Service details

IBM cloud

The process of getting the price information for a service in IBM cloud is to:

- First get the list of all resources by calling the catalog api
    - The catalog api returns all the possible resources that IBM cloud has, which might or might not have pricing information with them
- For those resources which have pricing information (can be checked through the kind tag), get the children url and then get the data from there
- If they plan kind of resources, get the id from those resources and call the pricing api, which returns rate metrics in all possible currency formats

Following are the different kind of resources that are available through the catalog API (examples for all these are in the resource example JSON file):

- ***buildpack***, this has children url which contains different plans which can be added to database
- **composite**, this has no children and does not have not have any price
- **geography**, information about different locations, does not have any billing information at all
- **iaas** does not have children and does not have any pricing information available
- **instance.profile**, this has information about different instance types that ibm hosts, there doesn't seem to be any pricing information available with it
- **runtime**, for different runtime like ASP.NET core that IBM hosts, no pricing information along with it either
- ***service***, has different child plan urls which have pricing information and can be added to the database
- ***ssm-service***, this also has children urls with plans that have pricing information and can be added to the database
- template, there is no plan or pricing information for template
- **volume.profile**, similar to instance.profile, this also does not have any kind of pricing information
- **plugin**, this does not have any information regarding pricing
- **catalog_schema**, **catalog_root**, these does not have any information regarding pricing either
- **cfee-buildpacks**, no pricing information
- **extension-point**, this is the children url of the plugin and does not have any pricing information
- ***application***, this has plan children urls and have pricing information
- ***plan***, this is usually in the children url of other kinds and has and id which can be used to get pricing information. A single plan can have multiple pricing tiers
- ***Pricing metric***, this has the actual information regarding the price of the plan

Digital Ocean

The process of getting pricing for Digital Ocean is rather simple. It provides an API to get paginated results of each droplet size that it provides. With each droplet config, it gives prices along with it. An example is:

```
{
  "slug": "512mb",
     "memory": 512,
     "vcpus": 1,
     "disk": 20,
     "transfer": 1,
     "price_monthly": 5,
     "price_hourly": 0.007439999841153622,
     "regions": [
         "ams2",
         "ams3",
         "blr1",
         "fra1",
         "lon1",
         "nyc1",
         "nyc2",
         "nyc3",
         "sfo1",
         "sfo2",
         "sgp1",
         "tor1"
     ],
     "available": true
 }
```

It provides databases and storage too, but there is no API to get database pricing in real time. Probably because it provides a simple database and storage pricing which is available here and here respectively.

Azure

Azure is a bit more complicated than Digital Ocean. It requires a bit of manual set up to configure it the first time. A good article explaining that set up is here. Everything under the **one time setup heading** has to be the first time setting it up, all the other steps can be done programmatically.

To get the information, we just have to call the API one time and it returns a huge json file with data about all different services. It contains metrics for different services that it provides. Here is a sample:

```
{
    "EffectiveDate": "2019-04-08T00:00:00Z",
     "IncludedQuantity": 0,
     "MeterCategory": "Azure Database for PostgreSQL",
     "MeterId": "9bd68f02-b6f0-41b2-984c-c4729b649088",
     "MeterName": "LRS Data Stored",
     "MeterRates": {
         "0": 0.057
     },
     "MeterRegion": "KR South",
     "MeterStatus": "Active",
     "MeterSubCategory": "Hyperscale (Citus) Backup Storage",
     "MeterTags": [],
     "Unit": "1 GiB/Month"
 }
```

The different categories that it returns are following:

- API Management
- Advanced Data Security
- Advanced Threat Protection
- App Center
- Application Gateway
- Application Insights
- Automation
- Azure API for FHIR
- Azure Active Directory B2C
- Azure Active Directory Domain Services
- Azure Analysis Services
- Azure App Service
- Azure Bastion
- Azure Blockchain
- Azure Bot Service
- Azure Cosmos DB
- Azure DDOS Protection
- Azure DNS
- Azure Data Explorer
- Azure Data Factory
- Azure Data Factory v2
- Azure Data Share
- Azure Database Migration Service
- Azure Database for MariaDB
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Databricks
- Azure DevOps
- Azure Firewall
- Azure Firewall Manager
- Azure Front Door Service
- Azure Lab Services
- Azure Machine Learning
- Azure Maps
- Azure Monitor
- Azure NetApp Files
- Azure Search
- Azure Site Recovery
- Azure Spring Cloud
- Azure Stack
- Azure Stack Edge
- Azure Stack Hub
- Backup
- Bandwidth
- BizTalk Services
- Cloud Services

- Cognitive Services
- Container Instances
- Container Registry
- Content Delivery Network
- Data Box
- Data Catalog
- Data Lake Analytics
- Data Lake Store
- Datacenter Capacity
- Digital Twins
- Dynamics 365 for Customer Insights
- Event Grid
- Event Hubs
- ExpressRoute
- Functions
- GitHub
- HDInsight
- HPCCache
- Insight and Analytics
- IoT Central
- IoT Hub
- Key Vault
- Load Balancer
- Log Analytics
- Logic Apps
- Machine Learning Service
- Machine Learning Studio
- Machine Learning service
- Media Services
- Microsoft Azure Peering Service
- Microsoft Genomics
- Multi-Factor Authentication
- Network Watcher
- Notification Hubs
- Power BI
- Power BI Embedded
- Redis Cache
- SQL DB Edge
- SQL Data Warehouse
- SQL Database
- SQL Server Stretch Database
- Scheduler
- Security Center
- Sentinel
- Service Bus
- Service Fabric
- Service Fabric Mesh
- SignalR
- Specialized Compute
- StorSimple
- Storage
- Stream Analytics

- Time Series Insights
- Traffic Manager
- VPN Gateway
- Virtual Machines
- Virtual Machines Licenses
- Virtual Network
- Virtual WAN
- Visual Studio Online
- Visual Studio Subscription
- Windows 10 IoT Core Services
- Xamarin University

What is needed to move further?

Next steps are:

- Identify which services are to be stored in DynamoDB for each cloud service
- Which fields and information are required to be stored alongside each service
- IBM provides prices in number of different currencies and locales, should only information from en-us and USD be stored?
- Would storing information regarding droplets from Digital Ocean suffice as it is the only API that it provides?