Deployment Plan – Movie Streaming Platform

1. Introduction

1.1 Purpose

This document outlines the deployment strategy for the Movie Streaming Platform, ensuring a smooth transition from development to production while maintaining reliability, security, and scalability.

1.2 Scope

The deployment plan covers:

- Release process
- Environments
- CI/CD pipelines
- Rollback strategy
- Monitoring and security measures

1.3 Target Audience

- DevOps Engineers
- System Administrators
- Developers
- QA Engineers

2. Deployment Architecture

2.1 Infrastructure Overview

- **Cloud Provider**: AWS / Google Cloud / Azure
- **Compute Services**: EC2 / Kubernetes / Docker Containers
- **Database**: MySQL (Managed Service)
- **Storage**: S3 for static assets, EFS for persistent data
- **Caching**: Redis / Memcached
- **Load Balancing**: Elastic Load Balancer (ELB) / Nginx / Cloudflare
- **Monitoring & Logging**: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana)

2.2 Deployment Environments

Environment	Purpose	Hosting Platform
Development	Ongoing development/testing	Local / Cloud Dev Instance
Staging	Pre-production testing	Cloud Staging Instance
Production	Live system for end users	Cloud Production Instance

3. Deployment Process

3.1 Continuous Integration & Deployment (CI/CD)

Tools Used: GitHub Actions / Jenkins / GitLab CI

Steps:

1. Code is pushed to repository (GitHub / GitLab / Bitbucket).

2. Automated tests are executed.
3. Code is built into Docker images.
4. Artifacts are stored in container registry.
5. Deployment to staging environment for final verification.
6. Manual or automated approval for production release.
7. Deployment to production using blue-green or rolling update strategy.

3.2 Deployment Strategies

Strategy	Description
Blue-Green	Two identical environments; traffic is switched to the new version after verification.
Rolling Update	Gradually deploys new versions, ensuring zero downtime.
Canary Release	Deploys to a small percentage of users first before full rollout.

3.3 Rollback Strategy

- **Database Backups**: Automatic backups before deployment.
- **Feature Flags**: Toggle features off if a failure occurs.
- **Versioned Deployments**: Ability to revert to the last stable version.
- **Monitoring Alerts**: Immediate alerts in case of failure.

4. Monitoring & Logging

4.1 Monitoring Tools

- **Application Performance**: New Relic / Datadog
- **Infrastructure Monitoring**: Prometheus / Grafana
- **Error Tracking**: Sentry / ELK Stack

4.2 Logging Framework

- **Centralized Logging**: Logstash & Kibana for log aggregation.
- **Retention Policy**: Logs retained for 90 days.

5. Security Considerations

5.1 Authentication & Authorization

- JWT-based authentication.
- Role-based access control (RBAC).

5.2 Data Protection

- Encrypted data at rest and in transit (AES-256, TLS 1.2+).
- Regular security audits and vulnerability scanning.

5.3 DDoS Protection

- Cloudflare / AWS Shield for mitigation.
- Rate limiting on API endpoints.

6. Backup & Disaster Recovery Plan

6.1 Backup Strategy

- Database Backups: Daily full backups, hourly incremental backups.
- File Storage Backups: Version-controlled backups of static assets.

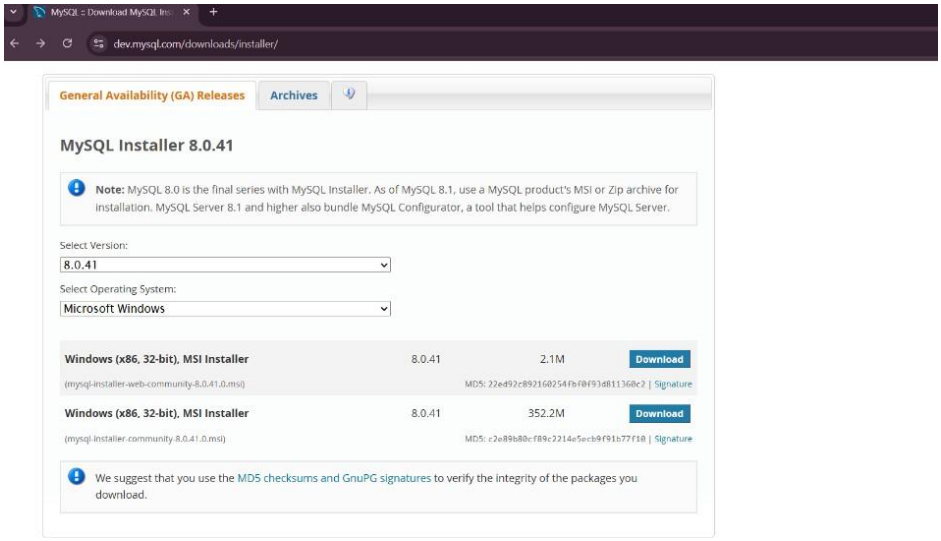
6.2 Disaster Recovery Plan

- Failover Strategy: Multi-region deployment for high availability.
- Recovery Time Objective (RTO): Less than 15 minutes.
- Recovery Point Objective (RPO): Less than 5 minutes.

