TECHEAZY
consulting

TechEazy Consulting

Free One Month Internship
Second Assignment

Skills needed for this project

📦 **Language (Java/Python/JavaScript)**

- **OOP concepts (Encapsulation, Inheritance, Polymorphism)**
- **Interfaces and Classes**
- **Exception Handling**
- **Collections and Generics**
- **Lambda & Streams (for filtering and mapping data)**
- **File I/O (for local image handling)**

Web application framework

- **Exposing and consuming REST API**
- **Writing service layer**
- **Configuration of database**
- **Using an ORM**
- **Defining entities and relationships between them**
- **CRUD operations with PostgreSQL**
- **DTOs and mapping**
- **Authentication & Authorization**
- **JWT integration (login, token generation, secured endpoints)**

⚛️ **ReactJS**

- Functional components and hooks
- Fetch API / Axios to call backend
- Form handling (upload photo form)
- Routing (React Router)
- JWT token handling (Login, LocalStorage)

☁️ **AWS Concepts (Optional and Good to Have)**

🔐 **IAM**

- Create users, roles
- Attach policies for S3, EC2, DynamoDB

🖼️ **S3**

- Create buckets
- Upload/download files
- Set permissions for public access

🧳 **EC2**

- Launch instance
- SSH into instance
- Host Spring Boot app with NGINX or directly using port mapping

🗃️ **DynamoDB**

- Create tables
- Insert and query records
- Understand primary keys and indexes

Skills needed for this project - Devops

AWS Cloud Services

- EC2 (launch, configure, user data)
- IAM (roles, instance profiles)
- S3 (private buckets, lifecycle rules)
- CloudWatch (logs, alarms)
- SNS, SQS (notifications & workflows)
- VPC (basic multi-tenant setup)

Infrastructure as Code

- **Terraform** or **CloudFormation** (parameterized EC2, IAM, S3 setup)

Scripting & Automation

- **Bash**, **AWS CLI**, or **Python (Boto3)** (provisioning, deployment, log upload)

CI/CD with GitHub Actions

- Auto deploy on push/tag
- Stage-based configs (dev, prod)
- SSH + remote script execution
- Secure secrets handling

Monitoring & Alerts

- CloudWatch Agent setup
- Error-based alarms
- Email alerts via SNS

Best Practices

- No hard coded secrets
- Config-driven setups
- Private/public repo access based on stage

Assignment: Zero Mile Delivery System — From Warehouse to Doorstep

Problem Statement

Build a **Zero Mile Delivery System** for a logistics company that handles **last-mile parcel delivery** from a central **warehouse**.

System Overview

- **Multiple Vendors** send a **Parcel List** to the **Warehouse**.
  - Each parcel list includes multiple parcels with:
    - Customer Name
    - Delivery Address
    - Contact Number
    - Parcel Size and Weight

- A **Web Portal** is used to:
  - Upload parcel lists by vendors
  - View all received parcels
  - Group parcels by delivery area and size
  - Plan **best possible delivery routes**
  - Assign available drivers and vehicles to each route
  - Track delivery status in real-time

1st Assignment for Developers

- Backend (Java/Spring Boot/Node/Python/FastAPI
  - Parcel Entity
    - Customer Name
    - Delivery Address
    - Contact Number
    - Parcel Size and Weight
    - Tracking Nunber
  - Parcel DTO
    - Customer Name
    - Delivery Address
    - Tracking Nunber
  - ParcelService
  - ParcelController
  - ParcelRepo
  - REST
    - GET list of all parcels
    - GET a parcel with given tracking ID
    - Create a parcel -save it in in memory
- FrontEnd (ReactJS/Angular)
  - A basic layout
  - Axios to connect to server
  - Form to create a parcel
  - Grid to show existing parcel
  - Option to delete and edit the parcel

