

## Step 1: Cloud Assessment:



### **Cloud Assessment**



- Asses Cost
- Asses Architecture
- Asses Security

The very first thing we must do is the below assessments and document everything!

#### **Asses Cost:**

We have to do below cost assessment and estimate the total cost of migration to make sure it doesn't exceed the assigned budget

- The cost of the team what working on the migration project
- The cost of creating infrastructure
- The cost of Cloud Service Provider (CSP)
- The Cost of Data Transfer
- Post-migration development & maintenance cost

## **Asses Architecture:**

We have to below Architecture assessment before starting actual cloud migration

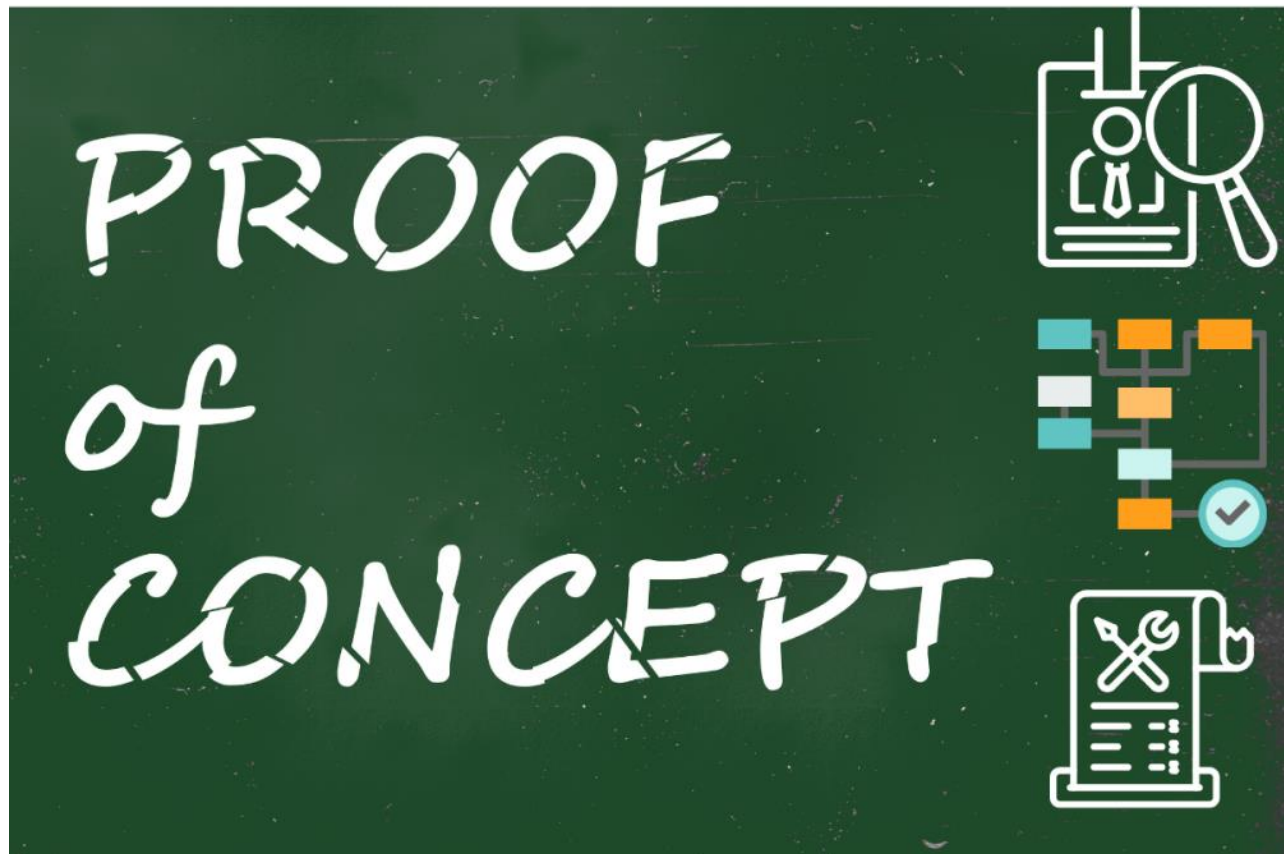
- Create a list of existing infrastructure resources such as servers RAM, CPU, storage, firewall, Databases, etc any and all IT Infra Resources.
- Create a list of expected resources that will need to be created in the remote Cloud Services Provider's platform, based on the existing list.
- Find out the way to save Infrastructure costs and what possible options are available and select the best possible options as per business budget and needs.
- Do a GAP analysis, to make a list of the things that need to be done before moving applications and data to the Cloud