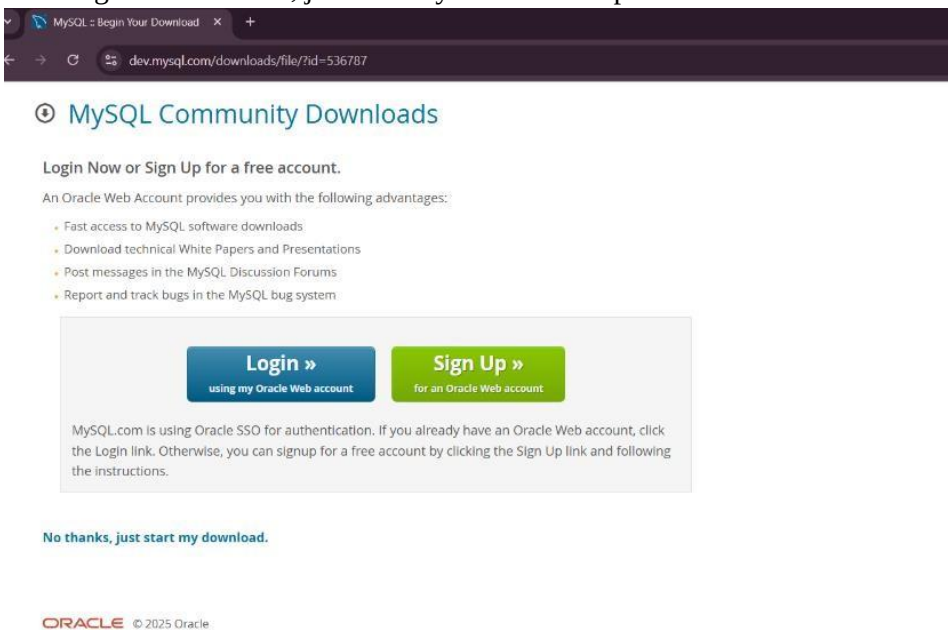
7. MySQL Installation

1. Download MySQL

- Go to the official : https://dev.mysql.com/downloads/installer/
- Choose the MySQL version and installer file that is compatible with your operating system to start the download.



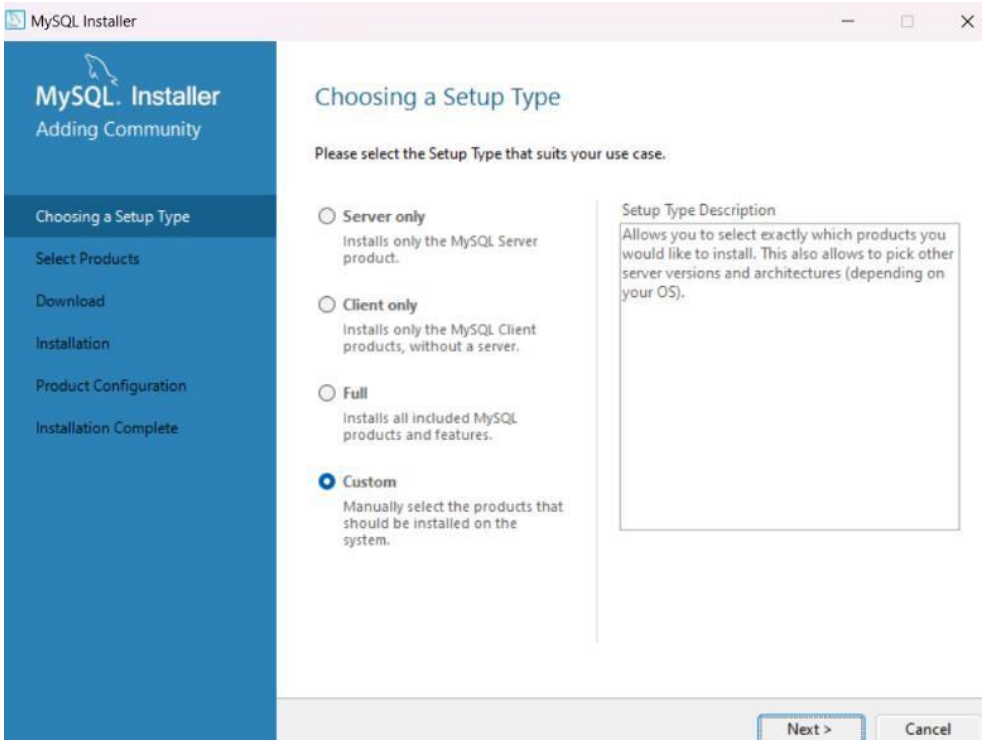
2. To download MySQL, you can either sign in with an Oracle account or proceed directly by selecting the "No thanks, just start my download." option.



