Accomplishments

Here are the list of contributions/accomplishments from the past 4 years in fidelity  Alternative investments:

* Cross-Team Support: Assisted multiple squads (how many teams, just Alts or anyone outside of Alts?) (1)  with pipeline creation and infrastructure utilization, source code development (Call wf and Sec liq) providing guidance and resolving doubts. (Did you learn how to do pipelines on your own and then shared with others?) (Was anything unique about these pipelines or infrastructure that was different and complex?)(2)
* Innovative Pipeline Development: Created utility pipelines from scratch, including those for uploading files to S3, writing automation test cases, injecting secrets into Jenkins Vault, and manipulating Terraform state, Promoting Artifacts. (Useful to all of alts) (What benefit did this bring to the squads? Time savings, how long does would this normally take if you didn't do this?) (3)
* Leadership in New Initiatives: Implemented the entire infrastructure for the Secondary Liquidity initiative, including blue-green v2 deployments and Jenkins pipelines Segments on my own by looking at backend code and figuring it out, without existing documentation.
* Problem Solving: Identified and resolved issues during multiple production deployments, not only for my own team but also for others. (Is it possible to list out what the issues were and why you were the only one who could solve) : Took care of more than 20 production deployments for our own squad and help other squads with their own production deployments as well when they were stuck. (What were other teams stuck on and do recall how many installs you assisted other teams with?) (4)
* Disaster Recovery Automation: Developed a disaster recovery solution using Lambda and Event Bridge to automate recovery processes on the Control-M infrastructure, eliminating the need for manual intervention. Designed and Developed common Control-M Infra to be shared across Alternative Investments, on my own that enables to trigger AWS batch using Route 53 as a hostname/Domain. : Implemented a solution that will allow the IP address of the ec2 instance to be fetched via a lambda and will auto update the route 53 IP address whenever the autoscaling group terminates and spins up a new ec2 instance through the use of an even bridge (what did this do for us? Did this reduce the time spent during install to perform DR test? Did this help R4? Any time savings here?) (5)
* End-to-End Lambda Development: Independently coded and deployed Lambda functions from scratch, managing the infrastructure setup, **source code development**, deployment, and Terraform imports. (Call wf) (Were you the only one on the squad that could do this? Was this a something you asked for to help you learn how to do it?) (6)
* Terraform Refactoring: Contributed significantly to refactoring Terraform modules to improve efficiency and maintainability and helped whole of alts with the same deployment. (Was this specific due to Cybersecurity or the move we did to go on the standard Terraform?) (7)
* Figured out a solution for OPA changes for doc mgmt. API (Only API that was breaking after implementing OPA changes, when Noone else from CAPE or ALM was able to provide a solution. (A struggle)
* Handled PR Reviews and Story creations, managed meetings (Internal review, Tech connect, DSU, External Reviews) in addition to my own development/story work
* Recognition and Approachability: Received multiple recognition awards from peers and leadership for my contributions and known for being approachable and always ready to help. Helped members of our team as well other teams like fccs, ai-pros, houseofscrum and folks outside of alts in some cases. (How many? You should be able to see this in Recognition Central) (8)

* What have you learned in the past 4 years? :

When I joined Fidelity via Capgemini in 2021, I was hired solely as a Node.js developer and was proficient only in Angular and React js at that time, leveraging my experience as a teaching assistant during my master’s program. Since then, I have expanded my skill set to include Java with Spring Boot and Spring Batch and gained extensive knowledge in AWS cloud infrastructure. I earned my AWS Cloud certification due to this, by giving the AWS Global Certified Cloud exam, with all that I learnt from here when implementing Call Workflow and became proficient in utilizing a variety of AWS services, creating, and managing them through Terraform scripting that I learnt proficiently as well and at that time was an entirely new concept to me. I soon learnt and mastered Groovy scripting, which is essential for writing the backend code that is used by all the pipelines under Fidelity. Additionally, I have enhanced my problem-solving abilities, learned to create architecture diagrams, learnt how come to up with innovative solutions, debugging errors, problem solving, critical thinking,  improved my documentation skills, communication skills, agile tools like JIRA, creating stories well and gained a new confidence and insights into the business aspects of technology solutions.

