

Abstract

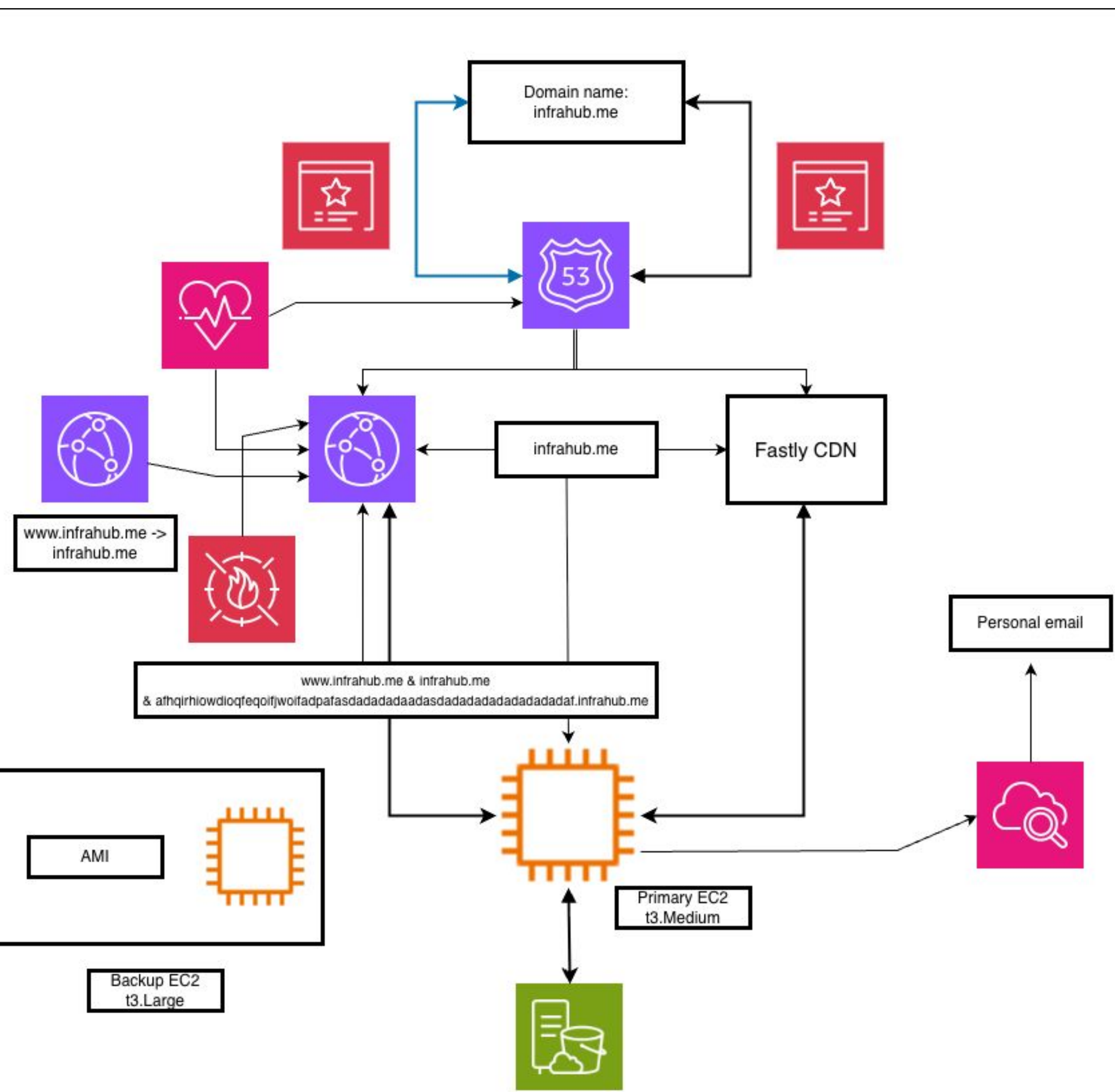
InfraHub is an IT automation platform designed to streamline infrastructure setup and deployment for organizations and IT teams. Our platform consolidates preconfigured OS installers, department-specific approved Docker scripts into a single centralized hub, enabling teams to deploy secure, reliable environments quickly and efficiently. InfraHub helps companies reduce setup time, enforce best practices, and ensure consistent, ready-to-use systems.

Objectives

- Provide fully reproducible development environments.
- Reduce setup and onboarding time.
- Automate Docker installation using Ansible (Ubuntu)
- Enable users to generate compose files from presets.
- Support Python, Node, and full stack sample environments.
- Deliver clear documentation for operators and maintainers.

Methods

- Bash Automation (start.sh and generate-compose.sh).
- Dynamic Docker Compose generation using YAML presets.
- Dockerfile-based environment packaging.
- Optional Ansible playbooks for automated installation and configuration of workstations.
- Testing across Ubuntu, macOS, and Windows using Docker Desktop.
- Validation containers using docker compose build, up -d, and log inspection.



Team Roles / Contributions

Bryan Garcia - Poster / Documentations / Research
Tom Mizarachi - Frontend Developer
Andres Jaimes - Backend Developer / Research
Cesar Pereyra - Automation & Scripts Engineer
Bryan Hernandez - Poster / Team Lead / Presentation
Instructor: S. Masoud Sadjadi

Results

- Functional interactive launcher that builds and runs selected environments.
- Stable operational performance across supported OS platforms.
- Verified Python and Node sample applications running on localhost.
- Successful automated Docker installation via Ansible on Ubuntu.
- Portability and reproducibility validated using Composer presets.
- Simplified developer onboarding with “one command start” workflow.

Conclusion

InfraHub demonstrates that reproducible development environments can be package and deployed easily using Docker and simple automation tools. The toolkit ensures consistency across machines, minimizes setup errors, and provides an extensible foundation for teaching, development, and DevOps-aligned workflows. The project succeeds in delivering a modular, scalable, and user-friendly automation solution.

References

AWS:

- <https://docs.aws.amazon.com/>
- <https://docs.aws.amazon.com/s3/>
- <https://docs.aws.amazon.com/vpc/>
- <https://docs.aws.amazon.com/cloudfront/>
- <https://docs.aws.amazon.com/network-firewall/latest/developerguide/tls-inspection-configurations.html>
- <https://docs.aws.amazon.com/ec2/>
- <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>
- <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cloudfront-functions.html>

Fastly:

- <https://www.fastly.com/documentation/guides/>

Youtube:

- <https://youtu.be/eaicwmnSdCs?si=M8GWGH2er9tVA2GH>
- https://youtu.be/9t9Mp0BGnyI?si=qKZyFQqW_YDnRzO8

Books:

- Amazon Web Services in Action
- AWS Certified Solutions Architect Study Guide

Other:

- docs.ansible.com
- <https://git-scm.com/book/en/v2/Getting-Started>About-Version-Control>