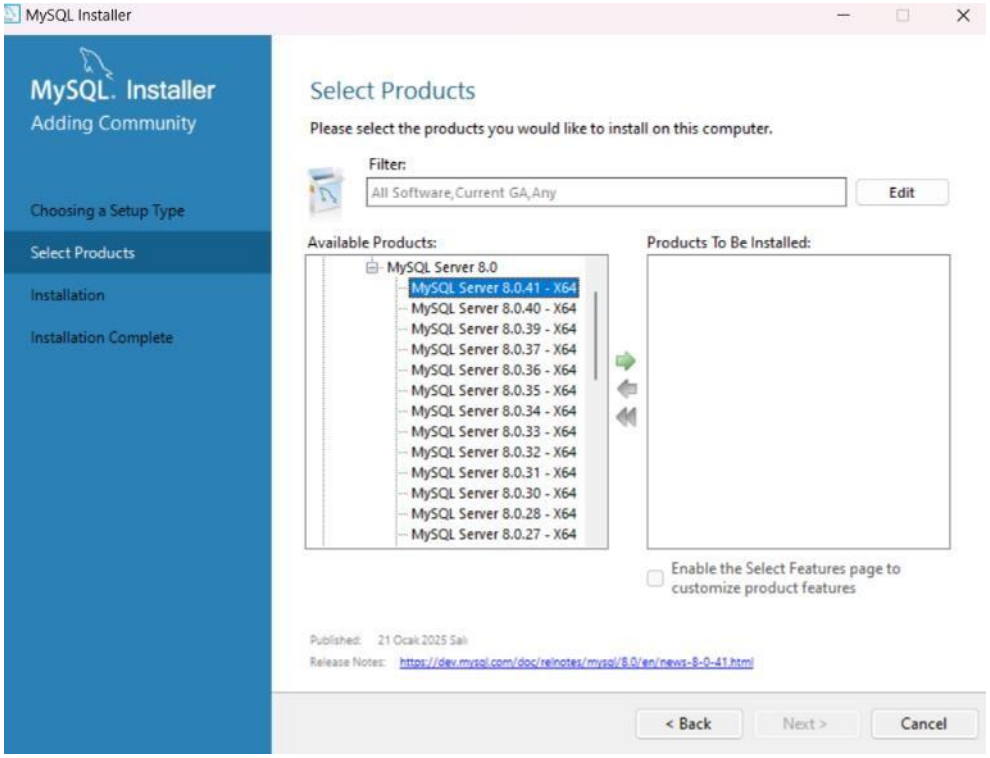
3. Install MySQL

Select the most suitable installation type for your needs to proceed with the MySQL installation.

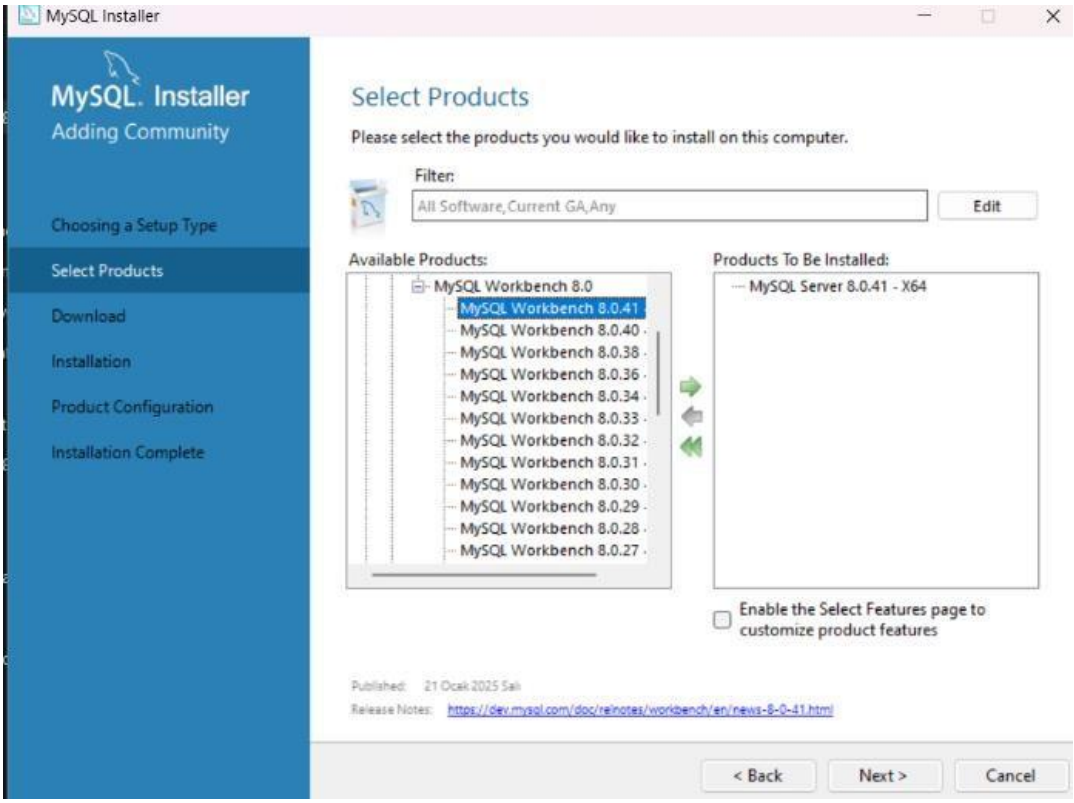
- *Developer Default* → Installs all components.
- *Custom* → Allows you to choose specific components (MySQL Server and Workbench are recommended).