2nd Assignment for Developers

- Backend (Java/Spring Boot/Node/Python/FastAPI
  - DeliveryOrder
    - A file containing list of parcels
      - Each line represents a Parcel Object (DTO to Entity)
    - Name of the vendor -like Myntra, Flipkart
    - Order delivery date
    - Vendor subscription type -ENUM for e.g. NORMAL, PRIME, VIP
  - DeliveryOrderDTO
    - Date
    - VendorName
    - TotalOrders
    - File link -which can be used by UI to download the file
  - DeliveryOrderService
  - DeliveryOrderController
  - DeliveryOrderRepo -Support Pagination
  - Vendor
    - All layers for Vendor -Support pagination
    - Relationship between Vendor and DeliveryOrder
  - REST
    - GET list of Delivery Orders for today
      - Add filters to get all Orders for a vendor and given date
    - Upload API which can be used by vendor to upload Order Details
    - Create a parcel -save it in in memory
  - **LOGIN API -Based on JWT token**
- FrontEnd (ReactJS/Angular)
  - A landing page
  - Login Link in navbar
  - Call server API to authenticate
  - Store token
  - Show following option to logged in user
    - Delivery Orders
      - Show all updated orders
      - Filter for vendor and date
    - Parcels
      - Today's parcel summary
      - Ability to group on pincode

RBAC -Hint for next Assignment

Vendor

- Upload a Order Details file
- Only See his own orders
- Ability to get historical records
- Ability to
  - Cancel
  - ReSchedule

Driver

- See parcels assigned to his vehicle
- Parcels can be across Vendor
- Mark Attendance
- See his delivery report

Admin

- See every vendor order
- Get Daily Insight
- Ability to ONBOARD
  - Vendor
  - Driver
- Ability to create Route

Customer

- Ability to see his parcels
- Reschedule drop time
- Raise support ticket
- Request call back

1st Assignment for DevOps

Assignment – Automate EC2 Deployment

```
% java -version
openjdk version "21.0.2" 2024-01-16
OpenJDK Runtime Environment (build 21.0.2+13-58)
OpenJDK 64-Bit Server VM (build 21.0.2+13-58, mixed mode, sharing)
```

