Servian Tech Challenge Assignment

GitHub Solution Repo: https://github.com/RaviCheetirala/TechChallengeApp

By Ravi Cheetirala
05-June-2021

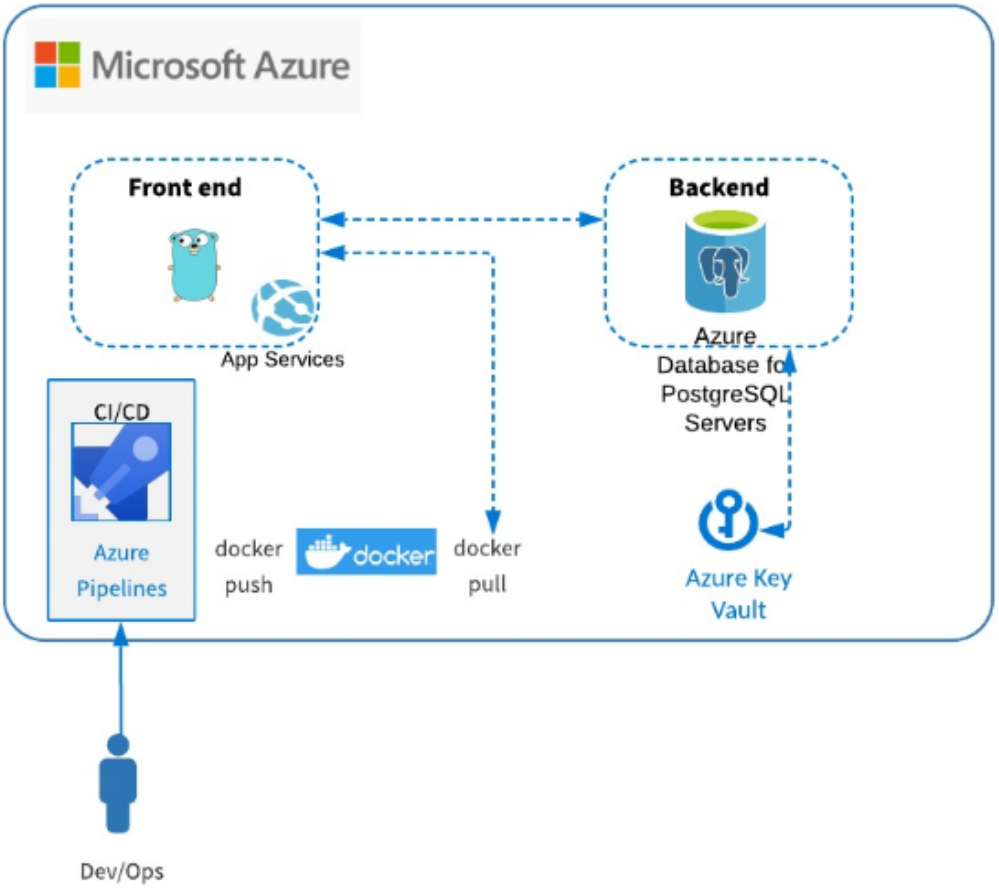
App Architecture:

Tech stack given :

Source Code Git:
https://github.com/servian/TechChallengeApp

Frontend	SPA Written in GoLang
Backend	PostgreSQL

Proposed Architecture:



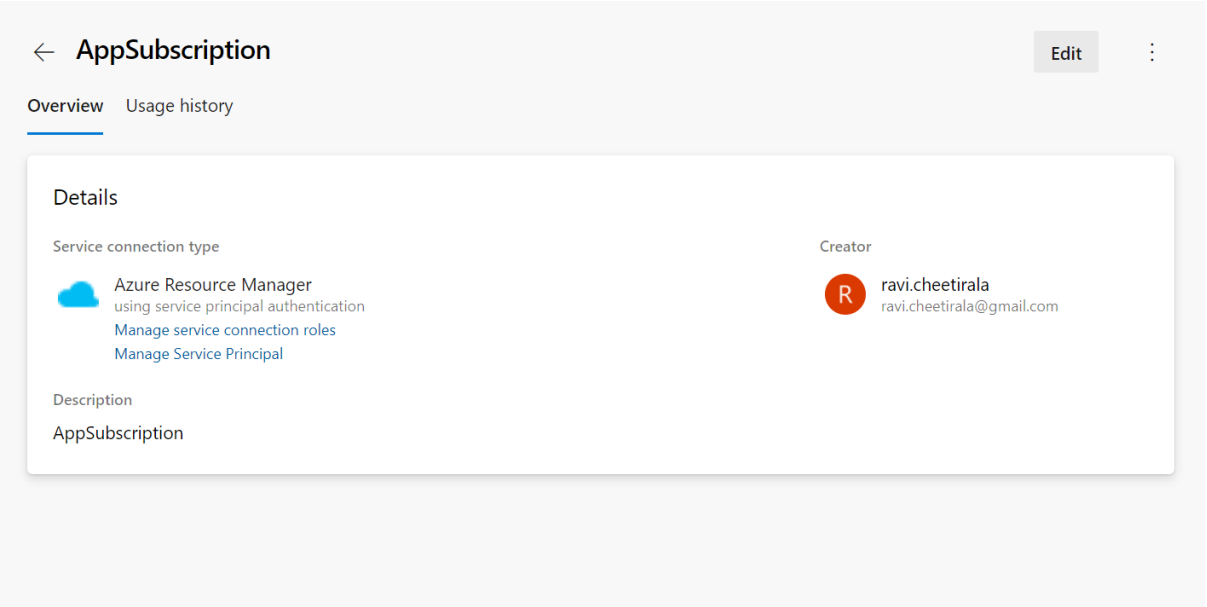
- The front end of the application is deployed in Azure App Service using the docker container.
- The Database of the application is deployed in Azure Database for PostgreSQL servers
- Both App Service and PostGreSQL are the Paas Services.
- CI/CD Pipeline is created to install both the Database and App Service ,along with the required configuration.
- App Service parameter file is hardcoded with the Docker image(318300/servianapp)