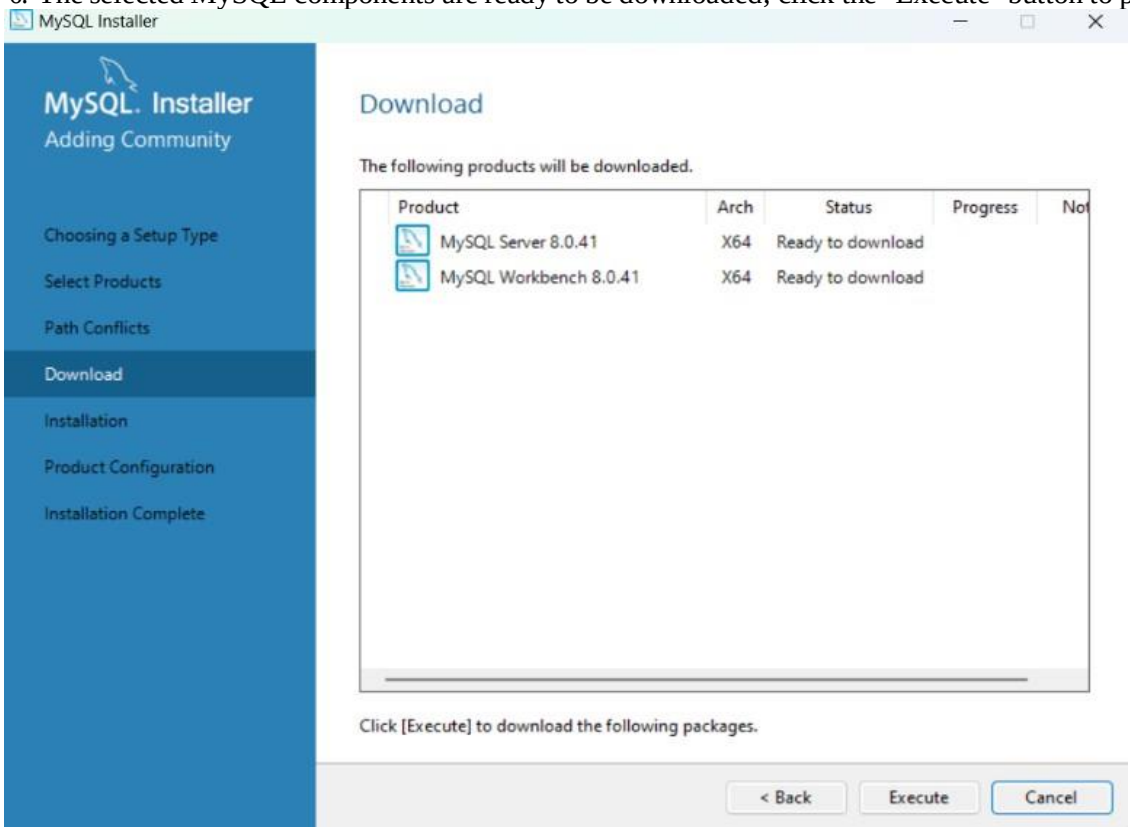
4. Choose the MySQL version to be installed and continue with the installation process.



