab
- **Message Queues**
  - Introduction to message queues
  - Simple Queue Service (SQS)
  - Kinesis Services
  - Kinesis Data Streams Use-Cases
  - Amazon Kinesis Pricing Method

NoSQL

- **MongoDB**
  - MongoDB introduction and architecture, components, deployment, indexes
  - MongoDB CRUD and Administrative Commands
- **Search Engines - ELK**
  - Introduction to search engines, Elasticsearch and typical use-cases
  - Basic setup and configurations
  - Indexes (Lucene) and mapping configurations
  - Querying: APIs, queries and complex queries, aggregations
  - Introduction to Logstash
  - Introduction to Kibana
- **Apache NiFi**
  - Introduction
  - Core Components
  - Parameter Context
  - Architecture
  - Registry
  - Limitations

Final Project