## **Asses Security**

Before moving to the cloud we have to consider below security assessments

- List existing security policies, frameworks, business mandatory policies which need special considerations.
- List how to achieve the above-existing infra requirements in the target cloud or how to improve it even further using available Cloud Service provider given tools
- List all the relevant Security options such as how to use them, are they incur the extra cost or are free of cost, how effective it is, etc

## **Step 2: Create Proof of Concept (POC):**



Before starting actual full-scale migration, there has to be a proof of concepts with at least to see if we can move core services and/or core data or not, otherwise if start full-scale migration and find out that some or most important services are incompatible or unable to migrate due to any reason, that might cause havoc loss to the company/project, so it's best practice to create a PoC before processing further! consider the below points while creating the Pilot project:

- After above three assessment it will be clear with which cloud provider the PoC need to be made i.e. with best possible less cost + more performance service + infrastructure + Best Security options
- Plan to move core service and core data + service first rather than making it feature-rich, try to make a smaller app/data migration only, just to verify what you are expecting to work, are working at all for your app/service or not
- Utilizing Cloud service provider provided tools to make this POC faster
- Not just depend on the Cloud Service Provided Tools, but also have a look at the best possible open-source options

### **Step 3: Data Migration:**



After PoC succeeds and is passed by top management, it's time to start the actual migration process and first, we will move the data:

### **Leverage Storage Options:**

- Select best possible storage option based on requirement + cost/budget + performance
- While selecting the storage option, check for tools provided by the cloud service provider for the data migration
- Check SLA, most of the service providers give SLA (Service Level Agreement) for their storage options, which might be a selection factor for mid/large/enterprise businesses.

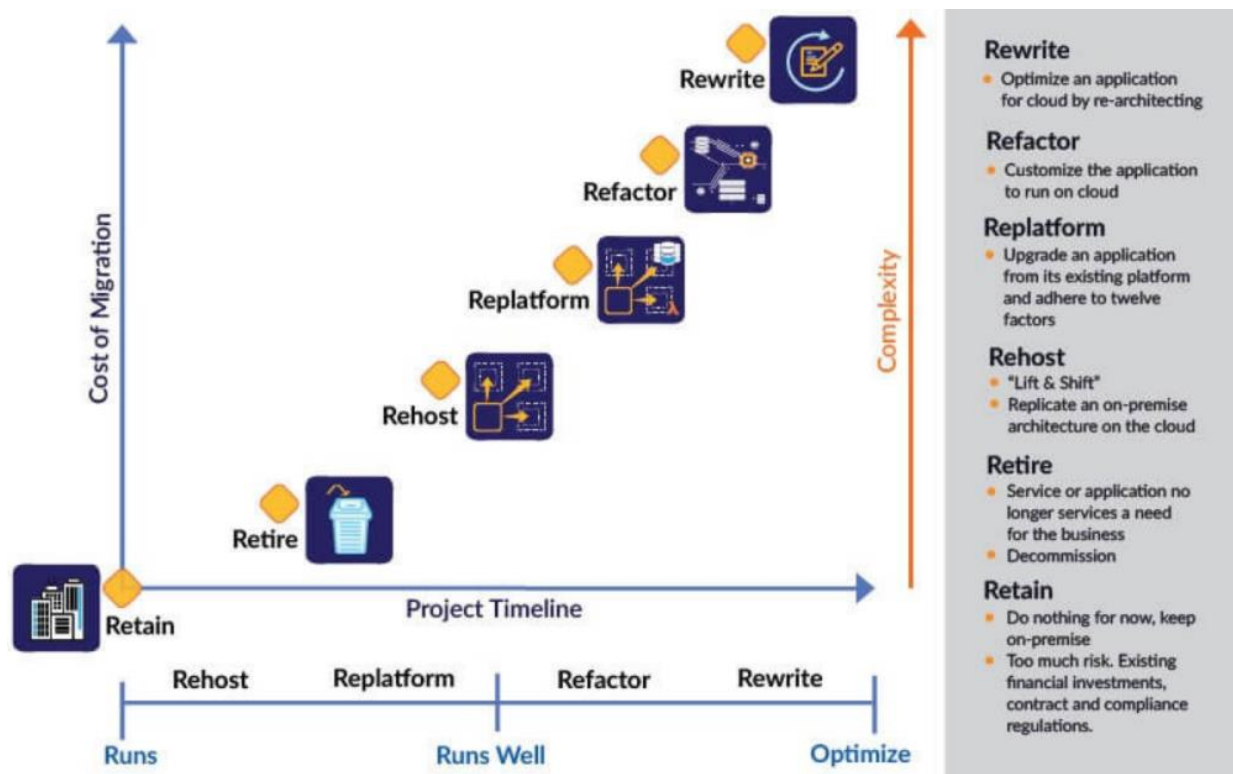
## **Migrate:**

After selecting the storage option, it's time to migrate actual data, and consider the below points:

