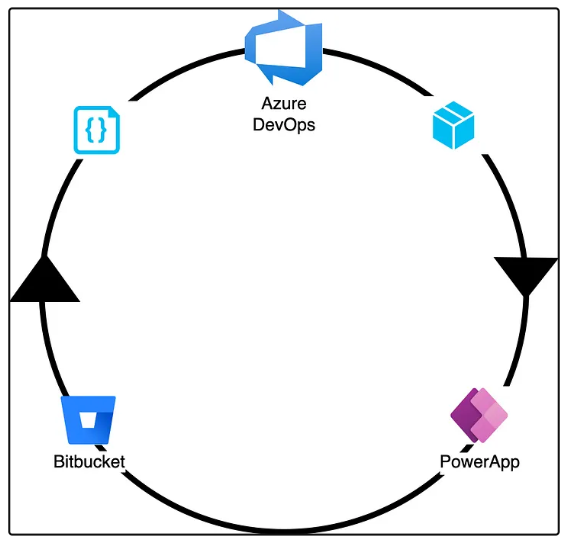
# **Azure DevOps with Power Automate**



How to use Azure DevOps to enable CI/CD for Microsoft PowerApps.

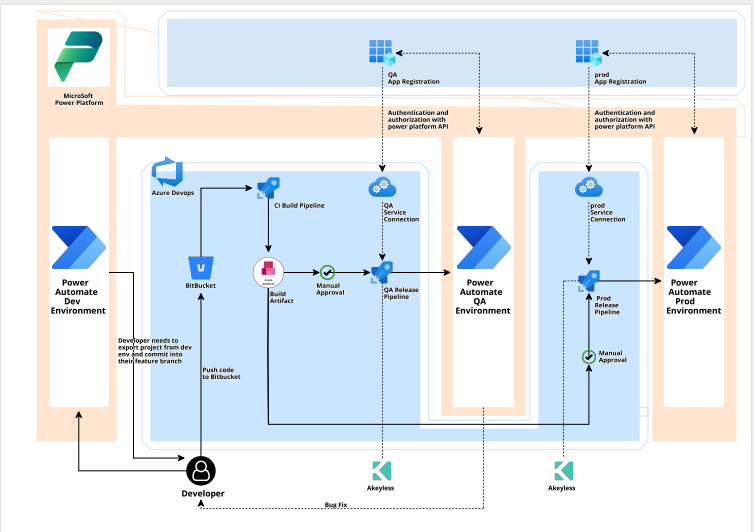
# **Prerequisites**

* Microsoft PowerPlatform Account
* An Azure DevOps Account
* Azure Account
* Bitbucket Account

**Main component**

1. Create Power Automate Environments
2. Create Power Automate in Dev Environment
3. Create Service Principals with PowerPlatform Permission
4. Create Power Platform Service Connections in Azure DevOps
5. Create Bitbucket Service Connection in Azure DevOps
6. Create Azure DevOps Build Pipeline
7. Create Azure DevOps Release Pipeline
8. Test the CI/CD Flow
9. Summary
10. References

# **Azure DevOps Architecture**



# **Process Overview**

1. Once developer has finished the development and wants to deploy to QA, developer will manually trigger the Build Pipeline.
2. The Build Pipeline will export the Power Automate Solution from Power Automate Dev Environment. It will create a solution file of power automate workflow. After the file creation the build pipeline commit the solution file to Bitbucket.
3. Then the Build Pipeline will create the Build Artifact using the solution file and store the artifact into Azure CI artifact.
4. When Build Artifact is created, it will automatically trigger the Release PipeLine for QA environment. Once approver approves the QA release, the QA release pipeline will deploy the Power Automate Workflow artifact into QA environment.
5. After QA testing is done if QA find any bug or any issue then the developer again need to make changes in dev environment and again trigger the build pipeline, and if all good from QA side, thenthe Approvers will approve the Pre -Prod environment release pipeline and it will deploy the artifacts into Pre-prod environment.
6. After the Pre Prod testing is done, the approver will approve the prod release pipeline and it will deploy the artifact into Prod environment.
7. So, for the pipeline and power automate authentication we have separate app registration along with service connection for each environment.
8. The app registration in azure works as a mediator between the release pipeline and power automate environment for authentication. So basically our release pipeline authenticates with the help of service connection to app registration and app registration had access to the power automate platform, so with of power automate API the app registration authenticates with power automate.