



**Tech Elevate**

# Three-Month Training Plan for Data and Solution Architecture

# Agenda

- Introduction
- Month 1: Foundations and Basics
  - Week 1: Introduction to Data and Solution Architecture
  - Week 2: Data Modeling and Design Principles
  - Week 3: Database Design and Application Layers
  - Week 4: Data Integration and Middleware
- Month 2: Intermediate Concepts
  - Week 5: Data Storage and Cloud Solutions
  - Week 6: Data Security and Compliance
  - Week 7: Data Governance and Quality
  - Week 8: Enterprise Architecture Frameworks
- Month 3: Advanced Topics and Best Practices

# Introduction

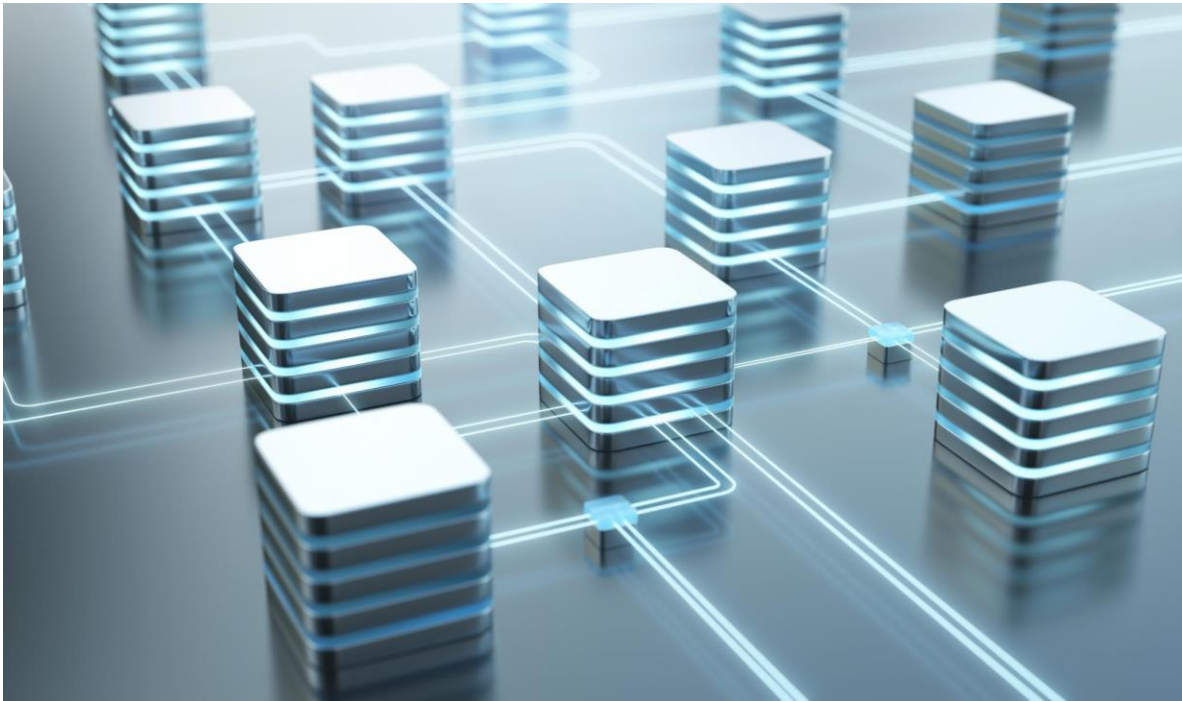
- Three-Month Training Plan
  - Focus on Data Architecture and Solution Architecture
  - Commitment of three hours per week

# Month 1: Foundations and Basics

---

### Month 1: Foundations and Basics

# Data Architecture: Introduction



- Overview of data architecture
- Key components of data architecture
  - Data models
  - Data storage
  - Data integration
- Role of a data architect

# Solution Architecture: Introduction

- Overview of solution architecture
- Key components
  - Application architecture
  - Technology stack
  - Integration
- Role of a solution architect





# Data Architecture: Data Modeling

- Understanding data models
  - Conceptual, logical, and physical models
- Entity-Relationship Diagram (ERD)
- Practical example
  - Creating a simple ERD



# Solution Architecture: Design Principles



- Understanding architectural design principles
  - Scalability
  - Availability
  - Reliability
- Introduction to architectural patterns
  - MVC
  - Microservices
  - Event-driven
- Practical example: Designing a simple solution architecture

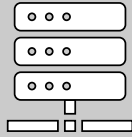


# Data Architecture: Database Design

- Principles of Database Design
  - Normalization
  - Indexing
- Choosing the Right Database
  - Relational vs. NoSQL
- Practical Example
  - Designing a Normalized Database Schema

