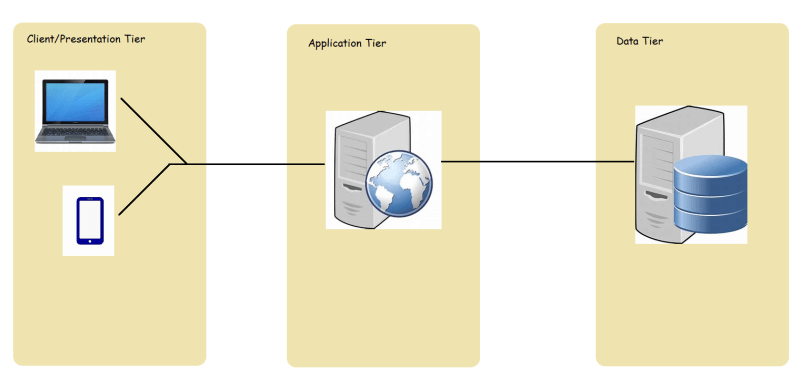
**Understanding Free tier Account**

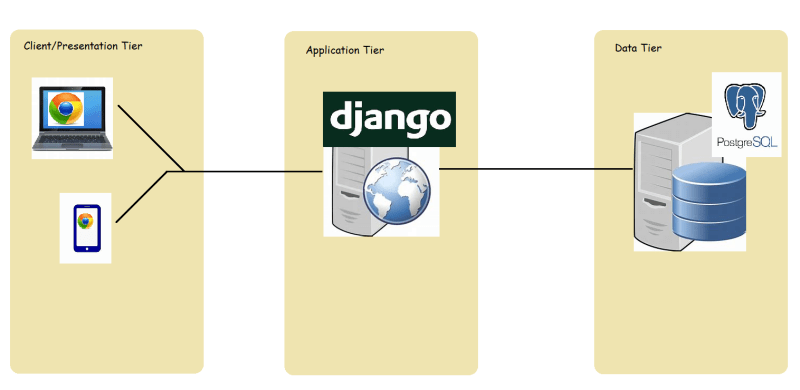
* AWS Free tier Account offers [https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\*all&awsf.Free%20Tier%20Categories=\*all](https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all)
  + some services for free over the period of 12 months
  + Trail versions
  + Some Service free for the life time

**Configuring API Credentials and AWS Idenity And Access Management**

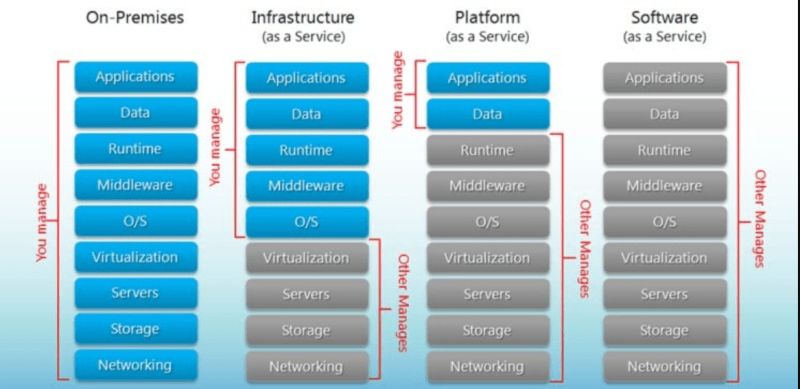
* The user with which you have created AWS Free trail (AWS Account) is called as *root account*
* For enabling programmatic access it is not recommended to use *root account*
* We should be creating another user for programmatic access which are referred as IAM Users.
* For Programmatic access to AWS, the crdentials of user is composed of two distinct parts
  + Access Key Id
  + Secret Access Key
* Each user may have up to two Active Access keys at any time.
* These access keys are long-term credentials and remain valid till you revoke them
* Create an IAM User: <https://serverless-stack.com/chapters/create-an-iam-user.html> for image demonstration
* Exercise: Create an IAM User with Adminstrator access and preserve access key and secret key.
* These Access Keys will be used by code or AWS CLI to access AWS Account to create/modify/delete resources

**Scenario: Classic Three Tier Applications**

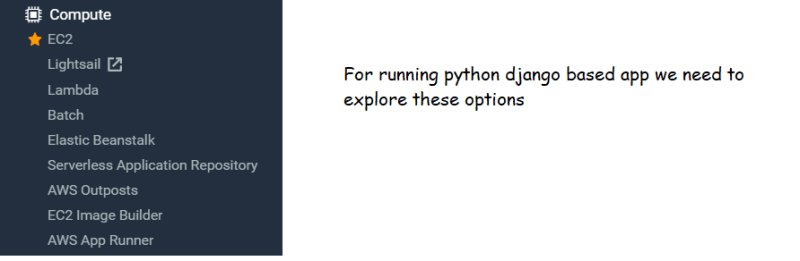
Consider the following simple three tier application 

Consider the following image for technical stack 

* To deploy the above-mentioned architecture in AWS what are the possible options.
* Before we get into AWS Options, let’s try to understand Cloud Service Models
* There are three types of cloud service models
  + Infrastructure as a Service
  + Platform as a Service
  + Software as a Service



Now in this architecture we have-

Application running on python django 

postgres database-