- Estimate the time it's going to take to complete the Data Migration, let the Operations team and management know, and get concept/approval
- Make Schema / Format ready (if needed) in the remote cloud storage
- Check if data pre-processing is needed, if so, do make processing is done before starting the migration process, e.g. for DB storage export data to CSV or JSON, and for an object at such as pic and doc file to zipping before moving, etc
- Find out when application traffic is lowest, and notify customers that you are going to move data in that time, make an application in maintenance mode so that it doesn't create new data, then start the migration and keep the service unavailable during the migration time
- After the data migration is done, don't just start using the new storage, instead use live data migration tools, which keep the old data storage and new data storage in sync until completely move to the new data storage option, and only move to the new data storage option till fully satisfied with the performance, security and stability of new storage, for this either use cloud service provided tool, or open sources data migration tools or just create your own custom solution!
- If the data is too huge that it might not be wise to do a live migration, instead it will be better to use physical data move using cloud service provided paid data migration service, where they will come to you and copy all the data into some large drive and then take it back to their data center where they do copy data from the drive, it's faster and then you can do live migration for any new data.

## **Step 4: Application Migration:**





There are 6 R's of cloud migration define cloud migration strategies, which include — Rehost, Refactor, Revise, Rebuild, Replace, and Retiring.

- **Rehost** - Rehosting can be thought of as "the same thing, but on cloud servers". Companies that choose this strategy will select an IaaS (Infrastructure-as-a-Service) provider and recreate their application architecture on that infrastructure. in short: **moving applications to the cloud as-is.**
- **Refactor** - Companies that choose to refactor will reuse already existing code and frameworks, but run their applications on a PaaS (Platform-as-a-Service) provider's platform – instead of on IaaS, as in rehosting. in short: **changing the app's code to better support the cloud environment**

- **Revise** - This strategy involves partially rewriting or expanding the code base, then deploying it by either rehosting or refactoring. in short: **moving applications to the cloud without major changes but taking advantage of the benefits of the cloud environment.**
- **Rebuild** - To "rebuild" means rewriting and re-architecting the application from the ground up on a PaaS provider's platform. This can be a labor-intensive process, but it also enables developers to take advantage of modern features from PaaS vendors. in short: **rewrite the application from scratch.**
- **Replace** - Businesses can also opt to discard their old applications altogether and switch to already-built SaaS (Software-as-a-Service) applications from third-party vendors. in short: **retire the application and replace it with a new cloud-native application**
- **Retiring** — After looking over everything in your environment, you're likely to discover there are things you can remove without consequences, doing it can save money, improves security, and gives your people one less tool they have to learn to use. in short: **Remove unnecessary resources**

## **List of Available Cloud Migration Tools & Service**

Leading Cloud service providers such as AWS, Azure, GCP gives us excellent migration tools as follows:

### **AWS cloud migration tools:**

AWS Provides below Cloud Migration tools:

- [\*\*AWS Migration Hub\*\*](#) — This hub lets you track the progress of migrations across AWS solutions, helping you pick the right tools, track metrics, and more.
- [\*\*AWS Application Delivery Service\*\*](#) — Plan for your migration by letting AWS review your on-prem data setup. Collected data is encrypted and accessible from the Migration Hub.
- [\*\*AWS Server Migration Service\*\*](#) — This service makes it easy and quick to move workloads to AWS, particularly when dealing with large-scale server migrations.
- [\*\*AWS Database Migration Service\*\*](#) — Easily and securely move your databases to AWS. The source database remains functional throughout the migration, minimizing downtime.
- [\*\*CloudEndure Migration\*\*](#) — This automated lift-and-shift solution is free for 90 days.
- [\*\*Migrating data and files: AWS Snowball, AWS Snowball Edge, AWS Snowmobile, AWS DataSync, and AWS Transfer for SFTP.\*\*](#) Read more [here](#)