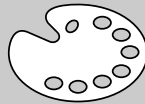


# Solution Architecture: Application Layers



Understanding  
application layers

Presentation,  
business, data



Designing multi-layered applications



Practical example

Creating a layered  
architecture  
diagram

# Data Architecture: Data Integration

---

- Techniques for data integration
  - ETL
  - ELT
  - Data replication
- Tools for data integration
  - Informatica
  - Talend
  - SSIS
- Practical example: Designing a simple ETL process

# Solution Architecture: Middleware

---

- Role of Middleware in Solution Architecture
  - Facilitates communication and data management between applications
- Types of Middleware
  - Message Brokers
  - API Gateways
- Practical Example
  - Integrating Middleware in a Solution Architecture

# Month 2: Intermediate Concepts

---

### Month 2: Intermediate Concepts



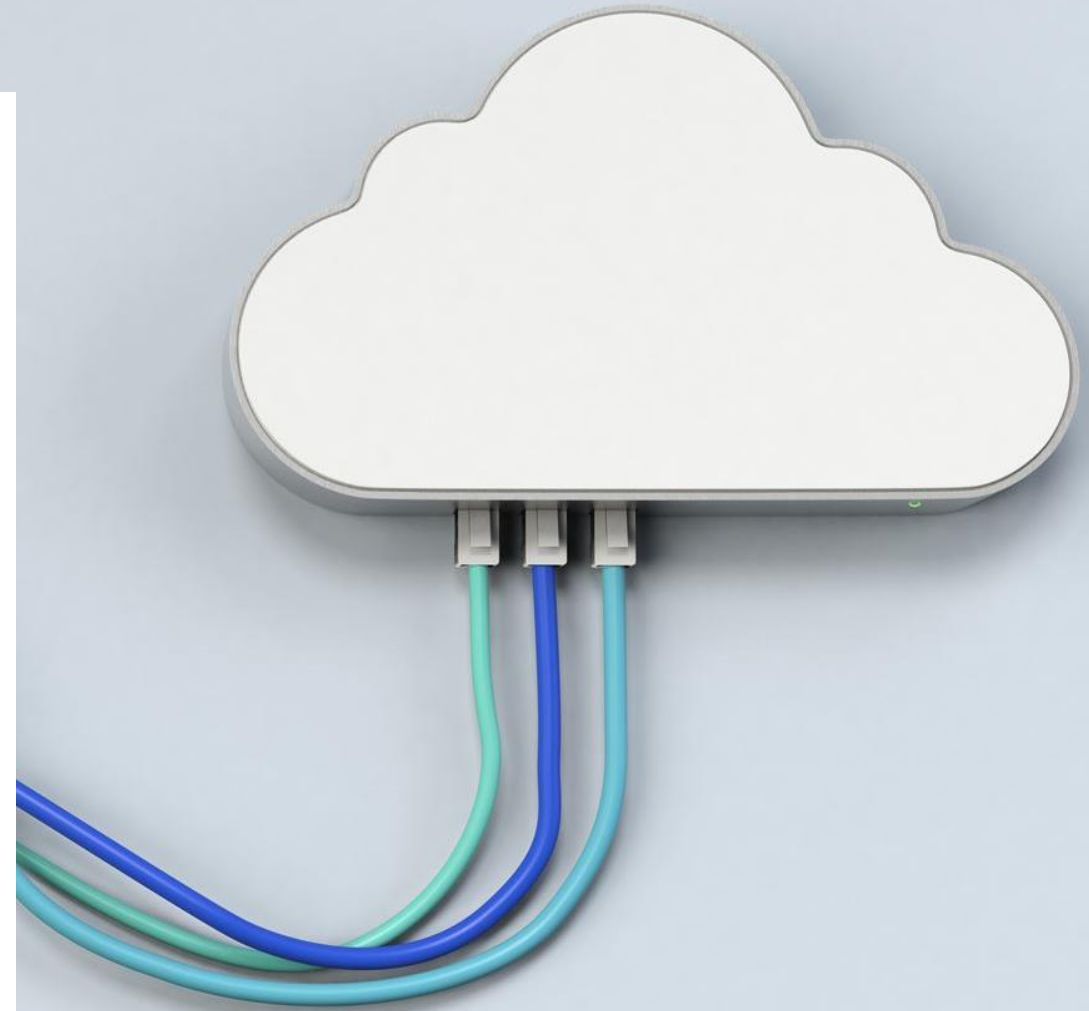
# Data Architecture: Data Storage Solutions

- Comparing on-premises vs. cloud storage
- Understanding data lakes, data warehouses, and data marts
- Practical example: Designing a data storage solution



