

Case Study

Building a Scalable and Future-Proof Coding Platform for K-12

Magic
EdTech

The Client

The client is an edtech company specializing in teaching K-12 students coding, computer science, and STEM. They offer standards-based curricula in computer science, esports, and STEM, combining critical thinking with engaging, project-based learning.

The Challenge

The client faced the challenge of building a highly scalable platform capable of handling thousands of concurrent users with minimal latency. The platform needed to ensure high availability to accommodate spikes in user demand without service interruptions.

Critical Success Parameters

- ✓ Build a highly scalable environment capable of handling thousands of concurrent users with minimal latency.
- ✓ Ensure flexibility in the platform architecture to easily support additional programming languages beyond Python, such as C# and Java.
- ✓ Seamlessly integrate the backend API, frontend interface, and execution environment to provide a cohesive user experience.
- ✓ Implement robust security measures and ensure compliance with best practices to protect the shell access environment.
- ✓ Guarantee high platform availability to accommodate user demand spikes without service disruptions.

Our Approach

- ✓ Built a containerized execution environment to securely run Python code and efficiently handle high loads, with scalability for future language support.
- ✓ Developed an API using Node.js to connect the ReactJS frontend with the backend execution environment.
- ✓ Provisioned an infrastructure using Terraform, ensuring high availability, security, and ease of management.
- ✓ Established four separate environments (Dev, CI, UAT/Pre-Prod, Prod) using Terraform to enable smooth deployment and large-scale testing.
- ✓ Provided secure shell access to users, allowing interaction with the cloud server's working directory for a realistic coding experience.
- ✓ Implemented a decoupled architecture, separating Python-specific logic from the interface, enabling easy integration of new languages.
- ✓ Integrated Datadog for real-time monitoring of application performance, security, and usage to maintain platform stability under heavy load.



Key Result Highlights

Achieved **50%** faster deployment times through automated CI/CD pipelines, enabling frequent updates.

Increased resource utilization by **80%** with containerization and Kubernetes, ensuring scalability without added costs.

Reduced operational overhead by **30%** automating infrastructure provisioning and monitoring.

Maintained **99.9%** uptime, even during peak loads, ensuring high reliability and user satisfaction.

Future-proof architecture enabled seamless integration of new programming languages, boosting development agility by **30%**.