* Why do you want to remain with Fidelity? :

I want to stay with Fidelity because it’s a place where I’ve grown a lot, both in my skills and my career. The supportive environment and the opportunities to keep learning and tackling new challenges make me excited about continuing my journey here. I truly appreciate the company’s commitment to innovation and the way it values its employees, the work culture and how folks respect one another’s ideas. I’m looking forward to contributing even more to the team and growing alongside the organization.

* How many teams, just Alts or anyone outside of Alts? **(1)**

I’ve supported nearly all teams within the Alts program, both onshore and offshore, with the exception of Gamechangers, as I haven’t had much interaction with them. Outside of Alts, I’ve assisted two squads from (WI) with Control-M setup and three squads from PI regarding Terraform and CloudFormation configurations.

* Did you learn how to do pipelines on your own and then shared with others?) Was anything unique about these pipelines or infrastructure that was different and complex? **(2)**

Due to limited documentation from the DevOps teams, I had to figure out how to build these pipelines mostly on my own. I studied backend code from various repositories under the Fidelity-Green Org, experimented through trial and error, and worked through multiple challenges. Once I successfully implemented the solution, I documented the process and conducted knowledge-sharing sessions for my squad and other teams.

* What benefit did this bring to the squads? Time savings, how long does would this normally take if you didn't do this **(3)**

The utility pipelines I created are now widely used across Alts. They’ve significantly reduced manual effort—for example, artifact promotion that used to take about 20 minutes for five APIs can now be done in under 5 minutes using the pipeline. This also includes JC Vault secret injection tasks, which are now automated and easy to use, even for new team members unfamiliar with the process. Previously, these required setting up locally, installing software, and running commands manually.

* Is it possible to list out what the issues were and why you were the only one who could solve **(4)**

Over the years, I’ve resolved numerous deployment-related issues, including over weekends. While I can’t recall every specific incident, many were related to batch jobs, APIs, and Lambda deployments—often caused by AWS permission errors, gaps in documentation, infrastructure issues (Terraform), or human error. I’ve developed a strong understanding of the backend systems, which helps me trace issues from error logs and quickly identify root causes, whether they’re related to infrastructure, deployment, or source code.

* What did this do for us? Did this reduce the time spent during install to perform DR test? Did this help R4? Any time savings here? **(5)**

This automation helped both the R4 squad and others in Alts. Previously, Control-M infrastructure needed manual intervention during installs and DR testing. I developed a Lambda-based solution integrated with EventBridge, which made the system self-reliant. It now automatically recovers from production issues like image non-compliance or autoscaling failures—removing the need to raise CM tickets or manually redeploy.

* Were you the only one on the squad that could do this? Was this a something you asked for to help you learn how to do it? **(6)**

I built the first two Node.js Lambdas (File Validator and File Processor) and the Node.js Service Framework for Call WF from scratch—handling Terraform for infra, Groovy for CI/CD, and TypeScript for source code. At the time, no one else on the squad (besides our tech lead) had experience with Node.js. These implementations later served as a learning reference as the team transitioned to Java-based development.

* Was this specific due to Cybersecurity or the move we did to go on the standard Terraform? **(7)**

This was part of an enterprise-wide shift to standardize Terraform using CAPE modules. It involved deprecating our old modules and importing infrastructure into the new state—a complex process due to AWS permission restrictions in higher environments. I contributed significantly by troubleshooting these production deployment issues and helping establish a process for safely transitioning state resources.

* How many? You should be able to see this in Recognition Central? **(8)**

I’ve received 5 recognitions on Recognition Central and around 10-15 via email (*don’t remember, made a mistake of not saving those*), many of which I may still be able to retrieve from archives. Some might only be accessible by [@Patta, Venkata](mailto:venkata.patta@fmr.com) or [@Deenadayalan, Naganand](mailto:Naganand.Deenadayalan@fmr.com), since I left for a month in between cause of the visa situation.