





2025

PROJECT REPORT

SecureCloud Scheduler

Cloud Computing

Karan Dubey

Intern ID: IP-6581

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Karan Dubey | **Approved By:** | Harshada Topale |
| **Created On:** | 22-02-2025 | **Approved On:** | 24-04-2025 |

Page left blank intentionally

**INDEX**

1. [PROJECT DETAILS 2](#_bookmark0)
2. [SUMMARY 2](#_bookmark1)
3. [INTRODUCTION 2](#_bookmark2)
   1. [Background 2](#_bookmark3)
   2. [Stakeholders 2](#_bookmark4)
   3. [Objectives 3](#_bookmark5)
4. [METHODOLOGY 3](#_bookmark6)
   1. [Considerations & Assumption 3](#_bookmark7)
   2. [Approach 3](#_bookmark8)
   3. [Activities 4](#_bookmark9)
5. [TARGETTED V/S ACHIEVED OUTPUT 4](#_bookmark10)
6. [CONCLUSION 5](#_bookmark11)
7. [APPENDICES 6](#_bookmark12)
   1. [Appendix A – Title 6](#_bookmark13)

# PROJECT DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | SecureCloud Scheduler | | |
| **Project Sponsor** | Tushar Topale | | |
| **Project Manager** | Harshada Topale | | |
| **Start Date** | 22/02/2025 | **Completion Date** | 24/04/2025 |

# SUMMARY

This project focused on deploying the open-source scheduling application, Cal.com, on an Azure Linux Virtual Machine using Docker Compose. The solution ensures scalability and security by leveraging Azure resources such as Virtual Machine Scale Sets (VMSS), Network Security Groups (NSGs), and Caddy for HTTPS. The project also incorporated monitoring with Azure Alerts. This deployment will streamline scheduling and enable remote operations. The entire process has been documented for ease of future maintenance.

# INTRODUCTION

## Background

Mr. Tushar, the founder of Cloud Counselage Pvt. Ltd., is facing challenges managing his busy schedule with multiple meetings scheduled on different platforms. He has observed that his team and colleagues also encounter similar issues. After conducting thorough research, he identified an open- source calendar management project as a potential solution to this problem. To ensure confidentiality and usability, Mr. Tushar plans to host this calendar manager on a cloud platform tailored for his startup's needs.

## Stakeholders

Primary Stakeholder:

* + - Harshada Topale: The main individual overseeing and benefiting from the project's implementation.

Other Stakeholders:

* + - Mr. Tushar: A key user and contributor to the overall objectives.
    - Mr. Tushar's Team: Users of the calendar system.

## Objectives

1. Deploy the Cal.com application on an Azure Linux VM using Docker Compose to ensure portability and simplified management of containerized workloads.
2. Configure Caddy for HTTPS to secure access to the application and ensure data transmission confidentiality.
3. Utilize VM Scale Sets to dynamically scale resources based on user demand, ensuring cost efficiency and high performance.
4. Implement Azure Alerts to proactively monitor the application’s

uptime, resource utilization, and overall health.

1. Provide thorough documentation for knowledge transfer and efficient future operations.

# METHODOLOGY

## Considerations & Assumption

* + - **Continuous Availability:** The application must maintain 99.9% uptime to cater to users across different time zones.
    - **Security:** HTTPS is non-negotiable for securing sensitive scheduling data.
    - **Scalability:** Azure VM Scale Sets were assumed to handle increased load dynamically without requiring significant manual intervention.
    - **Portability:** Docker Compose enables consistent deployment across environments.

## Approach

A systematic and iterative approach was implemented:

1. **Requirement Analysis:** Identified the functional requirements, including scalability, security, and accessibility. Prepared a checklist of dependencies such as Docker, Caddy, and Azure CLI.
2. **Infrastructure Setup:** Created and configured Azure resources tailored to the project requirements, such as VM Scale Sets for scaling and NSGs for security.
3. **Application Deployment:** Cloned the Cal.com source code from its GitHub repository. Configured the environment variables for application setup and executed deployment using Docker Compose.
4. **Testing:** Conducted rigorous tests, including load testing, functional validation, and security audits, to ensure the deployment met all quality benchmarks.
5. **Documentation:** Detailed the installation, configuration, and troubleshooting steps in a structured deployment guide.

## Activities

### Initial Azure Setup:

* + Provisioned Linux Virtual Machines with optimized hardware configurations to support Docker workloads.
  + Configured Virtual Networks and Subnets for secure communication between resources.
  + Implemented VM Scale Sets for autoscaling capabilities.

### Security Configuration:

* + Established Network Security Groups to enforce role-based access control and prevent unauthorized access.
  + Configured firewall rules to allow HTTPS traffic while blocking unnecessary ports.

### Application Deployment:

* + Installed essential software, including Docker Engine and Docker Compose.
  + Pulled the source code using the command:
  + git clone https://github.com/calcom/cal.com.git
  + Configured environment variables such as database connections, API keys, and ports.
  + Deployed services using a docker-compose.yml file, which

defined the application’s multi-container architecture.

### SSL Integration:

* + Set up Caddy as a reverse proxy to manage SSL/TLS certificates and automate HTTPS configuration.

### Monitoring and Alerts:

* + Enabled Azure Alerts to notify the team about critical issues, including high CPU/memory utilization or downtime.
  + Set up log streaming and dashboard visualizations for continuous performance monitoring.

# TARGETTED V/S ACHIEVED OUTPUT

### Targeted Output

The project aimed to deliver the following outputs:

* + A secure, self-hosted scheduling application accessible via a custom HTTPS domain.
  + A scalable architecture capable of adapting to changing user demands.
  + Real-time monitoring and alerting mechanisms for efficient issue resolution.

### Achieved Output

* + **Scalability:** The deployed infrastructure automatically adjusts to changes in user traffic.
  + **Security:** HTTPS encryption is enabled using automated SSL certificates via Caddy.
  + **Accessibility:** The application is live and accessible 24/7 via a secure URL.
  + **Monitoring:** Alerts and monitoring dashboards are fully operational.

# CONCLUSION

* The deployment of Cal.com on an Azure Linux Virtual Machine using Docker Compose successfully addresses the scheduling and calendar management challenges faced by the stakeholders, particularly Harshada Topale. By leveraging secure communication via HTTPS (Caddy) and scalable Azure resources like Virtual Machine Scale Sets (VMSS), the project delivers a robust and efficient solution.
* The documentation of the deployment process ensures maintainability and facilitates knowledge transfer, while the integration of Azure Alerts enables proactive monitoring. This project demonstrates the effective use of cloud technologies to implement a practical, user-friendly, and secure open-source solution.
* The successful implementation highlights the potential for further enhancements, such as integrating CI/CD pipelines or Kubernetes, to future-proof the solution for evolving needs. Overall, the project meets the stakeholders' expectations and sets a strong foundation for efficient calendar management.

# APPENDICES

## Appendix A – Architecture Diagram

An architecture diagram is attached to illustrate the deployment’s key components, including Azure Virtual Machines, VM Scale Sets, Docker Containers, and monitoring tools.