- There is No separate build pipeline is created to build the docker image

CI/CD:

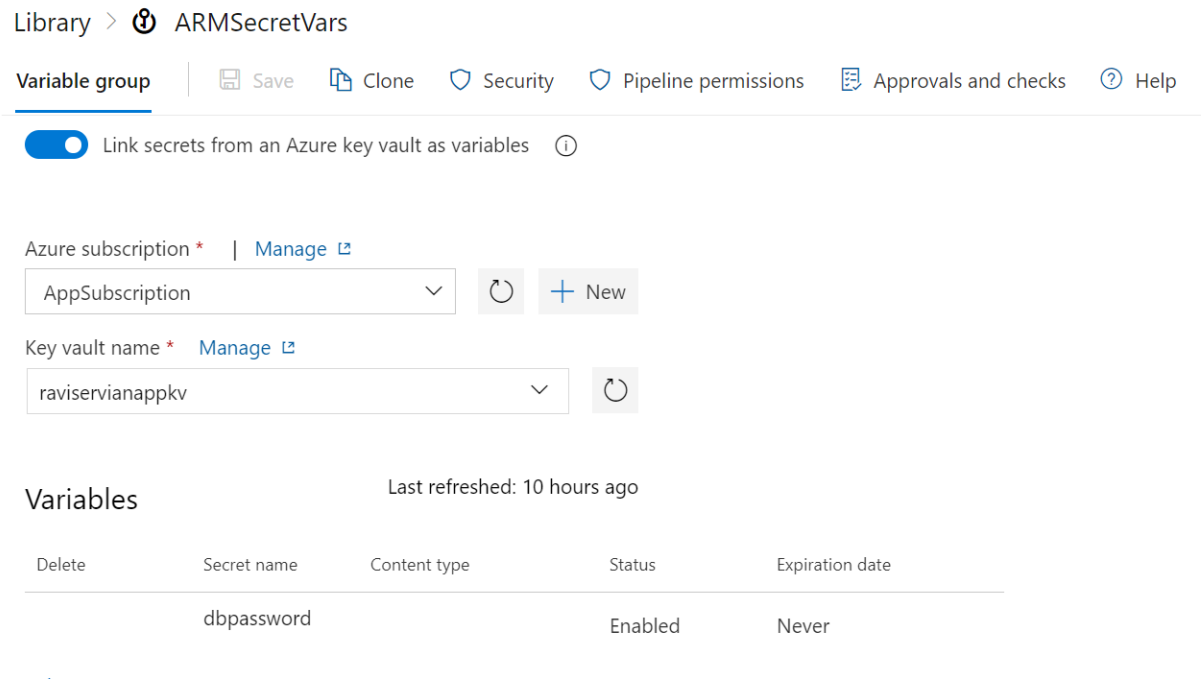
Pre-Requisites:

- Create a service connection with the name “AppSubscription” to connect to the subscription.

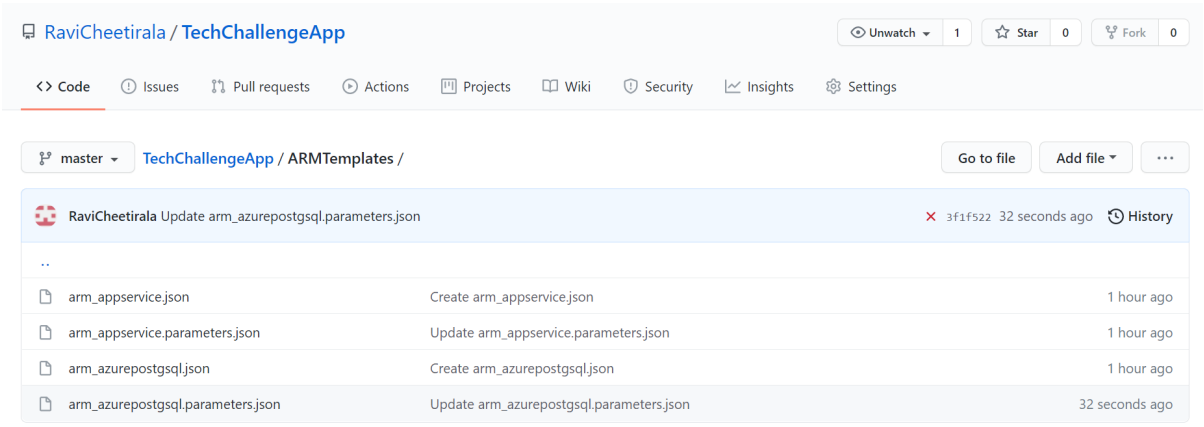


Note: We would need to create a Service Principal for the subscription to connect to the subscription from Azure DevOps assuming both Azure devops and App Subscription are different