### **Microsoft Azure cloud migration tools**

Azure provides below migrations tools/services

- [\*\*Azure Migrate\*\*](#) — Microsoft's built-in migration service serves as a central hub for tools, progress tracking, insights, and guidance to plan and successfully migrate to the cloud. Most other tools covered below are integrated into this central dashboard.
- [\*\*Azure Migrate: Server Assessment\*\*](#) and [\*\*Server Migration\*\*](#) — These tools let you assess and migrate servers to Azure, including physical servers and VMware, Hyper-V, public cloud, and other VMs.
- [\*\*Data Migration Assistant\*\*](#) — DMA helps locate compatibility issues that could derail your migration. It points out unsupported features, new features, and helps you plot a proper path for database migration.
- [\*\*Azure Database Migration Service\*\*](#) — Migrate on-prem databases to Azure VMs.
- [\*\*Azure Data Box\*\*](#) — Move large amounts of offline data to the Azure cloud.
- [\*\*Move\*\*](#) — This SaaS platform was acquired by Microsoft in 2019. It's a discovery solution that increases business intelligence to see and control environments across.

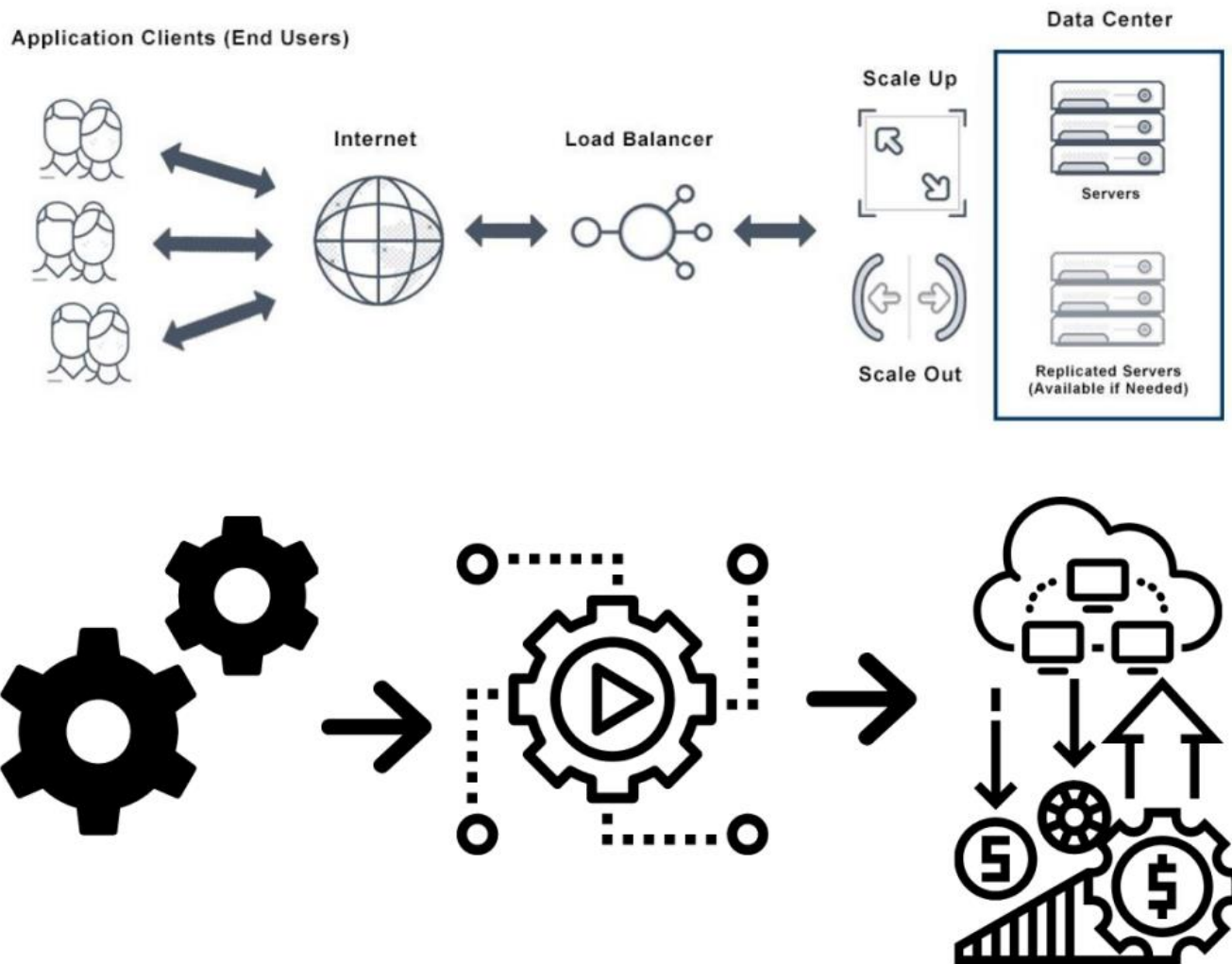
## **GCP cloud migration tools**

GCP Provides below migrations tools/services:

