**Government Processing Cloud Solution**

[Github Link](https://github.com/arshappleid/Cloud_Gov)

**Stakeholder List -**

* People who will use the cloud Solution.
  + Government Organization - Needs to use the cloud infrastructure to Manage all the cloud applications it offers to its service people.
  + Government Employees - People who work for the Organization, that help run day to day Services of the Organization.
    - Tier 1 Employees – Upper-Level Management, Has the capability to taken down the whole infrastructure with right Credentials.
      * Has Access to Backup database, PROD , DEV , TEST Environments.
    - Tier 2 Employees - Individual Team Manager, that help run individual projects/ teams inside the organization.
      * Only Have the capability to taken down their associates Projects / Services.
      * Should have access to monitor Individual Teams Expenses .
    - Tier 3 Employees - Team Members, that contribute to a teams projects. Require Access to the organizations data.
  + Organization Teams (Managed by Tier 2 Employees) -
    - Marketing - Have Access to the Front End Applications Only.
      * Will be gives access using IP addresses , and DNS names .
      * Only person with direct access to database.
    - IT - Have Access to all the resources , for maintenance purposes.
      * Only have access to DEV , STAGING database.
    - Development - Individual Development Teams , that develop applications that consume organizations data.
      * Can Have multiple Workloads (multiple apps).
      * Has Access to DEV, TEST , Environments.
      * No Direct Access to a database, only through an EC2 instance.
      * Ex. Front End Developers, Back End Developers , Full Stack Developers.
    - Data Analysis - Data Analysis Team that only has access to the DEV , PROD , TEST Database Environments.
      * No Direct Access to a database , only through an EC2 instance.
      * Ex. Data Engineers, BI Analysts , Data Analysts.
  + Normal People - People who will be able to access the front End Applications of Individual Services Offered by the organization.
    - Will be Given Access using DNS names .
* Workflow - (Record each in a Workbook)
  + Development Team requesting a new Environment for a new Service.
    - Provision DEV , TEST , PROD Environment.
    - Only the IT team should be able to change Resource Config.
      * Resource Config - Security Groups, Target Groups, Provisioned Instances.
      * If anyone Makes the Change they should be recorded in Cloud Trail, for Management to review when required.
    - Give Each Environment Access to appropriate Databases.
    - Configure END-to-END Tests that validate the Implemented Infrastructure.
  + Development Team Rolls out a new version of Current PROD app.
    - Considered Weight Deployment - Only Some traffic is routed towards the new instance.
    - Configure END-to-END Tests that Validate the resource has been Deployed.

**Architecture Functional Requirements -**

* Account Management through IAM.
  + Account Registration.
* Store data from different systems
  + Collection of Data, and store into database with transformations. Implement data validation.
  + Present the data, with dashboards.
  + Develop data Standardization procedures.
  + Scalable and secure data storage infrastructure.
* Implement predictive Analytics through Dashboards.
* Implement Secure Scalable File Storage System.
* Incorporate Business Intelligence and analytics Capabilities.
* Security and Privacy -
  + Compliance with data privacy and security intelligence.
  + Implement measures to protect the data.
  + The data must reside at rest and during transit within the 50 states, the District of Columbia, or outlying areas of the United States.
  + Establish access controls and user authentication mechanisms to ensure appropriate data access based on roles and privileges.
  + Comply with relevant data protection regulations and industry best practices.
* Scalability And performance -
  + Accommodate for growing Volume and Increasing User demands.
  + Optimize query performance through appropriate indexing, partitioning and caching strategies.

**Architecture Non-Functional Requirements -**

* Governance -
  + This solution does not implement any governance principles for now.
  + There Should be a disaster recovery strategy.
* Security – A separate VPC to monitor & record Incoming Traffic logs.
* Operational Excellence -
  + Implement Monitoring practices.
  + Define What needs to be recorded.
* Performance Efficiency – The System should have very low latency to respond to requests.
* Reliability – The system should provide 99.99 % reliability.
* Scalability – Infrastructure Should be able to scale to massive incoming traffic demands.
* Cost Optimized – The infrastructure should be designed to reduce costs, but performance should not be sacrificed.

**Database Requirements**

* One Distributed Relational database administered, available across multi-region for Low Latency.
  + Use Aurora Database that implements database backups, multiple read replicas, Separate End points for reading and Writing.
* One non relational database - to Store key-value indexed organizational Data.
* Backup Database (for the relational Database) that Copies the data from the main Database once every day.

The above requirements were understood. The databases were not implemented as Infrastructure as code because, there was no need to make changes once the database had been deployed. Also having the Database as Code, imposes the risk of them being easily deleted (i.e) , it is best if someone who wants to delete the database to do it from the AWS console.