5. Select MySQL Workbench to configure additional components during the installation.

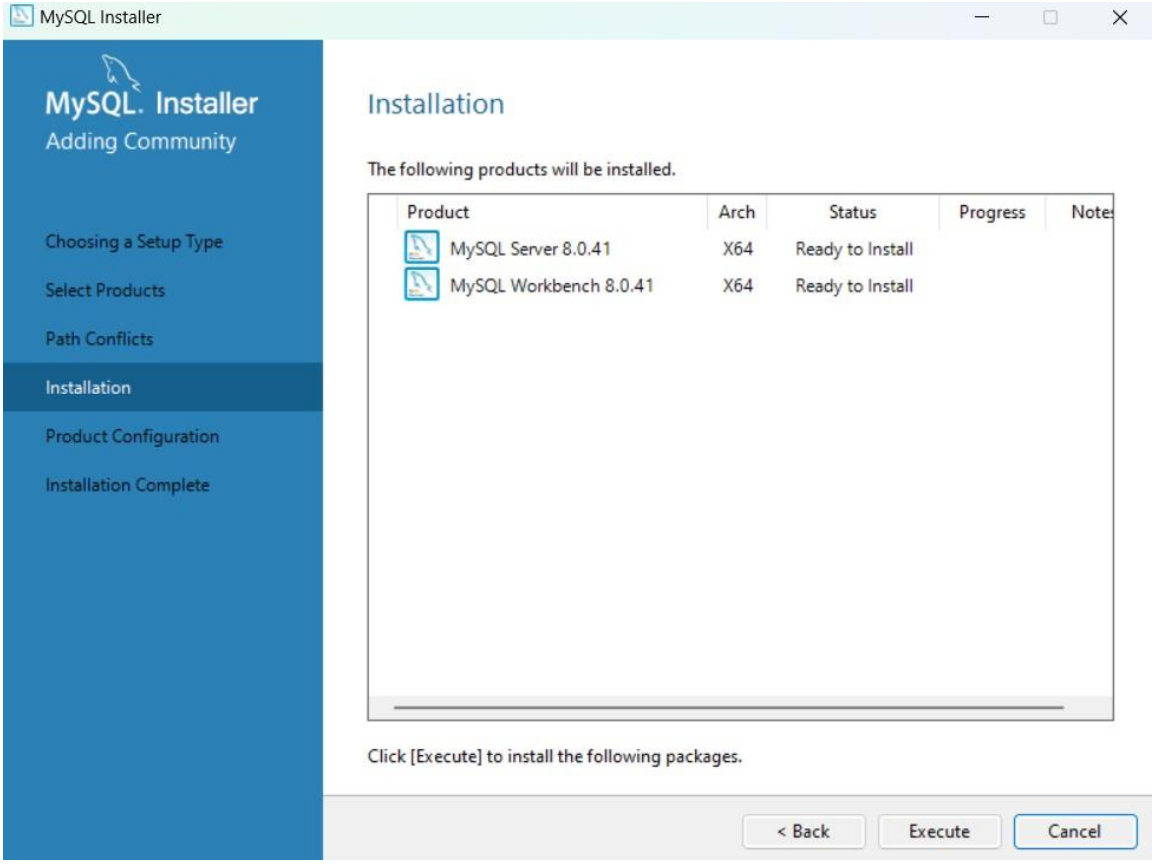


6. The selected MySQL components are ready to be downloaded; click the "Execute" button to proceed.

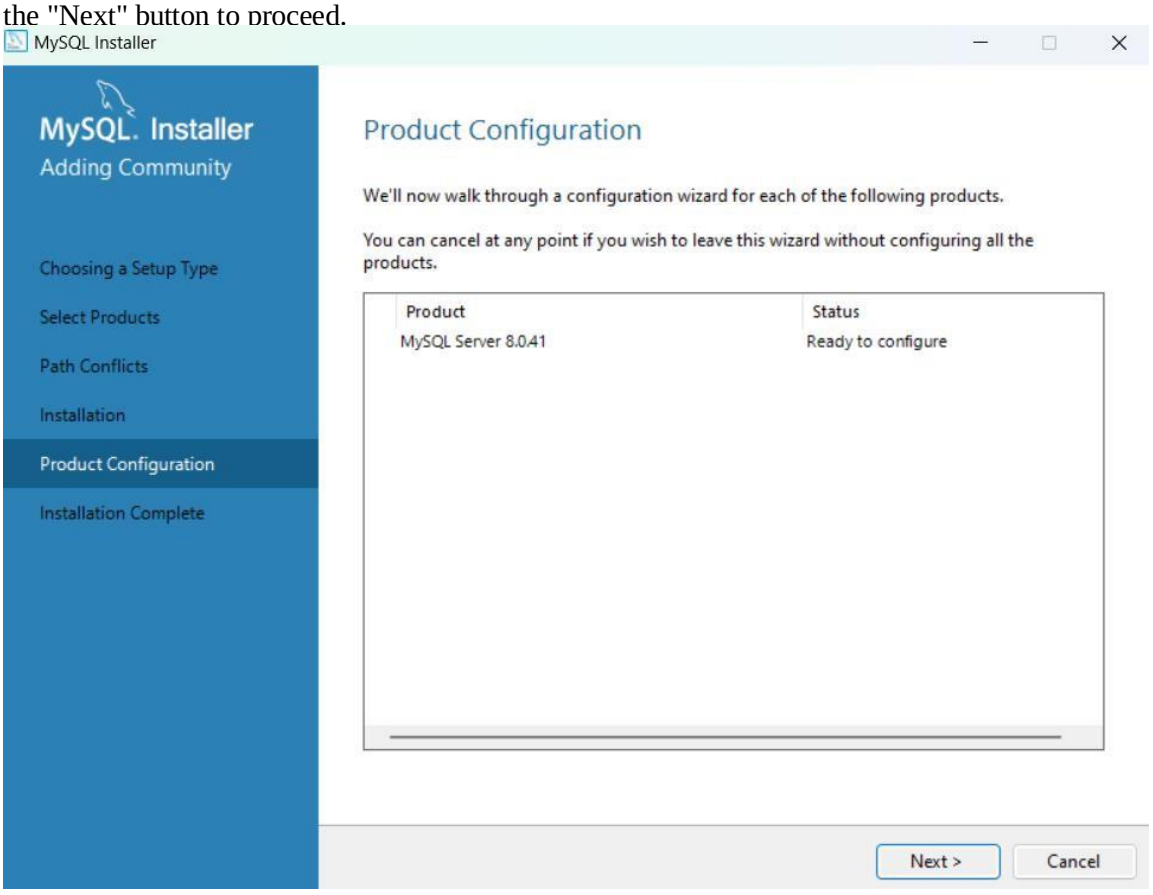


.

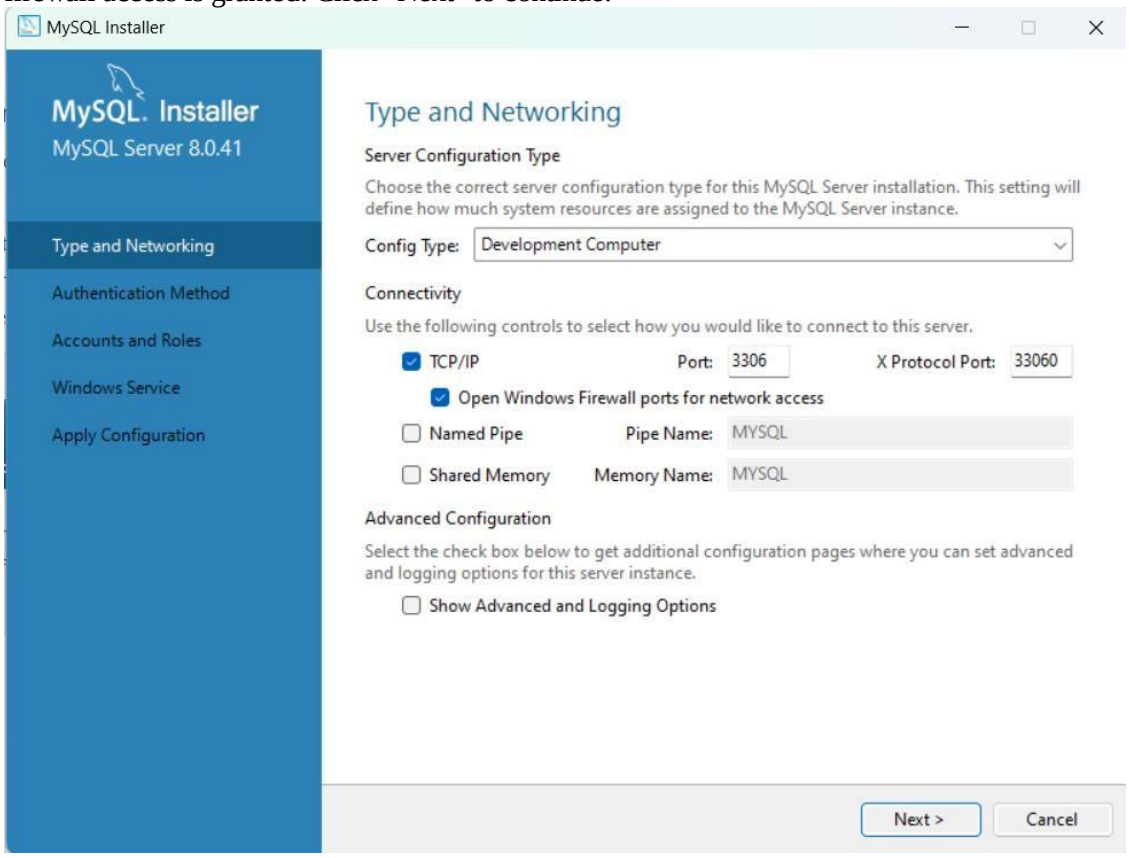
7. The download is complete, and MySQL Server and Workbench are ready for installation. Click the "Execute" button to start the installation