- [\*\*Transfer Service\*\*](#) — Execute large-scale data transfers from online and on-premises sources to Google Cloud Storage.
- [\*\*Transfer Appliance\*\*](#) — For offline bulk data migration, Transfer Appliance lets you securely capture, ship, and upload data using 100TB or 480TB models.
- [\*\*Migrate for Anthos\*\*](#) — Migrate and modernize existing workloads to containers.
- [\*\*Migrate for Compute Engine\*\*](#) — Get enterprise applications running in Google Cloud while data migrates in the background. Validate, run, and migrate applications without reworking them.
- [\*\*BigQuery Data Transfer Service\*\*](#) — Let your analytics team set the foundation for a BigQuery data warehouse and schedule and automate data transfers from your SaaS applications.

## **Step 5: Leverage Cloud:**





Finally, after migration over, remember migration is not just moving data and application to the cloud, don't forget your primary goal, which was to leverage full power, do below to harness it:

### Scale:

- So set auto-scaling policies properly, don't overestimate and don't underestimate, set auto-scaling policies based on application need.

- Over-provisioning costs more and under-provisioning costs clients, which can impact Company's Goodwill, so either is not an option, do experiment scaling as much as possible with Cloud Service provided tools and as well as open-source tools, practice **chaos engineering**!
- Look at **Multi-Cloud**, using containerization or function-based application running approach, we can separate codes(business logic) from Environment (OS/other dependencies), Cloud gives us awesome multi-cloud features, where we can leverage multi-cloud so that in case any outage in any cloud, there is another backup Cloud service provider ready to compensate that.

### **Automate:**

Cloud gives us wonderful automation options such as:

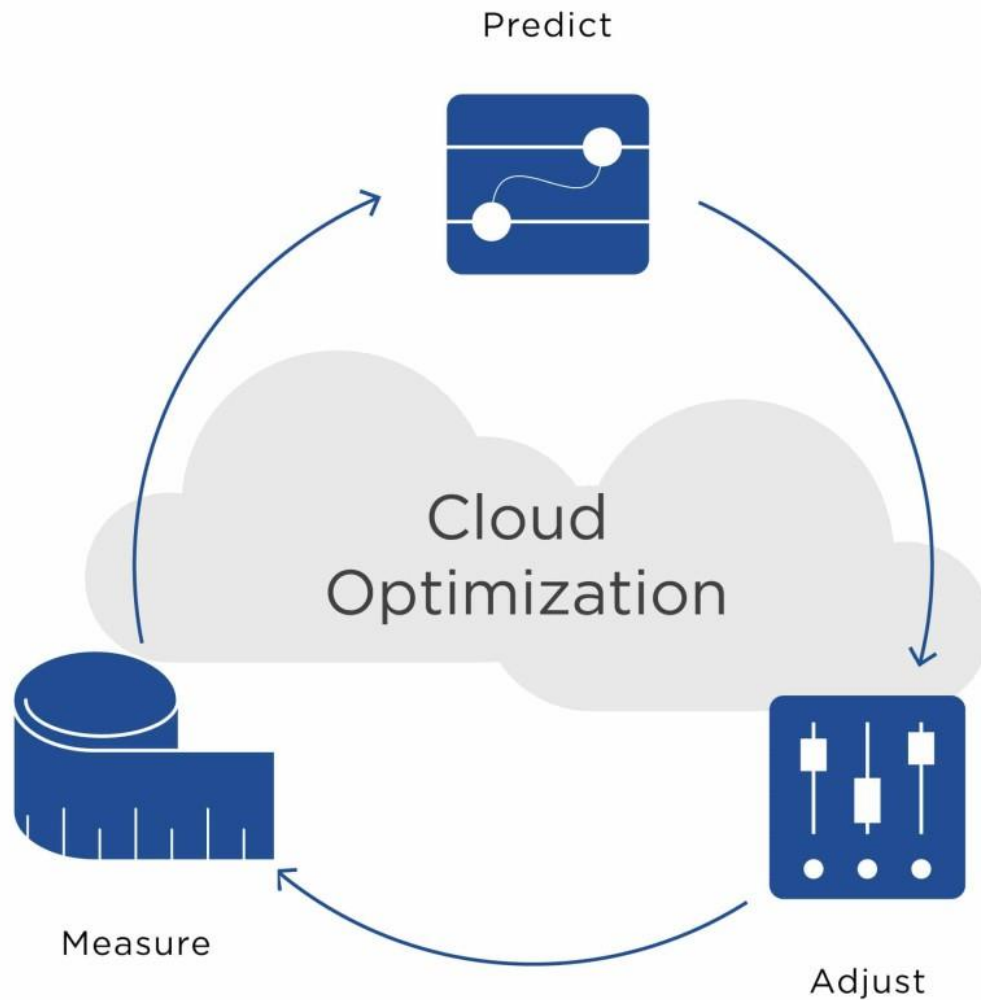
- We can **automate infrastructure** creation/provision, maintenance, upgrade/downgrade, security, kind of everything via Infrastructure as code tools/services such as Terraform, CloudFormations etc, we should and must utilize this awesome feature
- Using Automation, migration from **one cloud to another is also easy** (in case don't use the property tools/service, instead use opensource options)
- Using Automation tools, also help **estimate better** in cost calculation and lets us know what resources going to be created before creating it!
- Even using automated CI/CD process, from source code to deployment all can happen automation or using less manual intervention, making release process much faster and smooth

