GitHub FedRAMP Promotion

Dev Patel - 19IT092,
Student, B Tech - IT,
KDPIT, CHARUSAT

Hemant N. Yadav,
Assistant Professor,
KDPIT, CHARUSAT

Abstract

Compliance with rules and regulations that pertain to using the cloud is referred to as cloud compliance. The main idea behind cloud compliance is that any systems that are offered over the cloud must adhere to the same standards as the cloud clients. Many IT experts pay special attention to this since it is a crucial issue with new cloud computing services. Accessibility is provided by cloud installations, but they also provide open, decentralised networks that are more vulnerable. Frameworks for cloud compliance play a role in this. You may reduce the risks associated with using SaaS and third-party cloud infrastructure by aligning your data security policies and processes with cloud compliance frameworks.

Introduction

The Federal Risk and Authorization Management Program (FedRAMP) is a US government-wide program that delivers a standard approach to the security assessment, authorization, and continuous monitoring for cloud products and services. The governing bodies of FedRAMP include the Office of Management and Budget (OMB), US General Services Administration (GSA), US Department of Homeland Security (DHS), US Department of Defence (DoD), National Institutes of Standards & Technology (NIST), and the Federal Chief Information Officers (CIO) Council [1].

All cloud providers and US government agencies must comply with FedRAMP. FedRAMP is significant because it:

- When adopting standards set by the National Institutes of Standards & Technology (NIST) and FISMA, there is consistency and confidence in the security of cloud solutions.
- Transparency between the cloud service companies and US government.
- Automation and constant monitoring in close to real time.
- Utilizing evaluations and authorizations to adopt safe cloud solutions.

At Motorola Solutions, CIE projects that needs to be deployed have to comply with FedRAMP. And to review and monitor all the projects a separate environment is created where the final project is migrated and then deployed there after proper reviewing. Apart from that all the employees have a quarterly access review for the services that they have access to. For the employees with access to Federal Environment, a monthly access review is carried out by security compliance team.

Problem Statement

Only few employees have the access to Federal Environment who are mostly the reviewers and compliance team members.

So in order to allow the partner teams to migrate their projects to Federal Environment a provision has to be there where they can migrate their project to the federal environment by themselves. But along with this provision the code needs to be reviewed before deployment. So only migration needs to be done by the partners and then after a proper review they can be allowed to deploy the project.

The logs of the deployment pipeline runs should be saved and posted to corresponding endpoint.

Project Details

A pipeline task is created which the partner teams have to add to their pipeline along with their repository details, branch, organization, the deployment pipeline and the list if files that needs to be promoted.

The pipeline task triggers a REST request to the backend endpoint where the validation happens of input provided by the partners, contents of the files provided by the partners are fetched, a repository is created in the Federal Environment if not already present, a branch is created where all code gets pushed and commit is created.

For the review, a Pull Request is created and the reviews list is added to that who needs to approve the Pull Request only after which the code gets merged with the main branch. The deployment is limited to only main branch so the code has to be merged with main branch which can only be done by a Pull Request.

In order to maintain a consistency for the deployment code, partners provide their deployment pipeline with the pipeline task and we add some wrapper files which references the deployment pipeline provided by the partner. According to the type of deployment specific wrapper files needs to be triggered and according to that the logs are generated and stored by the backend service.

These wrapper pipelines also post the deployment pipeline logs to save them to their respective project’s endpoint. These logs are essential to debug on failure of the deployment pipeline.

Commits history is also fetched and added as comments to the Pull Request for the reference of the Security compliance team.

Technology Used

- Node.js is used for the development of the backend service.
- Azure DevOps Node API and GitHub API are used for carrying out the repository related tasks.
- Pipeline tasks are created for both Azure DevOps pipeline and GitHub workflows.
- For Azure promotion task is created as an Azure Marketplace Extension which can only be accessed by Motorola Solutions employee.
- For GitHub promotion task is created as a shareable action, which can be used in any repository inside Motorola Solutions organization repositories.

- Docker and Kubernetes are used for deployment of the backend service.

- Mocha, Chai and Sinon are used for writing unit tests and integration tests.

Use Case

- Partners can use this task for migrating their project and a repository of the same name is created if migration is done for the first time.

- If there are any changes or bug fixes done for a project than only specific files that are changed can be promoted.

- For the deployment pipeline if the partners don't provide any files for migration than the deployment pipeline is scanned and all the referenced files are also migrated. This is done recursively so any file that is referenced inside another file is migrated.

- The promotion task supports directories so that partners don't need to add the list of individual files.

- As promotion task is the only way for partner teams to communicate with the federal environment, they can also delete the already promoted files from federal environment using the promotion task.

Features

- Promotion of files and directories.

- Deletion of already promoted files and directories in the federal environments.

- Commits history is available to the reviews as comments in the Pull Request for ease in reviewing for the reviewers.

- Pipelines only run after an approval of the reviewers.

- Partners can also promote files from another repository by using a specified format consisting of project name, repository name and the file/directory path. These files are saved inside a folder with the repository name.

Conclusion

A pipeline task is created for the partner teams which triggers a backend service which fulfils the requirements of the partners making their deployment easier.

- Through this project I learnt about the best practices to be followed while developing any software.

- This project taught why DevOps is important and life of developers can be made easier by automating frequently carried out tasks.

- Regular feedback from the project users can help fix bugs and also optimize the tasks.

- Importance of unit testing and integration testing because they help you understand the code better and how it performs under different circumstances.

- Importance of code compliance and security at a project level and organizational level.

References

[1] "*What is FedRAMP?*" https://aws.amazon.com/compliance/fedramp/