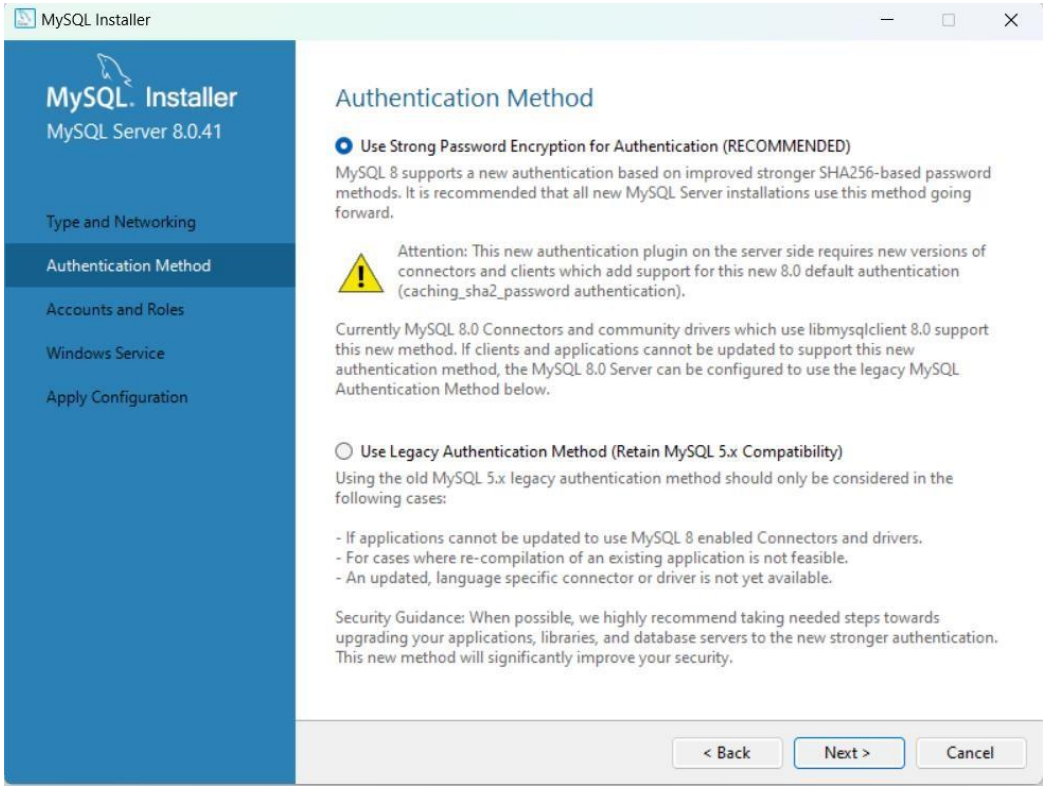
8. The installation is complete; now, the configuration process for MySQL Server 8.0.41 can begin. Click the "Next" button to proceed.