### **Test**

The test will get even easier:

- If we maintain infrastructure as code, we also can add automated test cases that will make cloud resource testing much easier
- If we maintain the automated CI/CD Process in the cloud, we can do easily add test cases in source code and do run the test cases after build successes and after testing, if passed, then moving to staging/production
- If during the test it failed, using a different Cloud feature we can notify the concerned team about the failure and as soon the fixes are done and codes pushed, the testing can happen automatically again

## **Step 6: Optimizations:**



After all the above in place, it's time to do the below:

### **Monitor(or Measure):**

- **Monitor cost**, do create budget and notification, as it's very easy to thing can go out of budget and hard to manage, so have to make sure it's always under control and keep it under control.
- **Monitor Security**, do make sure its least privileged approach is followed, monitor each and every access using cloud service proved tools, set alert for possible breaches, make

sure strict policies are in place and set an alert if anyone tries to deviate anything, should trigger alert and automation there to rectify it.

- **Monitor performance**, do monitor performance of each and every resource, check if any improvements if not, and do as needed
- **Monitor usage pattern**, create separate usage profiles, like many hours running, when the load is high or less, how much usage happening, over the month, quarter, year, etc, it will use in the next stage

## Redesign (or Adjust + Predict):

After motioning and tethering enough data :

- **Improve based on fact**, based on monitoring data, find out what resources needs improvements/upgrade/downgrade, etc
- **Look for Cost Saving opportunities**, by profiling data and requirements, we can predict the next usage pattern and by leveraging this knowledge we can save cost by either by doing reservations or purchasing savings plans provided by Cloud Service proves, for this we need how much to reserve and monitoring gives us that data.
- **Improve security**, by looking monitoring data & analyzing the data, find out anomalies, possible breaches, and find opportunities to improve it
- **Remove unnecessary resources**, based on monitoring remove the resources which is unnecessary and not getting used at all

## Migrations Challenges:



**Duration:** Although with help of modern tools and services migration takes less time, but in the case of legacy systems it takes a lot of time.

**Complexity:** Mapping existing systems and requirements to Cloud is no easy task and often it's very easy to get things more complicated, need very talented/experienced resource to handle the migration

**Huge Bills:** If not monitored and managed properly, cloud costs can go out of the hands

**Make Management Understand:** Let's agree, Cloud is complicated for non-tech / non-trained persons, management often doesn't know why and what, it's tough to make them understand what is needed and why is needed and its impact on laymen understandable way.

**Human Resource:** Probably **the most important is this**, Cloud migration need experienced knowledge hence experienced professional needed for do asses existing infra, plan the architecture of the cloud and not to mention actually implementing the infra/automate it and do the migration. For this we need, Cloud Architect, DevOps Engineers, Operations Engineer, etc, and since Cloud professionals are in demand, finding the right person is challenging.

## **Conclusion:**

Even though at a glance it looks like Cloud migration is moving resources to Cloud is just moving on-prem to Cloud, but it requires a lot of considerations before actually starting migrations and post migrations manage it.