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  13.797 s
[INFO] Finished at: 2025-06-29T18:18:30+05:30
[INFO] ------------------------------------------------------------------------
```

1. **Sign UP for your own AWS free tier**
2. **Spins up an EC2 instance** of a specific type
3. **Installs dependencies** -java 21
4. **Install maven - for MVN**
5. **Clones repo & deploys app** from GitHub
   a. Github repo link -https://github.com/Trainings-TechEazy/test-repo-for-devops
   b. **Clone it and then Build with** "mvn clean package"
   c. **Tested with Java version** "openjdk version "21.0.2" 2024-01-16"
   d. **To run** java -jar target/hellomvc-0.0.1-SNAPSHOT.jar
6. **Tests if app is reachable** via port 80 -http://<EC2-public-IP>/hello should return "**Hello from Spring MVC!**"
7. **Stops the instance** after a set time (for cost saving)
8. **No secret or Access KEY to be added in REPO -these will be read from ENV**
9. Create script in a way where a "Stage" parameter will be passed, like "Dev", "Prod" and it should pick a config file accordingly, like dev_config, prod_config

Make instance type, dependencies, and repo configurable, use defaults if not available.

- **Rule of thumb, if you are confused what type or value should I use, it is a candidate for config file**

2nd Assignment for DevOps

Tools Allowed:

- CloudFormation or Terraform
- AWS CLI or Python (boto3)
- Linux shell script (Bash)

Extend your previous automation to:

1. Create two roles
    a. A role having read only access on S3
    b. A role having permission to create bucket, upload files to it -NO read or down
2. Attach role 1.b to EC2 via instance profile (`IamInstanceProfile` in Terraform or CFN).
3. **Create private S3 bucket** (name should be configurable; if not provided, terminate with error).
4. **Upload EC2 logs** (e.g., `/var/log/cloud-init.log`, or your custom setup logs) to the S3 bucket **after instance shutdown** for archival.
5. Upload logs of app deployed in last assignment to bucket /app/logs
6. Add **S3 lifecycle rule** to delete logs after 7 days.
7. Use role 1.a to verify that files can be listed

Workflow - Hint for next DevOps Assignment

- User signups for a SaaS application -for e.g. Zero mile delivery system
  - Same software can be used in different region
  - Each Customer will have its own VPC
  - EC2 inside VPC
  - Private S3 bucket where data never leaves a particular region -GDPR
- A UI to collect user preference
  - A workflow should be triggered
  - A notification is send to Admin to verify request
  - Admin can ask for cost estimation
  - Send estimation to client
  - Client approve estimation
  - Admin approves the request for infra creation
  - Creates infra based on user preference
- Handle
  - Success scenario
  - Failed Scenario
- Components
  - VPC, Cost estimator, SNS, SQS

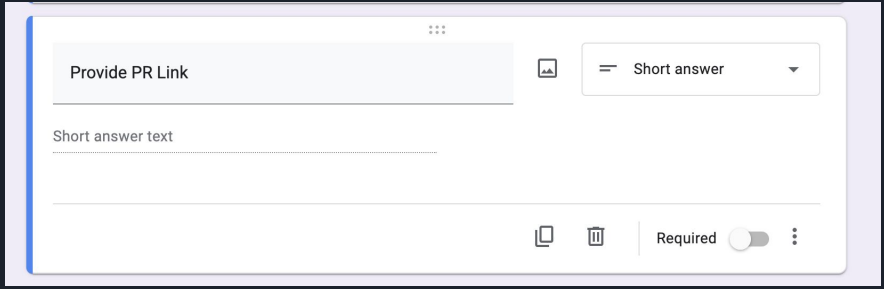
Create Your PR -after initial repo

Create PR

- git checkout -b feature/my-change
- Edit, add, or delete files as needed
- git add .
- git commit -m "Add my feature or fix"
- git push origin feature/my-change

Create Pull Request on GitHub

- Go to the GitHub repo.
- You'll see a prompt: **"Compare & pull request"** → Click it.
- Or go to **Pull Requests** tab → Click **New Pull Request**.
- Choose:
  - **base branch** = usually `main`
  - **compare branch** = `feature/my-change`
- Add a title and description.
- Click **"Create Pull Request"**.



Update:

- Postman Collection in `resources/`
- `README.md` with instructions

Send invite and add following as collaborators:

- Shivyandralwar2019@gmail.com
- dharesh.a.p@gmail.com
- cpandey05@gmail.com
- siddpawar583@gmail.com
- trainings.techeazyconsulting@gmail.com

Add PR link `in this form`
https://forms.gle/trPpGbr2Vudz8feWA

Timelines

	Type	Date & Time	Agenda
Orientation 1	Orientation	Sat, 19th July, 6.30 PM	Understand unique nature of this Internship
Orientation 2	Orientation	Mon, 21st July, 6.30 PM	Understand unique nature of this Internship
1st Scrum	Guidance Session -open to all	Mon, 21st July, 7.00 PM	• Discuss assignment • Open Session -QnA, guidance
2nd Scrum	Guidance Session -open to all	Tue, 22nd July, 7.00 PM	• **What is PR, how to raise PR** • Discuss final part of assignment
3rd Scrum	Guidance Session -open to all	Wed, 23rd July, 7.00 PM	• Open Session -QnA, Doubt clearance, guidance
Final Assignment Submission Date	Deadline	Fri, 25th July, 9.00 PM	Final assignment, PR submission link closes
Scrum Team For selected candidates	Internship Scrum -Selected Interns	Mon, 28th July, 6.00 PM	**Batch will be allotted** -No Session, meeting invite will be sent to selected candidates
Internship Start Date		**Sat, 6th Aug**	**Internships starts**, those who do not submit PR or do not participate will be removed.
Internship End Date		**Mon, 5th Sep**	**Problem discussion session will be open to all every week -you can complete the project without internship certificate**