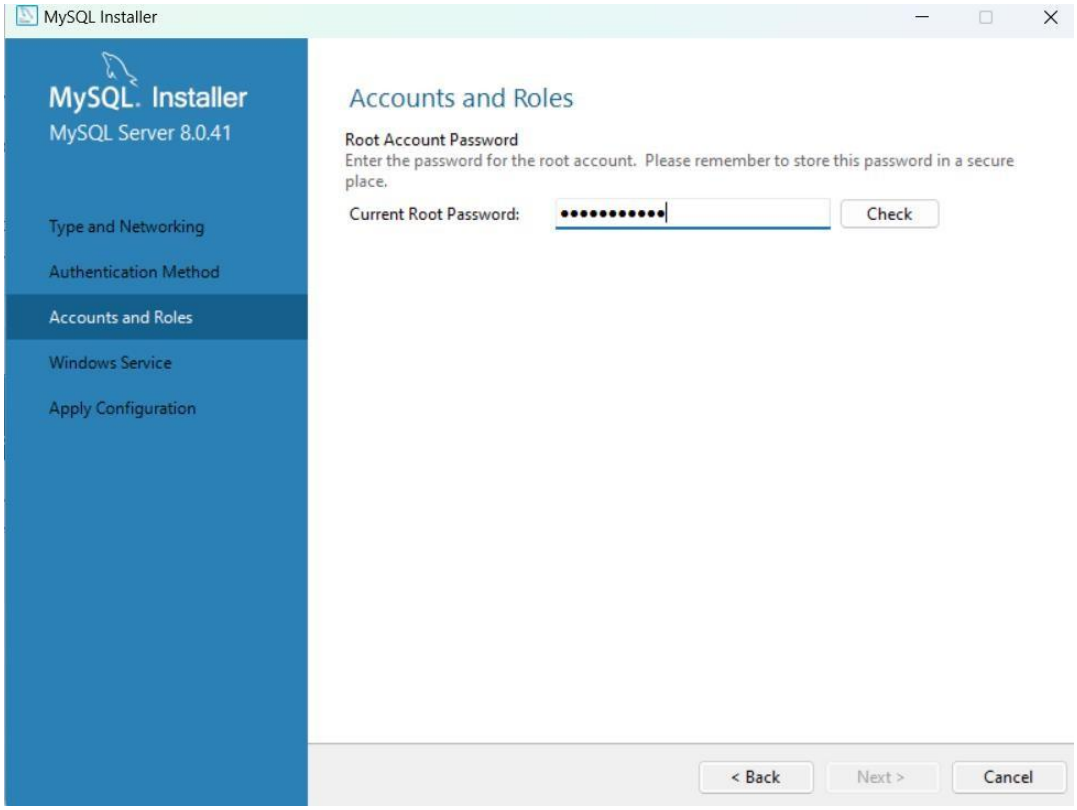
9. The MySQL Server configuration type is selected as "Development Computer." The default port is set to 3306, and the X Protocol Port is set to 33060. TCP/IP connection is enabled, and firewall access is granted. Click "Next" to continue.



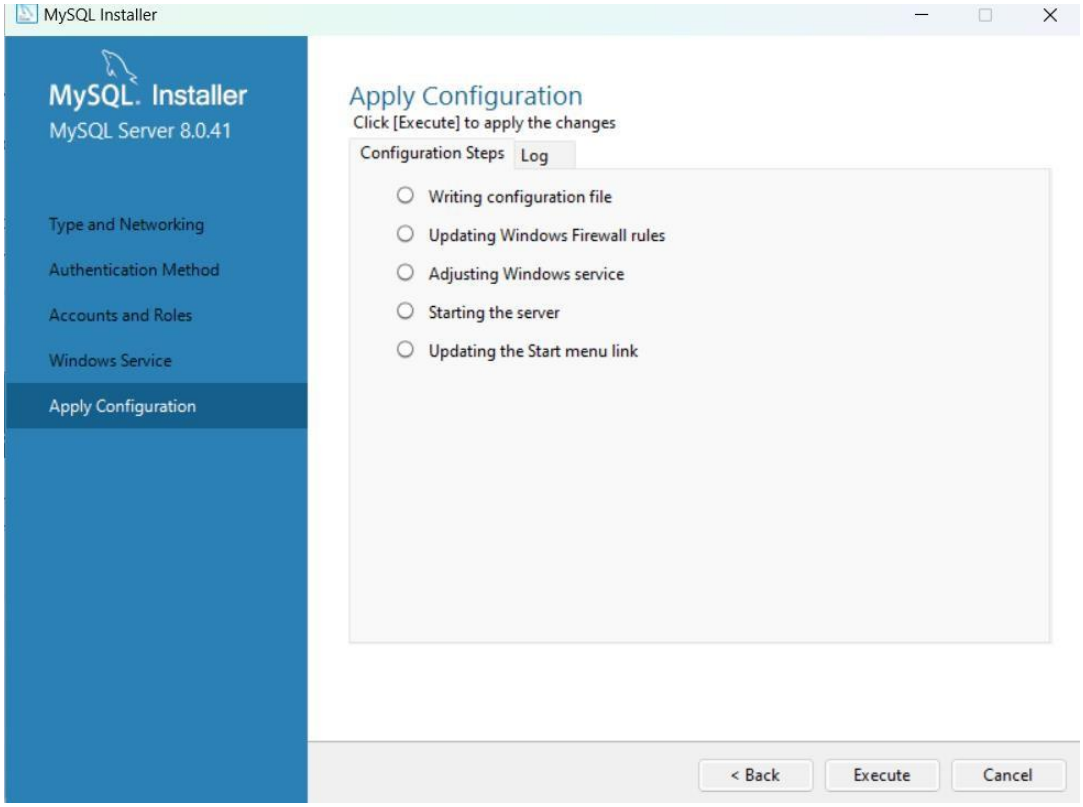
10. The authentication method selection screen for MySQL appears. Strong password encryption (SHA256-based) is recommended. If compatibility with older systems is needed, the "Legacy Authentication" option can be chosen. Click "Next" to proceed.