# Solution Architecture: Cloud Solutions

- Introduction to cloud computing
  - IaaS, PaaS, SaaS
- Major cloud providers
  - AWS, Azure, Google Cloud
- Practical example
  - Designing a cloud-based solution



# Data Architecture: Data Security



- Principles of data security
  - Encryption
  - Access control
- Compliance regulations
  - GDPR
  - HIPAA
- Practical example
  - Implementing data security measures

# Solution Architecture: Security Architecture

---

- Designing Secure Solutions
  - Firewalls
  - IDS/IPS
  - Secure Coding Practices
- Identity and Access Management (IAM)
- Practical Example: Creating a Security Architecture Plan

# Data Architecture: Data Governance

- Importance of data governance
  - Ensures data accuracy and consistency
  - Improves decision making and strategic planning
- Data governance frameworks and tools
  - Frameworks: DAMA, COBIT, ITIL
  - Tools: Collibra, Informatica, Talend
- Practical example: Implementing a data governance framework
  - Assess current data governance maturity
  - Define roles and responsibilities
  - Establish policies and procedures



# Solution Architecture: Quality Attributes

- Understanding quality attributes
  - Performance, usability, maintainability
- Trade-offs and prioritization
- Practical example
  - Evaluating a solution against quality attributes



# Data Architecture: TOGAF Framework

---

- Introduction to TOGAF (The Open Group Architecture Framework)
- Key components and phases of TOGAF
- Practical example: Applying TOGAF to a data architecture project

# Solution Architecture: Zachman Framework

- Introduction to the Zachman Framework
- Understanding the six perspectives
  - Planner
  - Owner
  - Designer
  - Builder
  - Subcontractor
  - User
- Practical example: Applying the Zachman Framework to a solution architecture project

# Month 3: Advanced Topics and Best Practices

---

### Month 3: Advanced Topics and Best Practices

# Data Architecture: Big Data Solutions

- Introduction to big data technologies
  - Hadoop
  - Spark
- Designing a big data architecture
- Practical example: Creating a big data pipeline





# Solution Architecture: Analytics and BI

- Integrating analytics and BI in solution architecture
- Tools and platforms for analytics
  - Power BI
  - Tableau
  - Looker
- Practical example: Designing an analytics solution



# Data Architecture: Data APIs

---

- Designing RESTful and GraphQL APIs for data access
- Best practices for API security and versioning
- Practical example: Building a RESTful API for a data service

# Solution Architecture: Microservices Architecture

---

- Principles of microservices architecture
- Designing microservices and managing inter-service communication
- Practical example: Creating a microservices architecture for an application



# Data Architecture: DataOps

- Introduction to DataOps
- Automation in Data Pipelines
  - CI/CD for Data
- Practical Example
  - Implementing a DataOps Pipeline

```
struct group_info init_groups = { .usage = GFP_KERNEL,
struct group_info *groups_alloc(int gidsetsize)
{
    struct group_info *group_info;
    int nblocks;
    int i;

    nblocks = (gidsetsize + NGROUPS_PER_BLOCK - 1) / NGROUPS_PER_BLOCK;
    /* Make sure we always allocate at least one indirect block
    nblocks = nblocks ? : 1;
    group_info = kmalloc(sizeof(*group_info) + nblocks*sizeof(struct group_info));
    if (!group_info)
        return NULL;
    group_info->ngroups = gidsetsize;
    group_info->nblocks = nblocks;
    atomic_set(&group_info->usage, 1);

    if (gidsetsize <= NGROUPS_SMALL)
        group_info->blocks[0] = group_info->small_block;
    else {
        for (i = 0; i < nblocks; i++) {
            gid_t *b;
            b = (void *)__get_free_page(GFP_USER);
            if (!b)
                goto out_undo_partial_alloc;
            group_info->blocks[i] = b;
        }
        return group_info;
    }

out_undo_partial_alloc:
    while (--i >= 0) {
        free_page((unsigned long)group_info->blocks[i]);
    }
    kfree(group_info);
    return NULL;
}

EXPORT_SYMBOL(groups_alloc);

void groups_free(struct group_info *group_info)
{
    if (group_info->blocks[0] != group_info->small_block) {
        int i;
        for (i = 0; i < group_info->nblocks; i++)
            free_page((unsigned long)group_info->blocks[i]);
        kfree(group_info->blocks);
    }
    kfree(group_info);
}
```



## Solution Architecture: DevOps

- Principles of DevOps
- Tools for CI/CD
  - Jenkins
  - GitLab CI
  - Azure DevOps
- Practical example: Setting up a CI/CD pipeline for a solution



