Graphical user interface, text, application, email

Description automatically generated

|  |
| --- |
| [AWS EC2 and Amazon Route 53 - reverse lookup - finding](https://confluence.refinitiv.com/display/RIANA/AWS+EC2+and+Amazon+Route+53+-+reverse+lookup+-+finding)  EC2 reverse lookup is only possible via ElasticIPs and a [special request form](https://portal.aws.amazon.com/gp/aws/html-forms-controller/contactus/ec2-email-limit-rdns-request). More details can be found here: <https://aws.amazon.com/ec2/faqs/?nc1=h_ls>  Route 53 Revere lookup zones capabilities are independent of EC2 reverse lookups of ElasticIPs.  Conclusion:  Current implementation of only forward lookup DNS within DDI environemnt conforms with Route 53.  Reverse lookup could be handled in the future as special case (similar to the approach of AWS). |
|  |

|  |
| --- |
| Diagram  Description automatically generated  Route 53 DNS Failover:  The solution below provides an active-active setup for WCNG API across two regions |
| 4.10 GIT Practice (Draft)  Usage of Branches  Multiple types of branches could exist:   * Master - released code. Deployed to Prod env * Develop - preproduction code * Feature or bugfix - under development / UT / QA * Developer's individual branch - individual code changes  | **Branch** | **Usage** | **Env In SDLC** | **Notes** | | --- | --- | --- | --- | | Master | Stable code base. Released, and deployed in Prod env | The release running in Prod env normally |  | | Develop | Relatively stable - Less frequent changes.  Branched from **Master**, for a release or multiple features.   * All code changes shall be merged to Develop branch after development/QA is done * Bugfixes can be branched from Develop branch too, like a feature branch | The release running in PPE env normally |  | | Feature Branch | Not stable - frequent changes when merging features / JIRAs  Branched from **Develop** branch. Contains | The releases running in Dev/QA env normally |  |   Below is a branching diagram showing the typical scenarios:   1. Feature development is preferred to be sequential. If multiple features are in parallel or have any overlap, the 2nd and further merges could have conflict and need to manage resolve 2. Hotfix could be branched from Develop, and need to merge back to Develop and Master when done |
|  |