# Abhishek Danalakota

DevOps Projects

## Client: Freelancer

**Description**: My client wants to change the parameter values of the target Synapse environment while deployment pipeline.

**Implementation**: As a DevOps engineer, I need to work on a solution to match the clients requirement.

1. I have got into a call with client to understand the requirements and bring an solution to his pain point.
2. After understanding the problem, I worked on my own for couple hours by reading through Microsoft blogs, similar community issues and official Microsoft documents.
3. I came with 2 approches for the problem one is to implement an inline PowerShell task which initial the git command, replace the parameters via PowerShell command and commit the solution back to repo before deployment.
4. Second one was to override the parameters by adding template.parameter.definition.json file onto the collaboration branch and refer those parameters in the Synapse task in the deployment pipeline.
5. Approach one wasn’t that useful as parameters might be infinite number and it will be mess by creating large set of parameters in the PowerShell script
6. Second approach was useful because it will eliminate the PowerShell task in the pipeline and at the same time custom parameters can be passed onto pipeline variables.
7. Client was very happy that he got the resolution and want to approach for future like invoking sql scripts in Synapse workspace using Azure CI/CD.

## Client Name: Local health sector

**Project**: Implementing DevOps CI/CD for Azure Functions

**Description**: Client has several azure function apps with node.js, dotnet & python programming languages.

1. Pain point for the customer is to move the function apps onto higher environments, initially they are using manually push from laptop using command line.
2. Client’s requirement is to automate the deployment process using Azure CI/CD

**Implementation**: As a DevOps Engineer I have conducted workshops with client to understand their goals and work together to achieve them.

1. Went through client’s code to see if there is any blockers like what version of node.js, dotnet & python they are using, is their application code uses an blob trigger, https trigger, Service Bus Trigger etc
2. As the client them self doesn’t have an Azure DevOps setup.
3. Created an application registration in the Azure portal, configured service principles to Azure DevOps.
4. Asked developers to push the code onto Azure repos
5. Started writing yaml code to create a build (CI) process for Azure function app, configured build tasks with repos, implemented branching strategy to avoid any unnecessary changes to the code.
6. Tested build process to see if the artifacts are generated.
7. Written yaml code/configured deploy tasks to deploy (CD) the code onto higher environments.
8. Asked testing team to perform testing on the application to check is any feature is missing.
9. Documented the whole CI/CD process in Azure Wiki for BAU purpose.
10. Enabled triggers to automate deployments. Enabled approvers to avoid any unnecessary deployments to higher environments.
11. Configured stages in the deploy pipeline so when a user wants to push the code onto specific environment, they can select the environment and deploy.

## Client Name: Public sector client

**Project**: Implementing CI/CD for Azure Data Factory

**Description**: Client new to Azure DevOps and wants someone to teach and implement CI/CD process for their existing solution.

**Implementation**: As a DevOps engineer I need to implement end to end CI/CD process for ADF

1. Analysed their ADF pipelines.
2. Configured service principle from Azure DevOps to Azure, Integrated git repos with ADF, configured git connection in ADF
3. Went through the requirements, researched on Microsoft articles, followed Microsoft best practices to implement the solution.
4. Written yaml file to for build (CI) and deploy (CD) pipelines, as MSFT recommends to add PowerShell tasks as pre and post deployment steps to ensure stopping, restarting trigger and performing clean-up tasks.
5. Configured tasks like, installing npm packages, Validate and generate ARM templates, copy ARM files, and push artifacts.
6. As the few of the parameters contain credentials of the environment, I have implemented Key Vault integration to the pipelines so no value is exposed while pipeline runs.
7. Along with testing team, helped them with testing of whole solution.
8. As part of knowledge transfer, I have documented all the steps involved in running the CI/CD pipelines and stored in wiki
9. Conducted KT sessions with in-house developers and explained the flow.

## Client Name: Local government body

**Project**: Implementing CI/CD for Power Apps and Dynamics 365

**Description**: Client requirement was to implement complete CI/CD process for their front end application and backend Dynamics 365 solution

**Implementation**: As a DevOps engineer, I have collaborated with their developers to identity the complete CI/CD process.

1. Understood current approach of moving solution to higher environment, found out that it would take around 4-5 hours to move the solution manually.
2. Started with basics steps like, App registration, service principle and register app reg as system administrator on the source and target environments in Power Platform Admin Centre.
3. As power apps portal/Dynamics 365 code comes as a solution.zip file, it is necessary to write an task which communicates with Power apps/Dynamics and copies the solution onto current git repo.
4. Configure build pipeline with tasks like powershell cli to initial git to copy files and save to current repo, install the power platform tools, power platform who I am tool tasks, solution export task, unpack solutions tasks and publish task to move the solution from lower to higher environments.
5. For the deploy pipeline, tasks like set artifacts, powerplatform import solution, powerplatform publish customisation are required.
6. To minimise the risk of deployment onto wrong environments, actions are taken as below.
7. Automatic triggers are customised so that each platform lead is responsible for the deployment by approving the pipeline by approver gates.
8. Branching strategy like deployments to production environment should be done via main branch etc.
9. After successful implementation of the CI/CD I have rolled out the knowledge transfer session with their developers to hand over the project.