

SUMMARY

- During the migration planning, teams currently managing on premises infrastructure were pivoted to manage the migration of services to AWS and should continue to manage cloud infrastructure
- It was desirable to pull servers under infrastructure as code, specifically Terraform, because the client had standardized on this tooling for all new cloud infrastructure, allowing migrated servers to be maintained as if they were originally cloud built
- The team took a 'teach while building' approach, allowing the client to pick up the necessary skills while maintaining migration velocity

CHALLENGE

- The client's infrastructure team was inexperienced in cloud technologies and infrastructure as code
- The migration timeline required quick implementation, so it was not feasible to stop migrations until teams were ready to fully manage this new infrastructure

SOLUTION

- We leveraged pair programming to start by showing our client's team how migrations are done, giving a space for questions to be raised in a controlled environment
- In addition to pair programming on the migrations being performed, the team set up time with these teams to give them exercises to strengthen their cloud expertise
 - This gave the client familiarity with Terraform, and also cloud resources themselves, and gave a forum to explain analogs between on AWS services and their on premises counterparts
 - Trust was built by tackling existing problems, even unrelated to migration. Solving for pain points that were the root cause of much of their needed support, freed time up to plan larger improvements to the infrastructure
- We followed the model of 'see one, do one, teach one' to improve skills, empowering the client's infrastructure team to gain migration experience and also practice teaching, mentoring, and providing a safe environment where questions could be asked and answered, without judgement