- Create a keyvault with the below CLI commands
  - az keyvault create --name raviservianappkv --resource-group ServianAssignmentRG --location "Australia East"
  - az keyvault secret set --name dbpassword --vault-name raviservianappkv --value *****
- Create variable group in Azure DevOps to fetch the secrets from the key vault in the pipeline

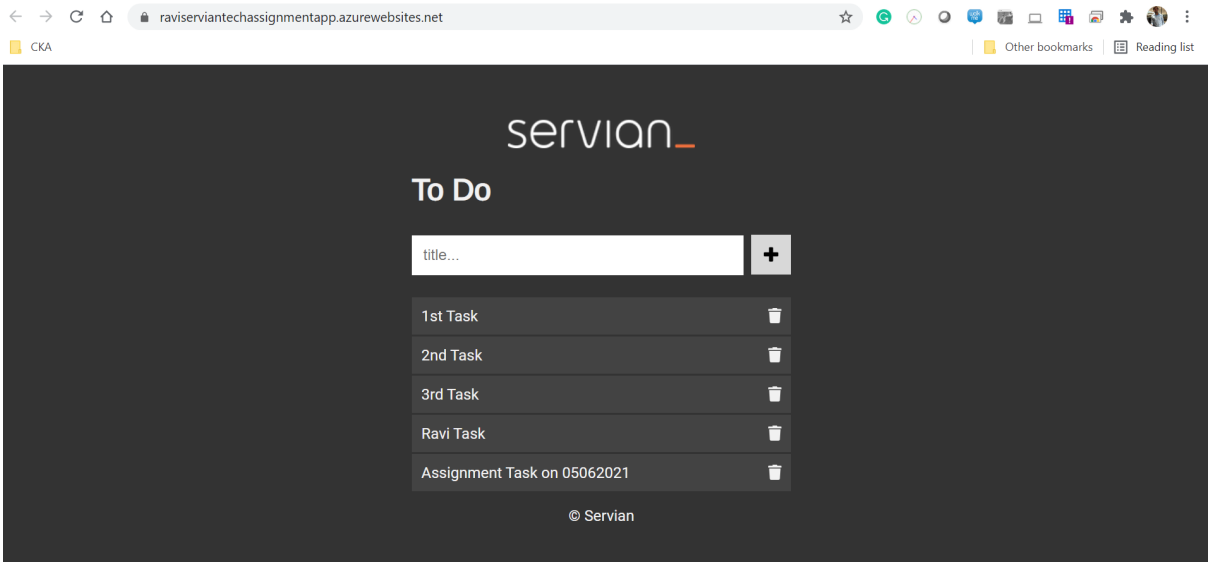


- ARM templates are created for all the required resources and available under the ARMTemplates folder ,which is under the root folder of the application.



Executing the app:

- Import the Github project in Azure Devops project
- Create the pipeline using the existing pipeline.yml(azure-pipelines.yml) ,which is available under the root folder of the repo.
- Trigger the pipeline manually,once the pipeline complete, please navigate to below url in browser
    - https://raviservianttechassignmentapp.azurewebsites.net/



Code/Config changes performed:

- Dockerfile is updated to add the port number and entrypoint is updated.
    - RUN echo "./TechChallengeApp updatedb; ./TechChallengeApp serve" > trigger.sh
    - EXPOSE 8080
    - ENTRYPOINT ["/bin/sh", "trigger.sh"]

- Conf.toml is updated with the below connection details.
    - "DbUser" = "servianappuser@ravixrca4postgresql"
    - "DbPassword" = "<keyvault key>"
    - "DbName" = "serviandb"
    - "DbPort" = "5432"
    - "DbHost" = "ravixrca4postgresql.postgres.database.azure.com"
    - "ListenHost" = "0.0.0.0"
    - "ListenPort" = "8080"
    - "sslmode" = "require"

- Db.go is updated as the AzurePostSQLDB creates the user with user@servername,while the database creation is failing with that name convention ,to avoid that,the owner is hardcoded and the same is being created in the CI/CD process. Also the table_namespace is not required to be given in the latest versions of DB.
- Docker image is in my public docker hub repo with name “318300/servianapp”

Some of the improvements to be/can be done:

- In the given solution,database is open for all the azure resources, which can be avoided by creating a private endpoints /vnet
- App Service is also not restricted to the world, any user through the internet can be accessed ,which can be avoided by an isolated SKU/ASE
- ARM template can be extended to run the scripts instead of a separate CLI task.
- Parameterisation of the data.