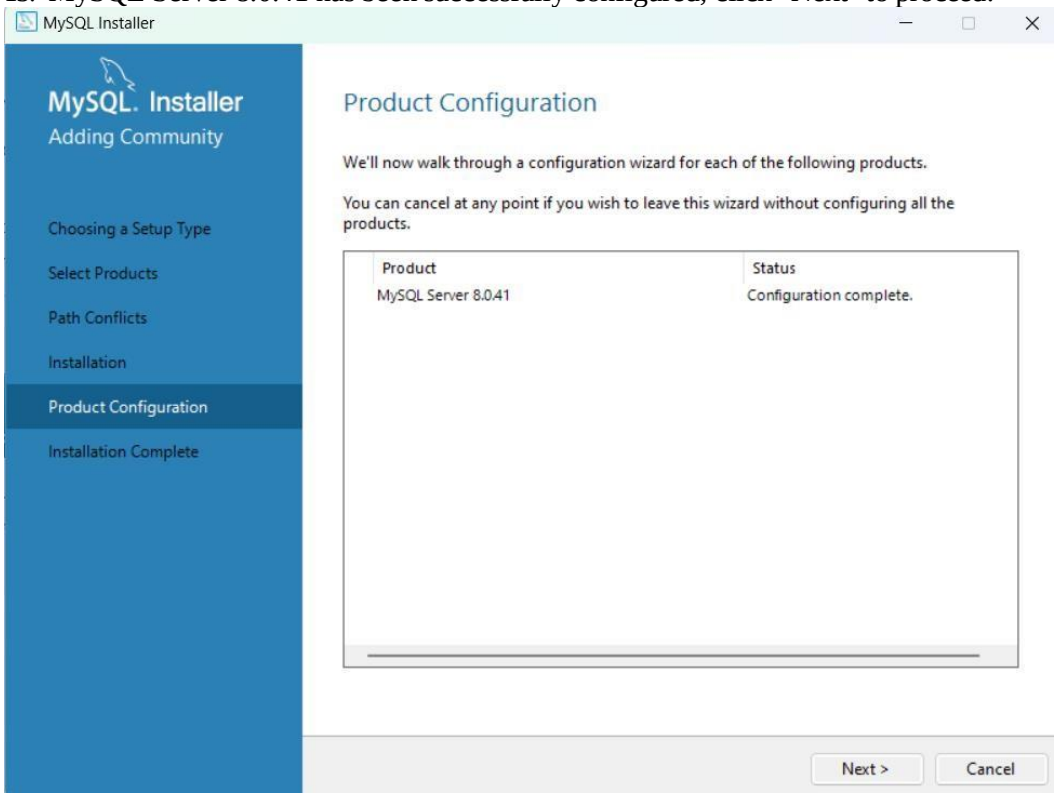
11. This is the root account password entry screen. The selected password should be stored securely, as the root account has full privileges on MySQL.



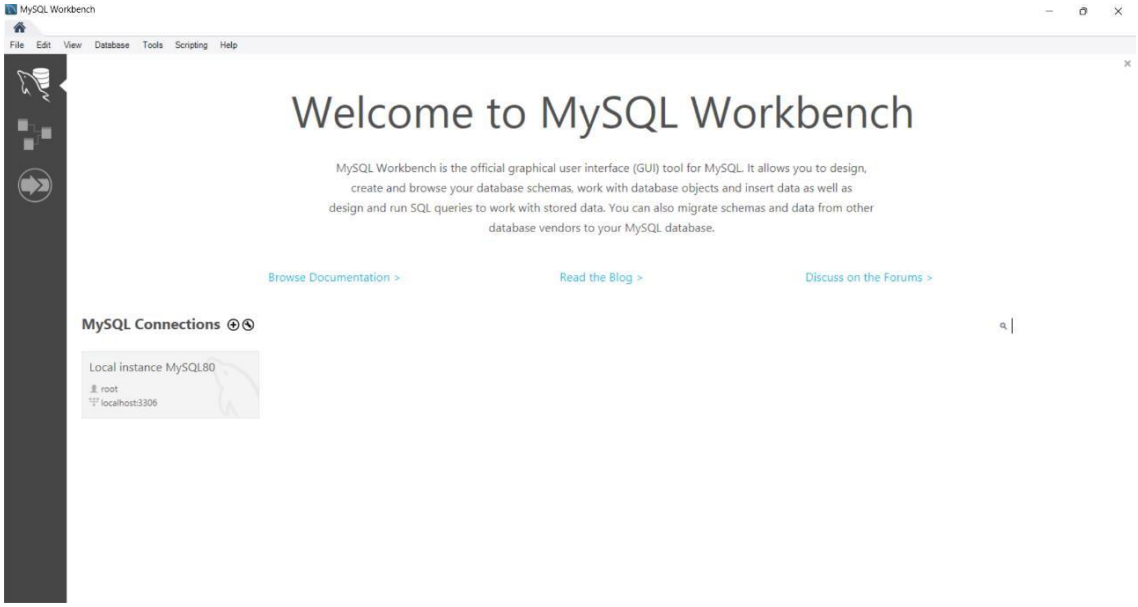
12. Click the "Execute" button to apply the changes and complete the MySQL configuration.



13. MySQL Server 8.0.41 has been successfully configured; click "Next" to proceed.



14. MySQL Workbench has been successfully launched. Click on the "Local instance MySQL80" connection to connect to your database.



8. Conclusion

This deployment plan provides a structured approach for the secure, scalable, and uninterrupted deployment of the Movie Streaming Platform**.** With robust CI/CD pipelines, monitoring systems, security measures, and rollback strategies**,** th**e** platform ensures high performance and reliability.