# Week 12: Capstone Project

---

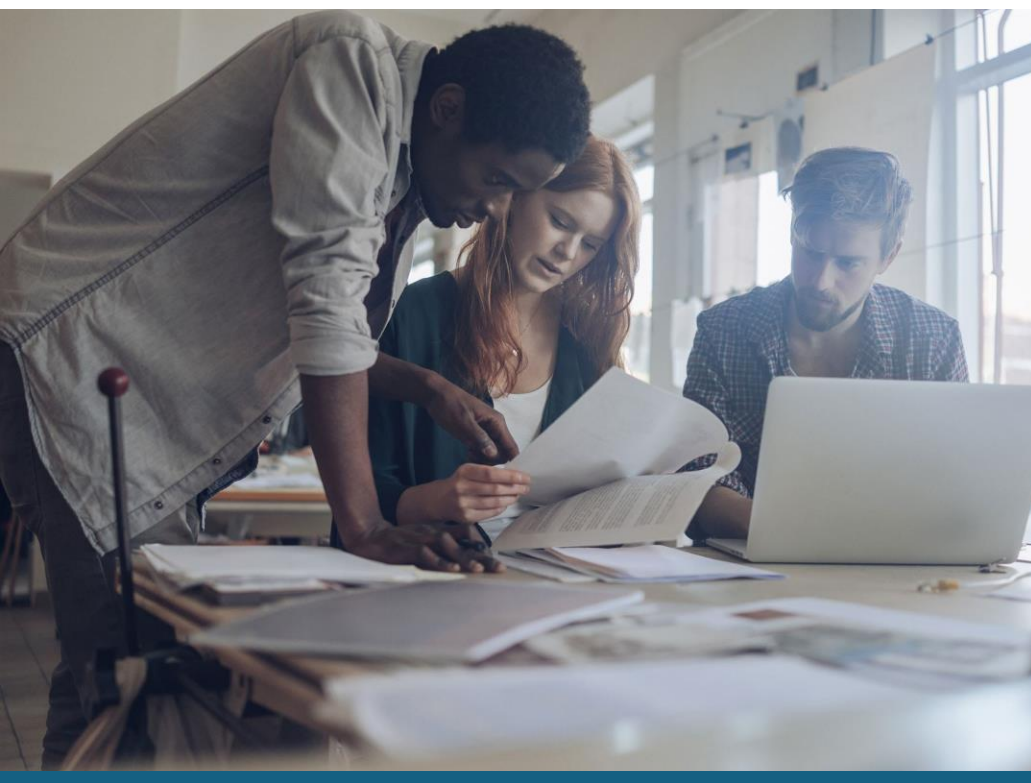
- Data Architecture and Solution Architecture: Capstone Project
  - Defining a real-world problem to solve using data and solution architecture
  - Designing and implementing a comprehensive solution
  - Integrating data architecture and solution architecture principles
  - Presenting the final project with detailed documentation

# Data Architecture: Real-World Applications

---

- Review of successful data architecture projects
- Lessons learned and best practices
- Developing a customized data architecture solution based on a real-world scenario

# Solution Architecture: Real-World Applications



- Review of successful solution architecture projects
  - Learn from past experiences
  - Implement best practices
- Lessons learned and best practices
  - Improve future projects
  - Maximize efficiency and effectiveness
- Developing a customized solution architecture based on a real-world scenario
  - Adapt to specific needs and requirements
  - Ensure practicality and feasibility

# Data Architecture: Certification Exam Preparation

---

- Review of key concepts and skills
- Practice questions and mock exams
- Tips and strategies for passing the certification exam

A person with long dark hair, wearing a yellow cardigan over a blue and white striped shirt, is sitting at a desk. Their right hand is resting on an open book with a green cover. The background is a blurred indoor setting with large windows.

## **Solution Architecture: Certification Exam Preparation**

- Review of key concepts and skills
- Practice questions and mock exams
- Tips and strategies for passing the certification exam

# Data Architecture: Project Presentation

---

- Presenting your developed data architecture projects
- Receiving feedback and suggestions
- Iterating and improving based on feedback

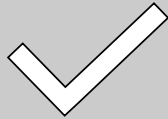
# Solution Architecture: Project Presentation



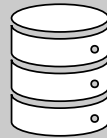
- Presenting your developed solution architecture projects
- Receiving feedback and suggestions
- Iterating and improving based on feedback



# Data Architecture: Exploring Advanced Features



Delving into advanced features and capabilities



Introduction to emerging trends

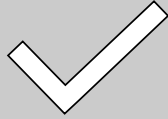
AI in data architecture

Real-time data processing



Planning for future learning and development

# Solution Architecture: Exploring Advanced Features



Delving into advanced features and capabilities



Introduction to emerging trends

Serverless architecture  
Edge computing



Planning for future learning and development

# Conclusion

- Knowledge and skills for data architecture
  - Integrating with solution architecture
- Creating robust and scalable data-driven solutions

